

Experimental Validation of Safe MPC for Autonomous Driving in Uncertain Environments

Ivo Batkovic^{ID}, Ankit Gupta^{ID}, Mario Zanon^{ID}, Member, IEEE, and Paolo Falcone^{ID}

Abstract—The full deployment of autonomous driving systems on a worldwide scale requires that the self-driving vehicle can be operated in a provably safe manner, i.e., the vehicle must be able to avoid collisions in any possible traffic situation. In this article, we propose a framework based on model predictive control (MPC) that endows the self-driving vehicle with the necessary safety guarantees. In particular, our framework ensures constraint satisfaction at all times while tracking the reference trajectory as close as obstacles allow, resulting in a safe and comfortable driving behavior. To discuss the performance and real-time capability of our framework, we provide first an illustrative simulation example, and then, we demonstrate the effectiveness of our framework in experiments with a real test vehicle.

Index Terms—Autonomous driving, nonlinear predictive control, recursive feasibility, safety, uncertain constraints.

I. INTRODUCTION

IN THE past decade, autonomous systems have attracted the attention of many researchers and companies on a global scale. The research questions and problems posed by the autonomous driving field have in particular been in the research focus since there is a great potential in enabling such technologies. Already today, advanced driver assist systems (ADASs) are being available for highway-like situations and have been proven to reduce the frequency and severity of some traffic accidents. To that end, it is believed that fully autonomous driving systems are a key component in reaching the vision of eliminating all fatal road traffic accidents [1], [2].

However, in order to fully deploy highly automated driving technologies in more general settings, such as urban driving,

Manuscript received 17 September 2022; revised 27 April 2023; accepted 27 April 2023. Date of publication 27 July 2023; date of current version 18 August 2023. This work was supported in part by the Knut and Alice Wallenberg Foundation through the Wallenberg Artificial Intelligence, Autonomous Systems and Software Program (WASP). Recommended by Associate Editor F. Dabbene. (*Corresponding author: Ivo Batkovic*.)

Ivo Batkovic is with the Mechatronics Group, Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden, and also with the Research Department, Zenseact AB, 417 56 Gothenburg, Sweden (e-mail: ivo.batkovic@chalmers.se).

Ankit Gupta is with the Research Department, Zenseact AB, 417 56 Gothenburg, Sweden (e-mail: ankit.gupta@zenseact.com).

Mario Zanon is with the IMT School for Advanced Studies Lucca, 55100 Lucca, Italy (e-mail: mario.zanon@imtlucca.it).

Paolo Falcone is with the Mechatronics Group, Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden, and also with the Dipartimento di Ingegneria “Enzo Ferrari,” Università di Modena e Reggio Emilia, 42121 Reggio Emilia, Italy (e-mail: falcone@chalmers.se).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCST.2023.3291562>.

Digital Object Identifier 10.1109/TCST.2023.3291562

the autonomous system needs not only to reliably sense the surrounding environment but also to safely interact with it. To that end, the open problems related to safe autonomous driving span from the design of cost-effective, robust, and reliable sensors (e.g., cameras, light detection and ranging (LiDAR), radars, GPS, and HD-maps) to the development of robust perception and motion planning and control algorithms. In this article, we assume that the state-of-the-art sensing and localization algorithms are solved and only consider the open problem of ensuring safe autonomous driving in uncertain environments where the future motion of other road users cannot be known *a priori* but must instead be estimated.

To solve the problem of “how” self-driving vehicles should behave, i.e., the planning and control of the vehicle motion, optimization-based techniques have proven to be a favorable choice. In particular, model predictive control (MPC) has been commonly used since it accommodates the control of (non)linear systems that can be subject to time-varying references and (non)linear constraints. Common approaches in the literature use MPC to approach the problems from different points of view, such as planning and control [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], energy consumption minimization [14], [15], [16], [17], and optimal coordination [18], [19], [20]. In most of these settings, a reference trajectory (or path) is already given, and the optimization problem is designed to minimize a performance criterion while satisfying constraints that are either imposed by the internal system or by the environment. Notable examples include: 1) maneuvering large vehicles in highly constrained environments [13], [21], [22]; 2) controlling vehicles in an intersection safely and efficiently [14], [20]; and 3) controlling a vehicle in environments with uncontrollable moving obstacles [4], [5]. However, while it is well known how to design an MPC controller such that closed-loop stability with respect to a reference trajectory (or path) is obtained and that constraint satisfaction holds at all times [23], [24], the results in the literature rely on assumptions that can be challenging, or even impossible, to satisfy for practical autonomous driving settings. In other words, one cannot directly apply the developed MPC theory to ensure that a self-driving vehicle will make safe decisions at all times, i.e., ensuring recursive feasibility of the MPC controller still remains an open problem for environments where other noncontrollable road users are present. We note that while the work in [20] showed that it is possible to formulate conditions for safe optimal coordination of multiple vehicles in intersections, it relied on an assumption that there

exists a “coordinator” that decides how and when each vehicle should enter the intersection. To that end, such an approach is not directly applicable, or practical, for general autonomous driving settings, since a self-driving vehicle needs to interact with other human-driven vehicles, cyclists, and pedestrians.

In order to also account for unforeseen events, contingency MPC (CMPC) was introduced in [25] and [26], where the MPC problem instead predicts two trajectories that are connected by their first input, where one of the trajectories acts as a contingency plan accounting for the worst case outcome, while the other trajectory considers the nominal situation. The authors have shown both through simulations and real experiments that the CMPC framework can serve as a credible strategy to augment a deterministic MPC controller with robustness. However, while the additional contingency trajectory helps against unforeseen events, the CMPC framework does not provide recursive feasibility guarantees for the autonomous driving setting. In a similar light, Brüdigam et al. [27], [28] formulated a two-step approach that accounts for safety. In this framework, a stochastic MPC (SMPC) problem is first solved for the nominal case. Then, the input of the SMPC problem is used only if a backup MPC formulation, which is more conservative and accounts for worst case outcomes, can verify that it is safe. Otherwise, the trajectory of the backup MPC takes over and steers the system to a safe configuration. While Alsterda et al. [25], Alsterda and Gerdes [26], and Brüdigam et al. [27], [28] addressed safety concerns using an MPC framework, they either rely on extending the state space of the MPC problem or solving two different problems separately.

In this article, we address the problem of ensuring the safe operation of an autonomous vehicle in uncertain environments, e.g., how a self-driving vehicle should be operated so that it never collides with other road users. We base our approach on the novel safe model predictive flexible trajectory tracking control (MPFTC) framework [29] and show in practice how a safe vehicle controller can be designed for urban autonomous driving settings. In particular, our contributions are the following. First, we introduce an MPC problem formulation that ensures robust constraint satisfaction for the self-driving vehicle while only slightly modifying the standard MPC design procedure and relying on a set of (verifiable) assumptions. Note that, contrary to our contribution, works [25], [26], [27], [28] do not provide a general framework for safe autonomous driving. In addition, we show through simulations that safety, even in trivial traffic situations, can become jeopardized with an MPC controller if the required assumptions do not satisfy the safe MPC formulation. Finally, by deploying the safe MPC framework on a real test vehicle, we show through a real experiment the performance and real-time capabilities of the framework.

Contributions of This Article: The work presented in [29] provides a generic safe framework, which is not trivial to apply to the case of autonomous driving. This article shows how to specialize the generic formulation in [29] to urban autonomous driving situations, and that it can indeed be used to guarantee safety. In particular, we show the effectiveness of the framework in simulations and also deploy it on a real vehicle.

Article Structure: Section II introduces the MPC problem statement and the assumptions needed to satisfy the safety of the control algorithm, while Section III discusses and shows how to enforce all required assumptions in practice. In particular, Section III-A introduces the considered system model and controller details, Section III-B discusses how to model the a priori unknown constraints in order to ensure robust constraint satisfaction, and Section III-C shows how to derive the terminal conditions needed to ensure recursive feasibility of the controller. In Sections IV and V, we show the performance of the proposed controller through simulations and real experiments, respectively. Finally, in Section VI, we draw conclusions and outline future work directions.

II. PROBLEM FORMULATION

Our objective is to design a controller for urban autonomous driving applications that allows for comfortable driving in environments where other road users are present, e.g., pedestrians, cyclists, and other vehicles. Before we state the problem formulation, we first introduce terms and notations that will be used throughout this article.

A. Notation

We consider a discrete-time nonlinear system defined by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \quad (1)$$

where $\mathbf{x}_k \in \mathbb{R}^{n_x}$ denotes the state and $\mathbf{u}_k \in \mathbb{R}^{n_u}$ denotes the control input at time k . The system is subject to state and input constraints of two categories: a priori known constraints $h(\mathbf{x}, \mathbf{u}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_h}$, which might be time-varying but which are fully known beforehand; and a priori unknown constraints $g(\mathbf{x}, \mathbf{u}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_g}$, whose functional form is known a priori, but whose value is not. In both cases, the state and input must satisfy $h(\mathbf{x}, \mathbf{u}) \leq 0$ and $g(\mathbf{x}, \mathbf{u}) \leq 0$, and all inequalities are defined elementwise. Examples of a priori known constraints include engine torque limitations and braking force, while examples of a priori unknown constraints include collision avoidance with other road users.

Due to the nature of the known and unknown constraints, we use $g_{n|k}(\mathbf{x}, \mathbf{u})$ to denote function g at time n , given the information available at time k . Moreover, we will denote by $g_n(\mathbf{x}, \mathbf{u}) := g_{n|\infty}(\mathbf{x}, \mathbf{u}) = g_{n|k}(\mathbf{x}, \mathbf{u})$, $\forall k \geq n$ the real constraint, since in general, $g_{n|k}(\mathbf{x}, \mathbf{u}) \neq g_n(\mathbf{x}, \mathbf{u})$, $\forall k < n$. Note that instead for a priori known constraints, $h_{n|k}(\mathbf{x}, \mathbf{u}) := h_n(\mathbf{x}, \mathbf{u})$ holds $\forall k$ by definition. Throughout the remainder of this article, we also apply the same notation to the predicted state and inputs, e.g., $\mathbf{x}_{n|k}$ and $\mathbf{u}_{n|k}$ denote the predicted state and input at time n given the current time k , respectively. In addition, we will use $\mathbb{I}_a^b := \{a, a + 1, \dots, b\}$ to denote a set of integers.

