Automated Driving

The Role of Forecasts and Uncertainty - A Control Perspective

Ashwin Carvalho^{a,*}, Stéphanie Lefèvre^a, Georg Schildbach^a, Jason Kong^a, Francesco Borrelli^a

^a*Department of Mechanical Engineering, University of California Berkeley, USA*

Abstract

Driving requires forecasts. Forecasted movements of objects in the driving scene are uncertain. Inevitably, decision and control algorithms for autonomous driving need to cope with such uncertain forecasts. In assisted driving, the uncertainty in the human/vehicle interaction further increases the complexity of the control design task.

Our research over the past ten years has focused on control design methods which systematically handle uncertain forecasts for autonomous and semi-autonomous vehicles. This article presents an overview of our findings and discusses relevant aspects of our recent results.

1. Introduction

Two assumptions facilitate the control design for autonomous cars. First, the driver does not have control of the vehicle. Second, there is precise knowledge of the ego vehicle's location, and current and future locations of objects around the ego vehicle. If both assumptions are satisfied, control design for autonomous driving can be addressed with basic control techniques. Extensive tuning and switching between local controllers is necessary to handle heterogeneous driving scenarios. With high probability, mass-production autonomous driving in the next years will happen in scenarios satisfying both assumptions, such as specific maneuvers on dedicated lanes.

This article focuses on the challenges of control design when one or both assumptions cannot be satisfied, i.e., there is interaction between the autonomous control systems and the driver (we refer to this as semi-autonomous driving or assisted driving), and/or there is high uncertainty in the forecasted locations of objects in the driving scene. Driver manuals around the world always warn drivers of wrong predictions. As an example, the driving rules in Australia require the driver to look around for “children (who) are small and can be unpredictable” (Government of South Australia, 2015). The New Jersey's driver manual (State of New Jersey Motor Vehicle Comission, 2015) classifies “hazards .. into three groups” with one of them being “pedestrians

*Corresponding author

Email address: ashwinc@berkeley.edu (Ashwin Carvalho)

and animals” which “are characterized by unpredictability and quick movement.”. Undoubtedly, driving requires forecasts, and forecasts can be highly uncertain in some driving scenarios. The simplest way to deal with uncertain forecasts in autonomous driving is to reduce the vehicle speed and wait until the uncertainty becomes negligible. In general, this might not be always possible (for instance when driving at high speed on a highway), may generate deadlock scenarios (waiting forever at an intersection) and is not the preferred driving mode of the vast majority of the drivers. For such scenarios, control design for assisted or autonomous driving is a real challenge.

Our research over the past ten years has focused on control design methods which systematically integrate the driver and the autonomous vehicle controller, and systematically handle uncertain forecasts. This paper presents a unified and systematic way of addressing uncertainty in intelligent vehicles. We will present a framework where uncertainty is introduced in the modeling phase and possibly learned from data. The uncertain models are used by a stochastic Model Predictive Controller (MPC) for safe and comfortable assisted or autonomous driving. The goal is to provide an overview of our findings, point to relevant literature and discuss relevant aspects of our recent results.

The paper starts with a motivating example in Section 2, discusses modeling in Section 3 and control design in Section 4, and presents simulation and experimental results in Section 5. The modeling section addresses separately the ego vehicle model (also referred to as the controlled vehicle), the driver model and the environment model. The control design section focuses on computationally tractable approaches for guaranteeing safe driving under uncertain forecasts.

1.1. Notation

Throughout this paper, the superscripts v , d and e are used to denote quantities corresponding to the vehicle, driver and environment, respectively.

The subscripts k and t represent time steps. That is, $k + 1$ denotes the time instant following k . The sampling times Δt^v , Δt^d and Δt^e are allowed to be different from each other and to vary with time. In the most general case, the dynamics of the vehicle, driver and environment subsystems are expressed in the belief space, that is, the space of probability distributions. $x_k^\star$ and $u_k^\star$ denote the belief state (discrete or continuous) and the control input, respectively, at time k ($\star \in \{v, d, e\}$). Uncertainty in the dynamics is represented by the disturbance vector $d_k^\star$.

At any time instant t , the predicted value of the variable z at time $t + k$ is denoted by $z_{k|t}$. For state variables, the measured or estimated value at time t is denoted by x_t . For input variables, the actuated value at time t is denoted by u_t .

1.2. System Architecture

Figure 1 shows a schematic of the intelligent vehicle architecture which will be used throughout this paper. The *Map-based Localization and Perception* module combines information received from sensors and digital maps to estimate relevant states representing the driving scenario. Here, x_t^v denotes the estimated state of the ego vehicle at time t , and x_t^e denotes the estimated state of the environment at time t (e.g., positions of other vehicles and pedestrians, road geometry).

Figure 1: Intelligent vehicle system overview

The *Planning and Control* module takes as an input the estimated states of the ego vehicle and the environment, and generates control inputs to the vehicle u_t^v . This task is performed by three subsystems, which will be described next. At time t , the *Reference Generation* module computes a sequence of reference states $\{x_{k|t}^{\text{ref}}\}_{k=0}^{T-1}$ and/or a sequence of reference control inputs $\{u_{k|t}^d\}_{k=0}^{T-1}$, where T is the prediction horizon of the controller. Reference states are generated by a path planner, while reference inputs are generated by a driver model. For driving assistance applications, the driver model predicts the most likely control inputs that the driver will apply over the prediction horizon T . For autonomous driving applications, the driver model provides the controller with inputs learned from human driving which can be used as a reference over the prediction horizon T . This can be useful in designing autonomous cars with personalized driving styles. The *Environment Prediction* module predicts the future state of the environment over the prediction horizon T , generating a sequence $\{x_{k|t}^e\}_{k=1}^T$.

At each sampling time instant, the *MPC-based Controller* computes an optimal sequence $\{u_{k|t}^c\}_{k=0}^{T-1}$ of control inputs, by solving a Constrained Finite-Time Optimal Control (CFTOC) problem. This input sequence (called the *open-loop input sequence*), combined with the driver inputs $\{u_{k|t}^d\}_{k=0}^{T-1}$ in the case of assisted driving, is used to generate a sequence of commands $\{u_{k|t}^v = f(u_{k|t}^d, u_{k|t}^c)\}_{k=0}^{T-1}$ to the vehicle. The function $f(\cdot, \cdot)$ is a design choice and, in the simplest case, equals $u_{k|t}^c$ for autonomous driving and the sum of $u_{k|t}^c$ and $u_{k|t}^d$ for assisted driving. In general, while the driver can only

actuate the brake and gas pedals, and shift gears, the control inputs u^c can include individual braking torques at the four wheels, torque vectoring and active suspension.

The CFTOC problem is solved as a numerical optimization program at each time step. It can directly include actuator constraints and safe regions for the vehicle in its environment. Based on a model of the vehicle, the CFTOC problem implicitly computes predictions for the vehicle state $\{x_{k|t}^v\}_{k=0}^T$ (called the *open-loop state sequence*). Only the first control input $u_t^v = u_{0|t}^v$ is actually applied to the system, and the computations are repeated at the next time step in a receding horizon fashion.

The *Low-level Actuation* module transforms the commanded inputs u_t^v into actuator commands. In the test vehicle used for the results presented in this paper, longitudinal and lateral motion control is achieved via the on-board Adaptive Cruise Control (ACC) and Motor-Driven Power Steering (MDPS) systems, respectively. The aforementioned systems require the desired longitudinal acceleration and steering angle as inputs, respectively. Therefore, in applications involving both longitudinal and lateral control, the input commanded by the *Planning and Control* module takes the form $u_t^v = [\delta_t, a_t]$, where δ is the desired steering angle and a is the desired longitudinal acceleration. In applications where only the longitudinal motion is controlled, the input to the vehicle takes the form $u_t^v = a_t$.

2. Model Predictive Control for Active Safety: An Illustrative Example

The objective of this section is to motivate the use of forecasts in designing controllers for intelligent vehicles. We present the design of a linear MPC for an emergency brake assist system. The goal is to track the driver's intended acceleration and intervene only when a violation of the preset safety distance to the vehicle in front is anticipated. As the focus is on motivating the usefulness of predictive control approaches, the simplified example assumes that there is no uncertainty in the system. That is, the future positions of objects in the environment and the future inputs of the driver are known without uncertainty. In addition, the driver controls the steering such that the vehicle stays in the center of its lane.

The longitudinal motion of the vehicle is described by a double integrator model, where the control input is the longitudinal acceleration. We assume no mismatch between the simulated vehicle model and the one used in the predictive controller design. A learning-based approach is used to predict the driver's acceleration inputs $\{u_{k|t}^d\}_{k=0}^{T-1}$ over the prediction horizon T (the approach will be described in detail in Section 3.2).

Figure 2: Snapshots of simulation with the ego vehicle E in blue and the target vehicle V in red. The red line depicts the trajectory of the target vehicle, which is assumed to be known a-priori.

At each time step t , the predictive controller solves the following CFTOC problem,

$$\min_{u_{0|t}^c, \dots, u_{T-1|t}^c} \sum_{k=0}^{T-1} (u_{k|t}^v - u_{k|t}^d)^2 \quad (1a)$$

$$\text{s.t. } x_{k+1|t}^v = Ax_{k|t}^v + Bu_{k|t}^v \quad (1b)$$

$$x_{k+1|t}^v \leq x_{k+1|t}^{\max} \quad (1c)$$

$$u^{\min} \leq u_{k|t}^v \leq u^{\max} \quad (1d)$$

$$u_{k|t}^v = u_{k|t}^c + u_{k|t}^d \quad (1e)$$

$$(k = 0, \dots, T-1)$$

$$x_{0|t}^v = x_t^v, \quad (1f)$$

and implements the optimal input $u_t^v = u_{0|t}^{c,\star} + u_{0|t}^d$ to the double integrator system $x_{t+1}^v = Ax_t^v + Bu_t^v$, where

$$A = \begin{bmatrix} 1 & \Delta t^v \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \Delta t^v \end{bmatrix}. \quad (2)$$

The cost function in (1a) penalizes the deviation between the vehicle inputs $u_{k|t}^v$ and the predicted inputs $u_{k|t}^d$ given by the driver model. $x_{k|t}^{\max}$ in (1c) is the upper bound on the position and velocity due to the vehicle in front and the speed limit, respectively. (1d) represents actuator limits, and (1f) is the initial condition for the state.

Two snapshots of the scenario for the simulations are shown in Figure 2. The ego vehicle E is moving in the left lane with a slower vehicle V (also referred to as the target vehicle) in the neighboring lane. As E approaches V , the latter changes lanes into its path, requiring a response from the driver.

The acceleration inputs applied by the driver and controller are shown in Figure 3. Initially, the driver accelerates as there is no vehicle ahead, and the controller matches the driver's input exactly. When V changes lanes into the path of E at $t = 4.7$ s, the driver begins to brake. However, the controller anticipates a violation of the safety distance to the vehicle in front and applies a higher value of braking than the driver.

Figure 3: Driver and controller inputs during the simulation. The controller tracks the driver's acceleration until a violation of the safety distance is predicted around $t = 5$ s. The controller applies harder braking to keep the vehicle safe. The difference between the driver and controller inputs is shown by the black line.

This is seen by the discrepancy between the driver and the controller's inputs around $t = 5$ s in Figure 3. The difference between the two inputs is depicted by the black line.

The open-loop predictions of the relative distance to the vehicle in front at $t = 6$ s are shown in Figure 4. If the controller had not intervened, the driver's inputs would have caused the vehicle to violate the specified safety distance of 4 m at $t = 7.2$ s. Figure 5 shows that the controller does not allow the relative distance between the ego and target vehicles to go below the specified safety distance during the simulation.

In summary, the above example illustrates the use of predictions of the driver's inputs and the target vehicle's positions to compute safe control actions for the ego vehicle. However, two important elements are not considered by this example. Firstly, the vehicle model in (1) is oversimplified and does not capture the non-linear vehicle dynamics as well as the tire-road interaction. The use of a higher fidelity vehicle model would increase the complexity of the on-line optimization problem. This trade-off is discussed further in the next section, which presents vehicle models that are suitable for real-time MPC.

The second element not considered is the uncertainty in the measured and predicted values of the vehicle, driver and environment states. The ability of the controller to prevent a collision in the above example relies on perfect knowledge of the vehicle states, driver's input and position of the target vehicle. These assumptions are not realistic and can easily lead to unsafe control actions. In the next section, we will present models of the vehicle, driver and environment which include a description of the uncertainty in terms of set bounds or Probability Density Functions (PDFs). This requires MPC design methodologies that account for uncertainty without being overly conservative. This will be addressed in the following sections.

Figure 4: Open-loop predictions of the relative distance between the target and controlled vehicles at $t = 6$ s. The red line depicts what would have happened if the controller did not intervene. The controller plans to brake harder than the driver to prevent the relative distance going below the specified safety distance of 4 m.

Figure 5: Relative distance between target and controlled vehicles. The red circle at $t = 4.8$ s indicates the instant at which the target vehicle is detected in front of the ego vehicle.

