Stochastic Model Predictive Control with a Safety Guarantee for Automated Driving

Tim Brüdigam, Michael Olbrich, Dirk Wollherr, Marion Leibold

***Abstract*—Automated vehicles require efficient and safe planning to maneuver in uncertain environments. Largely this uncertainty is caused by other traffic participants, e.g., surrounding vehicles. Future motion of surrounding vehicles is often difficult to predict. Whereas robust control approaches achieve safe, yet conservative motion planning for automated vehicles, Stochastic Model Predictive Control (SMPC) provides efficient planning in the presence of uncertainty. Probabilistic constraints are applied to ensure that the maximal risk remains below a predefined level. However, safety cannot be ensured as probabilistic constraints may be violated, which is not acceptable for automated vehicles. Here, we propose an efficient trajectory planning framework with safety guarantees for automated vehicles. SMPC is applied to obtain efficient vehicle trajectories for a finite horizon. Based on the first optimized SMPC input, a guaranteed safe backup trajectory is planned using reachable sets. This backup is used to overwrite the SMPC input if necessary for safety. Recursive feasibility of the safe SMPC algorithm is proved. Highway simulations show the effectiveness of the proposed method regarding performance and safety.**

***Index Terms*—model predictive control, stochastic model predictive control, failsafe trajectory planning, automated vehicles**

I. Introduction

Within the past decades, research has made significant progress in the area of self-driving cars. A majority of road accidents is still caused by human errors, therefore, increasing the level of vehicle autonomy has great potential to reduce the overall number of accidents.

The safety of automated vehicles depends on the ability of the vehicle control algorithm to handle uncertainty of other traffic participants and the environment. While there are various control methods to plan vehicle trajectories, Model Predictive Control (MPC) has proved to be a suitable approach by iteratively solving an optimal control problem on a finite prediction horizon. Uncertainties in the prediction model are addressed by Robust Model Predictive Control (RMPC) [1].

RMPC approaches were designed for trajectory planning in automated vehicles [2], [3], however, robustly accounting for uncertainty yields conservative vehicle behavior. Conservatism resulting from robustly handling uncertainty in MPC is reduced by Stochastic Model Predictive Control (SMPC) [4], [5], where robust constraints are reformulated into probabilistic constraints. This probabilistic reformulation enables optimistic trajectory planning in a majority of scenarios, but it also allows a small probability of constraint violation, i.e, a probability of collision for vehicles [6], [7].

In comparison to SMPC, trajectory planning based on reachability analysis provides formal safety guarantees [8], [9]. Here, worst-case predictions are obtained for other surrounding vehicles in order to plan fail-safe vehicle trajectories, referred to as fail-safe trajectory planning (FTP).

In this work, we tackle the challenge of planning efficient and safe trajectories for automated vehicles. We present a novel MPC trajectory planner, which combines the advantages of SMPC and fail-safe trajectory planning for environments with uncertainty. A trajectory is planned with SMPC, providing optimistic and efficient planning. In a regular setting, the first optimized SMPC input is then applied to the vehicle and a new SMPC optimal control problem is solved at the next time step with a shifted horizon. In addition to SMPC, for every time step a fail-safe trajectory is planned, based on the first optimized SMPC input. The optimistic SMPC input is only applied to the vehicle if it is still possible to find a fail-safe backup trajectory after having applied the first SMPC input. This ensures that the efficient SMPC trajectory is executed as long as a backup exists, therefore guaranteeing safety. The proposed method is referred to as *Stochastic Model Predictive Control + fail-safe trajectory planning* (SMPC+FTP).

The contributions of this work are as follows.

- Novel SMPC+FTP method providing efficient and safe trajectory planning including lane change decisions.
- Proof of recursive feasibility of the SMPC+FTP method.
- Simulation study with complex highway traffic situations.

The proposed SMPC+FTP ensures safety for the vehicle while exploiting the benefits of efficient SMPC trajectory planning. The design of the SMPC+FTP method guarantees recursive feasibility, i.e., if a solution exists at a time step, it is guaranteed that a solution also exists at the next time step. A simulation study of two complex scenarios demonstrates the benefits of optimistic trajectory planning in a regular highway scenario, while the ability of SMPC+FTP to guarantee safety is shown in an emergency scenario.

An extended version of this work is available [10], providing more detailed derivations and analyses.

A. Related Work

Trajectory planning for automated vehicles is a widely studied research area. There are various methods in non-MPC related fields, such as using partially observable Markov

The authors gratefully acknowledge the financial and scientific support by the BMW Group.

T. Brüdigam, D. Wollherr, and M. Leibold are with the Chair of Automatic Control Engineering at the Technical University of Munich, Munich, Germany (email: {tim.bruedigam; dw; marion.leibold}@tum.de).

M. Olbrich is with the Department of Computer Science at the University of Augsburg, Augsburg, Germany (email: michael.olbrich@informatik.uni-augsburg.de).

decision processes (POMDP) [11] or reinforcement learning [12]. Learning based methods are also popular for autonomous racing [13]–[16]. When considering automated road vehicles, MPC is for example used for maneuver and trajectory planning in [17], [18].

The main focus of this work is trajectory planning with SMPC and FTP. Fail-safe trajectory planning is defined as planning collision-free vehicle trajectories, accounting for any legal future motion of surrounding vehicles [19]. For bounded uncertainties in real-world applications, FTP is applied based on worst-case uncertainty realizations. Combined with reachability analysis, formal safety guarantees are given [9]. The computation of these reachable sets is connected to control invariant sets in RMPC as stated in [20]. An approach to include reachability analysis into MPC is given in [21].

In [22] a method is proposed to compute the set of all future locations possibly occupied by traffic participants. The remaining safe space is admissible to plan emergency trajectories. This FTP is presented in [8]. First, given the most likely motion of surrounding vehicles, an optimal trajectory is determined. Then, an emergency trajectory is connected to the last point of the optimal trajectory. The fail-safe trajectory is generated in such a way that the controlled vehicle comes to a standstill. In [19] an FTP method is introduced that generates fail-safe trajectories in real-time. The method is tested in various simulations based on the CommonRoad benchmark framework [23]. A motion planning framework is introduced in [24], which combines reachability analysis with optimization-based trajectory planning.

SMPC has been intensively studied in the context of automated vehicles. These works focus on the trade-off between risk and conservatism, defined by probabilistic constraints, so called chance constraints [25]. A major challenge in SMPC is reformulating the probabilistic chance constraint into a tractable constraint, which can be handled by a solver.

An SMPC particle approach is shown in [26] with a simple vehicle braking scenario, where particles approximate the uncertainty. An SMPC trajectory planner for automated vehicles in the presence of fixed obstacles is presented in [27]. In [6] vehicle trajectories are planned with SMPC based on the most likely prediction for surrounding vehicles, assuming Gaussian uncertainty. Varying risk parameters are studied, illustrating the trade-off between risk and conservatism. In [28] an SMPC lane change controller is presented, where the lane change risk is considered using predicted time-to-collision.

A different SMPC approach is utilized in [7], [29], focusing on Scenario Stochastic Model Predictive Control (SCMPC). In SCMPC samples of the uncertainty are drawn, which must then satisfy the constraints to find a tractable chance constraint expression. Arbitrary probability distributions are handled by SCMPC, while standard SMPC usually requires Gaussian distributions to analytically reformulate the chance constraint. While [29] focuses on simple lane change scenarios, the work is extended in [7] and experimental results are presented.

A combination of SMPC and SCMPC is given in [30], exploiting the individual advantages of SMPC and SCMPC. A further approach to SMPC is presented in [31], where a grid-based SMPC method is applied to plan vehicle trajectories, based on occupancy grids [32], [33].

In summary, SMPC approaches provide efficient vehicle trajectories for the majority of uncertainty realizations in regular situations. However, for unlikely uncertainty realizations, safety issues occur.

In this work, the benefit of efficiently planning trajectories with SMPC is combined with the safety guarantee of FTP. The FTP in this work is inspired by the ideas of [8], [19], [22]. In the following, SMPC and FTP are introduced. Then, the proposed SMPC+FTP method is derived in detail.

II. Preliminaries

In the following, we briefly introduce the general MPC optimal control problems (OCPs) for SMPC and FTP.

MPC iteratively solves an OCP with a finite prediction horizon N subject to input and state constraints. After solving the MPC OCP, only the first input $\boldsymbol{u}_0$ of the optimized input sequence $\boldsymbol{U} = (\boldsymbol{u}_0, \dots, \boldsymbol{u}_{N-1})^\top$ is applied. At the next time step, the updated MPC OCP is solved again. We distinguish between regular time steps h and prediction steps k within the MPC OCP. If clear from context, we omit the time step h, at which the MPC OCP is computed. In the following, we only explicitly denote the prediction time step k.

A. SMPC with Chance Constraints

While standard MPC considers hard constraints, this is problematic if uncertainties are present. Hard constraints subject to uncertainty can be considered by chance constraints. This yields the SMPC OCP

$$V^* = \min_{\boldsymbol{U}} \sum_{k=0}^{N-1} l(\boldsymbol{\xi}_k, \boldsymbol{u}_k) + V_\mathrm{f}(\boldsymbol{\xi}_N) \tag{1a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{f}\left(\boldsymbol{\xi}_k, \boldsymbol{u}_k\right) \tag{1b}$$

$$\boldsymbol{u}_k \in \mathcal{U}_k, \quad \boldsymbol{\xi}_{k+1} \in \boldsymbol{\Xi}_{k+1} \quad \forall k \in \{0, \dots, N-1\} \tag{1c}$$

$$\Pr\left(\boldsymbol{\xi}_k \in \boldsymbol{\Xi}'_{k,\text{safe}}(\boldsymbol{w})\right) \geq \beta \ \ \forall k \in \{1, \dots, N\} \tag{1d}$$

with prediction step k, states $\boldsymbol{\xi}_k$, system dynamics $\boldsymbol{f}$, and the normally distributed, zero mean uncertainty $\boldsymbol{w} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\Sigma}^w\right)$ with covariance matrix $\boldsymbol{\Sigma}^w$. The cost function consists of the stage cost $l(\boldsymbol{\xi}_k, \boldsymbol{u}_k)$ and the terminal cost $V_\mathrm{f}\left(\boldsymbol{\xi}_N\right)$. States and inputs are bounded by the state and input constraint sets $\boldsymbol{\Xi}_k$ and $\mathcal{U}_k$, respectively, and the safety constraint $\boldsymbol{\Xi}'_{k,\text{safe}}(\boldsymbol{w})$ depends on the uncertainty $\boldsymbol{w}$. The probabilistic chance constraint is given by (1d). The safety constraint $\boldsymbol{\xi}_k \in \boldsymbol{\Xi}'_{k,\text{safe}}(\boldsymbol{w})$ is required to hold according to the risk parameter β. For $\beta < 1$ a non-zero constraint violation probability is therefore allowed.

B. Fail-safe Trajectory Planning

We also consider an MPC OCP for FTP, i.e., a fail-safe MPC OCP. In contrast to SMPC, FTP considers the worst-case realizations of the uncertainty, resulting in safe, yet conservative optimized inputs. While the general OCP remains similar to (1), the chance constraint (1d) is replaced by

$$\boldsymbol{\xi}_k \in \boldsymbol{\Xi}_{k,\text{safe}}(\boldsymbol{w}) \quad \forall k \in \{1, \dots, N-1\} \tag{2a}$$

$$\boldsymbol{\xi}_N \in \boldsymbol{\Xi}_{N,\text{safe}}(\boldsymbol{w}). \tag{2b}$$

The FTP safe set $\boldsymbol{\Xi}_{k,\text{safe}}(\boldsymbol{w})$ is constructed based on reachability analysis to ensure formal safety guarantees. In addition to constraint (2a), a terminal constraint (2b) is required, which ensures that the terminal prediction state $\boldsymbol{\xi}_N$ allows to remain in a safe state beyond the prediction horizon. Based on this safe terminal set $\boldsymbol{\Xi}_{N,\text{safe}}(\boldsymbol{w})$ it is guaranteed that there exist system inputs $\boldsymbol{u}_{k^+}$ with $k^+ > N$ resulting in safe states $\boldsymbol{\xi}_{k^+}$.

III. Vehicle Models

MPC requires a system model for the controlled vehicle, known as the ego vehicle (EV), and surrounding vehicles, referred to as target vehicles (TVs), in order to predict future states within the OCP.

A. Ego Vehicle Model

We use a kinematic bicycle model to predict the EV states on a finite horizon, as suggested in [34]. The continuous-time system is given by

$$\dot{s} = v\cos(\phi+\alpha), \tag{3a}$$
$$\dot{d} = v\sin(\phi+\alpha), \tag{3b}$$
$$\dot{\phi} = \frac{v}{l_\text{r}}\sin\alpha, \tag{3c}$$
$$\dot{v} = a, \tag{3d}$$
$$\alpha = \arctan\left(\frac{l_\text{r}}{l_\text{r}+l_\text{f}}\tan\delta\right), \tag{3e}$$

where l_r and l_f are the distances from the vehicle center of gravity to the rear and front axles, respectively. The state vector is $\boldsymbol{\xi} = [s, d, \phi, v]^\top$ and the input vector is $\boldsymbol{u} = [a, \delta]^\top$. The vehicle velocity is given by v, acceleration and steering angle are denoted by a and δ, respectively. We consider the longitudinal position s of the vehicle along the road, the lateral vehicle deviation d from the centerline of the right lane, and the orientation ϕ of the vehicle with respect to the road. The nonlinear vehicle model (3) is summarized as $\dot{\boldsymbol{\xi}} = \boldsymbol{f}^\text{c}(\boldsymbol{\xi}, \boldsymbol{u})$.

