2013 American Control Conference (ACC)
Washington, DC, USA, June 17-19, 2013

Modeling Driver Behavior During Complex Maneuvers

Theresa Lin[1], Eric Tseng[2] and Francesco Borrelli[1]

Abstract— **This paper focuses on the modeling of driver behaviors during maneuvers that involve tire saturation. Drifting maneuvers are performed by an expert driver on low friction surfaces. Driving data is recorded and used to aid the construction of two different driver models. The first model parallels the driver to a model predictive controller and the second model treats the driver as a switched linear state feedback controller. This paper compares the two approaches and presents a framework where the driver model is integrated into an autonomous controller in the form of an optimization problem.**

I. INTRODUCTION

With the increasing trend of human-machine interactions in our everyday life, the understanding, prediction, and improvement of human decision making has become a fundamental aspect of control system design. Human decision making can be modeled through classical control theories such as linear state feedback control [1], optimal control [2], machine learning [3], Bayesian decision making [4], and hybrid systems control [5]. On the contrary, cognitive architecture frameworks such as ACT-R, focus on the cognitive aspects and model decision making by developing independent sub-modules [6]. Human drivers modeled from this view point can be used to simulate a virtual driver but lacks the integrability with control frameworks. The reaching experiments presented in [7] suggests that human behavior involves the learning of a *mental model*. One important question is how to use the methodologies and theories developed for these simple tasks to explain more complex tasks, such as driving, which involves not only a more difficult motor skill but also a combination and integration of multiple sensory feedbacks and cognitive processes.

Our interest in developing a representative model of the driver is motivated by *smart cars*. The concept of smart cars encompasses a line of research ranging from the lowest to the highest level of interference the system imposes on the driver. Regardless of where researchers stand on the spectrum, there is a common issue that needs to be addressed: driver acceptance and adaptability to smart cars. With warning systems, it is of concern whether drivers find such system useful or distractive; with semi-autonomous systems, predictions of driver behavior are used to decide whether control interventions are necessary [8]; and with autonomous systems, there is a push towards making the vehicle behave closely to human drivers so the sense of *taking over* is less intrusive [9]. Therefore, it is important to formalize a systematic way of modeling the human driver.

As suggested in [1], human drivers can be modeled into a hierarchical scheme with multiple levels of abstraction. For example, at the maneuvering level, the driver makes decisions on which maneuvers to execute (i.e. lane change). This paper focuses on the driver at the lowest control level, where the steering, braking and acceleration inputs can be modeled using continuous dynamics, linear or nonlinear. A large amount of research has focused on modeling the human driver as a linear controller, (see the work in [1] and [10] for a good survey on this topic). It is commonly assumed that the vehicle stays within the linear regions of the tire forces, which is a reasonable assumption only when driving on high friction roads.

In this paper, we address the modeling of driver behavior on low friction roads. In particular, we would like to model extreme/agile maneuvers which include operations not only in the linear but also the nonlinear regions of the tire forces. We record the sequences of inputs of an expert driver performing a 180 degree drifting turn on snow (similar to the trajectory shown in Fig. 1), and discuss the modeling of this behavior via a model predictive control framework and a piecewise-affine system framework. Vehicle states and driver input data is obtained from a passenger car equipped with an Oxford Technical Solution (OTS) RT3002 sensing system, which measures the position and the orientation of the vehicle in the inertial frame, and the vehicle velocities in the vehicle body frame. The test trials are performed in Smithers testing facility in Sault Ste Marie, MI, which provides icy and snowy tracks with low road friction coefficient i.e. $\mu \approx 0.4$.

We collected 5 trials from an expert driver. These data are used in the identification of driver models. We acknowledge that we have limited amount of data available and the proposed driver models need to be validated in our future work with more trials and across different drivers.

The paper is structured as follows. Section II describes the dynamics behind the drift maneuver. Section III presents two methods of modeling the human driver. Section IV discusses how the identified driver model can be incorporated into control frameworks, and Section V concludes the paper.

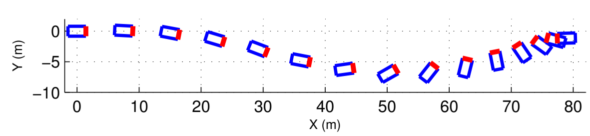

Fig. 1. X-Y trajectory of 180 degree drift maneuver.

This work was funded by the National Science Foundation and Ford Research Laboratories.