Finally, we provide the following definition of safety, which will be the foundation of our safety argumentation.

Definition 1 (Safety): A controller is said to be safe in a given set $\mathcal{S} \subseteq \mathbb{R}^{n_x}$ if $\forall \mathbf{x}_0 \in \mathcal{S}$ if it is robust control invariant for the closed-loop system, i.e., it generates control inputs $\mathbf{U} = \{\mathbf{u}_0, \dots, \mathbf{u}_\infty\}$ and corresponding state trajectories $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_\infty\}$ such that $h_k(\mathbf{x}_k, \mathbf{u}_k) \leq 0$ and $g_k(\mathbf{x}_k, \mathbf{u}_k) \leq 0$, $\forall k \geq 0$.

B. Problem Statement

Since our objective is to design a controller that is safe and comfortable, the controller must: 1) guarantee safety, in the sense of Definition 1, at all times and 2) control system (1) such that the state and input $(\mathbf{x}_k, \mathbf{u}_k)$ track a user-defined parameterized reference $\mathbf{r}(\tau) := (\mathbf{r}^x(\tau), \mathbf{r}^u(\tau))$ as closely as possible. The parameter τ can be interpreted as a “fictitious time,” as a natural choice for the reference in MPC is to use time as a parameter. Note that, if τ is selected to be time, its natural dynamics are given by

$$\tau_{k+1} = \tau_k + t_s \quad (2)$$

where t_s is the sampling time for sampled-data systems and $t_s = 1$ in the discrete-time framework.

To ensure that requirements 1) and 2) are satisfied in the urban autonomous driving setting, we use the framework proposed in [29], where the idea is to adjust the reference trajectory in order to avoid aggressive tracking behaviors. This is done by adapting the dynamics of the reference trajectory by means of the parameter τ , which acts as a fictitious time through the dynamics given by

$$\tau_{k+1} = \tau_k + t_s + v_k \quad (3)$$

where v is an additional auxiliary control input and τ comes as an auxiliary state. With this relaxed timing law of the reference trajectory, we proceed to formulate the practical version of MPFTC [29, Sec. V.A] through the following problem formulation:

$$\min_{\mathbf{x}, \mathbf{u}} \sum_{n=k}^{k+N-1} q_r(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \tau_{n|k}) + w v_{n|k}^2 \quad (4a)$$

$$+ p_r(\mathbf{x}_{k+N|k}, \tau_{k+N|k}) \quad (4b)$$

$$\text{s.t. } \mathbf{x}_{k|k} = \mathbf{x}_k, \tau_{k|k} = \tau_k \quad (4c)$$

$$\mathbf{x}_{n+1|k} = f(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}), n \in \mathbb{I}_k^{k+M-1} \quad (4d)$$

$$\tau_{n+1|k} = \tau_{n|k} + t_s + v_{n|k}, n \in \mathbb{I}_k^{k+M-1} \quad (4e)$$

$$h_n(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, n \in \mathbb{I}_k^{k+M-1} \quad (4f)$$

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0, n \in \mathbb{I}_k^{k+M-1} \quad (4g)$$

$$\mathbf{x}_{k+n|k} \in \mathcal{X}_r^s(\tau_{k+n|k}), n \in \mathbb{I}_k^{k+M-1} \quad (4h)$$

$$\mathbf{x}_{k+M|k} \in \mathcal{X}_{\text{safe}}(\tau_{k+M|k}) \subseteq \mathcal{X}_r^s(\tau_{k+M|k}) \quad (4h)$$

where k is the current time, N is the prediction horizon associated with a cost, $M \geq N$ is the full prediction horizon, q_r and p_r are cost functions that penalize deviations from the reference trajectory $\mathbf{r}(\tau) = (\mathbf{r}^x(\tau), \mathbf{r}^u(\tau))$, and $w > 0$ is associated with the cost of the auxiliary input $v_{n|k}$. The predicted state and control inputs are denoted by $\mathbf{x}_{n|k}$ and $\mathbf{u}_{n|k}$, respectively. Constraint (4b) ensures that the prediction starts at the current state, while constraints (4c) and (4d) enforce the system dynamics. Constraints (4e) represent known constraints, e.g., state and actuation limits, while constraints (4f) represent collision avoidance with respect to other road users, e.g., pedestrians, cyclists, and other vehicles. Finally, constraints (4g) and (4h) define a stabilizing set and a safe set, respectively. In order to provide additional intuition behind this problem formulation, we point out next the main differences between Problem (4) and a standard MPC formulation.

The first difference is that (4) uses the MPFTC framework, i.e., it depends on the auxiliary inputs τ and v , and the terminal sets are parameterized with the parameter τ . The motivation behind this design choice is that adapting τ offers flexibility in the tracking behavior whenever the constraints are causing the unmodified reference to become infeasible or the state \mathbf{x} is far from the reference, i.e., it allows the system to behave less aggressive when trying to keep up with the reference trajectory. The artificial reference only changes the cost of the MPC problem such that its introduction does not change how restrictive the approach is, as the constraints are unaffected. Note that, enforcing the constraint $v = 0$ forces the reference trajectory to move as in standard MPC.

The second difference is that Problem (4) can be seen as a two-stage problem, where the first stage $n \in \mathbb{I}_k^{k+N}$ defines the MPC problem and the second stage $n \in \mathbb{I}_{k+N+1}^{k+M}$ defines the terminal set implicitly, in which the properties of sets $\mathcal{X}_r^s(t)$ and $\mathcal{X}_{\text{safe}}(t)$ will be stated in Assumptions 3 and 5. This implicit terminal set formulation relates to the safety design, which ensures that the state $\mathbf{x}_{k+N|k}$ is able to reach the safe set $\mathcal{X}_{\text{safe}}(t_{k+M|k})$ in a finite amount of time $M - N \geq 0$ while satisfying the system dynamics and the a priori known and unknown constraint. For a more detailed discussion related to the design of Problem (4), we refer the reader to [29].

In order to prove that the controller based on Problem (4) is safe, we first need to introduce the following assumptions.

Assumption 1 (Regularity): The system model f is continuous, and the stage cost $q_r : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ and terminal cost $p_r : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ are continuous at reference and satisfy $q_r(\mathbf{r}^x(t_k), \mathbf{r}^u(t_k), t_k) = 0$ and $p_r(\mathbf{r}^x(t_k), t_k) = 0$. In addition, $q_r(\mathbf{x}_k, \mathbf{u}_k, t_k) \geq \alpha_1(\|\mathbf{x}_k - \mathbf{r}^x(t_k)\|)$ for all feasible $\mathbf{x}_k, \mathbf{u}_k$, and $p_r(\mathbf{x}_k, t_k) \leq \alpha_2(\|\mathbf{x}_k - \mathbf{r}^x(t_k)\|)$, where α_1 and α_2 are \mathcal{K}_{∞} -functions.

Assumption 2 (Reference Feasibility): The reference trajectory is feasible for the system dynamics, i.e., $\mathbf{r}^x(t_k + t_s) = f(\mathbf{r}^x(t_k), \mathbf{r}^u(t_k))$, and the reference satisfies the known constraints (4e), i.e., $h_n(\mathbf{r}^x(t_n), \mathbf{r}^u(t_n)) \leq 0$, for all $n \in \mathbb{I}_0^\infty$.

Assumption 3 (Stabilizing Terminal Conditions): There exists a stabilizing terminal set \mathcal{X}_r^s and a terminal control law $\kappa_r^s(\mathbf{x})$ yielding $\mathbf{x}_+^s = f(\mathbf{x}, \kappa_r^s(\mathbf{x}))$ and $t_+ = t_k + t_s$ such that $p(\mathbf{x}_+^s, t_+) - p(\mathbf{x}, t) \leq -q(\mathbf{x}, \kappa_r^s(\mathbf{x}), t)$ and $\mathbf{x}_+^s \in \mathcal{X}_r^s(t_+)$, and $h_n(\mathbf{x}, \kappa_r^s(\mathbf{x})) \leq 0$, for all $n, k \in \mathbb{I}_0^\infty$.

While Assumptions 1–3 coincide with those commonly used in standard MPC to prove stability (see, e.g., [24], [30]), we must also introduce the following assumptions in order to guarantee safety for the setting of autonomous driving situations where other road users are present.

Assumption 4 (Unknown Constraint Dynamics): The a priori unknown constraint functions satisfy $g_{n|k+1}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k})$ for all $n \geq k$.

This assumption only requires that the a priori unknown constraint $g_{n|k}$, which is associated with the predicted uncertainty of other road users, has a consistent characterization. In the context of autonomous driving, function g characterizes, among other constraints, the positions in which no collision with other road users occurs. A position is safe at a given time n if no other road user can be in that position at time n . By using a prediction model for other road users, one

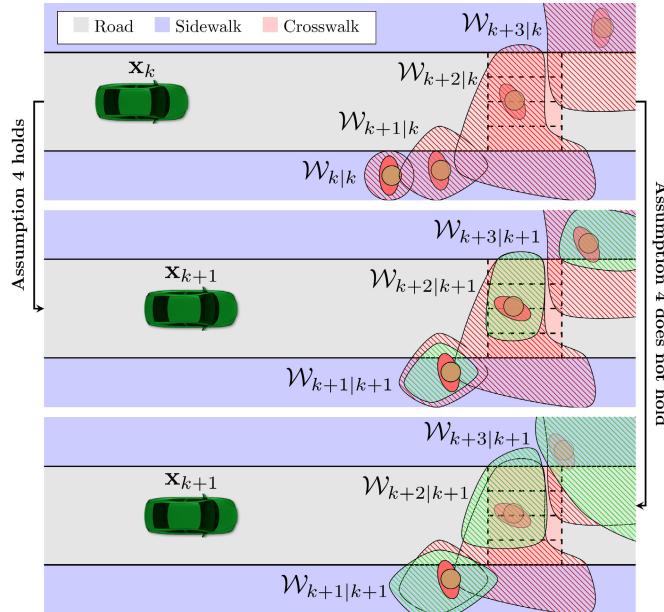


Fig. 1. Top: initial prediction for a pedestrian at time k . Middle and bottom: different predictions at time $k + 1$. The middle illustrates a model that satisfies (9), i.e., satisfying Assumption 4, while the bottom does not satisfy (9), and hence, Assumption 4 is also not satisfied.

must consider as unsafe those positions, which can be reached by other road users. Assumption 4 then requires that the reachable sets predicted at time k contain the corresponding reachable sets predicted at time $k + 1$, as shown in Fig. 1. This construction is discussed in depth in Section III-B.