Each MPC OCP is initialized with a linearization of the nonlinear prediction model (3) around the current vehicle state $\boldsymbol{\xi}^* = \boldsymbol{\xi}_0$ and the input $\boldsymbol{u}^* = [0, 0]^\top$. Selecting a non-zero reference input $\boldsymbol{u}^*$ often results in large differences $\Delta\boldsymbol{u} = \boldsymbol{u}_k - \boldsymbol{u}^*$ for prediction steps far ahead, increasing the inaccuracy of the linearization. The linearized continuous-time vehicle model is then given by

$$\dot{\boldsymbol{\xi}}^* + \Delta\dot{\boldsymbol{\xi}} = \boldsymbol{f}^\text{c}(\boldsymbol{\xi}^*, \boldsymbol{0}) + \boldsymbol{A}_\text{l}(\boldsymbol{\xi} - \boldsymbol{\xi}^*) + \boldsymbol{B}_\text{l}\boldsymbol{u} \tag{4}$$

with the Jacobian matrices

$$\boldsymbol{A}_\text{l} = \left[\frac{\partial \boldsymbol{f}^\text{c}}{\partial \boldsymbol{\xi}}\right]\Bigg|_{(\boldsymbol{\xi}^*, \boldsymbol{u}^*)}, \qquad \boldsymbol{B}_\text{l} = \left[\frac{\partial \boldsymbol{f}^\text{c}}{\partial \boldsymbol{u}}\right]\Bigg|_{(\boldsymbol{\xi}^*, \boldsymbol{u}^*)}. \tag{5}$$

A discrete-time model is required for MPC, therefore the linearized prediction model (4) is discretized with sampling time T. This yields the discrete states $\boldsymbol{\xi}_k = [s_k, d_k, \phi_k, v_k]^\top$ and inputs $\boldsymbol{u}_k = [a_k, \delta_k]^\top$ for prediction step k, as well as the linearized, discretized system

$$\boldsymbol{\xi}_{k+1} = \boldsymbol{\xi}_0 + T\boldsymbol{f}^\text{c}(\boldsymbol{\xi}_0, \boldsymbol{0}) + \boldsymbol{A}_\text{d}(\boldsymbol{\xi}_k - \boldsymbol{\xi}_0) + \boldsymbol{B}_\text{d}\boldsymbol{u}_k \tag{6a}$$
$$= \boldsymbol{f}^\text{d}(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k, \boldsymbol{u}_k) \tag{6b}$$

where $\boldsymbol{A}_\text{d}$ and $\boldsymbol{B}_\text{d}$ are matrices of the linearized system obtained from $\boldsymbol{A}_\text{l}$, $\boldsymbol{B}_\text{l}$ with zero-order hold. The nonlinear term $\boldsymbol{f}^\text{c}(\boldsymbol{\xi}^*, \boldsymbol{u}^*)$ in (4) is approximated by a forward Euler method since $\boldsymbol{\xi}_0$ is known. The linearized, discretized matrices $\boldsymbol{A}_\text{d}$ and $\boldsymbol{B}_\text{d}$ are given in the extended version [10]. In the following, for $k = 0$ in (6), i.e., $\boldsymbol{\xi}_k = \boldsymbol{\xi}_0$, the argument $\boldsymbol{\xi}_0$ is only mentioned once, i.e., $\boldsymbol{f}^\text{d}(\boldsymbol{\xi}_0, \boldsymbol{\xi}_0, \boldsymbol{u}_0)$ is abbreviated as $\boldsymbol{f}^\text{d}(\boldsymbol{\xi}_0, \boldsymbol{u}_0)$.

The following sections derive an SMPC method and constraints to avoid collisions with surrounding vehicles. However, even if no other vehicles are present, certain constraints are required. Acceleration and steering angle are bounded by

$$\boldsymbol{u}_\text{min} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_\text{max} \tag{7a}$$
$$\Delta\boldsymbol{u}_\text{min} \leq \Delta\boldsymbol{u}_k \leq \Delta\boldsymbol{u}_\text{max} \tag{7b}$$

with $\Delta\boldsymbol{u}_{k+1} = \boldsymbol{u}_{k+1} - \boldsymbol{u}_k$ and $\boldsymbol{u}_\text{max} = [a_\text{max}, \delta_\text{max}]^\top$, $\boldsymbol{u}_\text{min} = [a_\text{min}, \delta_\text{min}]^\top$. Further, road and velocity constraints are considered, resulting in

$$d_k \in \mathcal{D}^\text{lane} \tag{8a}$$
$$0 \leq v_k \leq v_\text{max} \tag{8b}$$

where $\mathcal{D}^\text{lane}$ represents road boundaries and v_max is the maximal velocity. Negative velocities are not allowed, i.e., $v_k \geq 0$.

In the following, we refer to input constraints by the set of admissible inputs $\mathcal{U}$ and state constraints are denoted by the set of admissible states $\boldsymbol{\Xi}$.

B. Target Vehicle Model

In order to avoid collisions, the EV is also required to predict the future states of surrounding TVs. The prediction model for the TVs used by the EV is a linear, discrete-time point-mass model given by

$$\boldsymbol{\xi}_{k+1}^\text{TV} = \boldsymbol{A}\boldsymbol{\xi}_k^\text{TV} + \boldsymbol{B}\boldsymbol{u}_k^\text{TV} \tag{9a}$$
$$\boldsymbol{u}_k^\text{TV} = \tilde{\boldsymbol{u}}_k^\text{TV} + \boldsymbol{w}_k^\text{TV} \tag{9b}$$

where $\boldsymbol{\xi}_k^\text{TV} = [x_k^\text{TV}, v_{x,k}^\text{TV}, y_k^\text{TV}, v_{y,k}^\text{TV}]^\top$ is the TV state with longitudinal position and velocity x_k^TV, $v_{x,k}^\text{TV}$ and lateral position and velocity y_k^TV, $v_{y,k}^\text{TV}$. The linear TV model allows to propagate the uncertainty, which is necessary for the MPC approach in the following sections. The TV model used in this work is only one possible option. Other linear TV prediction models can be utilized.

The system and input matrices are

$$\boldsymbol{A} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0.5T^2 & 0 \\ T & 0 \\ 0 & 0.5T^2 \\ 0 & T \end{bmatrix} \tag{10}$$

with sampling time T. The TV input consists of a feedback controller $\tilde{\boldsymbol{u}}_k^\text{TV}$ and a perturbation on the input, which is assumed to be an independent, identically distributed disturbance vector $\boldsymbol{w}_k^\text{TV}$. This setup assumes that the TV is following a given reference while deviations are allowed. The TV feedback controller is given by

$$\tilde{\boldsymbol{u}}_k^\text{TV} = \boldsymbol{K}\left(\boldsymbol{\xi}_k^\text{TV} - \boldsymbol{\xi}_{\text{ref},k}^\text{TV}\right) \tag{11}$$

with the TV reference $\boldsymbol{\xi}_{\text{ref},k}^\text{TV}$. The feedback matrix $\boldsymbol{K}$ is obtained by a linear-quadratic regulator strategy. If the TV input

computed by (11) exceeds the limits $\boldsymbol{u}_{\max}^{\mathrm{TV}} = [a_{\max}, a_{y,\max}]^\top$ and $\boldsymbol{u}_{\min}^{\mathrm{TV}} = [a_{\min}, a_{y,\min}]^\top$, summarized as $\mathcal{U}^{\mathrm{TV}}$, the TV inputs are bounded to satisfy $\mathcal{U}^{\mathrm{TV}}$.

We assume that $\boldsymbol{w}_k^{\mathrm{TV}}$ is subject to a Gaussian distribution with zero mean and covariance matrix $\Sigma_{\boldsymbol{w}}^{\mathrm{TV}}$, which is denoted by $\boldsymbol{w}_k^{\mathrm{TV}} \sim \mathcal{N}\left(0, \Sigma_{\boldsymbol{w}}^{\mathrm{TV}}\right)$. We also consider sensor noise in the measurement of the TV state, i.e.,

$$\hat{\boldsymbol{\xi}}_0^{\mathrm{TV}} = \boldsymbol{\xi}_0^{\mathrm{TV}} + \boldsymbol{w}_0^{\mathrm{sens}} \tag{12}$$

where $\hat{\boldsymbol{\xi}}_0^{\mathrm{TV}}$ is the measured initial state of the TV by the EV. The sensor noise $\boldsymbol{w}_0^{\mathrm{sens}} = [w_{0,x}^{\mathrm{sens}}, w_{0,v_x}^{\mathrm{sens}}, w_{0,y}^{\mathrm{sens}}, w_{0,v_y}^{\mathrm{sens}}]^\top$ is assumed to be a truncated Gaussian noise with $\boldsymbol{w}_0^{\mathrm{sens}} \sim \mathcal{N}\left(0, \Sigma^{\mathrm{sens}}\right)$ and $\boldsymbol{w}_0^{\mathrm{sens}} \in \mathcal{W}^{\mathrm{sens}}$, where $\mathcal{W}^{\mathrm{sens}}$ is a compact, convex and bounded set.

IV. Stochastic Model Predictive Control with Safety Guarantee

SMPC and fail-safe trajectory planning both have their individual advantages, i.e., efficient trajectories in an uncertain environment and guaranteed safe motion planning, respectively. In the following, we present a combined SMPC and FTP framework, SMPC+FTP, which exploits advantages of both methods to plan efficient and safe trajectories for autonomous vehicles. This section introduces the setup of the SMPC+FTP framework and gives a proof for recursive feasibility.

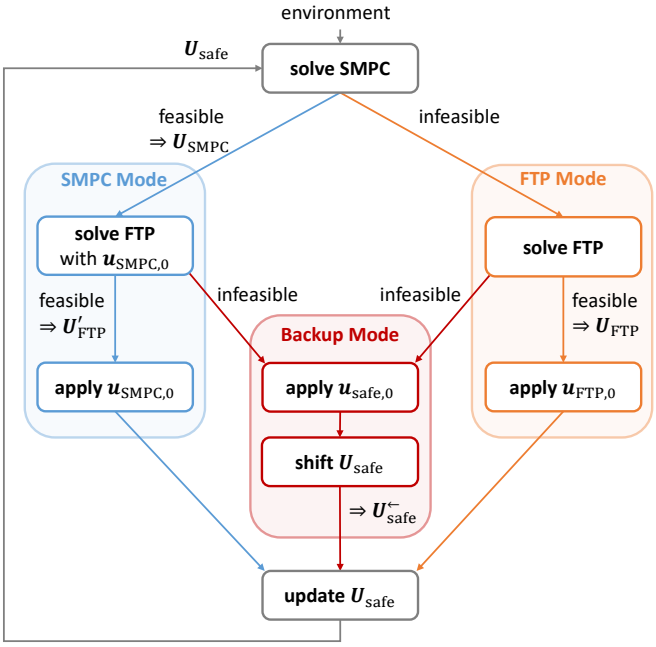

Fig. 1. SMPC+FTP procedure for each time step. Blue shows the ideal mode with an applied SMPC input, orange represents the safe alternative mode with an applied FTP input, and red indicates an infeasible FTP problem, which requires applying a safe backup input.

A. SMPC+FTP Method

Before presenting the SMPC+FTP method, we need to define requirements for a safe ego vehicle state as well as a safe input sequence $\boldsymbol{U}_{\mathrm{safe}} = [\boldsymbol{u}_{\mathrm{safe},0}, \boldsymbol{u}_{\mathrm{safe},1}, ..., \boldsymbol{u}_{\mathrm{safe},m}]^\top$ with $m+1$ individual inputs. Note that m is not directly related to the MPC prediction horizon.

Definition 1 (Safe State)**.** *The state of an ego vehicle, fully located in one lane, is considered to be safe if there is no lateral vehicle motion, i.e., $\phi = 0$, and if the ego vehicle velocity is lower than the velocity of the target vehicle in front on the same lane (or if the ego vehicle velocity is zero). The set of safe states is indicated by Ξ_{safe}.*

Definition 2 (Safe Input Sequence)**.** *An input sequence $\boldsymbol{U}_{\mathrm{safe}}$ is considered safe if consecutively applying all elements of $\boldsymbol{U}_{\mathrm{safe}}$ results in a state trajectory that avoids collisions and eventually leads to zero velocity.*

The definition of safe states and safe input sequences results in assumptions for TVs.