3. System Modeling

Predictive control schemes rely on models of system components to predict its evolution in response to control inputs and exogenous disturbances. The challenge in choosing abstractions useful for control design is the trade-off between accuracy and complexity. Models that accurately capture the dynamics of the system tend to yield computationally intensive algorithms. This section provides an overview of models of the three core components of the intelligent vehicle architecture - the vehicle, the driver, and the environment - with a focus on real-time implementation. The formulation of the safety constraints for the MPC-based controller design is also discussed.

3.1. Vehicle Model

In the context of path planning and control of autonomous vehicles, the goal of the vehicle model is to provide a relationship between physical inputs and the position and orientation of the vehicle. Vehicle models can be broadly classified, in an increasing order of complexity, into three categories:

1. *Point-mass models* treat the vehicle as a particle. Such models yield large tracking errors when used for path planning due to their inability to account for dynamic feasibility (Gao et al., 2010).
2. *Kinematic models* are a function of vehicle geometry, and can represent the vehicle motion in a range of conditions which does not involve highly dynamic maneuvers and/or tire force saturation (Kong et al., 2015).
3. *Dynamic models* rely on tire models to describe the interaction between the vehicle and the road. In this case, the complexity arises from the non-linear relationship between the tire forces and the vehicle states and inputs.

This section presents bicycle models of the vehicle dynamics, wherein the left and right wheels are represented by a single wheel. This approximation is a common approach in developing models suitable for control design (see e.g. Rajamani (2006); Gray et al. (2012); Gao et al. (2014); Carvalho et al. (2014)).

The discrete-time state update equations corresponding to the vehicle model are compactly written as

$$x_{k+1}^v = f^v(x_k^v, u_k^v, d_k^v), \quad (3)$$

where x_k^v and u_k^v are the vehicle state and input vectors, respectively. d_k^v denotes the disturbance vector, which is introduced to represent parametric uncertainty and modeling errors. The following subsections describe the kinematic and dynamic bicycle models in more detail. The notation used is depicted in Figure 6.

Kinematic bicycle model

The following set of differential equations describe the motion of the vehicle in the inertial frame shown in Figure 6:

$$\dot{X} = v \cos(\psi + \beta) \quad (4a)$$

$$\dot{Y} = v \sin(\psi + \beta) \quad (4b)$$

Figure 6: Notation used in the kinematic and dynamic bicycle models.

$$\dot{\psi} = \frac{v}{l_r} \sin(\beta) \quad (4c)$$

$$\dot{v} = a \quad (4d)$$

$$\beta = \tan^{-1} \left(\frac{l_r}{l_f + l_r} \tan(\delta) \right), \quad (4e)$$

where X and Y denote the inertial position coordinates, and ψ denotes the inertial heading angle. v and a denote the velocity and acceleration of the center of gravity, respectively. β denotes the angle between the velocity vector and the longitudinal axis, and δ denotes the front wheel steering angle with respect to the road. l_f and l_r denote the distances from the vehicle's center of gravity to the front and rear axles, respectively.

The discrete-time model (3) is obtained from (4) by Euler's method. The state and input vectors for the kinematic bicycle model are defined as $x_t^v = [X_t, Y_t, \psi_t, v_t]$ and $u_t^v = [\delta_t, a_t]$, respectively.

Dynamic bicycle model

The inertial position coordinates and heading angle in the dynamic bicycle model are defined in the same manner as those in the kinematic bicycle model. The differential equations in this case are given by,

$$\ddot{x} = \dot{\psi} \dot{y} + a \quad (5a)$$

$$\ddot{y} = -\dot{\psi} \dot{x} + \frac{2}{m} (F_{c,f} \cos \delta + F_{c,r}) \quad (5b)$$

$$\ddot{\psi} = \frac{2}{I_z} (l_f F_{c,f} - l_r F_{c,r}) \quad (5c)$$

$$\dot{X} = \dot{x} \cos \psi - \dot{y} \sin \psi \quad (5d)$$

$$\dot{Y} = \dot{x} \sin \psi + \dot{y} \cos \psi, \quad (5e)$$

where $\dot{x}$ and $\dot{y}$ denote the longitudinal and lateral speeds in the body frame, respectively and $\dot{\psi}$ denotes the yaw rate. m and I_z denote the vehicle's mass and yaw inertia,

respectively. $F_{c,f}$ and $F_{c,r}$ denote the lateral tire forces at the front and rear wheels, respectively, in coordinate frames aligned with the wheels.

The relationship between the lateral tire forces and vehicle states is described by tire models of the form

$$F_{c,i} = h(\alpha_i, \sigma_i, F_{z,i}, \mu_i), \quad (6)$$

where the index $i \in \{f, r\}$ distinguishes between the front and rear wheels. α_i and σ_i denote the tire slip angle and slip ratio, respectively, $F_{z,i}$ is the normal force and μ_i is the friction coefficient between the tire and the road surface. The slip angles α_i can be approximated in terms of the vehicle states and inputs as,

$$\alpha_f = -\delta + \tan^{-1} \left(\frac{\dot{y} + l_f \dot{\psi}}{\dot{x}} \right) \quad (7a)$$

$$\alpha_r = \tan^{-1} \left(\frac{\dot{y} - l_r \dot{\psi}}{\dot{x}} \right). \quad (7b)$$

Several approaches for modeling tire behavior have been presented in the literature. The Pacejka Magic Formula is a semi-empirical model commonly used in high-fidelity vehicle simulation models (Pacejka, 2005). Its main limitations are the large number of parameters to be identified and the black-box nature of the model. The Fiala tire model is more suited for control design due to its explicit representation involving few parameters (Hindiye and Gerdes, 2009; Carvalho et al., 2013). The linear tire model, wherein $h(\cdot)$ in (6) is a linear function of the slip angle α , is a common approach used in automotive active safety applications. Figure 7 shows the variation of lateral tire force with slip angle for the aforementioned tire models, for a given slip ratio, normal force and friction coefficient. The gray dots depict the lateral force values estimated from data collected during experiments on the winding road shown in Figure 9 and a ‘figure 8’ drift maneuver. The speed of the vehicle ranged from 10 to 35 m/s during these experiments. In theory, the lateral force vs slip angle curves vary throughout the tests due to changing wheel loads and slip ratios. These effects are neglected for the purpose of control design in this paper.

As in the case of the kinematic model, the discrete-time representation (3) is obtained by an Euler discretization of (5). The state and input vectors for the dynamic bicycle model are defined as $x_t^v = [\dot{x}_t, \dot{y}_t, \dot{\psi}_t, X_t, Y_t, \psi_t]$ and $u_t^v = [\delta_t, a_t]$, respectively.

Road-aligned coordinate frame

A simple transformation allows us to express the position and orientation of the vehicle in a coordinate frame aligned with the road or lane centerline (Rajamani, 2006). The vector of position and orientation coordinates at time t is given by

$$[\xi_t, \eta_t, \phi_t], \quad (8)$$

where ξ is the longitudinal position of the vehicle along the road, and η and ϕ denote the lateral position error and angular error with respect to the lane centerline, respectively. The notation is depicted in Figure 8.

Figure 7: Variation of lateral tire force with slip angle using different tire modeling approaches. The gray dots indicate the values estimated during a test on the winding road shown in Figure 9 and a 'figure 8' drift maneuver. The vehicle speed ranged from 10 to 35 m/s.

Figure 8: Position and orientation in road-aligned frame.

The modified state vector for the kinematic bicycle model is $x_t^v = [\xi_t, \eta_t, \phi_t, v_t]$, and that for the dynamic bicycle model is $x_t^v = [\dot{x}_t, \dot{y}_t, \dot{\psi}_t, \xi_t, \eta_t, \phi_t]$. The differential equations corresponding to the position and orientation in (4) and (5) can be modified accordingly (see e.g. Gao et al. (2012); Carvalho et al. (2013)). This transformation is useful in lane-keeping applications as the position reference becomes $\eta^{\text{ref}} = 0$ and lane boundaries can be expressed as simple bounds on the state η . Moreover, under the assumptions of constant speed, low road curvature and a linear tire model, the resulting dynamic bicycle model in the road-aligned frame is linear (Rajamani, 2006).

Comments on vehicle models

We use data collected from our experimental vehicle in order to compare the accuracy of the vehicle models presented in this section in normal driving conditions. The vehicle was driven on the winding track at Hyundai's California Proving Grounds depicted in Figure 9. The winding track is a road course with many turns and straight

Figure 9: Map of the winding track at Hyundai’s California Proving Grounds.

line paths. We performed two experiments. During Experiment 1, the average speed was 14.1 m/s with a standard deviation of 2.1 m/s. The maximum speed was 17.2 m/s. Experiment 2 was performed at higher speeds, and hence, more aggressive driving conditions. The average speed during the test was 19.4 m/s with a standard deviation of 4.1 m/s. The maximum speed was 26.2 m/s.

The accuracy of the kinematic bicycle model versus the dynamic bicycle model is analyzed as follows. At each time step, a forward simulation starting with the measured state and using the measured inputs is used to compute the predicted states over the next four time steps using either a kinematic or dynamic bicycle model discretized at 0.2 s. The *open-loop error* of a particular model is defined as the difference between the measured states and the predicted states obtained by that model. The percentage distributions of the open-loop position errors of the kinematic and dynamic bicycle models on the winding track are shown in Figure 10. Note that the dynamic bicycle model in the above analysis uses a linear tire model.

Figure 10a shows that both models perform similarly well in predicting the states of the vehicle at moderate speeds such as those in Experiment 1. However, the open-loop errors increase more rapidly with the prediction time step in the case of the kinematic model than in the case of the dynamic model. This difference is more pronounced in Experiment 2, which involves more aggressive driving maneuvers. In general, dynamic models are able to capture the vehicle dynamics in high tire slip conditions (Hindiye, 2013). Emergency collision avoidance maneuvers on low-friction surfaces such as snow in Carvalho et al. (2013) show that the tire slip angles during such maneuvers are close to, and often exceed, their saturation limits.

A drawback of dynamic vehicle models which include tire models is that they become singular at zero velocity. This is due to the longitudinal speed $\dot{x}$ appearing in

(a) Experiment 1: Mean speed = 14.1 m/s, Maximum speed = 17.2 m/s

(b) Experiment 2: Mean speed = 19.4 m/s, Maximum speed = 26.2 m/s

Figure 10: Percentage distributions of the open-loop position errors of the kinematic and dynamic bicycle models at time steps of 0.2 s. The mean and standard deviation of these errors are denoted by μ and σ , respectively. The subscripts K and D refer to the kinematic model and the dynamic model, respectively.

the denominator of the slip angle estimates in (7). Transient tire models resolve this issue, but increase the complexity of the vehicle model (Pacejka, 2005). Kinematic models, however, provide a more accurate representation of the nonholonomic motion of the vehicle at low speeds (Kong et al., 2015). Such models are especially useful for low-speed maneuvers such as parking.

Kinematic models only depend on the geometric parameters of the vehicle. On the other hand, dynamic models include additional parameters such as mass, rotational inertia, tire stiffness coefficients and the road friction coefficient. These parameters vary with operating conditions and require on-line non-linear parameter identification techniques (Hong et al., 2014).

In summary, the choice of vehicle model depends on the driving conditions. The control design strategies presented later in Section 4 can use any of the models presented in this section. This versatility is demonstrated by the use of both models in the Section 5 examples.

3.2. Driver Model

The objective of a driver model is to represent the process by which a driver transforms some perceived information about the driving situation into an action on the vehicle’s actuators (steering wheel, pedals). In this section, we discuss the motivations behind building driver models, formulate the problem and outline the different types of approaches proposed in the literature.

Motivation

Driver models are useful for both assisted and autonomous driving.

In the case of assisted driving, it is important to anticipate drivers’ actions in order to assist their decision and control. For example, a collision avoidance system needs to predict whether or not a driver will brake or steer in time to avoid an obstacle, in order to decide whether to intervene or not (McCall and Trivedi, 2007). In this case, a driver model makes the assistance function less intrusive by keeping the rate of false alarms low. Another example is energy management systems, which estimate the future desired speed to optimize the energy consumption of the engine (Yan et al., 2012). The driver models used by current semi-autonomous vehicles are generic, in the sense that they represent the way an average driver reacts to a given situation. However, a wide range of tools exist in machine learning which would allow to personalize driving assistance to various levels, e.g. by automatically clustering and classifying driving styles or by learning and adjusting driver models on-line. The advantage of personalization is that the response of the assistance system can be adapted to the specific driving habits of the driver, leading to safer and less intrusive interventions (Musicant et al., 2011).

In the case of autonomous driving, driver models can improve passengers’ comfort by making the driving feel as if the driver was in control. This can be achieved by tuning some parameters in the motion planner and the controller, or by using driver models to provide a reference of what a human driver would do in the current driving situation.

Formulation

While driving, human drivers use some features z_k about the driving situation to decide steering and acceleration commands $u_k^d = [\delta_k^d, a_k^d]$ to apply to the vehicle. Typical features include the road geometry, the state of the ego vehicle, the relative distance and velocity of other vehicles and obstacles. These features are contained in the vehicle state x^v and the environment state x^e . Therefore, in the formulation of the driver modeling problem, we will use $z_k = [x_k^v, x_k^e]$.