[1] {tlin051,fborrelli}@berkeley.edu, Department of Mechanical Engineering, University of California, Berkeley, CA, USA
[2] htseng@ford.com, Ford Research Laboratories, Dearborn, MI, USA

II. Vehicle dynamics

This section briefly discusses the vehicle dynamics during the drifting maneuvers. We use the four wheel vehicle model described in [11] where the six state nonlinear model $\dot{\xi}(t) = f(\xi(t), u(t))$ captures the lateral, longitudinal and yaw dynamics. Fig. 2 illustrates the system model. The state vector ξ is composed of the lateral velocity $\dot{y}$, the longitudinal velocity $\dot{x}$, the yaw angle ψ, the yaw rate $\dot{\psi}$, and the lateral and longitudinal coordinates Y and X. Here $\dot{y}, \dot{x}$ are in the body frame, the rest of the states are in the inertial frame. The input $u = \delta$ is the front steering angle. The tire slip angle $\alpha_\star = \arctan(v_{\star,c}/v_{\star,l})$, shown in Fig. 2, is the angle difference between the velocity vector of the tire and the orientation of the tire. The equations of motion w.r.t. the center of gravity of the vehicle are:

$$m\ddot{y} = -m\dot{x}\dot{\psi} + F_{y_{f,l}} + F_{y_{f,r}} + F_{y_{r,l}} + F_{y_{r,r}} \tag{1a}$$

$$m\ddot{x} = m\dot{y}\dot{\psi} + F_{x_{f,l}} + F_{x_{f,r}} + F_{x_{r,l}} + F_{x_{r,r}} \tag{1b}$$

$$\begin{aligned} I\ddot{\psi} = {} & a(F_{y_{f,l}} + F_{y_{f,r}}) - b(F_{y_{r,l}} + F_{y_{r,r}}) + \\ & + c(-F_{x_{f,l}} + F_{x_{f,r}} - F_{x_{r,l}} + F_{x_{r,r}}) \end{aligned} \tag{1c}$$

$$\dot{Y} = \dot{x}\sin\psi + \dot{y}\cos\psi \tag{1d}$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi \tag{1e}$$

where m is the vehicle's mass, I is the rotational inertial about the yaw axis, a, b and c are distances between the tires and the center of gravity (see Fig. 2). The x and y components of each individual tire forces, $F_{x_{\star,\bullet}}, F_{y_{\star,\bullet}}$ are modeled by using a nonlinear Pacejka tire model [12]. All the parameters required for the Pacejka tire model i.e. road friction coefficient, normal force, tire slip ratio are assumed to be constant, except for α which depends on $\dot{y}, \dot{x}, \dot{\psi}$ and δ [11]. Fig. 3 shows the lateral tire forces F_c as a function of α. Notice that the tire forces can be divided into a *linear region* where the relationship between α and F_c is approximately linear and a *saturated region* where F_c maintains roughly constant even when α is changing. As previously mentioned, many driver models have been developed for maneuvers operating in the linear region. We are interested in incorporating the saturated regions as well, which occur frequently in extreme maneuvers such as drifting.

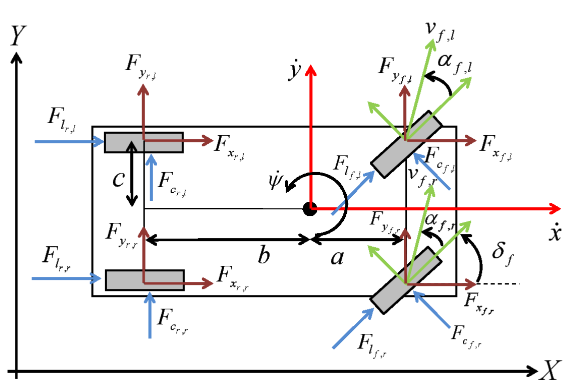

Fig. 2. Four wheel model illustrating the vehicle states.

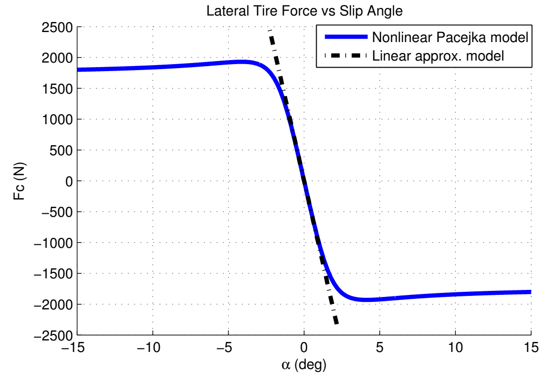

Fig. 3. Lateral tire forces as a function of α.