The final following assumption is needed to ensure that the controller based on Problem (4) is recursively feasible.

Assumption 5: There exists a robust invariant set denoted $\mathcal{X}_{\text{safe}} \subseteq \mathbb{R}^{n_x}$ such that, for all $\mathbf{x} \in \mathcal{X}_{\text{safe}}$, there exists a safe control set $\mathcal{U}_{\text{safe}} \subseteq \mathbb{R}^{n_u}$ entailing that $f(\mathbf{x}, \mathbf{u}_{\text{safe}}) \in \mathcal{X}_{\text{safe}}$ and $h_n(\mathbf{x}, \mathbf{u}_{\text{safe}}) \leq 0$, for all $\mathbf{u}_{\text{safe}} \in \mathcal{U}_{\text{safe}}$. Moreover, for all $\mathbf{x} \in \mathcal{X}_{\text{safe}}$, the a priori unknown constraints can never be positive, i.e., $g_{n|k}(\mathbf{x}, \mathbf{u}_{\text{safe}}) \leq 0$ holds for all $\mathbf{x} \in \mathcal{X}_{\text{safe}}$, and $\mathbf{u}_{\text{safe}} \in \mathcal{U}_{\text{safe}}$.

Note that Assumption 5 is the translation of Definition 1 in mathematical terms, which states that any set of states that are safe must be forward invariant and it must be such that the constraints $g_{n|k}$ are not violated. On the contrary, if no such safe configuration exists, then system (1) is intrinsically unsafe. Hence, Assumption 5 is essential for recursive feasibility of the controller based on Problem (4). In the context of autonomous driving, a safe set could involve stopping the car in a suitable location. We will provide more details regarding the safe set in Section III-C.

We are now ready to state the following result.

Proposition 1 (Recursive Feasibility): Suppose that Assumptions 1–5 hold and that Problem (4) is feasible for the initial state (\mathbf{x}_k, t_k) . Then, system (1) in a closed loop with the solution of (4) applied in receding horizon is safe (recursively feasible) at all times.

Proof: The full proof is given in [29, Th. 2]. \square

While this proposition states under which assumptions recursive feasibility (safety) can be ensured, it does not discuss how to enforce them in practice for our considered

autonomous driving setting. In Section III, we therefore show how to apply the theory from Proposition 1 in practice.

III. IMPLEMENTATION OF SAFE MPC

In this section, we discuss how our proposed framework can be applied in practice and how satisfaction of Assumptions 1–5 can be enforced. In Section III-A, we introduce the vehicle model, cost, and system constraints that satisfy Assumptions 1 and 2. Then, in Section III-B, we show that Assumption 4 can be satisfied by using prediction models of the environment to construct the a priori unknown constraints $g_{n|k}$. Finally, in Section III-C, we derive the terminal conditions that satisfy Assumptions 3 and 5.

A. Vehicle Model and System Constraints

We model the vehicle dynamics using the single-track vehicle model

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{\delta} \\ \dot{\alpha} \\ \dot{v} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} v \cos(\psi) \\ v \sin(\psi) \\ \frac{v}{l} \tan(\delta) \\ \alpha \\ w_0^2(\delta^{\text{sp}} - \delta) - 2w_0w_1\alpha \\ a \\ t_{\text{acc}}(a^{\text{req}} - a) \end{bmatrix} \quad (5)$$

where x and y denote the position coordinates in a global frame, ψ is the orientation angle, δ is the steering angle, α is the steering angle rate, v is the velocity, and a is the acceleration. The control inputs are given by the acceleration request a^{req} and steering angle setpoint δ^{sp} . Finally, w_0 , w_1 , and t_{acc} are model constants defining the steering actuator and acceleration dynamics.

In order to track the user-defined reference \mathbf{r} , we express the vehicle kinematics in the frame of the reference path [31]

$$\begin{bmatrix} \dot{s} \\ \dot{e}_y \\ \dot{e}_\psi \\ \dot{\delta} \\ \dot{\alpha} \\ \dot{v} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} v \cos(e_\psi)(1 - \kappa^{\text{r}}(s)e_y)^{-1} \\ v \sin(e_\psi) \\ vl^{-1} \tan(\delta) - sl^{-1} \tan(\delta^{\text{r}}(s)) \\ \alpha \\ w_0^2(\delta^{\text{sp}} - \delta) - 2w_0w_1\alpha \\ a \\ t_{\text{acc}}(a^{\text{req}} - a) \end{bmatrix} = \begin{bmatrix} \text{aux.} \\ \text{lat.} \\ \text{kin.} \\ \text{lon.} \\ \text{kin.} \end{bmatrix} \quad (6)$$

$$\mathbf{x} = [e_y \ e_\psi \ \delta \ \alpha \ v \ a]^T, \ \mathbf{u} = [a^{\text{req}} \ \delta^{\text{sp}}]^T$$

where s is the longitudinal position along the path, $\kappa^{\text{r}}(s)$ is the path curvature along the path, e_y is the lateral displacement error, e_ψ is the yaw error with respect to the reference \mathbf{r} , and δ^{r} is the reference steering angle. Since the reference in (6) becomes parameterized along the longitudinal position s , we consider s to be an auxiliary state [as opposed to τ in (4)], which we do not track, as we are only interested in tracking the velocity while remaining on the reference.

For both simulations and experiments in Sections IV and V, we consider that system (6) is subject to the following known constraints:

$$\begin{aligned} \|e_y\| &\leq \bar{e}_y, \quad \|e_\psi\| \leq \bar{e}_\psi, \quad \|\delta\| \leq \bar{\delta}, \quad \|\delta^{\text{sp}}\| \leq \bar{\delta} \\ 0 \leq v &\leq \bar{v}, \quad \underline{a} \leq a \leq \bar{a}, \quad -\underline{a} \leq a^{\text{req}} \leq \bar{a}, \quad \|\alpha\| \leq \bar{\alpha}. \end{aligned}$$

The costs q_r and p_r are chosen as quadratic costs with their minimum on the reference $\mathbf{r} = (\mathbf{r}^x, \mathbf{r}^u)$, i.e.,

$$q_r(\mathbf{x}_k, \mathbf{u}_k, s_k) := \begin{bmatrix} \mathbf{x}_k - \mathbf{r}^x(s_k) \\ \mathbf{u}_k - \mathbf{r}^u(s_k) \end{bmatrix}^\top W \begin{bmatrix} \mathbf{x}_k - \mathbf{r}^x(s_k) \\ \mathbf{u}_k - \mathbf{r}^u(s_k) \end{bmatrix}$$

$$p_r(\mathbf{x}_k, s_k) := (\mathbf{x}_k - \mathbf{r}^x(s_k))^\top P (\mathbf{x}_k - \mathbf{r}^x(s_k))$$

where $W := \text{blockdiag}(Q, R)$ is constructed using positive-definite matrices $Q \in \mathbb{R}^{6 \times 6}$ and $R \in \mathbb{R}^{2 \times 2}$. The computation of the terminal cost matrix $P \in \mathbb{R}^{6 \times 6}$ is discussed in Section III-C.

The next step in formulating Problem (4) is to construct the a priori unknown constraints $g_{n|k}$. Therefore, we provide next a discussion on how prediction models of the surrounding environment can be used.

B. Predictive Collision Avoidance Constraints

In order to apply the results from Proposition 1, one must first understand how constraints $g_{n|k}$ can be modeled such that Assumption 4 is satisfied. This section will therefore provide a formal description of how $g_{n|k}$ can be constructed and then show how a pedestrian prediction model can be derived.

We introduce function $\gamma(\mathbf{x}, \mathbf{u}, \mathbf{w}) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_g}$, with an uncertain variable \mathbf{w} that relates the uncertainty to the a priori unknown constraints, where the uncertainty for each prediction time has a bounded support, i.e., $\mathbf{w}_{n|k} \in \mathcal{W}_{n|k} \subseteq \mathbb{R}^{n_w}$. Using γ , we can then define the constraint g at each prediction time n as

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) := \max_{\mathbf{w}_{n|k} \in \mathcal{W}_{n|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}) \quad (7)$$

which implies robust constraint satisfaction, i.e.,

$$g_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \leq 0 \Leftrightarrow \left\{ \begin{array}{l} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k}) \leq 0 \\ \forall \mathbf{w}_{n|k} \in \mathcal{W}_{n|k}. \end{array} \right.$$

Furthermore, with a slight abuse of notation, we also denote

$$\gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathcal{W}_{n|k}) := \max_{\mathbf{w}_{n|k} \in \mathcal{W}_{n|k}} \gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}, \mathbf{w}_{n|k})$$

which will be used to simplify the discussion in Section III-B.

For our considered setting, the variable $\mathbf{w}_{n|k}$ represents the state of a dynamical system modeling the future motion of other road users and can be written as

$$\mathbf{w}_{n+1|k} = \omega(\mathbf{w}_{n|k}, \xi_{n|k}, \mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \quad (8)$$

where $\xi_{n|k} \in \Xi \subseteq \mathbb{R}^{n_\xi}$ is the associated control variable acting as a source of (bounded) noise. The function (8) includes an explicit dependence on $\mathbf{x}_{n|k}$ and $\mathbf{u}_{n|k}$ in order to model possible interactions between the uncertainty and the system (1). This allows one to, e.g., model the fact that a pedestrian decision might depend on how the vehicle moves. Whether such models can be derived and whether that is desirable or not is beyond the scope of this article.