Modeling human driving is a challenging task. The actual physical process involves human perception, information processing, decision making, and physical action execution, all of which are extremely complex to model and not fully understood. Instead, a number of approaches have been proposed in the literature to approximate the true process and reduce its complexity. Here, we discuss two of the more common modeling approaches: black boxes and stochastic hybrid systems.

Black box models take the form of a function

$$u_k^d = f^d(x_k^v, x_k^e, d_k^u), \quad (9)$$

which directly maps the current features $[x_k^v, x_k^e]$ to the driver inputs u_k^d with some uncertainty d_k^u .

In hybrid system models, the state of the driver is represented by a random variable x_k^d with an associated transition probability function $x_{k+1}^d \sim p(x_{k+1}^d | x_k^d, x_k^v, x_k^e)$. The variable x_k^d can be discrete, for example to represent different maneuvers, or continuous, for example to represent the driver's body pose. It is assumed that the driver commands are dependent on the driver state, and this dependency is modeled by a stochastic process $u_k^d \sim p(u_k^d | x_k^d, x_k^v, x_k^e)$. The equations defining the driver model are expressed as:

$$\begin{cases} x_{k+1}^d = f^d(x_k^v, x_k^e, x_k^d, d_k^d) \\ u_k^d = \kappa^d(x_k^v, x_k^e, x_k^d, d_k^u), \end{cases} \quad (10)$$

$$(11)$$

with d_k^d representing the uncertainty affecting x_{k+1}^d and d_k^u representing the uncertainty affecting u_k^d .

Typically, black box approaches are used only in applications which require a driver model for a specific maneuver, e.g. lane keeping. Stochastic hybrid systems are used to represent transitions between maneuvers in a systematic manner. They are the preferred choice for prediction problems involving multiple types of maneuvers. The following subsection reviews existing approaches for black box models and stochastic hybrid systems in the context of driver modeling.

Approaches

Black box approaches can be grouped into parametric and non-parametric approaches. In parametric approaches, the analytic expression of the function f^d in (9) is defined a priori, and the parameters are either set manually or learned from data. Many of these models were originally designed for microscopic traffic simulation and then became popular outside that field. Popular examples include the SUMO simulator model (Krajzewicz, 2010), the two-point visual control model (Salvucci and Gray, 2004), and the Intelligent Driver Model (Treiber et al., 2000), which are all inspired from stimulus-response models and optimal control theory. Non-parametric approaches do not make

a priori assumptions about the structure of the model; instead the function f^d is inferred from data using statistical machine learning approaches such as Artificial Neural Networks (Pomerleau, 1989) or Gaussian Mixture Regression (Lefèvre et al., 2014a). The advantage of non-parametric approaches is their ability to represent complex behaviors which cannot be captured by the simpler parametric approaches (Lefèvre et al., 2014b). However, like all data-driven approaches, their performance will vary significantly depending on the training data provided.

In *stochastic hybrid system* based approaches for driver modeling (10)–(11), the driver state x_k^d is often chosen as a discrete variable which characterizes the type of control strategy currently applied by the driver. Driver assistance systems often use the estimated driver state as an indicator to decide when to intervene, in addition to the predicted driver inputs.

The driver state x_k^d in (10) can be estimated using standard classification techniques. Examples in the literature include assessing the driver’s level of distraction from their body pose with k-means clustering (Shia et al., 2014), predicting a driver’s intention to change lanes from the vehicle motion and the driver’s actions with Support Vector Machines (Kumar et al., 2013), Relevance Vector Machines (Morris et al., 2011), or Hidden Markov Models (Lefèvre et al., 2014a), classifying drivers as compliant or violating at traffic lights with a Support Vector Machine (Aoude et al., 2012). More examples can be found in a recent survey on motion prediction for intelligent vehicles (Lefèvre et al., 2014).

For each possible driver state x^d , the function κ^d in (11) models the control behavior of the driver. Any of the black box models described in the previous paragraph can be used for κ^d .

When the stochastic hybrid system is used on-line to predict the future driver inputs, the first step is to compute the distribution over the driver state x_k^d , and the second step is to derive from it the distribution on the driver input u_k^d . Depending on the application and on the computational resources, one may decide to consider only the most likely driver state and/or driver input instead of probability distributions.

In past projects, our group has worked with both black box approaches (Gray et al., 2013a,b) and stochastic hybrid system approaches (Shia et al., 2014; Lefèvre et al., 2014a, 2015). We conducted an experimental comparison of black box approaches in Lefèvre et al. (2014b). Since stochastic hybrid system approaches can model more complex driving strategies including switching between different maneuvers, our recent work has focused on this area. Here, we present the approach used in our most recent work (Lefèvre et al., 2014a, 2015). We model the driver as a stochastic hybrid system switching between three control strategies - lane keeping, lane change left, lane change right - represented by the discrete driver state variable $x_t^d \in \{LK, CL, CR\}$.

In the first step of the training phase, each control strategy $i \in \{LK, CL, CR\}$ is modeled by a fully connected Hidden Markov Model (HMM) with the following variables:

- $m_t^i \in \{m^{i,j}\}_{j=1}^{M^i}$ is the hidden mode at time t . M^i is the number of possible hidden modes for control strategy i . The sequence $\{m_k^i\}_{k=0}^t$ is denoted by $m_{0:t}^i$ for visual clarity.
- $o_t = [z_t, u_t^d]$ are the observations at time t , where $z_t = [x_t^v, x_t^e]$, and u_t^d is the driver

input. For visual clarity, the sequences $\{o_k\}_{k=0}^t$, $\{z_k\}_{k=0}^t$ and $\{u_k^d\}_{k=0}^t$ are denoted by $o_{0:t}$, $z_{0:t}$ and $u_{0:t}^d$, respectively.

The joint distribution between the hidden modes $m_{0:t}^i$ and the observations $o_{0:t}$ is written as follows:

$$p(m_{0:t}^i, z_{0:t}, u_{0:t}^d) = p(m_0^i, z_0, u_0^d) \prod_{k=1}^t [p(m_k^i | m_{k-1}^i) p(z_k, u_k^d | m_k^i)], \quad (12)$$

where $p(\cdot)$ denotes a probability density function. A multivariate Gaussian distribution is assumed for $p(z_k, u_k^d | m_k^i)$. The parameters of each HMM are learned off-line from training data using the Expectation-Maximization algorithm and the Bayesian Information Criterion as described in Lefèvre et al. (2014a).

In the second step of the training phase, the transition probabilities between the three control strategies are learned from data and used to combine the three HMMs into one fully connected HMM representing the full driving process. The variables for the resulting HMM are therefore:

- $m_t \in \{m^{LK,j}\}_{j=1}^{M^{LK}}, \{m^{CL,j}\}_{j=1}^{M^{CL}}, \{m^{CR,j}\}_{j=1}^{M^{CR}}\}$ is the hidden mode at time t . This means that the hidden mode m_t can take any of the values learned for any of the HMMs representing the different control strategies.
- $o_t = [z_t, u_t^d]$ are the observations at time t , where z_t are the features representing the driving situation and u_t^d is the driver input.

The joint distribution between the hidden modes $m_{0:t}$ and the observations $o_{0:t}$ is written as follows:

$$p(m_{0:t}, z_{0:t}, u_{0:t}^d) = p(m_0, z_0, u_0^d) \prod_{k=1}^t [p(m_k | m_{k-1}) p(z_k, u_k^d | m_k)], \quad (13)$$

where the measurement model $p(z_k, u_k^d | m_k)$ was learned in the first stage of the training process, and the prior and transition probabilities $p(m_k | m_{k-1})$ are learned using a modified version of the Expectation-Maximization algorithm. This procedure is described in more details in Lefèvre et al. (2014a).

When presented with new data, the most likely current driver state x_t^d and the most likely current driver inputs u_t^d can be estimated by performing inference on the HMM.

The most likely current driver state is denoted X_t^d and is iteratively selected as the most likely control strategy given the history of driving situations:

$$X_t^d = \arg \max_{i \in \{LK, CL, CR\}} \sum_{j=1}^{M^i} p(m_t = m^{i,j} | z_{0:t}). \quad (14)$$

For clarity purposes, X_t will be simplified as X in the equations below.

The most likely current driver input is denoted U_t^d and is computed from the consecutive values of the driving situation using Gaussian Mixture Regression, considering

only the most likely current driver state. That is, U_t^d is computed as the conditional expectation of u_t^d given the sequence $z_{0:t}$:

$$U_t^d = E[u_t^d | z_0, \dots, z_t] \quad (15)$$

$$= \sum_{j=1}^{M_X} \beta_{j,t} (\mu_j^u + \Sigma_j^{uz} (\Sigma_j^z)^{-1} (z_t - \mu_j^z)), \quad (16)$$

where

$$\mu_j = \begin{bmatrix} \mu_j^z \\ \mu_j^u \end{bmatrix} \text{ and } \Sigma_j = \begin{bmatrix} \Sigma_j^z & \Sigma_j^{zu} \\ \Sigma_j^{uz} & \Sigma_j^u \end{bmatrix}$$

are the parameters of the multivariate Gaussian distribution learned for mode $m^{X,j}$. $\beta_{j,t}$ is the mixing coefficient for mode $m^{X,j}$ at time t , computed as the probability of being in mode $m_t = m^{X,j}$ at time t and observing the sequence $z_{0:t}$.

As was mentioned in the introduction of this section, the performance of learning-based approaches varies strongly with the data used in the training phase. In practice, this means that the ability of the learned model to accurately represent and reproduce human behavior will be affected by the amount of training data, especially by how well the feature space is covered in the training data. Ongoing research is focused on quantifying this using statistical analysis tools which are commonly used in machine learning applications (Cortes et al., 1995; Benjamini and Hochberg, 1995; Caballero et al., 2006).

3.3. Environment Model

Autonomous vehicles navigating in the presence of other objects such as cars and pedestrians must anticipate their future behavior in order to plan safe trajectories. An overview of existing approaches for vehicle motion modeling can be found in Lefèvre et al. (2014). The simpler approaches use *physics-based models*, which extrapolate the vehicle's movement based on kinematic or dynamic vehicle models with some assumptions on the driver's inputs. Longer prediction times require *maneuver-based models*, which use motion primitives to represent the various maneuvers that the driver can perform. *Interaction-aware models* yield an even higher level of realism as they base their predictions on an understanding of the interactions between vehicles.

Within each of the above categories, environment models can be further classified by their type of prediction into (i) deterministic models, (ii) stochastic models, and (iii) scenario-based models. Deterministic models provide a single prediction trajectory for all objects in the scene (Jansson, 2005). This trajectory is often used to quantify a threat level, e.g., by the time-to-collision (Labayrade et al., 2005). However, deterministic models cannot capture the significant uncertainty associated with different possible actions taken by the human driver, especially over longer time periods.

Stochastic models use standard PDFs to model the driver's behavior (Laugier et al., 2011). A deterministic trajectory can easily be obtained as the most likely outcome of the stochastic model. The stochastic approach is often implemented by using a set of standard distributions, typically Gaussian mixtures (Wiest et al., 2012). In addition, it

is difficult to include traffic rules and the interaction between vehicles in such models because it increases their complexity.

Scenario-based models can overcome these limitations by not stating these probability distributions explicitly (Broadhurst et al., 2005; Eidehall and Petersson, 2008). Rather, the uncertainty is described via a discrete number of possible future scenarios. This facilitates the modeling of vehicle interactions by excluding traffic scenarios that are unrealistic, e.g. if they cause a collision. Moreover, the scenarios may be drawn directly from collected real-world data and do not require the fitting of a probabilistic model.

The evolution of the environment over time is described by the following discrete-time state-space representation,

$$x_{k+1}^e = f^e(x_k^v, x_k^e, x_k^d, d_k^e), \quad (17)$$

where the disturbance d_k^e is introduced to model the uncertainty in the environment dynamics. Typically, the state vector x_k^e contains the position, orientation and velocity of each vehicle in the scene (we will refer to such vehicles as *target vehicles*). The control inputs to the target vehicles are implicitly modeled and hence, they do not explicitly appear in (17).

In this paper, we choose to present two approaches for modeling the behavior of target vehicles, and predicting their trajectories. These are described in the following subsections.

Multiple model filtering

Multiple model methods are popular in target tracking applications (Mazor et al., 1998). The idea is to model each target vehicle as a hybrid system, where the discrete modes correspond to various possible maneuvers. As discussed in Section 3.2, hybrid systems are a natural way of representing driving behavior. The approach developed in Carvalho et al. (2014) for target motion prediction in structured environments is discussed next.