Aggressive vehicle maneuvers, such as drifting, are performed by saturating the tires and deliberately causing the tires to loose traction in the lateral direction. Rally drivers often perform this maneuver around a curve to quickly align the orientation of the vehicle with the upcoming road [13]. Due to the nonlinear nature of the tire forces, the control sequence which achieves this maneuver is non-trivial and counter-intuitive for a novice driver.

We remark that in order to simplify our analysis and the complexity of the driver model, we have not included load transfer in our system dynamics. In addition, we have also limited the model to consider only the steering input, and have not included braking or acceleration in the model. Therefore, when obtaining the driver data, we asked the expert to perform the 180 degree turn by using only the steering wheel and not to use the brakes/throttles. This is achievable on low-friction roads such as ice/snow. Whereas on high-friction roads, a combination of steering and throttling/braking is required.

III. Driver modeling and identification

The human driver can be parallelled as various types of *controllers* depending on the level of abstraction at interest. In this section, we discuss two possible approaches. First, we treat the overall driver as an optimal controller where judgements are made based on predictions. Then we focus on modeling the driver at the control level, and view the driver as a state feedback linear controller.

A. Optimal control

Optimal control is a systematic way of generating an open-loop set of control sequence that are optimal for a particular cost function. The cost could be a function of the states, and/or control inputs. The predicted effects of control inputs/decisions on future state evolutions are also considered. The state predictions could either be obtained from a look-up table, from motion primitives [14], or from a model of the system dynamics. A model predictive controller (MPC) takes the model of the system dynamics to generate future state predictions. From an autonomous control perspective, MPC can be implemented to avoid obstacles [11] (discussed in the last section of the paper). From a human modeling perspective, learning to behave like an optimal controller involves learning the cost function (i.e. reference trajectory), and also learning either the motion primitives or the system dynamics (mental model) of the activity. We will focus on modeling the driver using the latter abstraction. To formally express this framework for both driver modeling (and for autonomous control [11]), we consider the following optimization problem:

$$\min_{U_t} \quad J_N(\bar{\xi}_t, U_t, \Delta U_t) \tag{2a}$$

$$\text{subj. to} \quad \xi_{k+1,t} = f(\xi_{k,t}, u_{k,t}) \quad k = t,\ldots,t+H_p-1 \tag{2b}$$

$$\Delta u_{k+1,t} = u_{k+1} - u_k \quad k = t,\ldots,t+H_p-2 \tag{2c}$$

$$u_{k,t} \in \mathcal{U} \quad k = t,\ldots,t+H_p-1 \tag{2d}$$

$$\Delta u_{k,t} \in \Delta\mathcal{U} \quad k = t+1,\ldots,t+H_p-1 \tag{2e}$$

$$\xi_{t,t} = \xi(t) \tag{2f}$$

H_p is the prediction horizon. $\bar{\xi}_t = [\xi_{t,t}, \xi_{t+1,t}, \ldots, \xi_{t+H_p-1,t}]$, $\bar{\xi}_t \in \mathbb{R}^{n\times H_p}$ is the sequence of states predicted at time t, based on the Euler discretized dynamics of the vehicle model (2b). $u_{k,t} \in \mathbb{R}^{m_r}$ (m_r is the number of inputs at each time instance) is the k^{th} column of the input sequence matrix $U_t = [u_{t,t}, u_{t+1,t}, \ldots, u_{t+H_p-1,t}]'$. The cost function (2a) is defined as

$$J_N(\bar{\xi}_t, U_t, \Delta U_t) = \sum_{k=t}^{t+H_p-1} \|\xi_{k,t} - \xi_{ref_{k,t}}\|_Q^2 + \|u_{k,t}\|_R^2 + \|\Delta u_{k,t}\|_S^2 \tag{3}$$

i.e. by minimizing a weighted combination of (i) deviations in the reference trajectory, (ii) the control input $u_{k,t}$ and (iii) the rate of change in $u_{k,t}$. Here $\|x\|_P^2 = \|Px\|^2, P = Q, R, S$.