Assumption 1 (Traffic Rules)**.** *Target vehicles adhere to the traffic rules.*

Assumption 2 (Vehicle Deceleration)**.** *The maximum absolute value of the ego vehicle deceleration is at least as large as the maximum absolute value of the target vehicle deceleration.*

Given a safe EV state, there exists a safe input sequence $\boldsymbol{U}_{\mathrm{safe}}$, consisting of deceleration and zero steering, which results in an EV zero velocity state in the current EV lane, i.e., zero velocity in x-direction and y-direction. This is based on Assumptions 1 and 2. TVs behaving against traffic rules cannot be reliably accounted for by any prediction and the deceleration assumption is necessary to avoid colliding with a braking TV in front.

In the following, the current EV state at time step h and prediction step $k = 0$ is denoted by $\boldsymbol{\xi}_0$. We omit explicitly denoting the current time step h within an OCP.

At the initialization of each OCP, the current EV state $\boldsymbol{\xi}_0$ and the current TV state $\boldsymbol{\xi}_0^{\mathrm{TV}}$ are known to the EV. Additionally, a safe input sequence $\boldsymbol{U}_{\mathrm{safe}}$ is available from the SMPC+FTP problem solved at the previous time step. Later, we will focus on obtaining a safe input sequence for the SMPC+FTP iteration at the next time step, given the safe input sequence of the current time step.

The SMPC+FTP method consists of two parts, SMPC and FTP, i.e., at every time step an SMPC OCP and an FTP OCP are solved. The general idea is that the first input $\boldsymbol{u}_{\mathrm{SMPC},0}$ of the SMPC input sequence $\boldsymbol{U}_{\mathrm{SMPC}} = [\boldsymbol{u}_{\mathrm{SMPC},0}, ..., \boldsymbol{u}_{\mathrm{SMPC},N-1}]^\top$ must only be applied if, based on the first SMPC input $\boldsymbol{u}_{\mathrm{SMPC},0}$, a fail-safe trajectory can be found. Compared to regular SMPC methods, this approach guarantees that applying the optimistic SMPC input $\boldsymbol{u}_{\mathrm{SMPC},0}$ does not lead to unsafe behavior. The algorithm outline is shown in Figure 1.

1) SMPC: In the first part of SMPC+FTP, an SMPC problem is solved on a finite horizon N_{SMPC}, yielding the input sequence $\boldsymbol{U}_{\mathrm{SMPC}} = [\boldsymbol{u}_{\mathrm{SMPC},0}, ..., \boldsymbol{u}_{\mathrm{SMPC},N_{\mathrm{SMPC}}-1}]^\top$. This SMPC optimization takes into account the uncertain environment and constraints due to other traffic participants, i.e., target vehicles. Collision constraints are formulated as chance-constraints, based on a probabilistic TV prediction. Therefore, the planned SMPC trajectory provides an efficient and optimistic future

trajectory for the EV, as it is not required to avoid collision with TVs for worst-case scenarios.

2) FTP: The second part of SMPC+FTP is based on FTP to ensure that the planned EV trajectory remains safe. First, a worst-case TV prediction is performed. Then, a fail-safe MPC problem on a finite horizon N_{FTP} is solved, resulting in an input sequence $\boldsymbol{U}_{\text{FTP}} = [\boldsymbol{u}_{\text{FTP},0}, ..., \boldsymbol{u}_{\text{FTP},N_{\text{FTP}}-1}]^\top$. The fail-safe trajectory is required to avoid collision with the worst-case TV prediction and after applying the full fail-safe input sequence $\boldsymbol{U}_{\text{FTP}}$, the terminal state $\boldsymbol{\xi}_{N_{\text{FTP}}}$ must be a safe state according to Definition 1. The exact FTP formulation depends on the feasibility of the SMPC OCP.

a) Feasible SMPC (SMPC Mode): If the SMPC OCP yields a solution, FTP is used to decide whether applying the first SMPC input $\boldsymbol{u}_{\text{SMPC},0}$ is safe. Therefore, an FTP OCP is formulated starting with the EV state obtained by applying the first SMPC input $\boldsymbol{u}_{\text{SMPC},0}$, i.e., the initial FTP OCP state is

$$\boldsymbol{\xi}_0' = \boldsymbol{f}\left(\boldsymbol{\xi}_0, \boldsymbol{u}_{\text{SMPC},0}\right) \tag{13}$$

with $\boldsymbol{f}\left(\boldsymbol{\xi}_0, \boldsymbol{u}_{\text{SMPC},0}\right)$ according to (6).

If feasible, the FTP OCP yields a fail-safe input sequence $\boldsymbol{U}_{\text{FTP}}'$, based on $\boldsymbol{\xi}_0'$. Therefore, the first element $\boldsymbol{u}_{\text{SMPC},0}$ of the SMPC input sequence is applied safely, as shown by the blue path in Figure 1. The resulting new safe input sequence is given by

$$\boldsymbol{U}_{\text{safe}} = [\boldsymbol{U}_{\text{FTP}}', \boldsymbol{U}_{\text{brake}}] \tag{14a}$$

$$\boldsymbol{U}_{\text{brake}} = \left[\begin{bmatrix} a_{\min} \\ 0 \end{bmatrix}, \begin{bmatrix} a_{\min} \\ 0 \end{bmatrix}, \dots\right] \tag{14b}$$

where $a_{\min}$ is the maximal deceleration and $\boldsymbol{U}_{\text{brake}}$ is a braking sequence to bring the EV to a standstill. The safe input sequence $\boldsymbol{U}_{\text{safe}}$ ensures a safe state after the full fail-safe input sequence $\boldsymbol{U}_{\text{FTP}}'$ was applied and then initiates braking to reach zero velocity. Note that $a_{\min}$ is only applied in $\boldsymbol{U}_{\text{brake}}$ until a standstill is reached, subsequently no deceleration is applied.

b) Infeasible SMPC (FTP Mode): If the SMPC OCP is infeasible, the FTP OCP is solved with initial state $\boldsymbol{\xi}_0$ for the FTP OCP. If an FTP solution $\boldsymbol{U}_{\text{FTP}}$ is found, the first element of $\boldsymbol{U}_{\text{FTP}}$, i.e., $\boldsymbol{u}_{\text{FTP},0}$, is applied, as indicated by the orange path in Figure 1. The updated safe input sequence follows from

$$\boldsymbol{U}_{\text{safe}} = [\boldsymbol{U}_{\text{FTP},1:N_{\text{FTP}}}, \boldsymbol{U}_{\text{brake}}] \tag{15}$$

with $\boldsymbol{U}_{\text{brake}}$ according to (14b) where

$$\boldsymbol{U}_{\text{FTP},1:N_{\text{FTP}}} = [\boldsymbol{u}_{\text{FTP},1}, ..., \boldsymbol{u}_{\text{FTP},N_{\text{FTP}}-1}] \tag{16}$$

consists of all input elements of $\boldsymbol{U}_{\text{FTP}}$ except the first input $\boldsymbol{u}_{\text{FTP},0}$.

3) Infeasible FTP (Backup Mode): In case of an infeasible FTP OCP, no new input is generated at the current time step h. However, by definition the safe input sequence obtained at the previous time step $h-1$ remains safe for the current time step h. Therefore, in case that no solution exists to the FTP OCP, the first element of the still valid, safe input sequence $\boldsymbol{U}_{\text{safe}}$ is applied, which is denoted by $\boldsymbol{u}_{\text{safe},0}$. This procedure is highlighted in red in Figure 1.

Continuously applying the elements of $\boldsymbol{U}_{\text{safe}}$ results in a safe trajectory according to Definition 2. If the FTP OCP remains infeasible for consecutive time steps, multiple subsequent input elements of a single safe input sequence are potentially applied until the FTP OCP becomes feasible again.

This procedure requires shifting $\boldsymbol{U}_{\text{safe}}$ after each SMPC+FTP iteration where the FTP OCP was infeasible, i.e., if the first input element $\boldsymbol{u}_{\text{safe},0}$ of $\boldsymbol{U}_{\text{safe}}$ was applied. The shifted updated input sequence is obtained by

$$\boldsymbol{U}_{\text{safe}}^{\leftarrow} = \boldsymbol{U}_{\text{safe}} \begin{bmatrix} \boldsymbol{0}_m \\ \boldsymbol{I}_m \end{bmatrix} = [\boldsymbol{u}_{\text{safe},1}, \boldsymbol{u}_{\text{safe},2}, ..., \boldsymbol{u}_{\text{safe},m}] \tag{17}$$

with $\boldsymbol{U}_{\text{safe}} \in \mathbb{R}^{2\times(m+1)}$, identity matrix $\boldsymbol{I}_m \in \mathbb{R}^{m\times m}$, and $\boldsymbol{0}_m \in \mathbb{R}^{1\times m}$. The shifted safe input sequence $\boldsymbol{U}_{\text{safe}}^{\leftarrow}$ consists of all elements of $\boldsymbol{U}_{\text{safe}}$ except the already applied input $\boldsymbol{u}_{\text{safe},0}$.

Then, the safe input sequence is updated at the end of the SMPC+FTP iteration by selecting

$$\boldsymbol{U}_{\text{safe}} = \boldsymbol{U}_{\text{safe}}^{\leftarrow}, \tag{18}$$

which initializes the safe input sequence for the next SMPC+FTP iteration.

4) Summary of SMPC+FTP: Within the SMPC+FTP method, four cases are considered. These cases are summarized in the following.

a) SMPC and FTP feasible (SMPC Mode): The first SMPC input $\boldsymbol{u}_{\text{SMPC},0}$ is applied and a new safe input sequence $\boldsymbol{U}_{\text{safe}}$ is obtained according to (14).

b) SMPC infeasible and FTP feasible (FTP Mode): The first FTP input $\boldsymbol{u}_{\text{FTP},0}$ is applied and a new safe input sequence $\boldsymbol{U}_{\text{safe}}$ is obtained according to (15).

c) SMPC feasible and FTP infeasible (Backup Mode): No new input sequence is obtained. The first input element of the safe input sequence $\boldsymbol{u}_{\text{safe},0}$ is applied. The safe input sequence $\boldsymbol{U}_{\text{safe}}$ remains valid for the next time step and is updated according to (18).

d) SMPC infeasible and FTP infeasible (Backup Mode): As in the previous case, no new input sequence is obtained. The input $\boldsymbol{u}_{\text{safe},0}$ is applied and $\boldsymbol{U}_{\text{safe}}$ is generated based on (18) for the next time step.

Following this procedure, in regular cases the SMPC inputs are applied, resulting in efficient performance, while FTP guarantees safety for all possible cases, including rare events.

B. Recursive Feasibility

A disadvantage of various SMPC algorithms is that recursive feasibility of the OCP cannot be guaranteed. In this section, recursive feasibility of the SMPC+FTP method is proved, i.e., if the optimization problem can be solved at step h, it can also be solved at step $h+1$ for all $h \in \mathbb{N}$. In this section, it is necessary to denote the time step h. The safe input sequence updated at time step h is denoted by $\boldsymbol{U}_{\text{safe},h}$.

Definition 3 (Safe Feasible Trajectory)**.** *Let there exist a safe set $\boldsymbol{\Xi}_{\text{safe}}$ and let $\boldsymbol{\Xi}_{\text{f}}$ be a control invariant set. Let $\boldsymbol{\chi}_h^{U_h} = [\boldsymbol{\xi}_h, ..., \boldsymbol{\xi}_{h+N}]$ denote a trajectory starting at initial state $\boldsymbol{\xi}_h$ at time step h with N trajectory steps obtained by applying the input sequence $\boldsymbol{U}_h = [\boldsymbol{u}_h, ..., \boldsymbol{u}_{h+N-1}]$*

with $\boldsymbol{\xi}_{h+1} = \boldsymbol{f}(\boldsymbol{\xi}_h, \boldsymbol{u}_h)$*. Then, the set* Γ_h *of safe feasible trajectories, leading into the set* $\boldsymbol{\Xi}_\mathrm{f}$*, is defined as*

$$\Gamma_h = \left\{ \boldsymbol{\chi}_h^{\boldsymbol{U}_h} \;\middle|\; \boldsymbol{\xi}_{h+i} \in \boldsymbol{\Xi}_\mathrm{safe},\ i \in \{0,\dots,N\},\ \boldsymbol{\xi}_{h+N} \in \boldsymbol{\Xi}_\mathrm{f} \right\}. \quad (19)$$

A safe feasible trajectory satisfies all constraints given by $\boldsymbol{\Xi}_\mathrm{safe}$ and ends in the control invariant set $\boldsymbol{\Xi}_\mathrm{f}$.

Assumption 3 (System Models)**.** *The ego vehicle system models* (3) *and* (6) *correspond to the dynamics of the real system. The target vehicle model* (9) *represents an over-approximation of the real target vehicle dynamics.*

Here, over-approximation means that the possible states reachable with the TV model include all possible states obtained with the real TV dynamics.