The j^{th} target vehicle is described by the following Markov jump affine system,

$$x_{j,k+1}^e = F_k^{(i)} x_{j,k}^e + G_k^{(i)} d_k^e + E_k^{(i)} \quad (18a)$$

$$o_{j,k}^e = H_k^{(i)} x_{j,k}^e + v_k^e, \quad (18b)$$

where the superscript (i) corresponds to the discrete mode $m^{(i)}$ in the model set $\mathcal{M} = \{m^{(j)}\}_{j=1}^M$. $o_{j,k}^e$ is the observation vector and v_k^e is the measurement noise. d_k^e and v_k^e are assumed to be normally distributed as $\mathcal{N}(0, \Sigma_d^e)$ and $\mathcal{N}(0, \Sigma_v^e)$ respectively. Transitions between modes are governed by a transition probability matrix $\pi \in \mathbb{R}^{M \times M}$, such that

$$\pi_{ij} = P(m_{k+1} = m^{(j)} | m_k = m^{(i)}). \quad (19)$$

The representation in (18) is chosen to be a function of the road geometry and other contextual information. For example, the set of longitudinal motion models can include a car-following model in addition to simple models such as constant velocity and constant acceleration. In Carvalho et al. (2014), the lateral motion model set consists

of state-feedback models wherein the lateral acceleration is given by a proportional derivative control law on the vehicle's offset from the lane centerline.

The problem of motion prediction for target vehicles is equivalent to estimating the discrete mode and continuous state of (18). The framework of Interacting Multiple Model Kalman filtering (IMM-KF) has been shown to be an efficient suboptimal solution for hybrid system state estimation (Mazor et al., 1998). At each time step, the IMM-KF computes the probabilities of the discrete modes and a mode-dependent state estimate, conditioned on the measurement history. These quantities along with the associated uncertainty can be easily propagated through the state update equations (18a) to obtain a distribution over the future positions of each target vehicle.

Sampling-based approach

Sampling-based or Monte Carlo approaches have been used for traffic predictions in Broadhurst et al. (2005); Eidehall and Petersson (2008). The idea is to represent the future states of objects in the environment by samples, which are referred to as scenarios. To generate scenarios, the first step is to sample possible feasible driver inputs from any given stochastic model and apply them to a dynamic model of the target vehicle. In the second step, scenarios resulting in unrealistic target trajectories are rejected, as suggested in Broadhurst et al. (2005). A particular scenario may be rejected if, for example, a target vehicle crosses the road boundaries, exceeds its physical limits (e.g., set by the road grip), or multiple vehicles collide. In this case, the scenario is rejected and replaced with a new scenario (this process is often called *rejection sampling*). This approach results in an interaction-aware target vehicle trajectory prediction model.

In more detail, the *speed profile* of a target vehicle j is generated by first sampling a target speed v_j^{ref} . A probabilistic model of v_j^{ref} considers speeds equal to the current speed as well as variations around it. The acceleration or braking of the ego vehicle $a_{j,k}$ at time k is then obtained by a proportional controller on the velocity $v_{j,k}$:

$$a_{j,k} = K_{\text{lon},j}(v_j^{\text{ref}} - v_{j,k}),$$

which is saturated at the acceleration limits. The gain $K_{\text{lon},j}$ is also sampled to capture variations in the rate of acceleration or braking.

The *steering profile* of a target vehicle j is obtained in a similar manner. First, the value of a random variable models whether the target vehicle intends to stay in its current lane or attempts a lane change to the left or the right. The center line of the target lane is the reference for the lateral position, η_j^{ref} . The reference is tracked by selecting the steering angle $\delta_{j,k}$ at time k by a scheduled proportional controller on the current lateral position $\eta_{j,k}$:

$$\delta_{j,k} = K_{\text{lat},j}(\eta_j^{\text{ref}} - \eta_{j,k}),$$

which is saturated according to steering and lateral acceleration constraints. The gain $K_{\text{lat},j}$ is also sampled to capture variations in the aggressiveness of the lane change maneuver.

3.4. Constraints

The goal of keeping the vehicle on the road while maneuvering safely and respecting the comfort of passengers can be expressed as constraints on the vehicle's states and inputs.

Actuator limits

Physical limitations on the actuators impose bounds on the control inputs and input slew rates,

$$u^{\min} \leq u_k^v \leq u^{\max} \quad (20a)$$

$$\dot{u}^{\min} \leq \dot{u}_k^v \approx \frac{u_k^v - u_{k-1}^v}{\Delta t^v} \leq \dot{u}^{\max} \quad (20b)$$

where Δt^v is the discretization time of the vehicle model (3).

Collision avoidance

The problem of avoiding collisions and staying clear of objects in the environment is challenging because the safe region in which the vehicle can move is often non-convex. Moreover, some formulations of collision avoidance yield non-differentiable constraints. This increases the complexity of the resulting on-line mathematical program. Various approaches for collision avoidance have been proposed in the literature. Obstacles are described as ellipses in Gao et al. (2014) resulting in non-convex but smooth constraints. Artificial potential field methods treat obstacles as repulsive forces and augment the objective function with a penalty due to collisions (see e.g. Khatib (1986); Gao et al. (2010)). Polytopic descriptions of objects require mixed-integer programming (Blackmore et al., 2011; Schouwenaars et al., 2001) or local collision checking approaches (Schulman et al., 2013). For structured environments such as highways, a simplified approach based on sampling the boundaries of the safe region on the road is presented in Anderson et al. (2010); Carvalho et al. (2013). The collision avoidance formulation used in the examples presented in Section 5 depends on the specific application under consideration. In general, the constraints are a non-linear function of the ego and target vehicle states, and are represented as

$$h(x_k^v, x_k^e) \leq 0. \quad (21)$$

Comfort and handling

For the comfort of passengers and to avoid skidding, the aggressiveness of the vehicle should be restricted. For kinematic vehicle models such as (4), this is enforced by constraining the lateral acceleration,

$$a_{\text{lat}}^{\min} \leq a_{\text{lat}} = \frac{v_k^2}{l_r} \sin(\beta_k) \leq a_{\text{lat}}^{\max}. \quad (22)$$

In the case of dynamic vehicle models such as (5), the slip angles at the front and rear wheels are constrained to be within bounds,

$$\alpha_i^{\min} \leq \alpha_{i,k} \leq \alpha_i^{\max}. \quad (i \in \{f, r\}) \quad (23)$$

In summary, safety and comfort requirements are enforced by constraints on the control inputs and states of the vehicle. The linear input constraints in (20) are concisely written as,

$$\mathbf{u}^v \in \mathcal{U} \quad (24)$$

where $\mathbf{u}^v = [u_0^v, \dots, u_{T-1}^v]$, and $\mathcal{U}$ is a polytope. The non-linear constraints in (21)–(23) arising from collision avoidance and comfort requirements are compactly expressed as

$$g_k(x_k^v, u_k^v, x_k^e) \leq 0. \quad (25)$$

4. Predictive Control Design for Autonomous Vehicles

The following CFTOC problem will be used to present our control approach,

$$\min_{\kappa_{0|t}^c, \kappa_{1|t}^c, \dots, \kappa_{T-1|t}^c} \sum_{k=0}^{k=T-1} E \left[\text{Cost}(x_{k|t}^v, u_{k|t}^v, x_{k|t}^{\text{ref}}, u_{k|t}^d) \right] \quad (26a)$$

$$\text{subj. to } x_{k+1|t}^v = f^v(x_{k|t}^v, u_{k|t}^v, d_{k|t}^v) \quad (26b)$$

$$x_{k+1|t}^d = f^d(x_{k|t}^v, x_{k|t}^e, x_{k|t}^d, d_{k|t}^d) \quad (26c)$$

$$x_{k+1|t}^e = f^e(x_{k|t}^v, x_{k|t}^e, x_{k|t}^d, d_{k|t}^e) \quad (26d)$$

$$u_{k|t}^v = f^u(u_{k|t}^d, u_{k|t}^c) \quad (26e)$$

$$u_{k|t}^d = \kappa_{k|t}^d(x_{k|t}^v, x_{k|t}^e, x_{k|t}^d, d_{k|t}^u), \quad (26f)$$

$$u_{k|t}^c = \kappa_{k|t}^c(x_{k|t}^v, x_{k|t}^e, x_{k|t}^d), \quad (26g)$$

$$\mathbf{u}_{k|t}^v \in \mathcal{U} \quad (26h)$$

$$g_{k|t}(x_{k|t}^v, u_{k|t}^v, x_{k|t}^e) \leq 0 \quad (26i)$$

$$x_{0|t} = x_t, \quad (26j)$$

where the symbol $v_{k|t}$ is the random variable v at time $t + k$ predicted at time t , and $E[v]$ denotes the expected value of v . Problem (26) seeks a set of feedback laws $\kappa_{k|t}^c(\cdot)$ which minimizes the average cost and satisfies constraints (26h)–(26i). With an abuse of notation and for the sake of simplicity, the notation $v_{k|t}$ will be replaced with v_k in the rest of the paper.

We denote by $\mathcal{D}^\star$ the set of all possible realizations of the uncertainty $d^\star$, with $\star$ equal to v , e or d . Due to the uncertainty in the predicted states $x_{k|t}^v$ and $x_{k|t}^e$, the constraints (26i) cannot be treated deterministically. Instead, constraint satisfaction can be enforced robustly for all admissible disturbances:

$$g_k(x_k^v, u_k^v, x_k^e) \leq 0 \quad \forall d^\star \in \mathcal{D}^\star, \quad (27)$$

or we can allow for a small probability of constraint violation:

$$P(g_k(x_k^v, u_k^v, x_k^e) \leq 0) \geq 1 - \varepsilon_k, \quad (28)$$

where $P(A)$ is the probability that the event A is true. Constraints (27) and (28) are referred to as *robust constraints* and *chance constraints*, respectively. If the constraints (26i) are replaced with (27), problem (26) is called a Robust MPC (RMPC) problem; if the constraints (26i) are replaced with (28), it is called a Stochastic MPC (SMPC) problem.

The solution to the MPC problem (26) comprises three main steps: (i) the translation of the optimization over *control policies* $\kappa^c(\cdot)$ into a finite-dimensional optimization problem, (ii) the propagation of the uncertain system states over the prediction horizon in order to translate the robust constraints (27) or probabilistic constraints (28) into deterministic constraints, and (iii) the solution of the resulting mathematical program.

With the exception of linear systems and special classes of distributions (e.g., normal distributions), steps (i) and (ii) are non-trivial. Various methods are reported in literature to handle steps (i) and (ii), and they affect the complexity of the mathematical program at step (iii) as well as the conservatism of the resulting control policy (Lee and Cooley, 1998; van Hessem et al., 2001; Primbs, 2007; Cannon et al., 2007; Hokayem et al., 2010).

Step (i) has different solutions (Bemporad et al., 2003). Open-loop prediction schemes are conservative since they look for one optimal open-loop control sequence that has to cope with all possible future uncertainty realizations, without taking future measurements into account. Closed-loop formulations overcome this issue but they can lead to computationally intractable problems. A compromise consists in fixing the control structure (e.g., affine state-feedback or affine disturbance feedback policies), parameterizing the control sequence in the feedback gains, and optimizing over these parameters (Kouvaritakis et al., 2011, 2010). The on-line computational complexity can be further reduced by computing the state or disturbance feedback gain off-line (e.g., using the Linear-Quadratic Regulator (LQR) algorithm).

The chance constraints (28) in step (ii) are translated into deterministic ones by enforcing tightened constraints on the expected values of the states and inputs. The tightening offset is computed based on the tails of the disturbance probability distributions (van Hessem and Bosgra, 2002; Skaf and Boyd, 2010). In practice, the PDFs have a finite support and are non-Gaussian. In addition, the vehicle models are non-linear, making it difficult to propagate the distribution of the vehicle states over the prediction horizon. Exact solutions to the SMPC problem for non-linear systems subject to non-Gaussian disturbances are, in general, computationally intractable for real-time implementation on current vehicle platforms. Obtaining a computationally tractable approximation to this problem is crucial for the real-time implementation of SMPC. Sample-based methods provide an alternative approach to transform the chance constraints for non-Gaussian random variables (Calafiore and Campi, 2006; Calafiore, 2010; Campi and Garatti, 2011a; Constantinescu et al., 2011; Zavala et al., 2009; Zavala, 2013). The approach consists in transforming the chance constraints (28) into deterministic counterparts by evaluating them at a large number of disturbance samples.

Non-Gaussian and finitely supported disturbances have been studied in Cannon et al. (2009b,a); Kouvaritakis et al. (2010, 2011); Calafiore and Campi (2006); Campi and Garatti (2011a) in the context of SMPC. Cannon et al. (2009a,b) use a tube-based method to translate the chance constraints. The work in Kouvaritakis et al. (2010, 2011) suggests to translate the chance constraints to deterministic ones by using tightening

offsets. The offsets are computed off-line using numerical approximations of convolution integrals. Chance-constrained predictive control for non-linear systems has been studied in Mesbah et al. (2014); Streif et al. (2014), wherein polynomial expansions are used to approximate probability distribution functions.

In this article, we give an overview of three approaches for vehicle control applications, (i) Tube-based RMPC, (ii) Linear time-varying SMPC, and (iii) Scenario-based SMPC.