Fig. 4 proposes a framework for modeling the human driver as a MPC controller. The model predictive controller block represents optimization problem (2). It tracks a reference trajectory and requires a model to make predictions. This leads to the following questions. Firstly, what *reference trajectory* is the driver trying to track? Is there some higher level path planner module which generates or selects this reference trajectory from memory? Secondly, how is the driver making *predictions of vehicle states*? In this paper, we propose the driver is predicting state evolutions from the nonlinear model described in section II. But the driver could be using a simpler dynamical model such as the point mass model [11] with a linear tire. This could possibly be the difference between expert and novice driving. The expert driver is able to make better predictions using a nonlinear model, whereas the novice driver is only using the linear model, and would fail to achieve drifting. Comparisons with novice driving data under similar conditions could be done in our future work to validate this hypothesis.

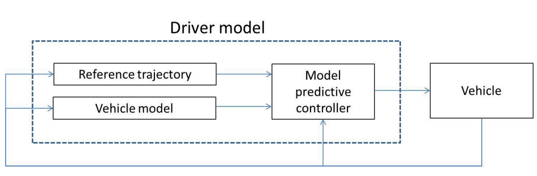

Fig. 4. Driver modeling in a model predictive control framework.

For the drift experiment described in this paper, we take the state trajectory of a successful maneuver which the driver has learned as the reference trajectory, and we model the driver at the control level by treating the driver as a MPC controller. Fig. 5 shows the steering input and the simulated state trajectories of a 180 degree drift turn when the MPC formulation in (2) is solved using the nonlinear solver NPSOL [15]. The sampling time is 50 ms and $H_p = 160$.

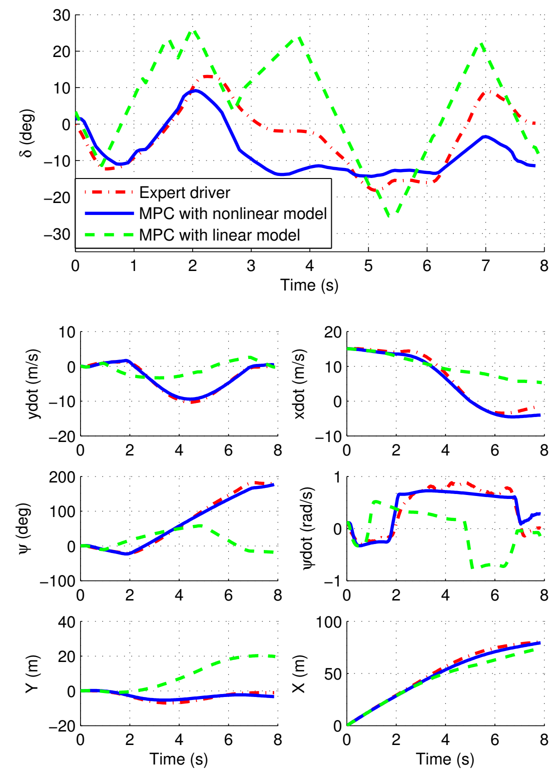

Fig. 5. Comparing expert driver and MPC result of drift maneuver: input trajectory δ and state trajectories $\xi = [\dot{y}, \dot{x}, \psi, \dot{\psi}, Y, X]$.

Fig. 5 indicates that an optimal control framework is suitable in replicating driver behavior. The solid line in Fig. 5 shows the simulated state and input trajectories when the nonlinear Pacejka tire model and (1) is used to make state predictions in (2b) and in the simulation. We can see that the simulated state trajectories tracks the reference trajectories reasonably well, and the input trajectory from the solver has a similar profile with the driver steering. Note that in the MPC formulation, the difference between solver steering and driver steering is not penalized in the cost function, that is why the steering profiles do not exactly match. Yet even though there are large quantitative differences, the qualitative behavior (the general shape/pattern of the steering profile), is quite similar. The quantitative differences are due to model mismatches between the test vehicle and the model used in the solver and the uncertainty in the actual road conditions. A more accurate model would reduce this mismatch.

Fig. 5 also shows that for complex maneuvers which involve the saturation regions of the tires, a high fidelity model is needed in the predictive framework. Most importantly, the nonlinearity in the tire forces needs to be recognized by the driver/solver in order for the drift maneuver to be performed. If the tire forces are treated to be only linear in the tire slip angle, then the driver/solver will not be able to track the reference trajectory that replicates this drifting maneuver. This is shown by the state and input trajectories represented by the dashed line in Fig. 5. The input trajectory is solved when the tire force in (1) is only obtained from a linear tire model (the dashed linear approximation in Fig. 3). The state trajectories are simulated by passing this input profile through the original nonlinear model.