Assumption 4 (Initial Safe Input Sequence)**.** *At the initial time step* $h = 0$ *the initial ego vehicle state is safe and there exists a known initial safe input sequence* $\boldsymbol{U}_\mathrm{safe,init}$*, such that* $\boldsymbol{\chi}_0^{\boldsymbol{U}_\mathrm{safe,init}}$ *is a safe feasible trajectory, i.e.,* $\boldsymbol{\chi}_0^{\boldsymbol{U}_\mathrm{safe,init}} \in \Gamma_0$.

We now show recursive feasibility of the proposed method.

Theorem 1. *Let Assumptions 3 and 4 hold. Then, for the SMPC+FTP approach there exists a feasible trajectory* $\boldsymbol{\chi}_h^{\boldsymbol{U}_h} \in \Gamma_h$ *that is guaranteed to be safe at all time steps* $h \in \mathbb{N}$.

Proof. The derivation of the proof is given in Appendix A. □

Note that the worst-case behavior of the TVs depends on the traffic rules. Therefore, safety and recursive feasibility of the SMPC+FTP method can only be guaranteed if surrounding TVs adhere to the underlying traffic rules, as stated in Assumption 1. However, no specific traffic rules are required to prove Theorem 1.

V. Trajectory Planning Algorithms

The two MPC OCPs, SMPC and FTP, are solved independently. In the following, the respective OCPs are derived.

A. Stochastic Model Predictive Control

SMPC solves an OCP with chance constraints, accounting for TV uncertainty, depending on a risk factor β. First, a safety area is defined around each predicted TV state, which accounts for the EV and TV shape. Then, this safety area is increased to account for TV uncertainty, given a predefined risk parameter. Eventually, a linear constraint is generated for each TV, depending on the positioning of the EV and the TV.

1) Deterministic Target Vehicle Prediction: For SMPC a simple TV prediction is applied, representing the most likely TV behavior with $\boldsymbol{w}_k^\mathrm{TV} = \boldsymbol{0}$, i.e., $\boldsymbol{u}_k^\mathrm{TV} = \tilde{\boldsymbol{u}}_k^\mathrm{TV}$. It is assumed that the current TV maneuver continues for the prediction horizon N_SMPC. Therefore, TV model (9) is applied where the TV reference $\boldsymbol{\xi}_{\mathrm{ref},k}^\mathrm{TV}$ depends on the current TV maneuver. The reference velocity $v_{x,\mathrm{ref},k}^\mathrm{TV}$ is set to the current TV velocity $v_{x,0}^\mathrm{TV}$. The TV reference lateral velocity is chosen to be $v_{y,\mathrm{ref},k}^\mathrm{TV} = 0$. The reference lateral position $y_{\mathrm{ref},k}^\mathrm{TV}$ is the current TV lane center. A new reference lane is selected if part of the TV shape lies in this adjacent lane and the lateral velocity moves the TV towards this adjacent lane.

2) Target Vehicle Safety Area: Collisions with TVs are avoided by ensuring the necessary distance between the EV and TV. Here, a safety rectangle around the TV is defined with length a_r and width b_r, based on a straight highway road.

Vehicle shapes do not intersect if the vehicle centers are distanced at least by the vehicle length l_veh and width w_veh. For the safety rectangle width this yields

$$b_\mathrm{r} = w_\mathrm{veh} + \varepsilon_\mathrm{safe} \quad (20)$$

where $\varepsilon_\mathrm{safe}$ is a possible additional safety margin.

Calculating the safety rectangle length a_r requires a velocity dependent part $\tilde{a}_\mathrm{r}\left(\boldsymbol{\xi}, \boldsymbol{\xi}^\mathrm{TV}\right)$, compensating for a potential velocity difference between the EV and the TV, resulting in

$$a_\mathrm{r} = l_\mathrm{veh} + \varepsilon_\mathrm{safe} + \tilde{a}_\mathrm{r}\left(\boldsymbol{\xi}, \boldsymbol{\xi}^\mathrm{TV}\right). \quad (21)$$

The velocity dependent part $\tilde{a}_\mathrm{r}$ needs to account for the difference in traveled distance between the EV and TV if both vehicles initiate maximal braking. It is obtained by

$$\tilde{a}_\mathrm{r}\left(\boldsymbol{\xi}, \boldsymbol{\xi}^\mathrm{TV}\right) = -\frac{1}{2a_\mathrm{min}} \max\left\{0, \left(v^2 - \left(v_x^\mathrm{TV}\right)^2\right)\right\} \quad (22)$$

where the max-operator ensures that the safety rectangle length does not decrease for $v_x^\mathrm{TV} > v^\mathrm{EV}$.

For the SMPC OCP, the safety rectangle is calculated for prediction time step k, based on the TV prediction $\boldsymbol{\xi}_k^\mathrm{TV}$ described in Section V-A1. However, only the initial EV state $\boldsymbol{\xi}_0$ is considered in the velocity depended part $\tilde{a}_\mathrm{r}$. This is necessary in order to generate linear safety constraints. The resulting safety rectangle parameters are

$$b_{\mathrm{r},k} = w_\mathrm{veh} + \varepsilon_\mathrm{safe} \quad (23a)$$

$$a_{\mathrm{r},k} = l_\mathrm{veh} + \varepsilon_\mathrm{safe} + \tilde{a}_\mathrm{r}\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k^\mathrm{TV}\right). \quad (23b)$$

3) Chance Constraint Reformulation: The TV safety rectangle given by (23) does not account for TV uncertainty. In the following, the safety rectangle is enlarged, depending on the TV uncertainty and a risk parameter β. The chance constraint, similar to (1d), is given by

$$\Pr\left(\boldsymbol{\xi}_k \in \boldsymbol{\Xi}'_{k,\mathrm{safe}}\left(\boldsymbol{w}_k^\mathrm{TV}\right)\right) \geq \beta \quad (24)$$

where the safe set $\boldsymbol{\Xi}'_{k,\mathrm{safe}}\left(\boldsymbol{w}_k^\mathrm{TV}\right)$ for the EV state depends on the previously defined safety rectangle parameters of (23) and the TV uncertainty $\boldsymbol{w}_k^\mathrm{TV}$.

The chance constraint (24) cannot be solved directly. We derive a deterministic approximation for this probabilistic expression, inspired by other SMPC approaches [6], [30].

According to (9) the TV state follows

$$\boldsymbol{\xi}_{k+1}^\mathrm{TV} = \boldsymbol{A}\boldsymbol{\xi}_k^\mathrm{TV} + \boldsymbol{B}\boldsymbol{K}\left(\boldsymbol{\xi}_k^\mathrm{TV} - \boldsymbol{\xi}_{\mathrm{ref},k}^\mathrm{TV}\right) + \boldsymbol{B}\boldsymbol{w}_k^\mathrm{TV}, \quad (25)$$

while the predicted TV state is given by

$$\hat{\boldsymbol{\xi}}_{k+1}^\mathrm{TV} = \boldsymbol{A}\hat{\boldsymbol{\xi}}_k^\mathrm{TV} + \boldsymbol{B}\boldsymbol{K}\left(\hat{\boldsymbol{\xi}}_k^\mathrm{TV} - \boldsymbol{\xi}_{\mathrm{ref},k}^\mathrm{TV}\right), \quad (26)$$

yielding the prediction error

$$\boldsymbol{e}_k = \hat{\boldsymbol{\xi}}_k^\mathrm{TV} - \boldsymbol{\xi}_k^\mathrm{TV}. \quad (27)$$

The TV prediction (26) is now split into a deterministic and a stochastic part

$$\hat{\boldsymbol{\xi}}_{k+1}^\mathrm{TV} = \boldsymbol{\xi}_{k+1}^\mathrm{TV} + (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\,\boldsymbol{e}_k - \boldsymbol{B}\boldsymbol{w}_k^\mathrm{TV} = \boldsymbol{\xi}_{k+1}^\mathrm{TV} + \boldsymbol{e}_{k+1} \quad (28)$$

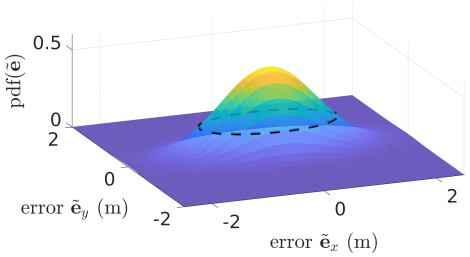

Fig. 2. Exemplary bivariate Gaussian probability distribution function of the prediction error $\tilde{\boldsymbol{e}}$, including an isoline (dotted black line).

which results in the prediction error update

$$\boldsymbol{e}_{k+1} = (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\,\boldsymbol{e}_k - \boldsymbol{B}\boldsymbol{w}_k^{\mathrm{TV}}. \tag{29}$$

Given the sensor noise $\boldsymbol{w}_0^{\mathrm{sens}}$ according to (12), the initial error follows $\boldsymbol{e}_0 \sim \mathcal{N}(0, \boldsymbol{\Sigma}_0^{\boldsymbol{e}})$ with $\boldsymbol{\Sigma}_0^{\boldsymbol{e}} = \boldsymbol{\Sigma}^{\mathrm{sens}}$. As we consider Gaussian distributions, a recursive computation of the prediction error covariance matrix $\boldsymbol{\Sigma}_k^{\boldsymbol{e}}$ is possible, yielding

$$\boldsymbol{\Sigma}_{k+1}^{\boldsymbol{e}} = \boldsymbol{B}\Sigma_{\boldsymbol{w}}^{\mathrm{TV}}\boldsymbol{B}^\top + (\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})\,\boldsymbol{\Sigma}_k^{\boldsymbol{e}}\,(\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K})^\top. \tag{30}$$

Based on the prediction error covariance matrix $\boldsymbol{\Sigma}_k^{\boldsymbol{e}}$, the TV safety rectangle is increased. Given a predefined SMPC risk parameter β, the aim is to find a region around the predicted TV state that contains the true TV state with probability β. As the TV safety rectangle only considers positions, we define the reduced error $\tilde{\boldsymbol{e}}_k = [e_{x,k}, e_{y,k}]^\top$ with the reduced covariance matrix

$$\tilde{\boldsymbol{\Sigma}}_k^{\boldsymbol{e}} = \mathrm{diag}(\sigma_{x,k}^2, \sigma_{y,k}^2) \tag{31}$$

with variances $\sigma_{x,k}^2$ and $\sigma_{y,k}^2$ for the longitudinal and lateral TV position, corresponding to the first and third diagonal element of $\boldsymbol{\Sigma}_k^{\boldsymbol{e}}$. The reduced error covariance matrix $\tilde{\boldsymbol{\Sigma}}_k^{\boldsymbol{e}}$ is now used to enlarge the safety rectangle to account for uncertainty.

The bivariate Gaussian distribution described by $\tilde{\boldsymbol{\Sigma}}_k^{\boldsymbol{e}}$ with mean $\boldsymbol{\mu} = [\mu_x, \mu_y]^\top = \boldsymbol{0}$ consists of independent random variables for longitudinal and lateral position. This allows to find a confidence region around the predicted TV state mean, bounded by an ellipsoidal isoline enclosing the highest density region, as illustrated in Figure 2. The aim is to find an isoline that contains the prediction error with a probability according to risk parameter β. The isoline ellipse equation is denoted by

$$(\tilde{\boldsymbol{e}}_k - \boldsymbol{\mu})^\top \left(\tilde{\boldsymbol{\Sigma}}_k^{\boldsymbol{e}}\right)^{-1} (\tilde{\boldsymbol{e}}_k - \boldsymbol{\mu}) = \kappa \tag{32a}$$

$$\frac{(e_{x,k} - \mu_x)^2}{\sigma_{x,k}^2} + \frac{(e_{y,k} - \mu_y)^2}{\sigma_{y,k}^2} = \kappa \tag{32b}$$

with tolerance level κ. The tolerance level κ depends on the risk parameter β and indicates the necessary constraint tightening in order to ensure that the prediction error remains below a probability β. The tolerance level κ is determined based on the cumulative distribution function $F(\kappa, n)$ of the *chi-square distribution* χ_n^2 with n degrees of freedom. In this case, $n = 2$ as the reduced error $\tilde{\boldsymbol{e}}_k$ consists of two elements. Given the risk parameter β and the quantile function F^{-1} of the *chi-square distribution* χ_2^2, it follows that

$$\kappa = F^{-1}(\beta, 2), \tag{33}$$

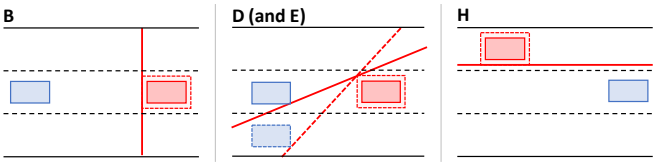

Fig. 3. Selected constraint generation cases for SMPC. Driving direction is from left to right. The EV and TV are shown in blue and red, respectively. The dashed red line represents the safety area around the TV.

which ensures that the probability of the true TV state lying within the isoline is $\beta \cdot 100\%$. The ellipse semi-major and semi-minor axes are then given by

$$e_{x,k,\kappa} = \sigma_{x,k}\sqrt{\kappa} \tag{34a}$$

$$e_{y,k,\kappa} = \sigma_{y,k}\sqrt{\kappa}. \tag{34b}$$

While an ellipse, according to (32), describes the desired confidence region, the constraint generation method used in this work requires a rectangular TV safety area. We therefore over-approximate the ellipse by a rectangle. In order to include this uncertainty consideration in the rectangle parameters $a_{\mathrm{r},k}$ and $b_{\mathrm{r},k}$ of (23), the rectangle parameters are increased based on the ellipse semi-major axis $e_{x,k,\kappa}$ and semi-minor axis $e_{x,k,\kappa}$, resulting in

$$b_{\mathrm{r},k} = w_{\mathrm{veh}} + \varepsilon_{\mathrm{safe}} + e_{x,k,\kappa} \tag{35a}$$

$$a_{\mathrm{r},k} = l_{\mathrm{veh}} + \varepsilon_{\mathrm{safe}} + \tilde{a}_{\mathrm{r}}\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k^{\mathrm{TV}}\right) + e_{y,k,\kappa}. \tag{35b}$$

The updated safety rectangle parameters are now used to generate the safety constraints for the SMPC OCP.