4.1. Tube-based RMPC

A tube-based robust non-linear MPC approach for autonomous lane-keeping and obstacle avoidance was presented in Gao et al. (2014). The main idea is to bound the maximum deviation of the system trajectory from the nominal trajectory under a precomputed affine feedback law. The bounds can be computed off-line for linear systems using the notion of robust positively invariant (RPI) sets. To account for the non-linear vehicle model, Gao et al. (2014) use a modified form of the dynamic bicycle model (5), which can be written as,

$$x_{k+1}^v = Ax_k^v + Bu_k^v + l(x_k^v) + d_k^v, \quad (29)$$

where $l(\cdot)$ is Lipschitz continuous, and $d^v \in \mathcal{D}$ is a set-bounded disturbance. The affine feedback law is given by,

$$u_k^v = \bar{u}_k^v + K(x_k^v - \bar{x}_k^v), \quad (30)$$

where $\{\bar{x}_k^v\}_{k=0}^T$ is the state sequence obtained by starting at the current state and applying the input sequence $\{\bar{u}_k^v\}_{k=0}^{T-1}$ to (29) assuming a zero disturbance sequence. The error $e_k := x_k^v - \bar{x}_k^v$ evolves as,

$$e_{k+1} = (A + BK)e_k + (l(x_k^v) - l(\bar{x}_k^v)) + d_k^v. \quad (31)$$

If a RPI set (denoted by $\mathcal{E}$) for (31) can be computed, robust constraint satisfaction is guaranteed by tightening the original state and input constraints by $\mathcal{E}$ and $K\mathcal{E}$, respectively, in the on-line constrained finite-time optimal control problem.

Computing the RPI set for a non-linear system is hard in general. The computation is simplified for systems of the form (31), if the non-linear terms are “small” compared to the linear term. In this case, the Lipschitz constant of $l(\cdot)$ can be used to bound the non-linearities, and suitably augment the disturbance set $\mathcal{D}$. The RPI set for the resulting linear system can be computed easily off-line. The on-line computations reduce to solving a standard non-linear MPC problem with tightened constraints to compute the nominal input sequence $\{\bar{u}_k^v\}_{k=0}^{T-1}$. A common choice of the feedback gain K in (30) is the infinite-horizon LQR gain for the pair (A, B) .

4.2. Linear Time-Varying SMPC

Analytical SMPC approaches are restricted to linear systems with normally distributed additive disturbances. The fact that linear transformations of Gaussian random variables are themselves normally distributed allows for the easy propagation of the uncertainty through the system dynamics. Moreover, an analytical expression for the

chance-constraint tightening can be obtained. Carvalho et al. (2014) use an affine time-varying description of the vehicle model (26b),

$$x_{k+1}^v = A_k x_k^v + B_k u_k^v + E_k + d_k^v, \quad (32)$$

where the disturbance sequence $\{d_k^v\}_{k=0}^{T-1}$ is assumed to be independent and identically distributed (i.i.d.). Each element of the sequence is a Gaussian random variable whose PDF is given by $\mathcal{N}(0, \Sigma^d)$. The linearization is performed on-line about predicted open-loop trajectories of the vehicle obtained from the previous time step of the MPC.

Closed-loop approach

Following the closed-loop approach presented in Kouvaritakis et al. (2010), the feedback law in (26g) is parameterized as,

$$u_k^v = K_k x_k^v + c_k. \quad (33)$$

The closed-loop system evolves as,

$$x_{k+1}^v = \Phi_k x_k^v + B_k c_k + E_k + d_k^v, \quad (34)$$

where $\Phi_k = A_k + B_k K_k$. If the current state x_0^v is normally distributed with mean $\hat{x}_0^v$ and covariance Σ_0^v , the predicted state x_k^v is distributed as $\mathcal{N}(\hat{x}_k^v, \Sigma_k^v)$, where

$$\hat{x}_{k+1}^v = \Phi_k \hat{x}_k^v + B_k c_k + E_k \quad (35a)$$

$$\Sigma_{k+1}^v = \Phi_k \Sigma_k^v \Phi^T + \Sigma^d. \quad (35b)$$

The benefit of using the closed-loop formulation (33) is emphasized by (35b). If the feedback gain is chosen to stabilize the system, the predicted covariance is smaller than that in the case of the open-loop formulation. A common choice of the sequence of feedback gains $\{K_k\}_{k=0}^{T-1}$ is the solution of the finite horizon LQR problem for the time-varying system (32).

Environment uncertainty representation

The predicted positions of target vehicles from the environment model are used to generate linear state constraints,

$$G_k^x x_k^v \leq h_k^x. \quad (36)$$

While collision avoidance constraints are non-convex in general, linear constraints of the form (36) can be obtained by, for example, linearizing ellipsoidal constraints or using the collision checking approach of Schulman et al. (2013). To account for the uncertainty in the target vehicles' predicted positions, the vector h_k^x is assumed to be normally distributed as $\mathcal{N}(\hat{h}_k^x, \Sigma_k^x)$. x_k^v and h_k^x in (36) are stochastic variables and hence, (36) is reformulated as the following chance constraint,

$$P(G_k^x x_k^v \leq h_k^x) \geq 1 - \varepsilon_k. \quad (37)$$

Dealing with the joint chance constraints (37) involves computing integrals over multivariate PDFs, which is computationally formidable. Van Hessem and Bosgra (2002) reformulate (37) as a second-order cone constraint. We employ Boole's inequality to transform the joint constraints (37) into univariate constraints,

$$P(g_{k,i}^x x_k^v \leq h_{k,i}^x) \geq 1 - \varepsilon_{k,i}, \quad (i = 1, \dots, m_k^x) \quad (38)$$

where $g_{k,i}^x$ and $h_{k,i}^x$ are the i^{th} rows of G_k^x and h_k^x , respectively, m_k^x is the number of rows in the matrix G_k^x , and $\{\varepsilon_{k,i}\}_{i=1}^{m_k^x}$ satisfies,

$$\sum_{i=1}^{m_k^x} \varepsilon_{k,i} \leq \varepsilon_k. \quad (39)$$

Intuitively, $\varepsilon_{k,i}$ represents the risk associated with the i^{th} constraint at time step k , and (39) constrains the total risk to be less than the desired value ε_k . While it is possible to optimize over $\{\varepsilon_{k,i}\}_{i=1}^{m_k^x}$, we use a fixed risk allocation procedure in this article. It can be shown that (38) is satisfied if and only if,

$$g_{k,i}^z \hat{z}_{k,i} + \gamma_{k,i} \leq 0, \quad (i = 1, \dots, m_k^x) \quad (40)$$

where,

$$\begin{aligned} \hat{z}_{k,i} &= \begin{bmatrix} \hat{x}_{k,i}^v \\ \hat{h}_{k,i}^x \end{bmatrix}, \quad g_{k,i}^z = \begin{bmatrix} g_{k,i}^x & -1 \end{bmatrix}, \\ \gamma_{k,i} &= \sqrt{2g_{k,i}^z \Sigma_{k,i}^z g_{k,i}^z} \operatorname{erf}^{-1}(1 - 2\varepsilon_{k,i}), \\ \Sigma_{k,i}^z &= \begin{bmatrix} \Sigma_k^v & 0 \\ 0 & \Sigma_{k,i}^x \end{bmatrix}, \end{aligned}$$

where $\operatorname{erf}^{-1}(\cdot)$ is the inverse error function, and $\Sigma_{k,i}^x$ denotes the i^{th} diagonal entry of Σ_k^x .

The optimization problem to be solved on-line is obtained by replacing the chance-constraints (37) with the deterministic constraints (40). If the cost function in (26a) is chosen to be quadratic in the states and inputs, the resulting mathematical program is a quadratic program (QP). Moreover, the complexity of the QP is the same as in the case of the nominal problem wherein all disturbances are replaced with their mean values.

4.3. Scenario-based SMPC

The approach builds on the latest developments in Scenario MPC (SCMPC) described in Blackmore et al. (2010); Schildbach et al. (2014). The SCMPC algorithm uses an representation of the uncertainty in the system via samples. This enables a direct connection of SCMPC, for instance, with scenario-based traffic predictions (see Section 3.3). In fact, any model-based or data-based approach can be used for generating a sufficient number of samples for all uncertainty variables d^v , d^e and d^d corresponding to the vehicle, environment and driver, respectively. Moreover, the uncertainty variables may follow an arbitrary distribution which need not be known explicitly.

More concretely, let $\{d_k^{v,(s)}\}_{s=1}^S$, $\{d_k^{e,(s)}\}_{s=1}^S$, $\{d_k^{d,(s)}\}_{s=1}^S$ be S i.i.d. samples of the random variables d_k^v , d_k^e and d_k^d , respectively. A *scenario* $\omega^{(s)}$ is defined as the full-horizon sample,

$$\omega^{(s)} = \{(d_0^{v,(s)}, d_0^{e,(s)}, d_0^{d,(s)}), \dots, (d_{T-1}^{v,(s)}, d_{T-1}^{e,(s)}, d_{T-1}^{d,(s)})\}. \quad (41)$$

Each scenario may lead to a different outcome of the vehicle dynamics (26b), the driver actions (26c), and/or the traffic environment (26d). Hence, the corresponding predictions of the vehicle states $\{x_k^{v,(s)}\}_{k=1}^T$, the driver states $\{x_k^{v,(s)}\}_{k=1}^T$, and the environment states $\{x_k^{e,(s)}\}_{k=1}^T$ become deterministic, but are dependent on the control inputs and the scenario $\omega^{(s)}$.

The basic idea of SCMPC is to make the constraints in the MPC problem (26) robustly safe in all of these scenarios. In other words, the state constraints (26i) are replaced with:

$$g_k(x_k^{v,(s)}, u_k^v, x_k^{e,(s)}) \leq 0 \quad \forall s = 1, \dots, S. \quad (42)$$

This approach is highly intuitive, flexible in handling many types of uncertainties, and generally easy to implement. Clearly, the higher the number of scenarios, the more conservative and safe will be the controller actions. Recent studies have established a direct connection between S and the probability level of chance constraint in closed loop (Schildbach et al., 2013, 2014). This theory is based on principles from scenario optimization (Campi and Garatti, 2008, 2011b). The number of required scenarios to achieve small probability of constraint violation is low, typically less than a hundred, making SCMPC attractive also from a computational standpoint. See Schildbach and Borrelli (2015) for further details.

5. Examples

In this section, we show some examples of how we address semi-autonomous and fully autonomous tasks using the architecture in Figure 1. For each example, the choice of vehicle, driver and environment models, and MPC approach depends on design requirements.

5.1. Experimental Setup

The vehicle used for experimental tests was a Hyundai Azera (total wheelbase of 2.843 m, $l_f = 1.105$ m and $l_r = 1.738$ m). An OTS (Oxford Technical Solutions) RT2002 sensing system, combined with a GPS base station, is used for the localization of the vehicle in the inertial frame. The OTS RT2002 system is comprised of a differential GPS (global positioning system), an IMU (inertial measurement unit) and a DSP (digital signal processor).

The positioning errors of the OTS system vary depending on the current mode of the GPS base station. The current mode depends on the number of available satellites and the communication between the vehicle and the base station. The possible modes in our tests are RTK (real-time kinematic) integer precision ($\sigma = 0.02$ m) or differential

Table 1: Velocity information about the winding track

Scaling	Avg. Speed	Speed Range	Std. Dev.
0.7	10.8 m/s	5-15.56 m/s	2.14 m/s
1	15.4 m/s	7.2-22.23 m/s	3 m/s

GPS precision ($\sigma = 0.4$ m). In both modes, the precision of the measured heading angle is the same ($\sigma = 0.1$ deg).

Computations related to the MPC-based controller, localization, and low-level controllers are performed on a dSpace MicroAutobox, which consists of an IBM PowerPC processor running at 900 MHz. The on-line non-linear optimization problem in the predictive controller is solved by using the general purpose solver NPSOL (Gill et al., 1986).

5.2. Path Following with Nominal MPC on Winding Track

This example presents the design of an autonomous path following controller. The winding track (Figure 9) is a road course with many turns and straight line paths, making it an ideal testbed for a path following controller. The variation in the track is a good sample of real life driving scenarios. We introduce the following definitions:

- $x_t^v = [X_t, Y_t, \psi_t, v_t]$ is the state of the ego vehicle at time t , and $u_t^v = [\delta_t, a_t]$ is the control input. The notation is introduced in Section 3.1. As the application in this example is fully autonomous, $u_t^v = u_t^c$.

The controller computes a sequence of control inputs $\{u_k^c\}_{k=0}^{T-1}$ to track the sequence of reference states $\{x_k^{\text{ref}}\}_{k=1}^T$.

Approach

Vehicle model: The kinematic bicycle model presented in Section 3.1 is used for the controller design. This example shows that the kinematic model performs well under a wide range of conditions.

Reference generation: A map of the winding track is created along with a speed reference at different points along the map. The speed reference on the map, or a speed map, is obtained by recording the driving data of a human driver on this track. For the experiments, the reference speed is obtained by scaling the speed map by a desired factor. In the first experiment, a scaling factor of 0.7 is used. The subsequent experiment follows the human driven speed reference by using a scaling factor of 1. Table 1 provides information on the reference speed range during both tests.