B. Piecewise affine model

We are interested in understanding how human drivers learn a *nonlinear* model of the vehicle, and whether or not drivers make *predictions* of the vehicle trajectory. In situations such as lane keeping, drivers can be modeled as a simple linear controller with a state feedback law [9], which does not have any predictive flavor. As for the drift maneuver, which utilizes saturation regions of the tire forces, modeling the human driver as a simple linear controller will not suffice.

Hybrid modeling involves switching between different modes of operation. Different modes might have different system models and/or control laws. This intuitively lines up with human decision making. Under different conditions, humans have corresponding control strategies. Following common intuitions, we propose to decompose the drifting maneuver into two control strategies:

1. The maneuver starts off with the vehicle traveling in a straight line. In order to *drift*, the controller is responsible for "destabilizing" the vehicle by saturating the tires, and placing the vehicle at a high lateral velocity and a large yaw rate.
2. Once the vehicle is destabilized, the controller needs to stabilize the vehicle again, more importantly, the vehicle has to end in the desired final state, i.e. heading angle of 180 degrees.

The conjecture is that if the human driver uses a switched control strategy in practice, we should be able to observe a similar behavior by modeling drifting by a human driver as a two-mode switched system. Therefore, instead of having a highly nonlinear and complicated mental model of the world, complex models can be broken down into different modes with simpler dynamics within each mode. Next we describe the framework of the piecewise affine (hybrid) model used to identify the human driver.

Formulation: We model driver behavior in input-output form with a special class of hybrid systems, namely, the *PieceWise AutoRegressive eXogenous* (PWARX) formulation, where the switching mechanism is determined by a polyhedral partition of the regressor domain [16]. For fixed model orders n_a and n_b, the regressor r_k is defined in terms of the input vectors $u_{k-i} \in \mathbb{R}^p, i = 0,1,...,n_b$ and past output vectors $y_{k-i} \in \mathbb{R}^q, i = 1,...,n_a$,

$$r_k = [y_{k-1}^T \; ... \; y_{k-n_a}^T \; u_k^T \; u_{k-1}^T \; ... \; u_{k-n_b}^T]^T \tag{4}$$

The current output y_k is expressed as a piecewise affine function of r_k,

$$y_k = \theta_{\sigma(k)}^T \begin{bmatrix} r_k \\ 1 \end{bmatrix} \tag{5}$$

where $\sigma(k)(\in \{1,...,s\})$, is the discrete state, s is the number of modes. $\theta_i (i = 1,...,s)$ is the matrix of parameters defining the system dynamics in each mode. The domain of the discrete modes are determined by a polyhedral partition of the regressor domain $\mathcal{R} \subset \mathbb{R}^d$, where $d = q \times n_a + p \times (n_b + 1)$. The discrete state $\sigma(k)$ is given by

$$\sigma(k) = i \quad \text{iff} \quad r_k \in \mathcal{R}_i \quad i = 1,...,s \tag{6}$$

and $\{\mathcal{R}_i\}_{i=1}^s$ is a complete partition of $\mathcal{R}$. Each region $\mathcal{R}_i$ is a convex polyhedron described by

$$\mathcal{R}_i = \{r \in \mathbb{R}^d : H_i \begin{bmatrix} r \\ 1 \end{bmatrix} \leq 0\} \tag{7}$$

Model identification: To identify the model of the human driver, input and output information are recorded during each trial and the parameters θ_i are identified for each mode. In this paper, we implemented the cluster-based procedure [17]. For comparison of this algorithm with other hybrid identification techniques, readers can refer to [16].

For the drifting scenario, we are interested in making predictions of the driver's steering based on state information. The hybrid model's output vector is the steering angle, $y_k = [\delta_k]$, and the input vector is defined from the vehicle states, $u_k = [\psi_k, \dot{\psi}_k, Y_k, \alpha_{f_k}, \alpha_{r_k}]^T$.

Using the PWARX formulation, we identify the parameters θ_i, predict the steering profile $\hat{y}_k = \theta_{\sigma(k)}^T [\hat{y}_{k-1}^T, u_k^T, 1]^T$, compare $\hat{y}_k$ with the actual driver steering y_k and analyze the effectiveness of introducing additional modes in the hybrid system. Three sets of models differing in the number of modes ($s = 1,2,3$) are identified. The maximum prediction error, $e = \|y_k - \hat{y}_k\|_\infty$, for each model is denoted respectively as e_1, e_2 and e_3 and presented in Fig. 6.