4) SMPC Constraint Generation: Given the safety rectangles for each TV, linear constraints to avoid collisions can be defined for each prediction step and for each TV. Each linear constraint is of the form

$$0 \geq q_y\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k^{\mathrm{TV}}\right) y_k + q_x\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k^{\mathrm{TV}}\right) x_k + q_t\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k^{\mathrm{TV}}\right) \tag{36}$$

where q_y and q_x are the coefficients for the EV states y_k and x_k, and q_t is the intercept. The coefficients q_y, q_x, and q_t of the linear constraint depend on the current EV state $\boldsymbol{\xi}_0$ and the predicted mean TV states $\boldsymbol{\xi}_k^{\mathrm{TV}}$. This results in multiple constraint generation cases, extending the cases in [7].

The cases are distinguished based on the initial vehicle configuration at the beginning of the OCP, i.e., $k = 0$. While the predicted TV state $\boldsymbol{\xi}_k^{\mathrm{TV}}$ is considered to build the constraint (36) at prediction step k for a specific case, only the initial EV state $\boldsymbol{\xi}_0$ is considered in order to allow generating linear constraints, as mentioned in Section V-A2.

We briefly discuss a shortened overview of constraint cases that are considered, summarized in Table I. Example cases are illustrated in Figure 3. A complete overview of cases, requirements, and constraint parameters q_x, q_y, q_t from (36) is found in the extended version [10].

In summary, no constraints are generated if the longitudinal distance between the EV and TV is larger than r_{lar} (case A). If the EV is close enough to the TV (longitudinal distance smaller than r_{close}), overtaking is possible by employing an inclined constraint (cases D and E). If the TV is located behind

TABLE I
CONSTRAINT GENERATION CASES

case	EV setting (w.r.t. TV)	SMPC	FTP
A, A*	large dist. ($> r_{\text{lar}}$)	no constraint	no constraint
B, B*	behind TV ($> r_{\text{close}}$)	vert. constraint	vert. constraint
C, C*	ahead of TV ($> r_{\text{close}}$)	vert. constraint	virtual TVs mixed constraints
D, D*	same lane as TV behind TV ($\leq r_{\text{close}}$)	incl. constraint	vert. constraint
E, E*	right lane next to TV behind TV ($\leq r_{\text{close}}$)	incl. constraint	vert. constraint
F, F*	left of TV close to TV ($\leq r_{\text{close}}$)	hor. constraint	hor. constraint
G, G*	2 lanes right of TV behind TV ($\leq r_{\text{close}}$)	hor. constraint	hor. constraint
H, H*	right of TV ahead of TV ($\leq r_{\text{close}}$)	hor. constraint	hor. constraint
J, J*	same lane as TV ahead of TV ($\leq r_{\text{close}}$)	no constraint	virtual TVs mixed constraints

the EV, no constraints are necessary as it is the responsibility of the TV to avoid a collision (case J). For all other cases, horizontal and vertical constraints are employed.

These constraint generation cases are now used to formulate safety constraints in the SMPC OCP.

5) SMPC Optimal Control Problem: With the definition of the safety constraints, the deterministic OCP replacing the SMPC problem is given by

$$V^* = \min_{\boldsymbol{U}} \sum_{k=1}^{N_{\text{SMPC}}} \|\Delta\boldsymbol{\xi}_k\|_{\boldsymbol{Q}} + \|\boldsymbol{u}_{k-1}\|_{\boldsymbol{R}} + \|\Delta\boldsymbol{u}_{k-1}\|_{\boldsymbol{S}} \tag{37a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{f}^{\text{d}}\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k, \boldsymbol{u}_k\right) \tag{37b}$$

$$\boldsymbol{\xi}^{\text{TV}}_{k+1} = \boldsymbol{A}\boldsymbol{\xi}^{\text{TV}}_k + \boldsymbol{B}\tilde{\boldsymbol{u}}^{\text{TV}}_k \tag{37c}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \quad \boldsymbol{\xi}_{k+1} \in \boldsymbol{\Xi} \qquad \forall k \in \{0, \dots, N_{\text{SMPC}} - 1\} \tag{37d}$$

$$0 \geq q_y\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}^{\text{TV}}_k\right) y_k + q_x\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}^{\text{TV}}_k\right) x_k + q_t\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}^{\text{TV}}_k\right) \quad \forall k \in \{0, \dots, N_{\text{SMPC}}\} \tag{37e}$$

with $\Delta\boldsymbol{\xi}_k = \boldsymbol{\xi}_k - \boldsymbol{\xi}_{k,\text{ref}}$, EV reference state $\boldsymbol{\xi}_{k,\text{ref}}$, and the linear function $\boldsymbol{f}^{\text{d}}$ according to (6). For the input difference $\Delta\boldsymbol{u}$, we set $\boldsymbol{u}_{\text{-1}}$ to the applied input of the previous time step. The cost function sum limits are shifted to include a terminal cost for $\boldsymbol{\xi}_N$. The weighting matrices are given by $\boldsymbol{Q}$, $\boldsymbol{S}$, and $\boldsymbol{R}$. We consider constant input constraints $\mathcal{U}$ according to (7) and state constraints $\boldsymbol{\Xi}$ according to (8).

The resulting SMPC OCP (37) is a quadratic program with linear constraints, accounting for uncertainty with the chance constraint reformulation described in Section V-A3. This OCP can be solved efficiently, where the major calculation steps to obtain the linear constraints (37e) are performed before the optimization starts.

B. Failsafe Trajectory Planning

While the SMPC algorithm only accounts for part of the TV uncertainty in order to plan an optimistic trajectory, the backup FTP algorithm needs to consider worst-case uncertainty realizations. This is achieved based on reachability analysis. First, the worst-case TV occupancy prediction is determined. Then, linear safety constraints are generated. Eventually, given a safe invariant terminal set, the FTP OCP is solved.

1) Target Vehicle Occupancy Prediction: Similar to the SMPC algorithm, a rectangular safety area surrounding each TV is defined. However, for the FTP the maximal reachable area needs to be determined. First, it is necessary to define certain traffic rules to which the TV adheres, according to Assumption 1:

- Road boundaries apply.
- Negative velocities are forbidden.
- Collisions with vehicles directly in front of the TV (in the same lane) must be avoided.
- Only a single lane change is allowed (within the prediction horizon).
- No lane change is allowed if the TV velocity is below a predefined minimal lane change velocity $v_{\text{LC,min}}$.
- No lane change is allowed if the distance to a vehicle on the new lane becomes too small.

As linear dynamics are assumed for the TV motion, the minimal and maximal possible TV inputs are used to determine the maximal reachable set, inspired by [8], [19], [22].

The set of all possible locations reachable for a TV at prediction step k is denoted by the reachable set $\mathcal{R}^{\text{TV}}_k$, including the TV and shape. While referring to $\mathcal{R}^{\text{TV}}_k$ as the reachable set of the TV, we additionally enlarge this set accounting for the EV shape. This is necessary as the set $\mathcal{R}^{\text{TV}}_k$ is later used to avoid collisions by keeping the EV center outside of $\mathcal{R}^{\text{TV}}_k$. Given the solution $\boldsymbol{\zeta}\left(\hat{\boldsymbol{\xi}}^{\text{TV}}_0, \boldsymbol{U}\right)$ to the TV dynamics (9) starting at the initial state $\hat{\boldsymbol{\xi}}^{\text{TV}}_0$ applying an input sequence $\boldsymbol{U}$, we define the reachable set

$$\mathcal{R}^{\text{TV}}_k = \left\{ \boldsymbol{\zeta}\left(\hat{\boldsymbol{\xi}}^{\text{TV}}_0, \boldsymbol{U}\right) \;\middle|\; \boldsymbol{U}(i) \in \mathcal{U}^{\text{TV}}\ \forall i \in \{0, ..., k-1\}, \quad \hat{\boldsymbol{\xi}}^{\text{TV}}_0 \in \boldsymbol{\Xi}^{\text{TV}}_0 \right\}. \tag{38}$$

The initial state for the reachable set $\mathcal{R}^{\text{TV}}_k$ is not the TV state $\boldsymbol{\xi}^{\text{TV}}_0$, but depends on the sensor uncertainty as well as the TV and EV shape. This initial set is given by

$$\boldsymbol{\Xi}^{\text{TV}}_0 = \left\{ \hat{\boldsymbol{\xi}}^{\text{TV}}_0 \;\middle|\; \boldsymbol{\xi}^{\text{TV}}_0 + \min\{\boldsymbol{w}^{\text{sens}}_0\} - [l_{\text{veh}}, 0, w_{\text{veh}}, 0]^\top \leq \hat{\boldsymbol{\xi}}^{\text{TV}}_0, \; \hat{\boldsymbol{\xi}}^{\text{TV}}_0 \leq \boldsymbol{\xi}^{\text{TV}}_0 + \max\{\boldsymbol{w}^{\text{sens}}_0\} + [l_{\text{veh}}, 0, w_{\text{veh}}, 0]^\top \right\}. \tag{39}$$

As we assume a linear TV prediction model, the reachable set $\mathcal{R}^{\text{TV}}_k$ is calculated for prediction steps $k > 0$ by applying the maximal and minimal inputs $\boldsymbol{u}^{\text{TV}} \in \mathcal{U}^{\text{TV}}$, while adhering to traffic rules.

The reachable set is only calculated at discrete time steps. In order to account for a continuous system, the final reachable set $\overline{\mathcal{R}}^{\text{TV}}_k$ is obtained by building a rectangular convex hull, covering two consecutive prediction steps, i.e.,

$$\overline{\mathcal{R}}^{\text{TV}}_k = \text{conv}\left\{\mathcal{R}^{\text{TV}}_{k-1}, \mathcal{R}^{\text{TV}}_k\right\} \tag{40}$$

where conv denotes the convex hull operation.

A special case is considered if the TV is located behind the EV in the same lane. The TV must not collide with the EV in the same lane, however, the TV is allowed to switch lanes in

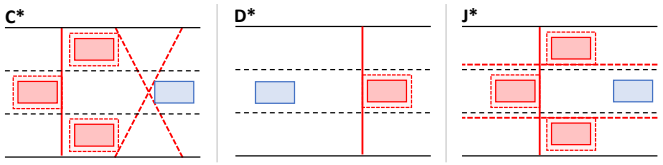

Fig. 4. Selected constraint generation cases for FTP. Driving direction is from left to right. The EV and TV are shown in blue and red, respectively. The dashed red line represents the safety area around the TV.

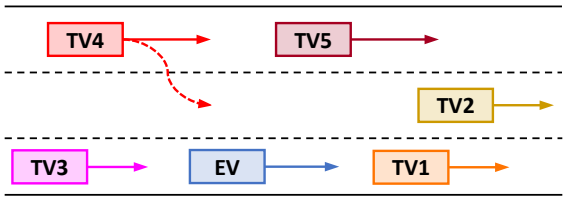

Fig. 5. Setup for both investigated scenarios (regular and emergency scenario).

order to pass the EV. Here, this is accounted for by treating this special case in the following way. Three placeholder TV reachable sets describe the possible TV behavior. The first placeholder TV reachable set is based in the EV lane such that collisions with the EV are avoided. The other two placeholder TV reachable sets cover the admissible adjacent lanes left and right of the EV, representing the reachable sets for a potential TV lane change.

2) FTP Constraint Generation: Once the reachable sets $\overline{\mathcal{R}}_k^{\mathrm{TV}}$ for each TV are determined, linear constraints are generated. We again consider different cases regarding varying EV and TV positions. The cases are similar to those of Section V-A4 with a few variations as stated in Table I. FTP cases are denoted with an asterisk. Exemplary FTP cases are illustrated in Figure 4. Again, a complete overview of the FTP cases is found in the extended version [10].

We briefly discuss the major differences to the SMPC constraint generation cases. Overtaking is not initiated given the vertical constraints in cases D* and E*. If the TV is located behind the EV, we consider possible TV lane changes by introducing placeholder TVs (cases C* and J*). Here, $r_{\mathrm{close}}^{\mathrm{FTP}}$ is used instead of r_{close}

Overall, the constraints generated for FTP are more conservative than for SMPC. This is due to the FTP aim of finding a trajectory that ends in a safe state. This would be complicated by incentivizing FTP to plan overtaking maneuvers. Details on finding a safe terminal state for the FTP OCP are given in the following.