Example 1: In the case where other road users are broadcasting their exact trajectory, function (8) simply becomes the system dynamics of the other road user, i.e., $\mathbf{w}_{n+1|k} = \omega(\mathbf{w}_{n|k}, \xi_{n|k})$.

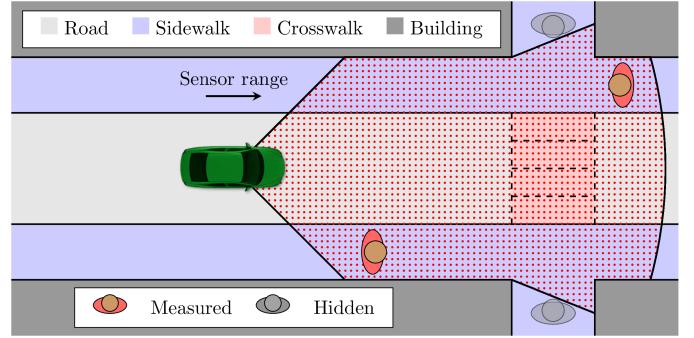


Fig. 2. Situation where the sensor range is not able to see pedestrians behind the corner of the building.

Furthermore, since (8) describes how the uncertainty $\mathbf{w}_{n|k}$ evolves in time, it becomes natural to rely on reachability analysis [32] to compute outer approximations of the uncertainty

$$\mathcal{W}_{n+1|k}(\mathbf{x}_{n|k}, \mathbf{u}_{n|k}) : \supseteq \{ \omega(\mathbf{w}_{n|k}, \xi_n, \mathbf{x}_{n|k}, \mathbf{u}_{n|k}) \mid \mathbf{w}_{n|k} \in \mathcal{W}_{n|k} \quad \forall \xi_n \in \Xi \} \quad (9)$$

for some initial $\mathcal{W}_{k|k} = \mathbf{w}_{k|k}$.

To further clarify the nature of the uncertainty sets (9), we provide the following example.

Example 2: In a noncooperative, nonconnected, and multiagent setting, the behavior of the other agents is not fully known. In this case, we must distinguish between two types of agents: 1) those that are detectable by the sensors and 2) those that are either beyond the sensor range or occluded by other obstacles (see Fig. 2). For type 1), we require that (8) does not underestimate the set of future states that can be reached by the other agents. For type 2), the uncertainty model must instead account for the possibility that agents could suddenly appear at the sensor range boundary or from behind an obstacle.

The following lemma proves that Assumption 4 is satisfied when the uncertainty sets are given by (9).

Lemma 1: Suppose that $g_{n|k}$ is defined according to (7) with $\mathcal{W}_{n|k}$ satisfying (9). Then, Assumption 4 holds.

Proof: The proof is given in [29, Lemma 1]. \square

Lemma 1 simply states that if the prediction model (8) is propagated using (9), then the uncertainty cannot increase as additional information becomes available (from either the onboard sensors or through communication links).

In order to simplify the following analysis, we will only consider pedestrian road users in the remainder of this article. Therefore, we introduce next a pedestrian prediction model that satisfies Assumption 4.

1) Pedestrian Prediction Model: In order to predict the future pedestrian motion, we propose a model that leverages the information from the road configuration, e.g., we want to use the information of available sidewalks and crosswalks since pedestrians are expected to use them when in traffic. Such information can be naturally represented as a graph of connected edges, where each edge i is associated with a reference path $\mathbf{r}_i^{\text{ped}}$. An example illustrating this is shown in Fig. 3.

Having access to such information makes it possible to decompose the pedestrian motion along each path $\mathbf{r}_i^{\text{ped}}$. For

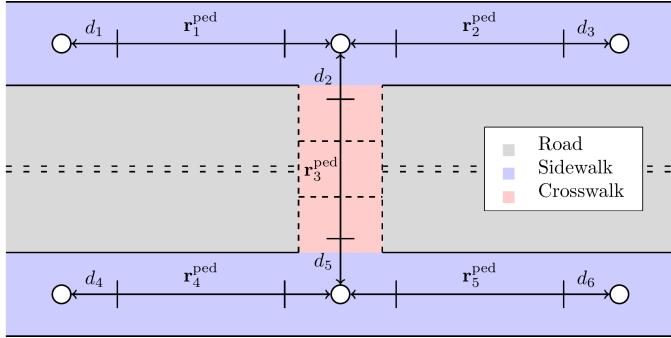


Fig. 3. Illustration of different road segments of an intersection can be represented as a connected graph.

In this specific setting, we assume that the pedestrian states are given by $\mathbf{w}_k := [w_k^{\text{lon}}, w_k^{\text{lat}}] \in \mathbb{R}^2$ and model the longitudinal and lateral position along path $\mathbf{r}_i^{\text{ped}}$, while ξ_k represents the walking speed in both the longitudinal and lateral directions. To contain the uncertainty of the future motion, we place the assumption that the lateral motion will be stabilized, i.e., we place the somewhat strict assumption that pedestrians are rational and abide by the road code. We, therefore, select the feedback law that guides the prediction to be

$$\xi_k = [v_i^{\text{ped}}, -K w_k^{\text{lat}}] \quad (10)$$

where v_i^{ped} denotes the reference walking speed along the path $\mathbf{r}_i^{\text{ped}}$ and $K > 0$ is a feedback gain. With this state representation, we can write the closed-loop pedestrian model as

$$\begin{aligned} \mathbf{w}_{k+1} &= \omega_i(\mathbf{w}_k, \xi_k) \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 - t_s K \end{bmatrix} \mathbf{w}_k + \begin{bmatrix} t_s \\ 0 \end{bmatrix} v_i^{\text{ped}} + \begin{bmatrix} t_s & 0 \\ 0 & t_s \end{bmatrix} \xi_k \end{aligned} \quad (11)$$

where $\xi \in \Xi \subseteq \mathbb{R}^2$, i.e., $\|\xi\|_\infty \leq \bar{\xi}$, is assumed to be some bounded noise with zero mean, representing some uncertainty in the future motion. The pedestrian motion can then be predicted using

$$\mathbf{w}_{n+1|k}^{\text{nom}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 - t_s K \end{bmatrix} \mathbf{w}_{n|k}^{\text{nom}} + \begin{bmatrix} t_s \\ 0 \end{bmatrix} r_v^{\text{ped},i} \quad (12)$$

for an initial $\mathbf{w}_{k|k}^{\text{nom}} = \mathbf{w}_k$. Finally, by using reachability analysis [32], [33], we can also predict the uncertainty sets $\mathcal{W}_{n|k}$ along an edge i similar to (9). As the longitudinal and lateral dynamics are decoupled for our proposed model, we can easily propagate the uncertainty in each direction, i.e., one can construct the uncertainty set at the next time step as

$$\begin{aligned} \mathcal{W}_{n+1|k} &:= \left\{ \mathbf{w} \mid \mathbf{w} \leq \begin{bmatrix} w_{n|k}^{\text{lon}} + t_s(v_i^{\text{ped}} + \bar{\xi}) \\ (1 - t_s K)w_{n|k}^{\text{lat}} + t_s \bar{\xi} \end{bmatrix}, \right. \\ &\quad \mathbf{w} \geq \begin{bmatrix} w_{n|k}^{\text{lon}} + t_s(v_i^{\text{ped}} - \bar{\xi}) \\ (1 - t_s K)w_{n|k}^{\text{lat}} - t_s \bar{\xi} \end{bmatrix}, \\ &\quad \forall \mathbf{w}_{n|k} := \left[w_{n|k}^{\text{lon}}, w_{n|k}^{\text{lat}} \right]^\top \in \mathcal{W}_{n|k} \left. \right\} \quad (13) \end{aligned}$$

with $\mathcal{W}_{k|k} = \mathbf{w}_k$. More complex models can be straightforwardly accommodated by applying standard computational geometry tools [23], [34], [35].

Note that (11) describes the predicted pedestrian motion along edge i only. Since the road map in Fig. 3 consists of a set of connected edges, the prediction will need to transition between the different edges eventually. To that end, similar to [36], we transition the dynamics from edge i to the neighboring connected edges, whenever the nominal prediction $\mathbf{w}_{n|k}^{\text{nom}}$ is within a transitioning threshold, e.g., whenever the nominal prediction is at a certain distance from the final node. Note that, if the final node is connected to multiple edges, then the future predictions are propagated along all these edges. This enables the model to account for all possible directions that a pedestrian might take, making it possible to consider multimodal predictions.

In reality, pedestrian walking behaviors may differ depending on numerous factors, e.g., personal preferences, the environment, and the presence of other pedestrians, which makes the future movement difficult to model in detail. While we proposed this pedestrian model for simplicity, we must stress that other frameworks/models can also be used as long as Assumption 4 is satisfied. In particular, the recently proposed methods by Koschi et al. [37] and Koschi and Althoff [38], where they use map-based information and incorporate traffic rules in their predictions could also be applied.

Next, we show how the a priori unknown constraint can finally be expressed using the derived prediction model (11).