The reference positions over the horizon are obtained as follows. First, the current position of the ego vehicle in the inertial frame is projected onto the lane centerline obtained from the map of the winding track. Starting from the projected position, points along the lane centerline are generated, where the spacing between the points is

Figure 11: Differences in the open-loop errors for four different scenarios. In the left column, test results for a straight road and a curvy road are shown for an experiment at a scaled speed of 0.7. On the right side are the same positions on the track with a scaled speed of 1.

a function of the desired speed. These points, along with the reference speed, constitute the reference state sequence $\{x_k^{\text{ref}}\}_{k=1}^T$. More details can be found in Kong et al. (2015). *Controller:* A nominal MPC-based controller is used to control the steering and acceleration for path following. No driver or environment model is used as the system is autonomous, and no other objects are considered in the driving scene.

Results

Figure 11 displays the open-loop predicted trajectories by the controller versus the closed-loop measured GPS positions on the winding track. On the straight line path, the kinematic bicycle model accurately models the predicted paths at both scaling factors. However, as is seen in Figure 11, the reference tracking at higher speeds is not as good due to the larger lateral acceleration.

Our results show that the proposed MPC-based controller can track the reference path well at various speeds, with better tracking at lower speeds. The average tracking error, i.e. the distance between the vehicle and the reference path, at a scaled speed of 0.7 and 1 is 0.26 m and 0.58 m, respectively. In Figure 12, the distributions of reference trajectory errors are shown for different scaled speeds.

Figure 12: Distribution of tracking errors relative to the reference trajectory for two reference speeds.

5.3. Autonomous Car Following

In this example, we address a longitudinal control problem during autonomous lane keeping on the highway. We introduce the following definitions:

- $x_t^v = [\xi_t^v, v_t^v]$ is the state of the ego vehicle at time t , and $u_t^v = a_t$ is the control input. The notation is introduced in Section 3.1. The application in this example is fully autonomous, hence, $u_t^v = u_t^c$.
- $x_t^e = [\xi_t^e, v_t^e]$ is the state of the environment at time t , where ξ_t^e is the longitudinal position of the preceding vehicle in a road-aligned coordinate system, and v_t^e is the longitudinal velocity of the preceding vehicle.
- $z_t = [\xi_t^r, v_t^r, v_t^v]$ are the features representing the current driving situation at time t , to be used by the driver model. $\xi_t^r = (\xi_t^e - \xi_t)$ is the relative distance to the preceding vehicle, and $v_t^r = (v_t^e - v_t)$ is the relative velocity to the preceding vehicle.

The driver model generates a human-like acceleration profile $\{u_k^d\}_{k=0}^{T-1}$, which is used by the controller as an input reference. The MPC-based controller solves a CFTOC problem to generate an open-loop acceleration sequence $\{u_k^c\}_{k=0}^{T-1}$ which guarantees the safety of the vehicle, also taking into account the predicted environment states $\{x_k^e\}_{k=1}^T$.

Approach

Vehicle model: A simple deterministic kinematic point-mass model was used for the ego vehicle in this example.

Driver model: We use the stochastic hybrid system approach presented in Section 3.2 to model the driver and generate the reference acceleration sequence $\{u_k^d\}_{k=0}^{T-1}$. Since the focus is on lane keeping, the driver state is set to $x_t^d = LK$ at each time step t .

Environment model: In this example, the predicted environment states are computed assuming constant speed for the preceding vehicle, using a kinematic model. The uncertainty associated with these predictions is not considered.

Controller: A nominal MPC-based controller is used to control the acceleration of the ego vehicle. The safety requirement of avoiding collisions with the preceding vehicle is enforced by constraining the ego vehicle’s longitudinal coordinate as,

$$\xi_t^v + \xi_{\text{safe}} \leq \xi_t^e, \quad (43)$$

where the safety distance ξ_{safe} is a design parameter.

Results

The proposed autonomous car following system was evaluated in two steps. First, we focused on the performance of our driver modeling approach and evaluated its ability to learn different driving styles from different drivers. Then, we focused on the performance of the model predictive controller and verified experimentally that the safety constraints are enforced at all times.

To evaluate the performance of the driver model, we collected highway driving data from 5 human drivers: Driver A, B, C, D, E. Each driver drove for 60 minutes in both low speed (congested) traffic and high speed (free-flow) traffic. For each driver, we learned a *personalized* driver model (using the data collected from that driver) and an *average* driver model (using the data collected from the other drivers). Off-line, we replayed the real-life driving situations collected on the highway, and simulated the behavior of the autonomous car. Using the inverse Time-To-Collision (TTCi) and the Vehicle Specific Power (VSP) as indicators to represent a driver’s style, we found that the *personalized* driver models always performed better at reproducing a specific driver’s driving style than the *average* driver models (see Figure 13).

As was mentioned in Section 3.2, the learned driver model will be representative of the driver’s behavior only if the training data contains samples from the entire feature space. In this work, this was ensured by plotting the distribution of the feature points in the training data and collecting more data in areas which were found to be not populated in the initial training dataset. We are currently investigating more general solutions to the problem of quantifying the adequacy of the training data.

We also demonstrated the ability of the controller to enforce safety constraints through real experiments on our test vehicle. For safety reasons, the experiments were performed at low speeds using virtual preceding vehicles. The safety distance was intentionally set to a high value of $\xi_{\text{safe}} = 5$ m in order to artificially create situations where the controller needs to deviate from the reference provided by the driver model to keep the vehicle safe. We performed experiments similar to the one described in Section 2, where a virtual preceding vehicle is suddenly introduced. We observed that the controller’s response adjusts to the situation. When the inputs provided by the driver model satisfy the safety constraints, the controller matches the reference exactly. When the controller detects a future violation of the constraints, it will apply a stronger braking to keep the ego vehicle in the safety zone. When the virtual car is introduced with initial unsafe conditions, the controller applies emergency braking. These results, obtained with the real vehicle, are similar to what was obtained in simulation in Section 2 (see Figures 3-5).

A more detailed description of our longitudinal controller and more thorough results can be found in Lefèvre et al. (2015).

Figure 13: Kolmogorov-Smirnov distance between the true and simulated driving sequences for two driving style indicators: TTCi and VSP. A *personalized* model and an *average* model were learned and tested for each driver.

5.4. Autonomous navigation via SMPC

In this example, we present the design of a controller for autonomous navigation in urban environments using the SMPC framework presented in Section 4.2. The approach is based on the work in Carvalho et al. (2014). The goal of the controller is to maintain a reference speed such as the speed limit while respecting the safety and comfort of passengers. If necessary, the vehicle is expected to go around slower vehicles in its lane. We introduce the following definitions:

- $x_t^v = [\dot{x}_t, \dot{y}_t, \dot{\psi}_t, \xi_t, \eta_t, \phi_t]$ is the state of the ego vehicle at time t , and $u_t^v = [\delta_t, a_t]$ is the control input. The notation is introduced in Section 3.1. As the application in this example is fully autonomous, $u_t^v = u_t^c$.
- $x_{j,t}^e = [\xi_{j,t}^e, v_{j,t}^e, a_{j,t}^e, \eta_{j,t}^e, \dot{\eta}_{j,t}^e]$ is the state of the j^{th} target vehicle at time t . $\xi_{j,t}^e$, $v_{j,t}^e$ and $a_{j,t}^e$ are the longitudinal position, speed and acceleration along the road, respectively. $\eta_{j,t}^e$ and $\dot{\eta}_{j,t}^e$ are the lateral position error and change rate, respectively, in a road-aligned coordinate frame. It is assumed that $o_{j,t}^e = [\xi_{j,t}^e, v_{j,t}^e, \eta_{j,t}^e]$ can be measured at each time step t .

The controller computes a sequence of control inputs $\{u_{t+k}^c\}_{k=0}^{T-1}$ to track the sequence of reference states $\{x_{t+k}^{\text{ref}}\}_{k=1}^T$.

Approach

Vehicle model: The dynamic bicycle model described in Section 3.1 with a linear tire model is used for the control design. The use of a dynamic model allows the controller to handle potential high slip conditions in the event of an emergency collision avoidance maneuver. An on-line linearization is performed analytically in order to use the SMPC approach presented in Section 4.2. Modeling and linearization errors are accounted for by an additive Gaussian disturbance in the model.

Reference generation: A simple reference generator which provides speed and lateral position set-points is used in this example. These set-points constitute the reference

state sequence $\{x_k^{\text{ref}}\}_{k=1}^T$. The proposed SMPC strategy can easily be extended to use the more complex driver models presented in Section 3.2 to generate a reference input sequence for the controller.

Environment model: The IMM-KF approach presented in Section 3.3 is used to estimate and predict the positions of target vehicles. Each target vehicle is represented by a kinematic point-mass model with an additive Gaussian disturbance to account for model mismatch. In this example, the longitudinal and lateral motion of each target vehicle are decoupled to reduce the complexity of the multiple model filter. Details of the motion models are presented in Carvalho et al. (2014). For the predictions, the dynamics corresponding to the most likely mode are used to propagate the state of each target vehicle and the associated uncertainty over the horizon.

Controller: We use the linear time-varying SMPC approach presented in Section 4.2. The collision avoidance constraints are formulated using the convex collision-checking approach of Schulman et al. (2013). Figure 14 shows the half-space representing the unsafe region for the depicted positions and orientations of the ego and target vehicles in two scenarios. The ego vehicle is labeled as E while the target vehicle is labeled as V . The dimensions of the target vehicle are suitably increased to account for a specified safety distance and the uncertainty in the target vehicle's position predicted by the environment model. Note that the constraint is on the ego vehicle's center of gravity (CoG). That is, the CoG of E must remain outside the gray shaded area in Figure 14 for a collision to be avoided.

Figure 14: Unsafe region for the depicted positions of the ego vehicle E and target vehicle V in two scenarios. The ego vehicle's center of gravity must lie outside the gray shaded area for a collision to be avoided.

Results

The scenario for the simulation is shown in Figure 15a. The ego vehicle E is moving at a speed of 10 m/s in the left lane with the objective of staying in the left lane and increasing its speed to 15 m/s. The target vehicles V_1 and V_2 are in the right lane at the start of the simulation. V_1 moves at a speed of 8 m/s and is in the process of changing its lane to the left lane. The lane change trajectory for V_1 is generated using a sigmoid function. V_2 moves in a straight line at a speed of 10.5 m/s. The SMPC problem is solved with a constraint violation probability $\varepsilon_k = 0.002$ in (37). The prediction horizon is $T = 20$ time steps at a sampling time of 0.1 s, resulting in a preview time of 2 s.

Snapshots of the simulation at various times are shown in Figure 15. In each sub-figure, blue and red circles depict the predicted positions of the ego and target vehicles, respectively. The predicted positions of E are obtained from the solution of the optimization problem, and those of V_1 and V_2 are given by the environment model.

Figure 15: Snapshots of simulation with the ego vehicle E in blue and the target vehicles V_1 and V_2 in red. The blue and red circles are the predicted positions of the ego and target vehicles, respectively, over the horizon.

As seen in Figure 15b, the environment model predicts a lane change maneuver for V_1 at $t = 2.5$ s. The potential constraint violation causes the controller to command a braking action and decrease the speed of E . To attain the objective of maintaining the reference speed, the controller attempts to plan a path around V_1 , but is prevented from doing so by the presence of V_2 (Figure 15c). When V_2 passes E and a sufficient safety distance is available, the controller plans and executes an overtaking maneuver as shown in Figures 15d and 15e. Finally, the controller plans a path in between V_1 and V_2 when feasible to cause E to return to its original lane and increase its speed (Figure 15f).

Risk vs conservatism in SMPC

An important element of the SMPC problem is the risk parameter ε_k , defined as the probability of violating the probabilistic safety constraints. In order to study the effect of varying ε_k on the performance of the controller, we repeat the simulation presented above with an increased value of $\varepsilon_k = 0.5$. This is equivalent to the nominal case where all disturbances are assumed to take on their mean values, as the amount of chance-constraint tightening in (40) is zero.

A snapshot of the simulation at $t = 4.5$ s is shown in Figure 16. The controller finds the more aggressive maneuver of changing lanes in between the two target vehicles to be feasible. This can be compared to the more conservative maneuver planned by the controller shown in Figure 15c, in which the vehicle stays behind V_1 and overtakes only when V_2 has passed.

In the context of urban driving, scenarios similar to the one described above are

Figure 16: Snapshot of simulation with the nominal controller ($\varepsilon_k = 0.5$) at $t = 4.5$ s. In this case, a more aggressive overtaking maneuver is found to be feasible by the controller, as compared to the maneuver shown in Figure 15c.

expected to occur often. A conservatively chosen risk parameter would result in the ego vehicle stopping or slowing down unnecessarily, leading to longer commute times. The key lies in finding the right balance between risk and conservatism.