3) Safe Invariant Terminal Set: In addition to the regular safety constraints, a safe invariant terminal set is required to ensure safe EV inputs after the finite MPC prediction horizon. The FTP inputs are designed in such a way that they remain safe over the prediction horizon. However, after N_{FTP} inputs are applied and no new FTP solution is obtained, an emergency strategy has to be applied to come to a standstill. This is achieved by braking, while maintaining a constant steering angle $\delta = 0$, according to (14) and (15). Therefore, the terminal state of the FTP OCP needs to fulfill certain requirements. First, the vehicle orientation must be aligned with the road, i.e., $\phi = 0$. This guarantees that braking and a constant steering angle $\delta = 0$ keep the EV within its current lane. Second, the distance to a TV in front of the EV must be large enough that no collision occurs if both vehicles initiate maximal deceleration. This is accounted for by

$$x_N \leq x_N^{\mathrm{TV}} - \Delta s_{N_{\mathrm{FTP}},\min} \tag{41a}$$

$$v_N \leq v_{N_{\mathrm{FTP}},\max} \tag{41b}$$

with the minimal terminal safety distance $\Delta s_{N_{\mathrm{FTP}},\min}$ and the maximal terminal safety velocity

$$v_{N_{\mathrm{FTP}},\max} = v_{N_{\mathrm{FTP}},\min}^{\mathrm{TV}} - \sqrt{2\Delta s_{N_{\mathrm{FTP}},\min} a_{x,\min}} \tag{42}$$

where $v_{N_{\mathrm{FTP}},\min}^{\mathrm{TV}}$ is the lowest predicted longitudinal TV velocity. Both (41) and (42) combined ensure that the minimal terminal safety distance $\Delta s_{N_{\mathrm{FTP}},\min}$ is large enough such that, given a maximal EV velocity $v_{N_{\mathrm{FTP}},\max}$, maximal deceleration of the EV guarantees collision avoidance for $k > N_{\mathrm{FTP}}$. This less intuitive terminal constraint again has the advantage of yielding linear constraints.

4) FTP Optimal Control Problem: An OCP with a similar structure compared to (37) is applied for the FTP, yielding

$$V^* = \min_{\boldsymbol{U}} \sum_{k=1}^{N_{\mathrm{FTP}}} \|\Delta\boldsymbol{\xi}_k\|_{\boldsymbol{Q}} + \|\boldsymbol{u}_{k-1}\|_{\boldsymbol{R}} + \|\Delta\boldsymbol{u}_{k-1}\|_{\boldsymbol{S}} \tag{43a}$$

$$\text{s.t. } \boldsymbol{\xi}_{k+1} = \boldsymbol{f}^{\mathrm{d}}\left(\boldsymbol{\xi}_0, \boldsymbol{\xi}_k, \boldsymbol{u}_k\right) \tag{43b}$$

$$\boldsymbol{u}_k \in \mathcal{U}, \quad \boldsymbol{\xi}_{k+1} \in \boldsymbol{\Xi} \qquad \forall k \in \{0, \ldots, N_{\mathrm{FTP}} - 1\} \tag{43c}$$

$$0 \geq q_y\left(\boldsymbol{\xi}_0, \overline{\mathcal{R}}_k^{\mathrm{TV}}\right) y_k + q_x\left(\boldsymbol{\xi}_0, \overline{\mathcal{R}}_k^{\mathrm{TV}}\right) x_k + q_t\left(\boldsymbol{\xi}_0, \overline{\mathcal{R}}_k^{\mathrm{TV}}\right) \qquad \forall k \in \{0, \ldots, N_{\mathrm{FTP}}\} \tag{43d}$$

$$x_N \leq x_N^{\mathrm{TV}} - \Delta s_{N_{\mathrm{FTP}},\min}, \quad v_N \leq v_{N_{\mathrm{FTP}},\max} \tag{43e}$$

with the linear function $\boldsymbol{f}^{\mathrm{d}}$ according to (6). The safety constraint (37e) is now changed to constraint (43d), accounting for the worst-case TV uncertainty realizations. Similar to the SMPC OCP, (43) is a quadratic program with linear constraints, which can be solved efficiently.

VI. Results

We evaluate the proposed SMPC+FTP algorithm in different settings. In the following, the simulation setup is introduced first. Then, SMPC+FTP is analyzed and compared to an SMPC approach and an FTP approach in two scenarios.

A. Simulation Setup

In this simulation section, we analyze the scenario illustrated in Figure 5. The EV is located on the right lane on a three-lane highway. We consider five TVs surrounding the EV on the highway. The goal for the EV is to maneuver safely and efficiently through traffic. The specific aims are to avoid collisions while maintaining a velocity close to a chosen reference velocity.

We consider two different scenarios:

1) Regular scenario: All TVs keep their initial velocities and lanes.

TABLE II
GENERAL SIMULATION PARAMETERS

scalars	vectors	matrices
$w_{\text{lane}} = 3.5$	$\boldsymbol{u}_{\max} = [5, 0.2]^\top$	$\boldsymbol{K} = \begin{bmatrix} 0 & -0.55 & 0 & 0 \\ 0 & 0 & -0.63 & -1.15 \end{bmatrix}$
$l_{\text{veh}} = 5$	$\boldsymbol{u}_{\min} = [\text{-}9, \text{-}0.2]^\top$	$\tilde{\boldsymbol{\Sigma}}_{\boldsymbol{w}}^{\text{TV}} = \text{diag}(0.44, 0.09)$
$w_{\text{veh}} = 2$	$\boldsymbol{u}_{\max}^{\text{TV}} = [5, 0.4]^\top$	$\boldsymbol{Q} = \text{diag}(0, 0.25, 0.2, 10)$
$l_{\text{f}} = l_{\text{r}} = 2$	$\boldsymbol{u}_{\min}^{\text{TV}} = [\text{-}9, \text{-}0.4]^\top$	$\boldsymbol{R} = \text{diag}(0.33, 5)$
$v_{\max} = 35$	$\boldsymbol{w}_0^{\text{sens}\top} =$	$\boldsymbol{S} = \text{diag}(0.33, 15)$
	$[0.25, 0.03, 0.25, 0.03]$	

2) Emergency scenario: One of the TVs (TV5) performs an emergency braking maneuver. This causes TV4 to avoid TV5 by moving to the center lane. This is followed by a soft braking maneuver of TV1 to account for possible hazards. Eventually, TV4 moves to the left lane again to pass TV2.

The first scenario represents a regular scenario with no unexpected TV behavior. The second scenario covers a rare case, where a series of unexpected TV actions results in a challenging situation for the autonomous EV.

The simulations are carried out in Matlab using the *fmincon* solver on a computer with an AMD Ryzen 7 1700X processor. The algorithms are based on the NMPC toolbox [35]. In the following, setup parameters are introduced that remain constant throughout the different simulations. All quantities are given in SI units. Units are omitted if clear by context.

All MPC algorithms use a sampling time $T = 0.2\,\text{s}$ with SMPC horizon $N_{\text{SMPC}} = 10$ and the FTP horizon $N_{\text{FTP}} = 10$. The linearized, discrete-time EV prediction model and constraints follow (6)-(8), whereas the TV prediction model is given by (9)-(11). Table II shows the other main simulation parameters. The lane boundaries follow from the lane and vehicle width. Additionally, the safety parameters are $\varepsilon_{\text{safe}} = 0.01$, $r_{\text{lar}} = 200$, $r_{\text{close}} = 90$, $r_{\text{close}}^{\text{FTP}} = \max\{10, |v_0 N_{\text{FTP}} T|\}$, $v_{\text{LC, min}} = 10$, and $\Delta s_{N,\min} = 22.5$.

In all scenarios, the initial EV reference is set to $[d_{\text{ref}}, \phi_{\text{ref}}, v_{\text{ref}}] = [0, 0, 27]$. While the reference orientation and velocity remain constant throughout the simulation, the EV reference for the lateral position is always set to the current EV lane center.

Whereas the MPC OCPs use the linearized, discrete-time prediction model (6), the inputs are applied to a simulation using the continuous-time system (3).

Given this simulation setup, we now investigate the individual scenarios and analyze the proposed SMPC+FTP method.

B. Regular Highway Scenario

We first analyze a regular highway scenario. The initial states of the vehicles are given in Table III. The five TVs shown in Figure 5 all maintain their initial velocities and lanes, therefore, $\boldsymbol{\xi}_{\text{ref},k}^{\text{TV}} = \boldsymbol{\xi}_0^{\text{TV}}$.

In the following, the SMPC+FTP solution is shown in detail and comparisons are made to an SMPC and an FTP method.

TABLE III
INITIAL VEHICLE STATES

vehicle	initial state	vehicle	initial state
EV	$[0, 0, 0, 27]^\top$	TV3	$[\text{-}245, 20, 0, 0]^\top$
TV1	$[70, 20, 0, 0]^\top$	TV4	$[\text{-}35, 32, 7, 0]^\top$
TV2	$[125, 20, 3.5, 0]^\top$	TV5	$[40, 32, 7, 0]^\top$

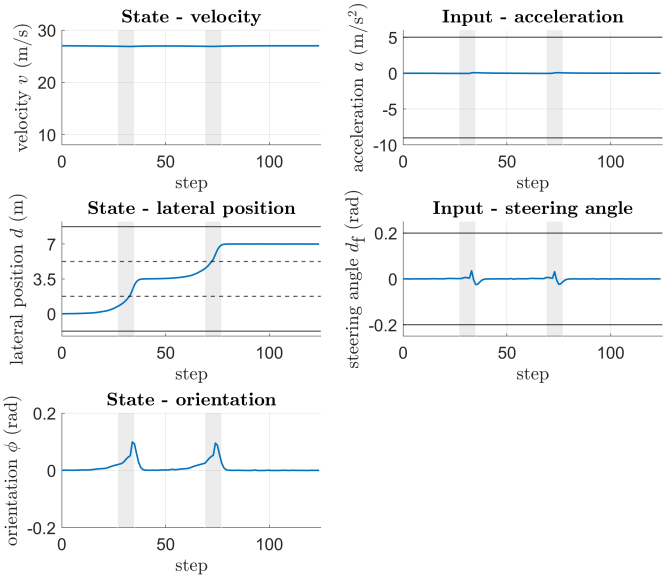

Fig. 6. SMPC+FTP states and inputs for the regular scenario. Vehicle motion in the gray areas is illustrated in Figure 7.

1) SMPC+FTP: Applying the proposed SMPC+FTP approach to the regular highway scenario yields efficient EV behavior in traffic. The SMPC risk parameter is chosen to be $\beta = 0.8$. The inputs and important states are shown in Figure 6, vehicle motion is displayed in Figure 7.

The EV approaches TV1 due to the velocity difference. The EV then changes lanes to the center lane with a moderate steering angle of $\delta < 0.04$. Once TV2 is reached, the EV again changes lanes and eventually passes TV2. The vehicle orientation remains at a limited level, i.e., $\phi < 0.11$. Through-

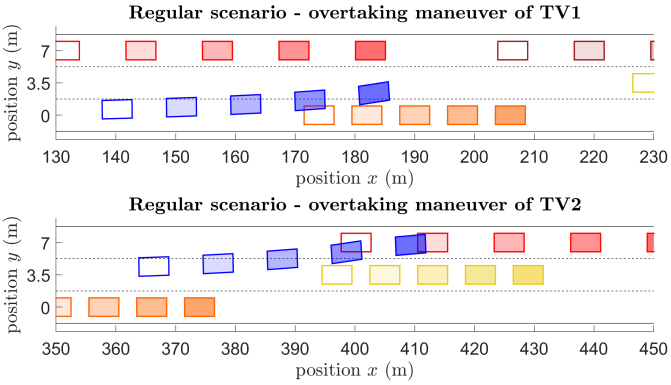

Fig. 7. Shots of the regular scenario with SMPC+FTP. Fading boxes show past states. The EV is shown in blue.

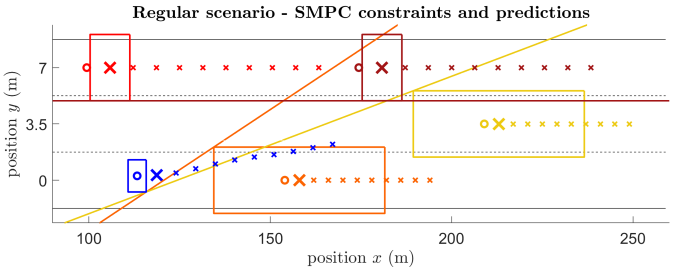

Fig. 8. SMPC constraints for the regular scenario at time step $h = 22$ and prediction step $k = 1$. The EV shape and planned trajectory are shown in blue. TVs as well as respective safety rectangles and constraints have the same color. Initial states are marked by a circle, prediction states are represented by crosses with a bold cross indicating the displayed prediction step.