2) *Constraint Generation:* For each road user i that is present in the environment, we need to identify if the vehicle potentially needs to interact with it, e.g., if the predicted path of the road user eventually crosses the planned path of the vehicle. Hence, having the uncertainty set $\mathcal{W}_{n|k}^i$ for each road user i , we want to avoid the positions where the uncertainty enters within a distance $\Delta := \bar{e}_y + \Delta_{\text{safe}}$ of the reference trajectory, where Δ_{safe} accounts for the dimensions of the ego vehicle, as well as the obstacle dimensions. More formally, the longitudinal positions that should be avoided can be expressed through the following set:

$$\begin{aligned} \mathcal{T}_r(\mathcal{W}_{n|k}) &:= \{s \mid \exists \mathbf{w}_{n|k} \in \mathcal{W}_{n|k} \text{ s.t.} \\ &\quad \|T^w(\mathbf{w}_{n|k}) - T^x(\mathbf{r}^x(s))\|_2 \leq \Delta\} \end{aligned} \quad (14)$$

where $T^x : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^2$ and $T^w : \mathbb{R}^{n_w} \rightarrow \mathbb{R}^2$ map the position components of the reference and the uncertainty set into a common global frame. A visual example of set (14) is shown in Fig. 4. Using (14), we can then express the collision avoidance set for each road user i as

$$C_{n|k}^i(\mathcal{W}_{n|k}^i) := \begin{cases} \left[\sigma_{n|k}^{\text{L},i}, \bar{\sigma}_{n|k}^{\text{U},i} \right], & \text{if } \mathcal{T}_r(\mathcal{W}_{n|k}^i) \neq \emptyset \\ \emptyset, & \text{otherwise} \end{cases} \quad (15)$$

where we define $\sigma_{n|k}^{\text{L},i}$ as

$$\sigma_{n|k}^{\text{L},i} := \min_s s, \quad \text{s.t. } s \in \mathcal{T}_r(\mathcal{W}_{n|k}^i)$$

and $\sigma_{n|k}^{\text{U},i}$ as the solution of the same problem with the minimization replaced by a maximization. To further clarify

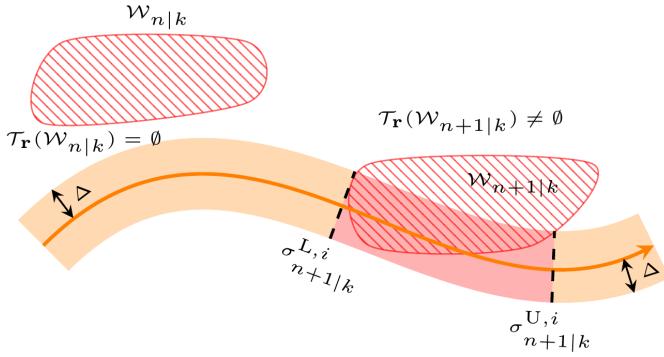


Fig. 4. Visual representation of (14) and (15). The orange line represents the vehicle's reference path $r^x(s)$, while the red regions represent the uncertainty sets $\mathcal{W}_{n|k}$ and $\mathcal{W}_{n+1|k}$.

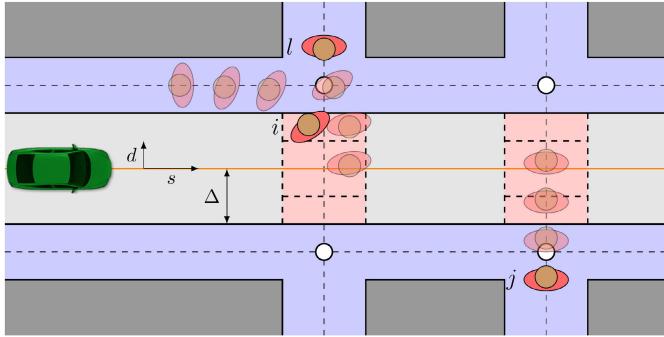


Fig. 5. Orange line represents the vehicle's reference path, while the black dashed lines represent the walkable road graph of the pedestrian model.

the meaning of the collision avoidance set $\mathcal{C}_{n|k}^i$, we provide the following example.

Example 3: Consider the setting presented in Fig. 5, where the future motions of three pedestrians are predicted. In this situation, pedestrian i is only within the driving corridor Δ at the initial time, and hence, $\mathcal{C}_{k|k}^i \neq \emptyset$ and $\mathcal{C}_{k'|k}^i = \emptyset$ for all $k' > k$, while the other two pedestrians i and j are instead predicted to enter the driving corridor at a later time. In this particular setting, the vehicle has three options to consider: 1) stop for all pedestrians; 2) yield to pedestrian i and j ; or 3) yield only to pedestrian i . Fig. 6 shows an illustration of the collision avoidance sets and shows the three possible decisions that the vehicle can take while avoiding collisions.

To formulate constraint $g_{n|k}$, we realize that the vehicle must decide between either driving ahead of the road user or if it should yield to it. To that end, for each road user i , we must avoid the positions contained within the collision avoidance set $\mathcal{C}_{n|k}^i$ and therefore formulate the following candidate collision avoidance function:

$$\gamma_{n|k}^i(s_{n|k}, \mathbf{u}_{n|k}, \mathcal{W}_{n|k}^i) := \begin{cases} s_{n|k} - \sigma_{n|k}^{U,i}, & \text{if yielding and } \mathcal{C}_{n|k}^i(\mathcal{W}_{n|k}^i) \neq \emptyset \\ \sigma_{n|k}^{L,i} - s_{n|k}, & \text{if not yielding and } \mathcal{C}_{n|k}^i(\mathcal{W}_{n|k}^i) \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

which finally allows us to write the resulting a priori unknown constraint as

$$g_{n|k}^i(s_{n|k}, \mathbf{u}_{n|k}) := \gamma_{n|k}^i(s_{n|k}, \mathbf{u}_{n|k}, \mathcal{W}_{n|k}^i). \quad (17)$$

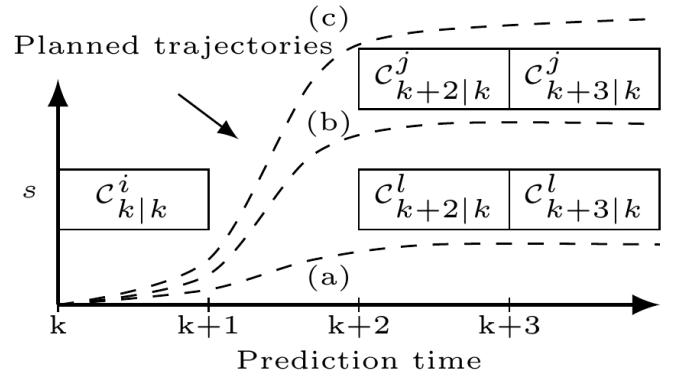


Fig. 6. Collision avoidance sets $\mathcal{C}_{n|k}^z$, $z \in \{i, j, l\}$ for the three pedestrians from Example 3. The boxes represent the longitudinal positions that the vehicle must avoid, while the dashed lines represent three different possible planned trajectories that the vehicle can make in this setting.

Remark 1: While the introduction of additional pedestrians could be challenging, one could use some strategy to reduce the complexity while still capturing the most important features of the problem. Note, however, that only a limited amount of pass/yield configurations can effectively exist in the prediction horizon, e.g., if too many pedestrians try to cross the road, the only feasible option might be to stop and wait for them to pass.

As a safe though possibly conservative approach to address the combinatorial complexity issue that Remark 1 mentions, we deploy a simple heuristic in this article that only considers the combinations of the closest intersection and always yield to road users further away. In particular, for the closest intersection, we sort the times when the road users enter and solve Problem (4) for the different combinations. Considering the setting presented in Example 3 and Figs. 5 and 6, this amounts to considering the options: 1) yield to both road users i and l and 2) yield to road users i and j .

Finally, as it was mentioned in Example 2, in order to ensure the validity of Assumption 4, the vehicle must also be ready to act for any road users that cannot be measured directly from its sensors, i.e., it must account for the possibility that other road users may be occluded similar to Fig. 2. To address this issue, we deploy a similar strategy as in [38], where virtual road users are placed wherever the sensor suite warns that a potential occlusion may occur. Then, for each virtual road user, we predict the future uncertainty sets $\mathcal{W}_{n|k}^{\text{virtual}}$ and form constraints (17).

While Sections III-A and III-B showed how the known and unknown constraints could be constructed for the vehicle model (1), the final step needed for Proposition 1 is to show how the terminal conditions can be derived.

C. Safe Terminal Conditions

This section addresses the derivation of the stabilizing terminal set needed for Assumption 3 and provides an example of a safe set that is suitable for urban autonomous driving and satisfies Assumption 5.

1) *Stabilizing Terminal Conditions:* In order to derive the terminal cost matrix P and the stabilizing terminal set \mathcal{X}_r^s

for Problem (4), we propose to first decouple the longitudinal and lateral motions of the nonlinear system (6). While the full derivation is provided in the Appendix, here we will only provide an overview of the steps for convenience. Hence, the reader is referred to the Appendix for the full details of the derivation.

Since the longitudinal kinematics of (6) are linear, i.e., the states $\mathbf{x}_{\text{lon}} := [v, a]^\top$, we use standard approaches in the literature to compute the stabilizing terminal conditions [39]. In particular, we use an linear quadratic regulator (LQR) controller with tuning $Q_{\text{lon}}^{\text{LQR}} \in \mathbb{R}^{2 \times 2}$ and $R_{\text{lon}}^{\text{LQR}} \in \mathbb{R}$ to obtain a feedback gain K_{lon} that stabilizes the system and then solve the linear matrix inequality (LMI) (30) to obtain the longitudinal terminal cost matrix P^{lon} . To get the corresponding longitudinal terminal set $\mathcal{X}_{\mathbf{r}}^{\text{lon}}$, we start from the admissible constraint set and iteratively compute backward reachable sets that are intersected with the admissible constraint set [34]. Repeating this process until convergence yields the following longitudinal terminal invariant set:

$$\mathcal{X}_{\mathbf{r}}^{\text{lon}}(s) := \left\{ v, a \mid H_{\text{lon}} \begin{bmatrix} v - v^{\mathbf{r}}(s) \\ a - a^{\mathbf{r}}(s) \end{bmatrix} \leq b_{\text{lon}} \right\}. \quad (18)$$

The lateral kinematics of (6), with the corresponding states $\mathbf{x}_{\text{lat}} := [e_y, e_\psi, \delta, \alpha]^\top$, can be viewed as a linear parameter-varying (LPV) system whose dynamics depend on the velocity. As such, we construct a polytopic approximation of the nonlinear system for a range of velocities v [40]. Then, similar to the longitudinal kinematics, we solve LMI (47) to obtain the terminal cost matrix and compute the lateral terminal set

$$\mathcal{X}_{\mathbf{r}}^{\text{lat}}(s) = \{e_y, e_\psi, \delta, \alpha \mid H_{\text{lat}}[e_y, e_\psi, \delta - \delta^{\mathbf{r}}, \alpha - \alpha^{\mathbf{r}}]^\top \leq b_{\text{lat}}\}.$$

The terminal cost matrix P and the stabilizing terminal set $\mathcal{X}_{\mathbf{r}}^s$ are then finally expressed as

$$\begin{aligned} P &= \text{blockdiag}(P_{\text{lat}}, P_{\text{lon}}) \\ \mathcal{X}_{\mathbf{r}}^s(s) &:= \left\{ \mathbf{x} \mid [v, a]^\top \in \mathcal{X}_{\mathbf{r}}^{\text{lon}}(s), [e_y, e_\psi, \delta, \alpha]^\top \in \mathcal{X}_{\mathbf{r}}^{\text{lat}}(s) \right\}. \end{aligned} \quad (19)$$

We remind the reader that the full derivation of the terminal conditions is presented in the Appendix.