5.5. Lane Change Assistance via Scenario MPC

In this example, the SCMPC approach discussed in Section 4.3 is applied for a lane change assistance system for autonomous highway driving. Given a lane change intention by the driver, the controller first determines whether it is safe to perform a lane change. If so, the controller executes the lane change maneuver such that the safety constraints are satisfied. More details can be found in Schildbach and Borrelli (2015). We introduce the following definitions:

- $x_t^v = [\xi_t, \eta_t, \phi_t, v_t]$ is the state of the ego vehicle at time t , and $u_t^v = [\delta_t, a_t]$ is the control input. The notation is introduced in Section 3.1. The application in this example is fully autonomous, hence, $u_t^v = u_t^c$.
- $x_{j,t}^e = [\xi_{j,t}^e, \eta_{j,t}^e, \phi_{j,t}^e, v_{j,t}^e]$ is the state of the j^{th} target at time t . The notation is similar to that for the ego vehicle. The position and orientation are expressed in the road-aligned coordinate frame introduced in Section 3.1.

The controller computes a sequence of control inputs $\{u_{t+k}^c\}_{k=0}^{T-1}$ to track the sequence of reference states $\{x_{t+k}^{\text{ref}}\}_{k=0}^T$.

Approach

Vehicle Model: The SCMPC approach is based on a kinematic bicycle model of the ego vehicle (see Section 3.1). For simplicity, no uncertainty in the ego vehicle's dynamics is assumed.

Reference generation: A simple reference generator is used, which sets the reference speed to v^{ref} . When a lane change is commanded by the driver, the reference generator sets the desired lateral position to η^{ref} . The aforementioned quantities constitute the reference state sequence $\{x_k^{\text{ref}}\}_{k=0}^T$.

Environment Model: The uncertainty in the environment at time t is represented by S sampled scenarios $\{\omega^{(s)}\}_{s=1}^S$ generated by the procedure described in Section 3.3. In this example, each scenario is a full-horizon sample given by $\omega^{(s)} = \{d_0^{e,(s)}, \dots, d_{T-1}^{e,(s)}\}$,

Figure 17: Construction of safety intervals for a particular scenario s . The controlled vehicle is shown in dark gray and three target vehicles in light gray. The hatched regions fall outside the safety intervals of the current lane $l = cl$ and the target lane $l = tl$, respectively.

resulting in a sampled sequence of environment states $\{x_k^{e,(s)}\}_{k=1}^T$ over the prediction horizon. The predicted environment states are used to formulate state constraints as described below.

State Constraints. In highway traffic, all drivers are required to maintain a minimum safety distance to the vehicle driving in front of them (California Department of Motor Vehicles, 2014, pp. 56 ff.)(Bundesministerium der Justiz und für Verbraucherschutz, 2013, §4(1)). In particular, the ego vehicle is responsible to maintain a safety distance d_{ev} to the target vehicle ahead of it. During a lane change maneuver, the ego vehicle must also respect the safety distance d_{tv} of the target vehicle that drives behind it in the target lane. The safety distances d_{ev} and d_{tv} depend on the relative speed between the ego and target vehicles. They may occasionally be violated if the target vehicles perform unusual maneuvers.

Given a particular scenario $\omega^{(s)}$, the predicted environment states $\{x_k^{e,(s)}\}_{k=1}^T$ can be translated into time-varying constraints on the longitudinal and lateral position of the ego vehicle. This is illustrated in Figure 17 for a given time step in the prediction horizon. The lanes are denoted by $l \in \{cl, tl\}$, where cl and tl are the current and target lanes, respectively. For a scenario $\omega^{(s)}$, the bounds on the lateral and longitudinal position of the ego vehicle in lane l are given by,

$$\xi_k \in [\xi_k^{\min,l,(s)}, \xi_k^{\max,l,(s)}] \quad (44a)$$

$$\eta_k \in [\eta_k^{\min,l,(s)}, \eta_k^{\max,l,(s)}]. \quad (44b)$$

The final limits on the longitudinal and lateral position in lane l are defined as the tightest safety interval over all scenarios, that is,

$$\xi_k^{\min,l} = \min_{s=1,\dots,S} \xi_k^{\min,l,(s)}, \quad \xi_k^{\max,l} = \max_{s=1,\dots,S} \xi_k^{\max,l,(s)} \quad (45a)$$

$$\eta_k^{\min,l} = \min_{s=1,\dots,S} \eta_k^{\min,l,(s)}, \quad \eta_k^{\max,l} = \max_{s=1,\dots,S} \eta_k^{\max,l,(s)}. \quad (45b)$$

In addition, the algorithm features an outer loop to find the earliest times k_{lc} at which a safe lane change is possible (that is, under all S scenarios). The bounds on ξ_k

Figure 18: Computed safe trajectories by SCMPC for three situations, showing the ego vehicle in gray and the target vehicles $j = 1, 2, 3, 4$ with their respective speeds v_1, v_2, v_3, v_4 . The future trajectories are indicated for some of the vehicles.

then correspond to the boundaries of the current lane for $k \leq k_{lc}$, and both the current and target lane when $k > k_{lc}$:

$$\xi_k^{\min} := \begin{cases} \xi_k^{\min, cl} & \text{if } k \leq k_{lc} \\ \max\{\xi_k^{\min, cl}, \xi_k^{\min, tl}\} & \text{if } k > k_{lc} \end{cases}, \quad (46a)$$

$$\xi_k^{\max} := \begin{cases} \xi_k^{\max, cl} & \text{if } k \leq k_{lc} \\ \min\{\xi_k^{\max, cl}, \xi_k^{\max, tl}\} & \text{if } k > k_{lc} \end{cases}. \quad (46b)$$

The bounds on η_k are computed similarly:

$$\eta_k^{\min} := \begin{cases} \eta_k^{\min, cl} & \text{if } k \leq k_{lc} \\ \min\{\eta_k^{\min, cl}, \eta_k^{\min, tl}\} & \text{if } k > k_{lc} \end{cases}, \quad (47a)$$

$$\eta_k^{\max} := \begin{cases} \eta_k^{\max, cl} & \text{if } k \leq k_{lc} \\ \max\{\eta_k^{\max, cl}, \eta_k^{\max, tl}\} & \text{if } k > k_{lc} \end{cases}. \quad (47b)$$

The resulting longitudinal and lateral state constraints can be summarized as

$$x_k^{\min} \leq x_k^v \leq x_k^{\max}. \quad (k = 1, \dots, T) \quad (48)$$

With the above definition of the state constraints, the optimization problem (26) becomes a deterministic non-linear program which can be solved by standard algorithms (Nocedal and Wright, 2006; Luenberger and Ye, 2008).

Without loss of generality, the simulations presume that the ego vehicle receives a lane change command at time $t = 0$ s. The prediction horizon is $T = 50$ steps at a sampling time of $t_s = 0.1$ s, giving the SCMPC a total preview time of 5 s. Eight candidate lane change times $t_{lc} \in \{0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5\}$ seconds are considered; i.e., eight instances of the optimization problem (26) must be solved. Clearly, the earliest lane change time is preferred.

The reference speed v^{ref} is set to the speed limit of 22.4 m/s, in order to advance as fast as possible on the highway. Small weights on the control inputs a and δ are used to make the control actions smooth and energy efficient. The number of scenarios has been selected as $S = 19$, corresponding to a violation probability of $\varepsilon_k = 5\%$ in the chance constraint (28).

Results

The safe trajectories for three different cases are shown in Figure 18. A sample scenario for the target vehicles is also depicted, to serve as a reference. The depicted scenario assumes constant speed (as indicated in Figure 18) and no lane changes for each of the target vehicles.

In Case 1, the ego vehicle attempts to overtake the slower target vehicle 1 in front of it. In order not to violate the front safety distance of target vehicle 4, the ego vehicle needs to accelerate before merging into the faster lane. The SCMPC finds that the earliest lane change time $t_{lc} = 1$ s, up to which the ego vehicle needs to accelerate in its own lane. Observe that the safe trajectory in Figure 18(a) remains on the original lane up to t_{lc} , when the lane change is initiated. The speed of the ego vehicle increases rapidly up to t_{lc} to enable a safe lane change. It increases more slowly thereafter, in order to move closer to the reference speed.

In Case 2, the ego vehicle intends to merge into a slower lane with target vehicle 2 limiting its speed. In order not to violate its own front safety distance, the ego vehicle needs to slow down before merging into the target lane. The SCMPC finds that a lane change is possible immediately, $t_{lc} = 0$ s, as can be observed from the safe trajectory in Figure 18(b). However, the ego vehicle has to brake for more than 1 s to maintain its front safety distance in the target lane. It adjusts to the speed of target vehicle 2 thereafter.

In Case 3, the ego vehicle attempts to merge into a faster lane, blocked by target vehicle 2 and target vehicle 4 going at higher speeds. The ego vehicle would need to accelerate before merging into the target lane. However, the SCMPC concludes that this is not possible within the preview time of 5 s. Therefore, the safe trajectory in Figure 18(c) remains in the original lane. Nonetheless, the ego vehicle has to brake quickly for more than 4 s in order to maintain its front safety distance to target vehicle, which has a lower speed.

6. Conclusions and Outlook

The experimental results in Section 5 show that the proposed approach can systematically solve the control design problem for autonomous and semi-autonomous vehicles. Data collected during driving is used to calibrate models and quantify their

uncertainty. Uncertain models are used by a SMPC scheme which takes into consideration computational complexity and safety constraints.

In the near future, the uncertainty associated with the environment and its prediction will be reduced with the use of better sensors, algorithms, and vehicle-to-vehicle and vehicle-to-infrastructure communication. Guaranteeing safety, reliability and a tractable computational complexity of the on-line computations are issues to be addressed.

Guaranteeing safety is known in the MPC community as the problem of persistent feasibility. It has been well-studied in literature (see Mayne et al. (2000) for a survey). A suitably chosen terminal cost function and robust control invariant (RCI) terminal constraint can ensure stability and persistent feasibility of the MPC problem. However, in automated driving, the computation of the RCI set is challenging due to the non-linear system dynamics, and the non-convex and time-varying state constraints. Outer-approximations of the terminal set based on conservative approximations (e.g. the vehicle in front applies full braking at the end of the prediction horizon) will lead to overly conservative behavior. Moreover, the approximations can only be scenario specific at best. In practice, as with other control design techniques, extensive simulations and experimental results are necessary for validating the safety of the resulting closed-loop scheme. Reducing the validation and verification time through the application of advanced theory would be invaluable to the automotive industry.

In general, a longer prediction horizon is desirable in order to be able to react early to dangerous situations. This comes at the price of increased computational complexity of the on-line optimization problem. General purpose optimization solvers such as NPSOL (Gill et al., 1986) can handle arbitrary cost functions, dynamics and constraints, but do not scale well with the prediction horizon. The reason is that such solvers use dense linear algebra and finite difference methods to approximate the gradient of the cost function and the Jacobian of the constraints. On the other hand, state-of-the-art non-linear solvers such as IPOPT (Wächter and Biegler, 2006) use a sparse formulation of the MPC problem, yielding algorithms that scale well with the horizon. However, such solvers require the aforementioned gradient and Jacobian to be provided by the user, which can be hard with certain choices of the dynamics and constraints. Automatic or algorithmic differentiation tools are useful for this purpose, but are restricted to systems satisfying some smoothness conditions.

The proposed modeling approach makes use of data collected during real driving. This implies that a vehicle would change its behavior after being sold, as it encounters different driving scenarios. The reliability of an autonomous or semi-autonomous system which evolves with time is a challenge for this industry. Continuous communication of the vehicle with the web and cloud computing could facilitate the implementation of this feature.

Acknowledgments

This material is based upon work partially supported by the National Science Foundation under award number 1239323 and the Office of Naval Research MURI N00014-13-1-0341. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the

National Science Foundation. The work was also supported by the Hyundai Center of Excellence at UC Berkeley, and by the Ford Motor Company.