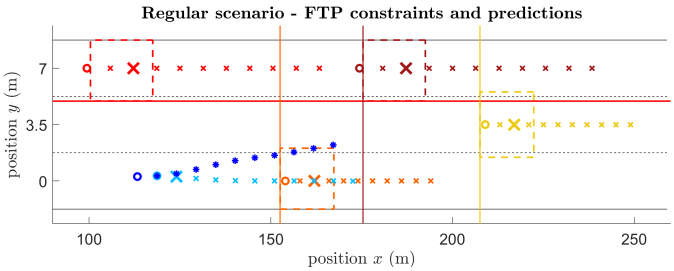

Fig. 9. FTP constraints for the regular scenario at time step $h = 22$ and prediction step $k = 1$. The EV is shown in blue. TVs as well as respective reachable sets and constraints have the same color. Initial states are marked by a circle. The initial FTP state starts after the first SMPC input is applied. Prediction states are represented by crosses with a bold cross indicating the displayed prediction step. For reference, the planned SMPC trajectory is given by dark blue asterisks with a dark blue circle indicating the initial EV state.

out the scenario, the EV maintains the reference velocity, and acceleration inputs are small. The average computation time to solve the SMPC and FTP OCPs are $0.11\,\mathrm{s}$ and $0.15\,\mathrm{s}$, respectively. Lower computational effort is possible with other solvers. If applied in a setting that requires online computation, OCPs with computation times exceeding the requirements are considered as infeasible. In this case, the previously calculated, still valid safe input sequence would be used.

We will now take a closer look at the constraints for SMPC and FTP. SMPC constraints for time step $h = 22$ are illustrated in Figure 8. For TV1 in the same lane as the EV, an inclined constraint is generated (case D). At each prediction step, the constraint connects the initial EV shape with the TV1 safety rectangle at the predicted position. The predicted SMPC trajectory for the EV stays above the constraint line. It is to note that only the respective predicted state must satisfy the illustrated constraint. Predicted states farther in the future satisfy respective constraints depending on a TV safety rectangle for a predicted TV position farther ahead. For TV2 case E is active, also resulting in an inclined constraint. Both TV4 and TV5 are two lanes left of the EV, yielding cases G and H, resulting in horizontal constraints to the right side of the TVs. TV3 is not shown in Figure 8 due to clarity.

The FTP constraints at step $h = 22$ are shown in Figure 9. The constraints are more conservative compared to the SMPC constraints. The reachable TV sets extend further to the back than the front, as maximal deceleration is larger than maximal acceleration. Additionally, the convex hull of reachable sets over two consecutive steps is considered. Constraints for TV1 and TV2 are built according to cases D^* and B^*, respectively. Both constraints for TV4 and TV5 are generated given cases H^* and G^*. While the SMPC trajectory moves towards the center lane to overtake TV1, the FTP trajectory finds a vehicle motion that, for the final prediction step, remains in the current lane with $\phi = 0$ and enough distance to TV1, i.e., a safe terminal state. As the FTP OCP yields a solution, the first input $\boldsymbol{u}_{\mathrm{SMPC},0}$ of the planned SMPC trajectory is then applied.

2) Comparison to SMPC and FTP: Throughout the entire simulation, both the SMPC and FTP OCPs remain feasible. Therefore, the SMPC inputs are always applied. Only applying an SMPC algorithm without FTP would therefore yield the same result for this regular scenario.

Unlike SMPC, applying only FTP results in a different solution. As the constraints are more conservative compared to SMPC, the EV never changes lanes to overtake. As indicated by the FTP prediction in Figure 9, the FTP constraints keep the EV in its current lane.

We will use the following metric to compare the performance of SMPC+FTP and FTP. Based on the cost function of the OCP, the applied inputs and resulting states for the entire simulation are analyzed according to

$$J_{\mathrm{sim}} = \sum_{k=1}^{N_{\mathrm{sim}}} \|\Delta\boldsymbol{\xi}_k\|_{\boldsymbol{Q}} + \|\boldsymbol{u}_{k-1}\|_{\boldsymbol{R}} + \|\Delta\boldsymbol{u}_{k-1}\|_{\boldsymbol{S}} \qquad (44)$$

with the simulation steps N_{sim}.

The overall cost for SMPC+FTP is $J_{\mathrm{sim}} = 11.32$, while the overall FTP cost is $J_{\mathrm{sim}} = 4.03\mathrm{e}4$. The cost comparison shows that the SMPC+FTP approach yields a more efficient behavior than an FTP approach. In this case increased efficiency results from keeping the velocity close to the reference velocity.

3) Risk Parameter Variation: In the previously discussed simulation, the risk parameter was chosen to be $\beta = 0.8$. Here, we briefly analyze the effect of varying risk parameters on the EV performance. The risk parameters analyzed range from $\beta = 0.8$ to $\beta = 0.999$. The overall simulation cost, according to (44), for each risk parameter is given in Table IV. The overall costs of the simulation results show that the SMPC behavior and costs for this regular scenario are very similar. However, it can be beneficial regarding the cost to choose a larger risk parameter, as inputs are changed more smoothly. In all five examples the EV behavior is almost similar.

TABLE IV
RISK PARAMETER ANALYSIS

risk parameter β	0.8	0.9	0.95	0.99	0.999
cost J_{sim}	11.21	11.35	11.58	11.34	11.31

4) Varying Simulation Settings: In the previous analysis, only one vehicle configuration is considered. In order to show that the SMPC+FTP method is suitable for various scenarios, we ran 1000 simulations, each consisting of 125 simulation steps, with randomly selected initial vehicle positions and velocities for each simulation run. The EV is located on

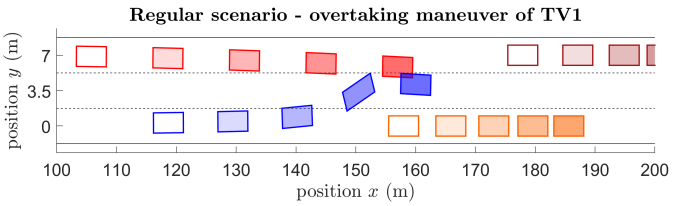

Fig. 10. Shots of the emergency scenario collision applying only SMPC. Fading boxes show past states. The EV is shown in blue.

one of the three lanes, i.e., $d_0 \in \{0, 3.5, 7\}$, with initial longitudinal position $s_0 = 0$ and velocity $v = 27$. The five TVs are randomly placed on one of the three lanes with an initial longitudinal position $x_0^{\text{TV}} \in [-100, 200]$, constant velocity $v_x^{\text{TV}} \in [20, 32]$, and constant $v_y^{\text{TV}} = 0$. It is ensured that all vehicles positioned on the same lane have an initial longitudinal distance $\Delta x \geq 50$ and that TV velocities are chosen such that TVs do not collide with each other.

The SMPC+FTP method successfully handled all 1000 simulation runs and no collisions occurred.

C. Emergency Highway Scenario

After having shown the efficient SMPC+FTP planning for a regular highway scenario, we now illustrate the safety property of the proposed algorithm in an emergency scenario. The initial vehicle states are the same as in the regular scenario. However, in this emergency scenario the TVs change their velocities and lateral positions. Starting at time step $h = 20$, TV5 initiates an emergency braking maneuver with maximal deceleration until reaching a complete halt. This causes TV4 to change lanes to the center lane in order to avoid TV5. TV1 reduces its velocity to $v_x^{\text{TV1}} = 10\,\text{m}\,\text{s}^{-1}$. After having passed TV5, TV4 moves to the left lane to then pass the slower TV2. TV1 also increases its velocity to $v_x^{\text{TV1}} = 20\,\text{m}\,\text{s}^{-1}$.

In the following, SMPC without FTP is analyzed first. Then, the solution of the SMPC+FTP algorithm is presented.

1) SMPC: Applying only SMPC results in optimistic EV trajectory planning, while not considering highly unlikely events. Even though TV4 is slowly moving to the center lane, the EV still moves to the center lane to overtake TV1, as a TV4 lane change is still unlikely. However, at step $h = 25$, TV4 continues to increase its lateral velocity towards the center lane. At this point, there exists no feasible SMPC solution anymore that satisfies the chance constraint. This causes the EV to collide with TV4. The collision sequence is illustrated in Figure 10. While SMPC performs well in regular scenarios without unlikely uncertainty realizations, these rare situations cause major safety issues.

2) SMPC+FTP: We now show how the proposed SMPC+FTP method handles the emergency scenario. The EV states and inputs are given in Figure 11.

Initially, the EV attempts to switch lanes and overtake TV1. However, at step $h = 27$, the SMPC is unable to find a solution. The FTP problem is still solved successfully and the first planned FTP input is applied. For the next four steps, the SMPC OCP remains infeasible, indicated by the pink lines in Figure 11, and the FTP inputs are applied, which are obtained

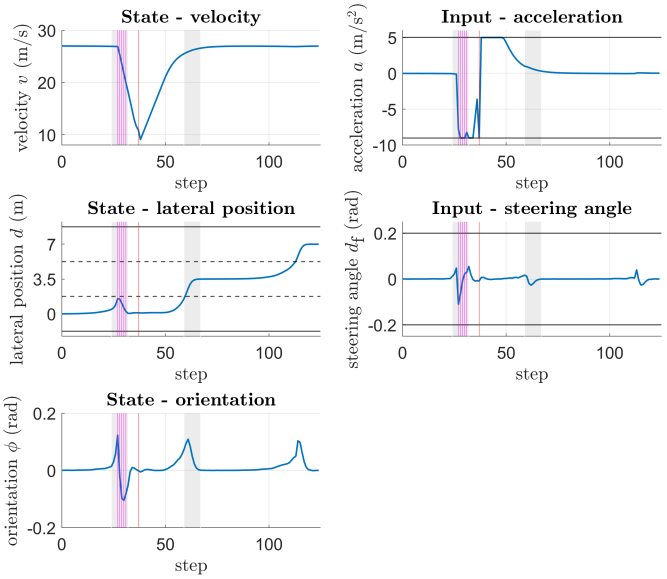

Fig. 11. SMPC+FTP states and inputs for the emergency scenario. Pink vertical lines represent infeasible SMPC and feasible FTP solutions (FTP Mode), red vertical lines show infeasible FTP solutions (Backup Mode). Vehicle motion in the gray areas is illustrated in Figure 12.

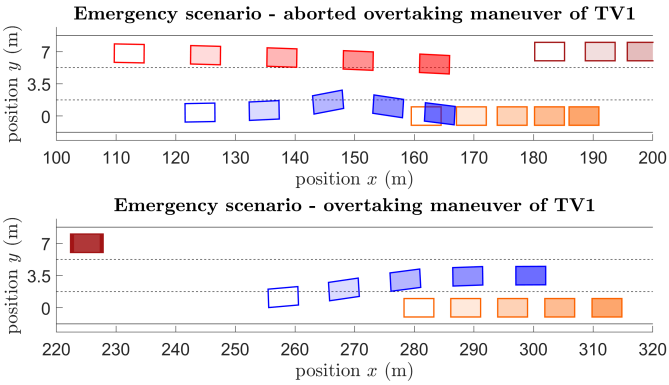

Fig. 12. Shots of the emergency scenario with SMPC+FTP. Fading boxes show past states. The EV is shown in blue.

by successfully solving the FTP OCPs (FTP Mode). The EV slows down and returns to the right lane, as illustrated in the first shot of Figure 12. At step $h = 37$, the SMPC problem is feasible and the EV plans to overtake TV1 again. However, as TV4 is still too close, the FTP is unable to find a new safe backup trajectory if the next planned SMPC input were applied, i.e., the FTP OCP becomes infeasible. The safe input sequence obtained at the previous time step $h = 36$ is applied to the EV (Backup Mode), as indicated by the red line in Figure 11. The EV remains in the right lane until TV4 is far enough away to safely change to the center lane, as shown in the second shot of Figure 12. Eventually, the EV passes TV2 by smoothly switching to the left lane with a small steering angle change. The average computation time for solving the SMPC and FTP OCPs are $0.15\,\text{s}$ and $0.22\,\text{s}$, respectively. The values are higher compared to the regular scenario, as the computation time for infeasible OCPs is significantly larger.

It is also possible to only apply FTP in this emergency scenario. While this leads to safe vehicle behavior throughout

the simulation, the EV does not overtake TV1 and TV2. Comparing the cost yields the following result. Applying FTP to the emergency scenario yields a cost of $J_{\text{sim}} = 4.28\text{e}4$, while the SMPC+FTP cost is $J_{\text{sim}} = 3.34\text{e}4$.

In summary, the simulation scenarios in this section have shown the benefits of the proposed SMPC+FTP method. SMPC optimistically plans trajectories, which are executed as long as there always exists a safe backup trajectory, computed by FTP. In regular scenarios, SMPC+FTP provides benefits known from SMPC. In emergency scenarios, the safety guarantee of FTP holds while the EV is still more efficient compared to applying pure FTP.