The one-mode system relates to a single linear controller. For the one-mode hybrid system, the prediction error is comparatively higher. This confirms our previously discussed conjecture that in order to drift, the driver has to understand the nonlinearity involved, and therefore a simple linear control strategy will not suffice. Notice that e_1 in trial 1 is smaller than subsequent trials. State trajectories for trial 1 shows that the driver failed to achieve the maneuver. This indicates that in trial 1, the driver is likely to be still using a single-mode strategy and only learns to switch between modes through subsequent trials.

Comparing e_2 and e_3, it can be seen that, only a very small improvement is obtained when an additional mode is added. This confirms our conjecture that it is sufficient to decompose the drifting maneuver into two modes of control.

Fig. 7 compares the actual steering angle with the predicted steering angle from the two-mode hybrid model. Notice that the state evolution began in mode 1, switched to mode 2 at $t = 4.8s$, and stayed in mode 2 without any further switching. This confirms our hypothesis of the decomposition of the control strategy in the drifting maneuver.

θ_1, the parameter of mode 1 of a two-mode model, obtained from model fitting of trial 5 data is validated on two other trials. Keeping θ constant means we are not allowing the predicted driver to switch his strategy. Fig. 8 shows the comparison between the recorded steering profiles and that predicted from θ_1. We can see that the first half of the steering profile makes very good predictions but the second half, where the driver should be switching modes, fits poorly. The results of our model validation conforms with the hypothesis of a two-mode switching control strategy.

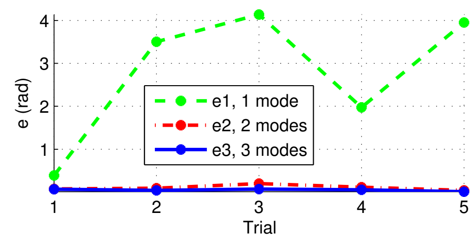

Fig. 6. Maximum prediction error for 1, 2 and 3 mode hybrid models.

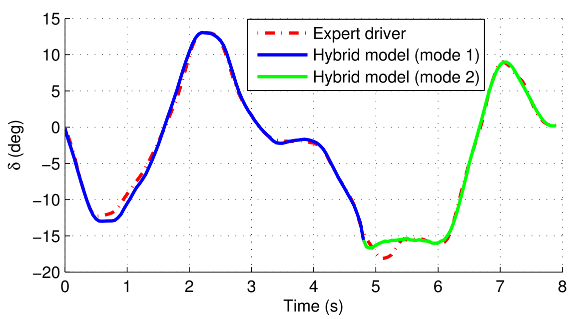

Fig. 7. Predicted vs actual expert driver steering angle for drift maneuver.

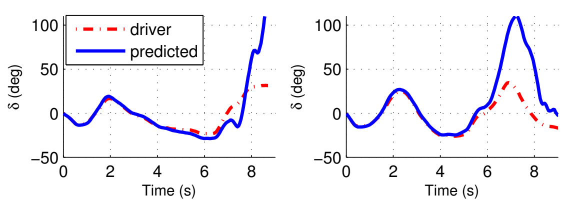

Fig. 8. Model validation: θ_1 from trial 5 validated on data of trial 2 and 3.

IV. Incorporating Driver Model into Control Frameworks

As mentioned in the introduction, it is of interest to incorporate human driver models into the design of autonomous and semi-autonomous controllers. In this section, we investigate how the driver models can be applied for intelligent control. Often, autonomous controllers will work well to perform certain tasks such as lane-keeping or obstacle avoidance without the aid of a driver model. For example, a nonlinear solver can be used to find a solution to an optimal control problem where the obstacle is represented as either a cost term or a constraint [11]. Again, the solutions do not need to utilize the saturated regions of the tires. For maneuvers which involve the saturated regions, such as drifting or last minute obstacle avoidance on slippery surfaces, the solutions would live closer to the unstable equilibrium points of the system dynamics [18]. It would be much harder for the nonlinear solver to find such a solution.