2) *Safe Set Formulation*: For many practical applications where safety is emphasized, a system is generally considered to be safe at steady state, i.e.,

$$\begin{aligned} \mathcal{X}_{\text{safe}}(s_k) &:= \{ \mathbf{x} \mid \mathbf{x} = f(\mathbf{x}, \mathbf{u}), \\ h_k(\mathbf{x}, \mathbf{u}) &\leq 0, m_k(\mathbf{x}, \mathbf{u}) \leq 0 \} \end{aligned} \quad (20)$$

where the function m_k may define additional constraints needed for the set definition. This general definition can readily be applied to autonomous driving settings, since a vehicle is generally considered safe if it has come to a full stop, i.e., a vehicle that is parked in a parking lot, or other safe configurations modeled by m_k are considered safe and not responsible for collisions with other road users. While the safe set in (20) serves as an example in this article, we must stress that other formulations of the safe set are possible. For instance, one could replace (20) with a time-to-collision constraint set, provided that one is able to guarantee that this does yield the required safety guarantees.

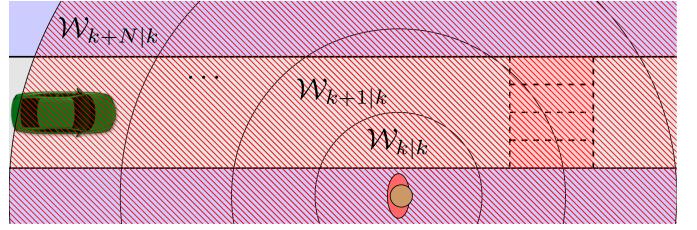


Fig. 7. As the predicted uncertainty $\mathcal{W}_{n|k}$ grows in time, the collision-free region of the vehicle drastically shrinks.

In general, most processes controlled either by humans or by automatic controllers do have emergency procedures that are triggered whenever safety is jeopardized. Assumption 5, i.e., the existence of a safe set, is meant to cover all these situations. Furthermore, we must stress that this assumption also entails that $g_{n|k}$ directly depends on the safe set since it is assumed that $g_{n|k}$ may never be active when $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(s_n)$. Hence, this assumption entails that when using the uncertainty model presented in Section III-B, function γ used in (16) is not simply the output of the uncertainty model but also includes the information that $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(s_{n|k})$ implies $\gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{\text{safe}}, \mathbf{w}_{n|k}) = 0$.

To further motivate the choice of a safe set at steady state, consider the setting presented in Fig. 7, where a prediction of the future pedestrian motion (red shaded region) is growing unbounded in time. For such a setting, one can argue that the safest option that the vehicle can take is to come a complete stop since the uncertainty allows the future pedestrian position to be anywhere on the road. However, this choice does not entail that the pedestrian will never hit the vehicle since the model still predicts that this can happen. The second part of Assumption 5 is meant to cover this case by imposing that $\mathbf{x}_{n|k} \in \mathcal{X}_{\text{safe}}(s_{n|k})$ implies $\gamma_{n|k}(\mathbf{x}_{n|k}, \mathbf{u}_{\text{safe}}, \mathbf{w}_{n|k}) = 0$, which entails that the model predicts that the pedestrian will avoid hitting the vehicle if it is stopped. This observation makes the assumption more acceptable, as it yields a more refined pedestrian model that accounts for the fact that pedestrians do not walk over stopped vehicles. In addition, this is also abiding by the standards of the road code, for which a parked vehicle is not responsible for collisions that might occur. Clearly, more complex situations (in terms of responsibilities in case of collisions) arise when the vehicle needs to stop in a crossroad in order to yield to another vehicle having priority or at a zebra crossing to yield to a pedestrian. Covering all possible situations in the best way is beyond the scope of this article and the subject of future research.

In this article, we express the safe set as all states where the vehicle has come to a full stop. We motivate this choice by noting that human drivers, in general, already plan their driving such that they can safely come to a complete stop in case of an emergency situation, i.e., reaching a zero velocity is considered to be safe. While what we propose here might be oversimplified for real-world autonomous driving applications, future work will carefully evaluate whether more refined strategies are necessary. Note that other formulations of the safe set might be better suited for settings outside the scope of urban driving, e.g., for highway driving, one could instead rely on a safe set that considers a specific relative distance

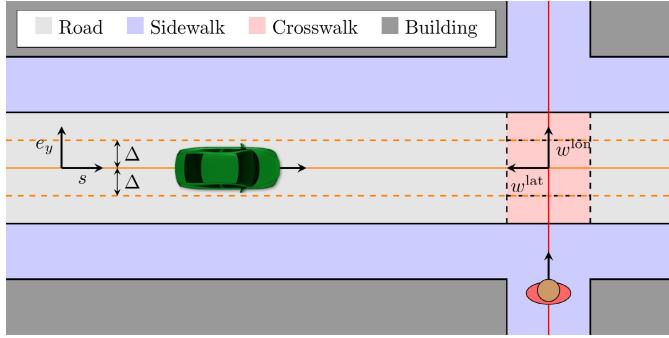


Fig. 8. Straight road simulation setting with a crossing pedestrian that is occluded by the environment.

TABLE I
PARAMETER VALUES FOR SECTIONS IV AND V

w_0	20	[s^{-1}]	w_1	0.9	[-]
t_{acc}	1.8	[s^{-1}]	\bar{e}_y	0.4	[m]
$\bar{\psi}$	0.61	[rad]	δ	0.53	[rad]
\bar{v}	15.28	[$m \cdot s^{-1}$]	a	-5	[$m \cdot s^{-2}$]
\bar{a}	2	[$m \cdot s^{-2}$]	$\bar{\alpha}$	0.35	[rad $\cdot s^{-1}$]

toward the lead vehicle while keeping a zero difference in velocity.

We now have all the necessary results needed to formulate (4) and implement the theory from Proposition 1. In Section IV, we deploy the controller based on Problem (4) and show the importance of Assumption 4 and how recursive feasibility (safety) can be jeopardized if occluded obstacles are not considered. Then, in Section V, we deploy the framework in a real vehicle at a test track to verify the real-time performance of the controller.

IV. SIMULATION RESULTS

In this section, we consider the straight road setting presented in Fig. 8 in order to illustrate how the controller based on Problem (4) behaves in a situation when a pedestrian crosses the road. While this setting is indeed very simple, we will show that an occluded pedestrian that might appear from the bottom in Fig. 8 may render the controller unsafe if the assumptions needed for Proposition 1 do not hold. Therefore, to illustrate the implications that Proposition 1 has on safety, we will show two simulations: one where Proposition 1 does not hold (Assumption 4 is not satisfied) and one where Proposition 1 does hold.

For both simulations, we consider the parameter values presented in Table I and we formulate the MPC controller (4) in the multiple shooting framework using a sampling time of $t_s = 0.05$ s and prediction horizons $N = 20$ and $M = 100$. We used the following stage cost matrices:

$$Q = \text{diag}(1, 1, 10, 1, 1, 1), \quad R = \text{diag}(4, 10) \quad (21)$$

and the terminal matrix P from (49). The discretized dynamics of (6) were obtained using a fourth-order Runge–Kutta integrator with five steps per control interval. The OCP was then solved using Acados [41] together with the HPIPM solver [42] by deploying a sequential quadratic programming (SQP) approach with a real-time iteration (RTI) scheme [8], [43].

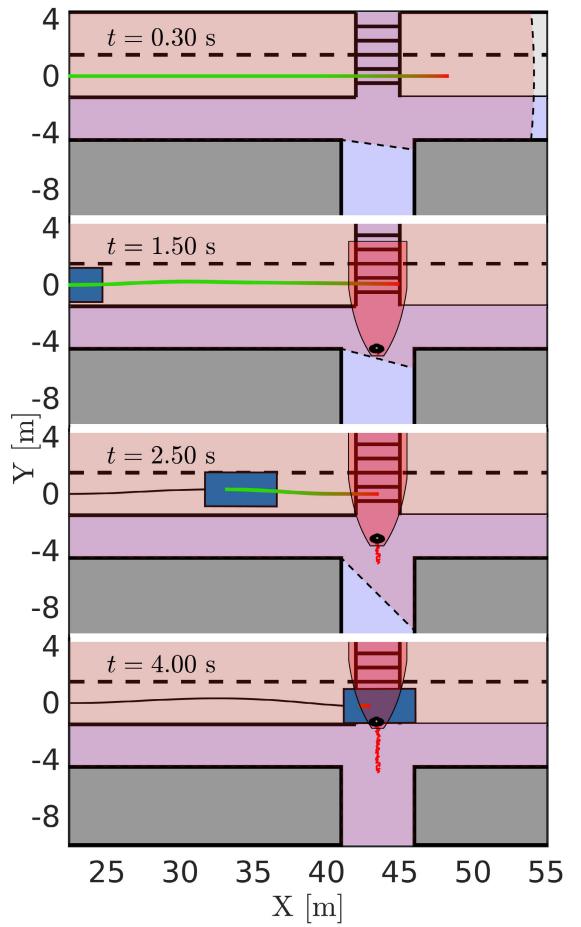


Fig. 9. Four different time instances of the simulation environment. The first panel shows that the sensors (pink shaded region) cannot see behind a wall and that the vehicle, therefore, plans a trajectory (red/green line indicating low/high speed) within the sensing range. The last three panels show that a pedestrian, who was not visible to the sensors earlier, appears and forces the vehicle to perform emergency braking, which results in a collision.