References

- Anderson, S., Peters, S., Pilutti, T., Iagnemma, K., 2010. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios. *International Journal of Vehicle Autonomous Systems* 8 (2), 190–216.
- Aoude, G. S., Desaraju, V. R., Stephens, L. H., How, J. P., 2012. Driver behavior classification at intersections and validation on large naturalistic dataset. *IEEE Transactions on Intelligent Transportation Systems* 13 (2), 724–736.
- Bemporad, A., Borrelli, F., Morari, M., September 2003. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Transactions on Automatic Control* 48 (9), 1600–1606.
- Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society B* (57), 289–300.
- Blackmore, L., Ono, M., Bektassov, A., Williams, B. C., 2010. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics* 26(3), 502–516.
- Blackmore, L., Ono, M., Williams, B. C., 2011. Chance-constrained optimal path planning with obstacles. *IEEE Transactions on Robotics* 27 (6), 1080–1094.
- Broadhurst, A., Baker, S., Kanade, T., 2005. Monte carlo road safety reasoning. In: 2005 IEEE Intelligent Vehicles Symposium. pp. 319–324.
- Bundesministerium der Justiz und für Verbraucherschutz, 2013. Straßenverkehrs-Ordnung.
URL www.gesetze-im-internet.de/stvo.2013/index.html
- Caballero, Y., Bello, R., Taboada, A., Nowe, A., Garcia, M., Casas, G., 2006. A new measure based in the rough set theory to estimate the training set quality. In: *Proc. of International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*. pp. 133–140.
- Calafiore, G., 2010. Random convex programs. *SIAM Journal on Optimization* 20 (6), 3427–3464.
- Calafiore, G., Campi, M., May 2006. The scenario approach to robust control design. *IEEE Transactions on Automatic Control* 51 (5), 742–753.
- California Department of Motor Vehicles, 2014. California driver handbook.
URL apps.dmv.ca.gov/pubs/dl600.pdf

- Campi, M., Garatti, S., 2011a. A sampling-and-discarding approach to chance-constrained optimization: Feasibility and optimality. *Journal of Optimization Theory and Applications* 148 (2), 257–280.
- Campi, M. C., Garatti, S., 2008. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization* 19, 1211–1230.
- Campi, M. C., Garatti, S., 2011b. A sampling and discarding approach to chance-constrained optimization: Feasibility and optimality. *Journal of Optimization Theory and Applications* 148, 257–280.
- Cannon, M., Couchman, P., Kouvaritakis, B., 2007. MPC for stochastic systems. In: *Assessment and Future Directions of Nonlinear Model Predictive Control*. Vol. 358 of *Lecture Notes in Control and Information Sciences*. Springer, pp. 255–268.
- Cannon, M., Kouvaritakis, B., Ng, D., 2009a. Probabilistic tubes in linear stochastic model predictive control. *Systems & Control Letters* 58 (10-11), 747 – 753.
- Cannon, M., Ng, D., Kouvaritakis, B., 2009b. Successive linearization NMPC for a class of stochastic nonlinear systems. In: *Nonlinear Model Predictive Control*. Vol. 384 of *Lecture Notes in Control and Information Sciences*. Springer, pp. 249–262.
- Carvalho, A., Gao, Y., Gray, A., Tseng, H. E., Borrelli, F., 2013. Predictive control of an autonomous ground vehicle using an iterative linearization approach. In: *16th IEEE Conference on Intelligent Transportation Systems*. pp. 2335 – 2340.
- Carvalho, A., Gao, Y., Lefèvre, S., Borrelli, F., 2014. Stochastic predictive control of autonomous vehicles in uncertain environments. In: *12th International Symposium on Advanced Vehicle Control*.
- Constantinescu, E., Zavala, V., Rocklin, M., Lee, S., Anitescu, M., 2011. A computational framework for uncertainty quantification and stochastic optimization in unit commitment with wind power generation. *IEEE Transactions on Power Systems* 26 (1), 431–441.
- Cortes, C., Jackel, L. D., Chiang, W.-P., 1995. Limits on learning machine accuracy imposed by data quality. In: *Proc. of Advances in Neural Information Processing Systems*. pp. 239–246.
- Eidehall, A., Petersson, L., 2008. Statistical threat assessment for general road scenes using Monte Carlo sampling. *IEEE Transactions on Intelligent Transportation Systems* 9(1), 137–147.
- Gao, Y., Gray, A., Frasca, J., Lin, T., Tseng, E., Hedrick, J., Borrelli, F., 2012. Spatial predictive control for agile semi-autonomous ground vehicles. In: *11th International Symposium on Advanced Vehicle Control*.
- Gao, Y., Gray, A., Tseng, H. E., Borrelli, F., 2014. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics* 52 (6), 802–823.

- Gao, Y., Lin, T., Borrelli, F., Tseng, E., Hrovat, D., 2010. Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads. In: 3rd Annual ASME Conference on Dynamic Systems and Controls. pp. 265–272.
- Gill, P. E., Murray, W., Saunders, M. A., Wright, M. H., 1986. User's guide for NPSOL (version 4.0): A Fortran package for nonlinear programming. Tech. rep., DTIC Document.
- Government of South Australia, 2015. The driver's handbook: Care, courtesy, common Sense.
URL mylicence.sa.gov.au/road-rules/the-drivers-handbook/care-courtesy
- Gray, A., Ali, M., Gao, Y., Hedrick, J. K., Borrelli, F., 2012. Integrated threat assessment and control design for roadway departure avoidance. In: 15th International IEEE Conference on Intelligent Transportation Systems. IEEE, pp. 1714–1719.
- Gray, A., Gao, Y., Hedrick, J., Borrelli, F., 2013a. Robust predictive control for semi-autonomous vehicles with an uncertain driver model. In: 2013 IEEE Intelligent Vehicles Symposium. pp. 208–213.
- Gray, A., Gao, Y., Lin, T., Hedrick, J., Borrelli, F., 2013b. Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model. In: 16th IEEE Conference on Intelligent Transportation Systems. pp. 2329–2334.
- Hindiyeh, R., 2013. Dynamics and control of drifting in automobiles. Ph.D. thesis, Stanford University.
- Hindiyeh, R., Gerdes, J., 2009. Equilibrium analysis of drifting vehicles for control design. In: 2nd Annual ASME Conference on Dynamic Systems and Controls. pp. 181–188.
- Hokayem, P., Cinquemani, E., Chatterjee, D., Ramponi, F., Lygeros, J., 2010. Stochastic receding horizon control with output feedback and bounded control inputs. In: 49th IEEE Conference on Decision and Control. pp. 6095–6100.
- Hong, S., Lee, C., Borrelli, F., Hedrick, J., 2014. A novel approach for vehicle inertial parameter identification using a dual kalman filter. IEEE Transactions on Intelligent Transportation Systems PP (99), 1–11.
- Jansson, J., 2005. Collision avoidance theory with application to automotive collision mitigation. Ph.D. dissertation, Linköping University, Linköping, Sweden.
- Khatib, O., 1986. Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research 5 (1), 90–98.
- Kong, J., Pfeiffer, M., Schildbach, G., Borrelli, F., 2015. Autonomous driving using model predictive control and a kinematic bicycle vehicle model, Accepted to 2015 IEEE Intelligent Vehicles Symposium.
- Kouvaritakis, B., Cannon, M., Raković, S. V., Cheng, Q., 2010. Explicit use of probabilistic distributions in linear predictive control. Automatica 46 (10), 1719–1724.

- Kouvaritakis, B., Cannon, M., Yadlin, R., 2011. On the centres and scalings of robust and stochastic tubes. United Kingdom Automatic Control Council (UKACC) International Conference on Control.
- Krajzewicz, D., 2010. Traffic simulation with SUMO - Simulation of urban mobility. In: Fundamentals of Traffic Simulation. No. 145 in International Series in Operations Research & Management Science. Springer New York, pp. 269–293.
- Kumar, P., Perrollaz, M., Lefèvre, S., Laugier, C., 2013. Learning-based approach for online lane change intention prediction. In: 2013 IEEE Intelligent Vehicles Symposium. pp. 797–802.
- Labayrade, R., Royere, C., Aubert, D., 2005. A collision mitigation system using laser scanner and stereovision fusion and its assessment. In: 2005 IEEE Intelligent Vehicles Symposium. pp. 441–446.
- Laugier, C., Paromtchik, I. E., Perrollatz, M., Yong, M., Yoder, J.-D., Tay, C., Mekhnacha, K., Nègre, A., 2011. Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety. IEEE Intelligent Transportation Systems Magazine 2(4), 4–19.
- Lee, J., Cooley, B., 1998. Optimal feedback control strategies for state-space systems with stochastic parameters. IEEE Transactions on Automatic Control 43 (10), 1469–1475.
- Lefèvre, S., Vasquez, D., Laugier, C., 2014. A survey on motion prediction and risk assessment for intelligent vehicles. Robomech Journal 1(1), 1–14.
- Lefèvre, S., Carvalho, A., Borrelli, F., 2015. Autonomous car following: A learning-based approach, Accepted to 2015 IEEE Intelligent Vehicles Symposium.
- Lefèvre, S., Gao, Y., Vasquez, D., Tseng, E., Bajcsy, R., Borrelli, F., 2014a. Lane keeping assistance with learning-based driver model and model predictive control. In: 12th International Symposium on Advanced Vehicle Control.
- Lefèvre, S., Sun, C., Bajcsy, R., Laugier, C., 2014b. Comparison of parametric and non-parametric approaches for vehicle speed prediction. In: 2014 American Control Conference. pp. 3494–3499.
- Luenberger, D., Ye, Y., 2008. Linear and Nonlinear Programming, 3rd Edition. Springer.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., Scokaert, P. O., 2000. Constrained model predictive control: Stability and optimality. Automatica 36 (6), 789–814.
- Mazor, E., Averbuch, A., Bar-Shalom, Y., Dayan, J., 1998. Interacting multiple model methods in target tracking: A survey. IEEE Transactions on Aerospace and Electronic Systems 34 (1), 103–123.
- McCall, J., Trivedi, M., 2007. Driver behavior and situation aware brake assistance for intelligent vehicles. Proceedings of the IEEE 95 (2), 374–387.

- Mesbah, A., Streif, S., Findeisen, R., Braatz, R., 2014. Stochastic nonlinear model predictive control with probabilistic constraints. In: 2014 American Control Conference. pp. 2413–2419.
- Morris, B., Doshi, A., Trivedi, M., 2011. Lane change intent prediction for driver assistance: on-road design and evaluation. In: 2011 IEEE Intelligent Vehicles Symposium. pp. 895–901.
- Musicant, O., Bar-Gera, H., E, S., 2011. Individual driver’s undersirable driving events - A temporal analysis. In: International Conference on Road Safety and Simulation. pp. 14–16.
- Nocedal, J., Wright, S. J., 2006. Numerical Optimization, 2nd Edition. Springer Verlag.
- Pacejka, H., 2005. Tire and vehicle dynamics. Elsevier.
- Pomerleau, D. A., 1989. Alvin: An autonomous land vehicle in a neural network. In: Advances in Neural Information Processing Systems. pp. 305–313.
- Primbs, J., 2007. A soft constraint approach to stochastic receding horizon control. In: 46th IEEE Conference on Decision and Control. pp. 4797–4802.
- Rajamani, R., 2006. Vehicle Dynamics and Control. Springer.
- Salvucci, D. D., Gray, R., 2004. A two-point visual control model of steering. Perception 33 (10), 1233–1248.
- Schildbach, G., Borrelli, F., 2015. Scenario model predictive control for lane change assistance on highways, Accepted to 2015 IEEE Intelligent Vehicles Symposium.
- Schildbach, G., Fagiano, L., Frei, C., Morari, M., 2014. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. Automatica 50(12), 3009–3018.
- Schildbach, G., Fagiano, L., Morari, M., 2013. Randomized solutions to convex programs with multiple chance constraints. SIAM Journal on Optimization 23(4), 2479–2501.
- Schouwenaars, T., De Moor, B., Feron, E., How, J., 2001. Mixed integer programming for multi-vehicle path planning. In: 2001 European Control Conference. Vol. 1. pp. 2603–2608.
- Schulman, J., Ho, J., Lee, A., Awwal, I., Bradlow, H., Abbeel, P., 2013. Finding locally optimal, collision-free trajectories with sequential convex optimization. In: Robotics: Science and Systems. Vol. 9. pp. 1–10.
- Shia, V., Gao, Y., Vasudevan, R., Campbell, K., Lin, T., Borrelli, F., Bajcsy, R., 2014. Semiautonomous vehicular control using driver modeling. IEEE Transactions on Intelligent Transportation Systems 15 (6), 2696–2709.

- Skaf, J., Boyd, S., November 2010. Design of affine controllers via convex optimization. *IEEE Transactions on Automatic Control* 55 (11), 2476–2487.
- State of New Jersey Motor Vehicle Commission, 2015. The New Jersey driver manual. URL state.nj.us/mvc/pdf/Manuals/drivermanual.pdf
- Streif, S., Karl, M., Mesbah, A., 2014. Stochastic nonlinear model predictive control with efficient sample approximation of chance constraints. *arXiv preprint arXiv:1410.4535*.
- Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E* 62 (2), 1805–1824.
- van Hessem, D., Bosgra, O., 2002. A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints. In: 41st IEEE Conference on Decision and Control. Vol. 4. pp. 4643–4648.
- Van Hessem, D., Bosgra, O., 2002. A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints. In: *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*. Vol. 4. IEEE, pp. 4643–4648.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1185110
- van Hessem, D., Scherer, C., Bosgra, O., 2001. LMI-based closed-loop economic optimization of stochastic process operation under state and input constraints. In: 40th IEEE Conference on Decision and Control. Vol. 5. pp. 4228–4233.
- Wächter, A., Biegler, L. T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* 106 (1), 25–57.
- Wiest, J., Höffken, M., Kreßel, U., Dietmayer, K., 2012. Probabilistic trajectory prediction using gaussian mixture models. In: 2012 IEEE Intelligent Vehicles Symposium. pp. 141–146.
- Yan, F., Wang, J., Huang, K., 2012. Hybrid electric vehicle model predictive control torque-split strategy incorporating engine transient characteristics. *IEEE Transactions on Vehicular Technology* 61 (6), 2458–2467.
- Zavala, V., 2013. Real-time optimization strategies for building systems. *Industrial & Engineering Chemistry Research* 52 (9), 3137–3150.
- Zavala, V. M., Constantinescu, E. M., Krause, T., Anitescu, M., 2009. On-line economic optimization of energy systems using weather forecast information. *Journal of Process Control* 19, 1725–1736.