Autonomous system: In Section III-A, we formulated a MPC problem and discussed how it can be used to parallel driver behavior. From a control perspective, we can also view this MPC formulation as an autonomous controller capable of performing extreme maneuvers. In this first formulation, the optimization variables are the control variable $u_{k,t}$ and are only constrained by the polytopic bounds on $u_{k,t}$ and $\Delta u_{k,t}$. Here we include the following two-mode switched linear model,

$$u_{k,t} = \theta^T_{\sigma(k,t)} \begin{bmatrix} r_{k,t} \\ 1 \end{bmatrix} \tag{8a}$$

$$\sigma(k,t) = \begin{cases} 1 & \text{if } k < T_{sw} \\ 2 & \text{if } k \geq T_{sw} \end{cases} \tag{8b}$$

as an additional constraint to the MPC problem, and check if the references can still be tracked when this two-mode switching control strategy is embedded into the optimization problem. The optimization variables of the new MPC are now the feedback gains θ_1 and θ_2. Note that since the results of the hybrid model fitting showed that the switching only occurs once in the trajectory, we will simplify the mode switching condition to be dependent on the time index k instead of the state as in (7).

Similar to the results presented in section III-A, the new MPC formulation is solved using NPSOL with a sampling time of 50 ms and $H_p = 160$ steps. The reference trajectory comes from recorded expert driving data. Note T_{sw} could be posed as an optimization parameter, but to avoid having to solve a mixed-integer program, we fixed T_{sw} and tried a few values. For the following results, we have fixed $T_{sw} = 83$.

Fig. 9 shows the state and steering trajectories predicted by the optimal control problem. The state trajectories verify that even with the additional state feedback constraint (8a), the solver is still able to find a feasible solution to track the reference trajectories. In addition, the general shape of the input trajectory also conforms with the steering pattern of initializing drift and countersteering.

Semi-autonomous system: In the final section of this paper, we propose ways to incorporate predictions of the driver input into semi-autonomous control systems. The predicted driver input $\hat{u}_{k,t}$ is a function of the current time index, the vehicle states and the past inputs. In Section III, $\hat{u}_{k,t}$ can be an open loop sequence (solutions to the MPC), or in the form of a state feedback law. For a semi-autonomous control system, we are interested in minimizing controller intervention. Therefore, we pose a similar MPC problem to (2) and express the control variable as $u_{k,t} = \hat{u}_{k,t} + \tilde{u}_{k,t}$, where $\tilde{u}_{k,t}$ is the optimization variable to the MPC which measures how much the controller intervenes. By adding this

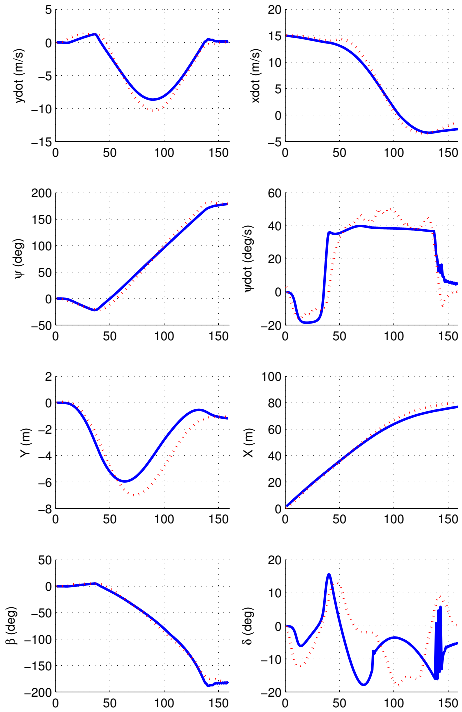

Fig. 9. State and input trajectory: NPSOL solution of new optimization formulation (blue solid line) vs expert driver trajectories (red dotted line).

intervention term in the cost function, the nonlinear solver will find a solution which minimizes controller intervention while still tracking the desired trajectory and satisfying state and input constraints.

Fig. 10 shows the controller intervention $\tilde{u}$ for three different MPC formulations. The solid line represents the formulation where $\tilde{u}$ is not penalized in the cost function. The dashed and dashed-dotted lines represents the formulation where $\tilde{u}$ is penalized. The dash-dotted line represents the formulation where predicted driver steering is taken from the open loop sequence, which in this case is just the recorded driver inputs from trial 5. In general, this open loop prediction could be obtained from any higher level module which computes a sequence of driver predictions. The dashed line represents the formulation where driver steering is predicted in closed-loop with the state predictions inside the optimization problem using the hybrid model identified in section III-B. Fig. 10 shows that $\tilde{u}$ for the latter two formulations are smaller. We remark that there is a tradeoff between controller intervention and tracking error and is highly dependent on the weights between the respective terms in the cost function.