A. Compromised Safety

Fig. 9 shows four time instances of the MPC controller that only reacts to what it can sense directly, i.e., it does not expect that an occluded road user may appear at the intersection. To that end, it is visible from the first time instance in Fig. 9 that the vehicle assumes that it can pass the intersection without causing any collision. Around time $t = 1.5$ s, the occluded pedestrian is eventually detected by the sensor system, and since the vehicle cannot clear the intersection safely, it applies full braking. However, due to the late detection of the road user, the vehicle does not manage to come to a full stop before colliding with the road user. Fig. 10 shows the closed-loop states of the controller, which shows that around $t = 1.5$ s, the vehicle applies full braking and tries to steer to marginally increase the traveled distance before colliding with the pedestrian. While this behavior may not be desirable in practice, it can be addressed by modifying the vehicle model and/or system constraints, e.g., through the inclusion of tire force constraints. Note that even at time $t = 4$ s (some time after the initial collision), the velocity is still around 3 m/s (≈ 10 km/h).

What caused the controller to become unsafe in this simulation setting is the fact that Assumption 4 was not satisfied.

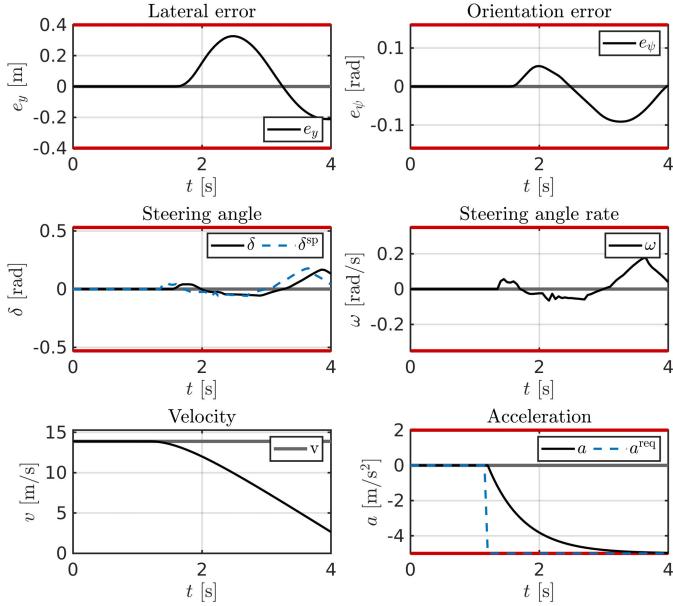


Fig. 10. Closed-loop evolution of the unsafe MPC controller shown in Fig. 9. Just before $t \leq 2$ s, the pedestrian becomes visible and forces the vehicle to perform an emergency braking. The red lines denote the state and input bounds, while the gray line denotes the reference trajectory.

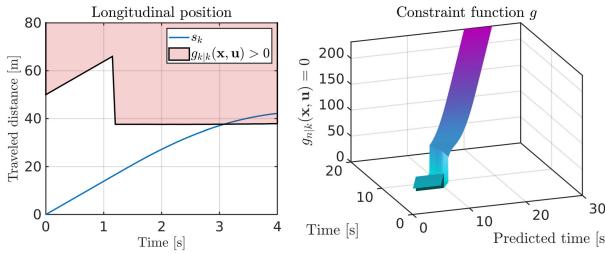


Fig. 11. Time evolution of $g_{n|k}$ for the unsafe MPC controller in Section IV-A. Left: constraint $g_{k|k}$. Right: predicted $g_{n|k}$.

Fig. 11 shows how $g_{k|k}$ and $g_{n|k}$ evolves in time. From the left plot, it is visible that the upper bound on the longitudinal position (red region) drastically shrinks around $t = 1.5$ s. This is also visible in the right plot where the predicted constraint does not satisfy $g_{n|k+1} \leq g_{n|k}$. The feasibility of the problem was preserved by relaxing constraint $g_{n|k}$ with an exact penalty [44].

Next, by considering the same example, we show that safety does not become jeopardized for the same traffic situation if Proposition 1 holds.

B. Safe Crossing

Fig. 12 shows four time instances of the same MPC controller, and however, now, occluded pedestrians are anticipated to appear from behind the corner. This is visible from the first two time instances in Fig. 12, as a “virtual” pedestrian is predicted to appear from behind the corner and move toward the intersection. This forces the vehicle to approach the intersection by reducing its speed, as it can be seen in Fig. 13. While the vehicle approaches the intersection at a reduced speed (compared to the unsafe controller), the occlusion reduces and the real pedestrian is eventually detected. After $t > 6$ s, the vehicle is free to pass as it can see that there are no more occlusions, and the real pedestrian has

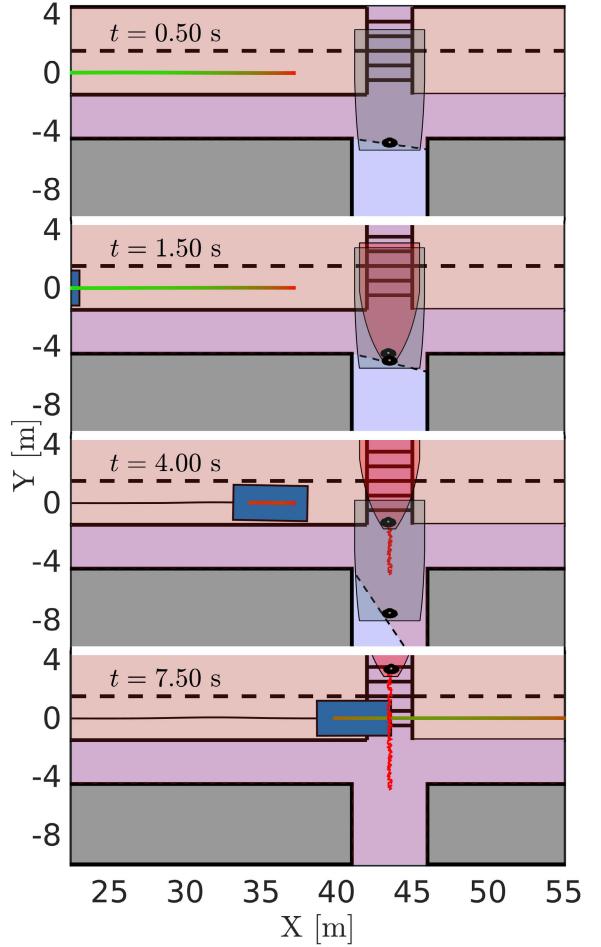


Fig. 12. Four different time instances of the simulation environment. The two top panels show that the sensors (shaded region) cannot see behind a wall; however, the vehicle plans a trajectory as if there were a pedestrian behind the corner. The two last panels show that a pedestrian, who previously was not visible, shows up. However, since the vehicle already anticipated that a pedestrian might appear, it manages safely adjust its speed and yield to the pedestrian.

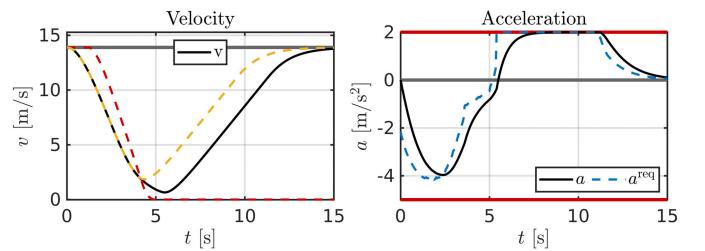


Fig. 13. Closed-loop evolution of the safe MPC controller shown in Fig. 12. The vehicle approaches the intersection by reducing its velocity. After $t = 6$ s, the pedestrian passes, and the vehicle is free to accelerate again. The yellow dashed line in the left plot shows the closed-loop velocity of the same controller in the case that no pedestrian appears at the intersection, while the red dashed line shows a trajectory comparison of the unsafe controller in Section IV-A.

already crossed the road. The yellow dashed line shows the closed-loop velocity of the same MPC controller in the case that no pedestrian appears. Due to the uncertainty, the safe MPC controller, much like a human driver, slows down and is prepared to come to a stop if needed. However, as the vehicle approaches the intersection and the sensor view opens

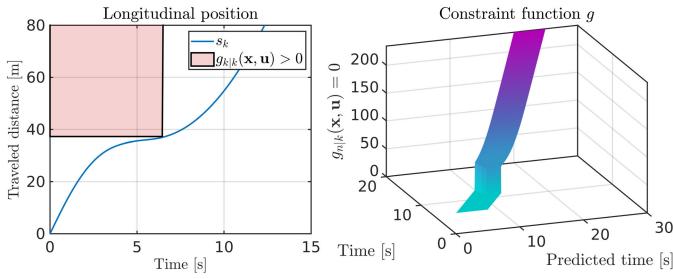


Fig. 14. Constraint evolution for the safe MPC controller in Fig. 12. Left: constraint $g_{k|k}$. Right: time evolution of the predicted constraint $g_{n|k}$.

up, it becomes clear that it can cross the intersection safely without stopping unnecessarily.

In Fig. 14, we can see that the a priori unknown constraint $g_{n|k}$ satisfies Assumption 4 compared to the unsafe controller presented in Section IV-A. Here, it is visible that the constraint does not suddenly shrink and that $g_{n|k+1} \leq g_{n|k}$ for all times.

This section served as a simple example of the importance that Proposition 1 has on safety. In Section V, we deploy the same controller in a real vehicle platform.

V. EXPERIMENTAL RESULTS

We deployed the safe MPC controller on a full-scale Volvo XC90 T6 petrol-turbo SUV at a closed test track in order to verify the performance in practice. The vehicle offers an actuation interface that accepts longitudinal acceleration requests and steering wheel angle setpoint requests. To access the vehicle CAN bus and sensor system, we used the open-source software OpenDLV¹ and the middleware library Cluon,² which enabled reading the sensor data and the sending of actuation requests through the vehicle CAN bus.