VII. Discussion

In some FTP approaches it is required that the vehicle comes to a standstill at the end of the fail-safe trajectory. Here, we only require a certain distance to vehicles ahead and zero orientation offset with respect to the road for the terminal state. This enables the use of a relatively short FTP horizon.

It is possible to get oscillating behavior between applied SMPC inputs and the activation of FTP. This can be avoided by designing the SMPC controller and its constraints less aggressively, as done in the simulation study.

The applied vehicle inputs in the emergency scenario lead to relatively high steering angles. This is not ideal for a smooth vehicle motion. Even though this behavior is acceptable in rare cases, the motion could be optimized by defining more cases for the constraint generation.

The properties of the combined SMPC+FTP method are not restricted to the suggested SMPC and FTP trajectory planners described in Section V-A and Section V-B, respectively. Other SMPC or FTP approaches can be applied.

In dense traffic or unclear traffic situations, humans often do not wait until the desired maneuver is entirely realizable. Instead, humans often slowly initiate maneuvers, causing other vehicles to react. For example, cutting into a lane is often preceded by slight motion towards the other lane so that other vehicles leave extra space. Therefore, it is possible to execute the lane change maneuver successfully, even though it was not possible to safely plan the entire lane change maneuver initially. The SMPC+FTP framework enables automated vehicle motion that comes close to this efficient human behavior.

VIII. Conclusion

In this work we presented a safe and efficient SMPC+FTP method for self-driving vehicles. While SMPC is used to plan optimistic, efficient vehicle trajectories, a fail-safe trajectory planning (FTP) MPC problem ensures that only those SMPC inputs are applied that keep the vehicle in a safe state.

The efficiency of the SMPC+FTP method depends on the proposed constraint generation. Extending and refining the case differentiation will have a positive effect on efficiency. Considering urban automated driving, the SMPC+FTP approach remains valid, however, the case differentiation must be adapted to fit the urban environment. Furthermore, it is also possible to extend the application area to non-transportation applications, such as human-robot collaboration, where uncertainty is always present while safety must still be guaranteed.

Appendix A
Proof of Theorem 1

Proof. Recursive feasibility is proved by induction by showing that $\Gamma_h \neq \emptyset \Rightarrow \Gamma_{h+1} \neq \emptyset$ for all $h \in \mathbb{N}$.

At time step $h = 0$ it holds that $\boldsymbol{\chi}_0^{\boldsymbol{U}_{\text{safe,init}}} \in \Gamma_0$, i.e., an initially safe trajectory exists according to Assumption 4. If the FTP OCP can be solved at step $h = 0$, a new safe input set $\boldsymbol{U}_{\text{safe},0}$ is obtained according to (14) or (15). This new safe input set $\boldsymbol{U}_{\text{safe},0}$ remains valid at step $h = 1$ and ensures that a safe trajectory exists, i.e., $\boldsymbol{\chi}_1^{\boldsymbol{U}_{\text{safe},0}} \in \Gamma_1$. If the FTP OCP is infeasible at step $h = 0$, the shifted previous safe input set remains valid, i.e., $\boldsymbol{U}_{\text{safe},0} = \boldsymbol{U}_{\text{safe,init}}^{\leftarrow}$ according to Section IV-A3. In this case, the shifted safe input set $\boldsymbol{U}_{\text{safe},0} = \boldsymbol{U}_{\text{safe,init}}^{\leftarrow}$ guarantees that $\boldsymbol{\chi}_1^{\boldsymbol{U}_{\text{safe},0}} \in \Gamma_1$. Therefore, $\Gamma_0 \neq \emptyset \Rightarrow \Gamma_1 \neq \emptyset$.

For $h = 1$ it holds that $\boldsymbol{\chi}_1^{\boldsymbol{U}_{\text{safe},0}} \in \Gamma_1$. A feasible FTP OCP yields the new safe input sequences $\boldsymbol{U}_{\text{safe},1}$, such that there exists a safe trajectory $\boldsymbol{\chi}_2^{\boldsymbol{U}_{\text{safe},1}} \in \Gamma_2$. If the FTP OCP is infeasible, reusing the still valid previous safe input set $\boldsymbol{U}_{\text{safe},0}$, i.e., setting $\boldsymbol{U}_{\text{safe},1} = \boldsymbol{U}_{\text{safe},0}^{\leftarrow}$, ensures that $\boldsymbol{\chi}_2^{\boldsymbol{U}_{\text{safe},1}} \in \Gamma_2$.

For time step $h \geq 2$ it holds that $\boldsymbol{\chi}_h^{\boldsymbol{U}_{\text{safe},h-1}} \in \Gamma_h$. If the FTP OCP is feasible, this yields the new safe input sequences $\boldsymbol{U}_{\text{safe},h}$, such that there exists a safe trajectory $\boldsymbol{\chi}_{h+1}^{\boldsymbol{U}_{\text{safe},h}} \in \Gamma_{h+1}$. If the FTP OCP is infeasible, the previous safe input set $\boldsymbol{U}_{\text{safe},h-1}$ is still valid and choosing $\boldsymbol{U}_{\text{safe},h} = \boldsymbol{U}_{\text{safe},h-1}^{\leftarrow}$ ensures that $\boldsymbol{\chi}_{h+1}^{\boldsymbol{U}_{\text{safe},h}} \in \Gamma_{h+1}$.

Therefore, $\boldsymbol{\chi}_{h+1}^{\boldsymbol{U}_{\text{save},h}} \in \Gamma_{h+1}$ holds for all $h \in \mathbb{N}$, i.e., the proposed method is safe and recursively feasible. □

Acknowledgment

The authors thank Daniel Althoff, Matthias Althoff, and Christian Pek for valuable discussions.

References

[1] D.Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967 – 2986, 2014.

[2] R. Soloperto, J. Khler, F. Allgwer, and M. A. Mller. Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control. In *2019 18th European Control Conference (ECC)*, pages 811–817, 2019.

[3] S. Dixit, U. Montanaro, M. Dianati, D. Oxtoby, T. Mizutani, A. Mouzakitis, and S. Fallah. Trajectory planning for autonomous high-speed overtaking in structured environments using robust mpc. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2310–2323, 2020.

[4] A. Mesbah. Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems*, 36(6):30–44, Dec 2016.

[5] M. Farina, L. Giulioni, and R. Scattolini. Stochastic linear model predictive control with chance constraints a review. *Journal of Process Control*, 44(Supplement C):53 – 67, 2016.

[6] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli. Stochastic predictive control of autonomous vehicles in uncertain environments. In *12th International Symposium on Advanced Vehicle Control*, Tokyo, Japan, 2014.

[7] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli. Scenario model predictive control for lane change assistance and autonomous driving on highways. *IEEE Intelligent Transportation Systems Magazine*, 9(3):23–35, Fall 2017.

[8] S. Magdici and M. Althoff. Fail-safe motion planning of autonomous vehicles. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 452–458, 2016.

[9] S. Söntges and M. Althoff. Computing the drivable area of autonomous road vehicles in dynamic road scenes. *IEEE Transactions on Intelligent Transportation Systems*, 19(6):1855–1866, 2018.

[10] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold. Stochastic model predictive control with a safety guarantee for autonomous driving, 2020. arXiv: 2009.09381.

[11] C. Hubmann, J. Schulz, M. Becker, D. Althoff, and C. Stiller. Automated driving in uncertain environments: Planning with interaction and uncertain maneuver prediction. *IEEE Transactions on Intelligent Vehicles*, 3(1):5–17, 2018.

[12] B. Mirchevska, C. Pek, M. Werling, M. Althoff, and J. Boedecker. High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2156–2162, 2018.

[13] U. Rosolia, A. Carvalho, and F. Borrelli. Autonomous racing using learning model predictive control. In *2017 American Control Conference (ACC)*, pages 5115–5120, 2017.

[14] J. Kabzan, L. Hewing, A. Liniger, and M. N. Zeilinger. Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, 4(4):3363–3370, 2019.

[15] A. Wischnewski, J. Betz, and B. Lohmann. A model-free algorithm to safely approach the handling limit of an autonomous racecar. In *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, 2019.

[16] T. Stahl, A. Wischnewski, J. Betz, and M. Lienkamp. Multilayer graph-based trajectory planning for race vehicles in dynamic scenarios. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 3149–3154, 2019.

[17] B. Gutjahr, L. Grll, and M. Werling. Lateral vehicle trajectory optimization using constrained linear time-varying mpc. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1586–1595, June 2017.

[18] B. Yi, P. Bender, F. Bonarens, and C. Stiller. Model predictive trajectory planning for automated driving. *IEEE Transactions on Intelligent Vehicles*, 4(1):24–38, 2019.

[19] C. Pek and M. Althoff. Computationally efficient fail-safe trajectory planning for self-driving vehicles using convex optimization. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 1447–1454, Nov 2018.

[20] Jeremy H. Gillula. *Guaranteeing Safe Online Machine Learning via Reachability Analysis*. Dissertation, Stanford University, 2013.

[21] B. Schrmann, N. Kochdumper, and M. Althoff. Reachset model predictive control for disturbed nonlinear systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 3463–3470, 2018.

[22] M. Althoff, D. Heß, and F. Gambert. Road occupancy prediction of traffic participants. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 99–105, 2013.

[23] M. Althoff, M. Koschi, and S. Manzinger. Commonroad: Composable benchmarks for motion planning on roads. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 719–726, Los Angeles, USA, 2017.

[24] S. Manzinger, C. Pek, and M. Althoff. Using reachable sets for trajectory planning of automated vehicles. *IEEE Transactions on Intelligent Vehicles*, pages 1–1, 2020.

[25] A.T. Schwarm and M. Nikolaou. Chance-constrained model predictive control. *AIChE Journal*, 45(8):1743–1752, 1999.

[26] L. Blackmore, M. Ono, A. Bektassov, and B.C. Williams. A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *Trans. Rob.*, 26(3):502–517, June 2010.

[27] D. Lenz, T. Kessler, and A. Knoll. Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 292–297, Seoul, South Korea, 2015.

[28] J. Suh, H. Chae, and K. Yi. Stochastic model-predictive control for lane change decision of automated driving vehicles. *IEEE Transactions on Vehicular Technology*, 67(6):4771–4782, June 2018.

[29] G. Schildbach and F. Borrelli. Scenario model predictive control for lane change assistance on highways. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 611–616, Seoul, South Korea, 2015.

[30] T. Brüdigam, M. Olbrich, M. Leibold, and D. Wollherr. Combining stochastic and scenario model predictive control to handle target vehicle uncertainty in autonomous driving. In *2018 21st IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pages 1317–1324, 2018.

[31] T. Brüdigam, F. di Luzio, L. Pallottino, D. Wollherr, and M. Leibold. Grid-based stochastic model predictive control for trajectory planning in uncertain environments. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8, 2020.

[32] S. Steyer, G. Tanzmeister, and D. Wollherr. Grid-based environment estimation using evidential mapping and particle tracking. *IEEE Transactions on Intelligent Vehicles*, 3(3):384–396, Sep. 2018.

[33] S. Steyer, C. Lenk, D. Kellner, G. Tanzmeister, and D. Wollherr. Grid-based object tracking with nonlinear dynamic state and shape estimation. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–20, 2019.

[34] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli. Kinematic and dynamic vehicle models for autonomous driving control design. In *2015 IEEE Intelligent Vehicles Symposium (IV)*, pages 1094–1099, Seoul, South Korea, 2015.

[35] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control*. Springer-Verlag, London, 2017.

Tim Brüdigam received bis B.S. and M.S. degree in electrical engineering from the Technical University of Munich (TUM), Germany in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree at TUM. During his studies, he was a research scholar at the University of South Carolina in 2014 and the California Institute of Technology in 2016. He joined the Chair of Automatic Control Engineering at TUM as a research associate in 2017. His main research interest lies in advancing Model Predictive Control (MPC), especially stochastic MPC, with possible application in automated driving.

Michael Olbrich received his B.S. and M.S. degree in electrical engineering from the Technical University of Munich in 2017 and 2019, respectively. He is currently a Research Assistant with the Department of Computer Science, Chair of Control Engineering at the University of Augsburg, where he is pursuing his Ph.D. degree. His research interests include model predictive control and optimal trajectory planning and control, particularly in its application to microactuators.

Dirk Wollherr received the Dipl.-Ing. (2000), Dr.-Ing. (2005) and Habilitation (2013) degree in electrical engineering from Technical University of Munich, Germany. He is a Senior Researcher in robotics, control and cognitive systems. His research interests include automatic control, robotics autonomous mobile robots, human-robot interaction and socially aware collaboration and joint action. He is on the management boards of several conferences and Associate Editor of the IEEE Transactions on Mechatronics.

Marion (nee Sobotka) Leibold received the Diploma degree in applied mathematics from the Technical University of Munich, Germany, in 2002, and the Ph.D. degree in motion planning and control of legged robots from the Faculty of Electrical Engineering and Information Technology, Technical University of Munich, in 2007. She is currently a Senior Researcher with the Faculty of Electrical Engineering and Information Technology, Institute of Automatic Control Engineering, Technical University of Munich. Her research interests include optimal control and nonlinear control theory, and the applications to robotics.