V. Conclusions & Future Work

We have shown through system identification on real driving data, that extreme maneuvers performed by an expert driver can be modeled at the lowest control level as a MPC controller or a two-mode switched system. In the MPC framework, the reference trajectory and the vehicle model are important issues that need to be considered. In the hybrid framework, we showed that since we need to consider the

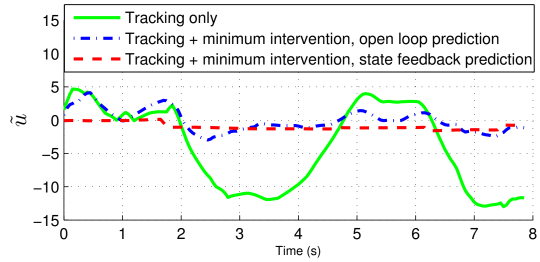

Fig. 10. Controller intervention $\tilde{u}$: difference between controller input and predicted driver input. Predictions are open loop or through state feedback.

saturation regions of the tire, it is not sufficient to model the human as a one-mode linear feedback controller. We have proposed and implemented how predictions of driver steering could be incorporated into an autonomous controller.

For future work, we plan to model the higher level modules which the driver uses in generating the reference trajectory. More driving data with different drivers will be collected in order to better validate the models, and used to parameterize the different characteristics of the driver model as a function of driver skills/habits.

References

[1] C. Cacciabue, Modelling driver behaviour in automotive environments, Springer, 2007
[2] E. Todorov, Optimality principles in sensorimotor control. Nature Neuroscience, 2004, 7, 907-915.
[3] P. Abbeel, and A.Y. Ng, Apprenticeship learning via inverse reinforcement learning. Proc. 21st Int. Conf. on Machine Learning, 2004, pp.1-8
[4] K. P. Kording, D. M. Wolpert, Bayesian decision theory in sensorimotor control. Trends Cogn. Sci. DOI:10.1016/j.tics.2006.05.003
[5] Z. Kong, and B. Mettler, Interaction patterns in agent-environment dynamics and formal modeling of human guidance behavior, AEM Report 2012-1.
[6] D. D. Salvucci, Modeling driver behavior in a cognitive architecture, Human Factors, 2006, 48(2), pp. 362-380.
[7] R. Shadmehr, and S. P. Wise, The computational neurobiology of reaching and pointing, The MIT Press, 2005.
[8] R. Vasudevan, V. Shia, Y. Gao, R. Cervera-Navarro, R. Bajcsy, and F. Borrelli, Safe semi-autonomous control with enhanced driver modeling, American Controls Conference, 2012.
[9] A. Gray, M. Ali, Y. Gao, J. K. Hedrick, and F. Borrelli, Integrated threat assessment and control design for roadway departure avoidance, Intelligent Transportation Systems Conference, 2012.
[10] C. C. MacAdam, Understanding and modelling the human driver, Vehicle System Dynamics., 2003, 40, pp. 101-134.
[11] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads, Dynamic Systems and Control Conference, 2010.
[12] E. Bakker, L. Nyborg, and H. B. Pacejka, Tyre modeling for use in vehicle dynamics studies, 1987, SAE paper No. 870421.
[13] E. Velenis, P. Tsiotras, and J. Lu, Modeling aggressive maneuvers on loose surfaces: the cases of trail-braking and pendulum-turn, European Control Conference, 2007.
[14] A. Gray, Y. Gao, T. Lin, J. K. Hedrick, E. Tseng, and F. Borrelli, Predictive control for agile semi-autonomous ground vehicles using motion primitives, American Control Conference, 2012.
[15] P. Gill, W. Murray, M. Saunders, and M. Wright, NPSOL Nonlinear Programming Software. Stanford Business Software, Inc., Mountain View, CA, 1998.
[16] S. Paoletti, A. Lj. Juloski, G. Ferrari-Trecate, and R. Vidal, Identification of hybrid systems: A tutorial. Eur. J. Control, 2007, 513, 242 260.
[17] G. Ferrari-Trecate, Hybrid Identification Toolbox (HIT), 2005, www-rocq.inria.fr/who/Giancarlo.Ferrari-Trecate/HIT_toolbox.html
[18] R. Y. Hindiyeh, and J. C. Gerdes, Equilibrium analysis of drifting vehicles for control design, Dyn. Sys. and Controls Conference, 2009.