A. Platform Limitations

In order to localize the vehicle, we used the onboard real-time kinematic (RTK) GPS unit and the built-in vehicle inertial measurement unit (IMU) sensors to form an estimate of the initial state \mathbf{x}_k . However, at the time of testing, no real-time corrections were available, which resulted in a lowered accuracy in the positioning. Furthermore, the interface between OpenDLV and the vehicle showed a time delay of 150 ms when sending and reading signals related to the steering actuator. Since the steering actuator dynamics were fast and already accounted for in (6), we used dead reckoning to estimate both the steering angle δ and the steering angle rate α . Furthermore, to account for the input delay, we augmented the state-space vector in (6) with additional time-delayed states for the steering angle, similar to [5].

The test vehicle also included a safety feature that limited the steering wheel actuator for different velocities. To capture this limitation, an additional velocity-dependent constraint of the following form was added:

$$|\delta_k| \leq (10^4(1 + e^{0.5v})^{-1} + 40) \frac{16.8\pi}{180} \text{ rad.} \quad (22)$$

¹See <https://opendlv.org/> for more information.

²See <https://github.com/chrberger/libcluon> for more information.



Fig. 15. View of the test vehicle that is being controlled by solving Problem (4) on a laptop computer and sending actuation requests over a network connection through OpenDLV.

The final limitation at the time of testing was that the onboard cameras did not have access to a computer vision software stack, i.e., the detection of the environment, the drivable road, and other road users were not possible. We, therefore, relied only on GPS measurements to localize the vehicle with respect to the reference path and simulated pedestrians that were moving in the environment. While these limitations reduced the experimental scope, we must stress that this setup made it possible to conduct all experiments in a safe and controlled manner. Furthermore, we ought to stress that the main scope of this article was to deploy the MPC controller (4) in a real vehicle platform to test the real-time controller performance, while testing the perception layer of the vehicle is out of the scope of this article.

B. Controller Details

Similar to Section IV, we consider the parameter values from Table I when formulating (4) in the multiple shooting framework with a sampling time of $t_s = 0.05$ s and use a fourth-order Runge–Kutta integrator with five steps per control interval. We used the stage cost (21) and the terminal cost (49), together with prediction horizons $N = 65$ and $M = 100$, and solved the OCP (4) using Acados [41] and HPIPM [42] with an SQP-RTI scheme. Note that the only difference between the experimental setup and the simulation setup is the different value on N .

To address the mixed-integer problem introduced by constraint $g_{n|k}$, i.e., deciding for which road user to yield and for which to not yield, we formulate (4) for each constraint configuration. In the experiments, the number of combinations for crossing pedestrians was at most three. To that end, we formulated three OCPs in parallel with each constraint configuration. Then, the control action corresponding to the OCP with the best cost value was applied to the vehicle.

To deploy the controller to the test vehicle, a C++ interface was written to connect Acados and HPIPM with the OpenDLV framework. A Linux laptop (i9 2.4 GHz, 32-GB RAM) was then used to solve the resulting OCPs in parallel and send the control inputs to the vehicle using OpenDLV (see Fig. 15).

C. Results

In order to evaluate the proposed safe MPC framework, we deploy the controller in a four-way intersection with

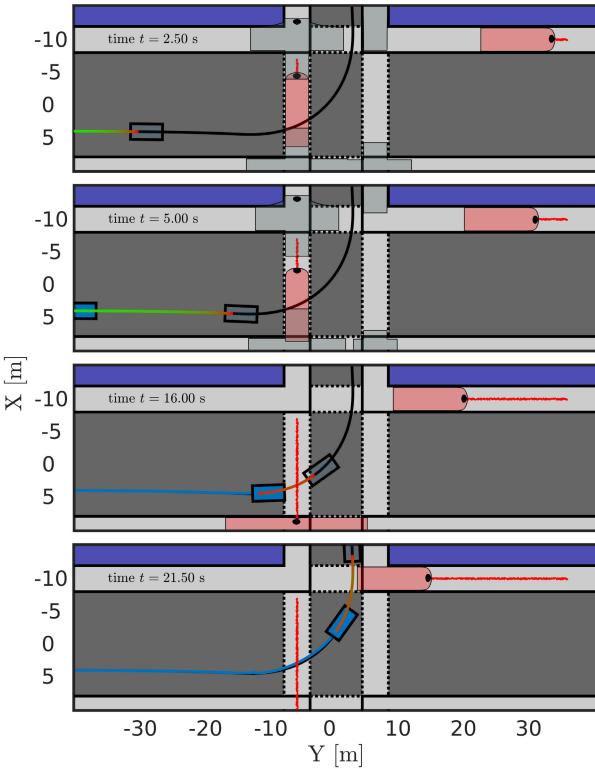


Fig. 16. Vehicle bounding box is illustrated by the blue box, while the predicted pedestrian motion is illustrated by the red regions for the real pedestrians and gray regions for the virtual pedestrians. The opaque box represents the terminal state, and the line connecting the vehicle bounding box and the opaque box illustrates the predicted open-loop solution. The blue and red lines denote the traveled history of the vehicle and pedestrians, respectively.

(simulated) moving pedestrians. In this setting, the vehicle needs to perform a left turn while safely avoiding any collision with moving pedestrians at two different crosswalks. Indeed, while more general settings would include other road users, we consider only pedestrians in this experiment for simplicity and to be able to clearly illustrate the results.

Fig. 16 shows the open-loop solutions across four different time instances. The blue line shows the past positions of the ego vehicle, while the red lines show the past positions of the pedestrians. The red regions denote the future predicted pedestrian uncertainty $\mathcal{W}_{n|k}$ of the measured pedestrians, while the gray regions show the corresponding uncertainty for the “virtual” pedestrians. The blue bounding box denotes the ego vehicle at the current time instance, while the opaque box shows the ego vehicle position at time $k + M$, i.e., $\mathbf{x}_{k+M|k} \in \mathcal{X}_{\text{safe}}$. The red and green lines connect the blue box and the opaque box denotes the open-loop solution in the xy plane, where the red and green indicate a low and high velocity, respectively.

The vehicle approaches the first intersection cautiously and yields to the first pedestrian. Then, once the pedestrian has crossed the intersection, the vehicle accelerates up to the next intersection, which also has a pedestrian that moves toward it. In this case, the vehicle approaches the next intersection cautiously and decides to cross only if it can clear the intersection before the pedestrian is predicted to enter. Hence,

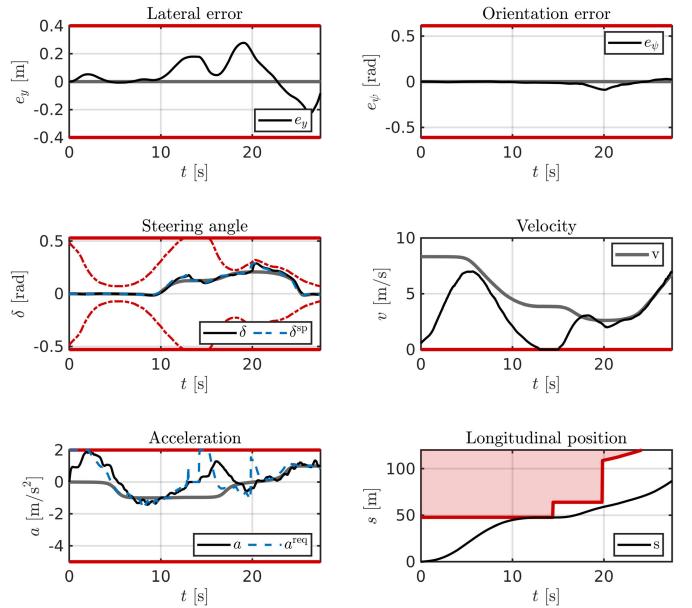


Fig. 17. Closed-loop states from Fig. 16. The black lines represent the corresponding state and input reference, while the blue and orange lines show the state and control inputs.

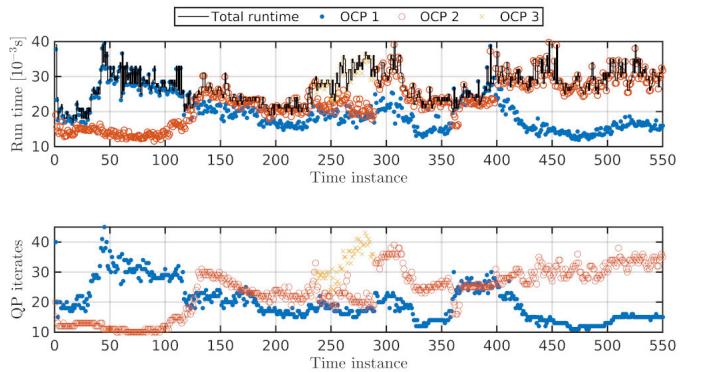


Fig. 18. Black line in the top plot shows the total runtime of the framework for each time instance. The markers on the other hand show the solution time for different configurations of (4). The bottom plot shows the number of QP iterations that the solver took for each OCP.

as the vehicle comes close to the next intersection, it realizes that it can clear the intersection safely and therefore accelerates past it in the final frame at $t = 21.5$ s.

Fig. 17 shows the corresponding closed-loop trajectories for Fig. 16. Here, it is visible that the vehicle starts far from the reference trajectory, which it begins to track. Around time $t = 6$ s, the vehicle is forced to start deviating from the reference trajectory in order to yield for the first pedestrian and, therefore, comes to a complete stop. After $t = 15$ s, when the first pedestrian has crossed, the vehicle is allowed to accelerate again and approaches the next intersection, which can be cleared without needing to wait for the other pedestrian to cross. After $t = 22$ s, it is visible that the velocity is being precisely tracked again.

Fig. 18 shows the solution times of the OCPs and the number of QP iterations it took for the solver for the experiment presented in Figs. 16 and 17. Due to the mixed-integer nature of constraint $g_{n|k}$, the number of OCPs that were solved was at most 3 in this particular setting. It is visible that the solution

time for all OCPs was at most 40 ms, which was well below the sampling time $t_s = 50$ ms.

VI. CONCLUSION

In this article, we showed that a safe real-time capable MPC controller can be designed for urban autonomous driving applications. In particular, we have shown how such an MPC framework can be formulated in practice by using a safe set that ensures robust constraint satisfaction at all times, which guarantees the recursive feasibility (safety) of the controller. Since one part of the safety guarantees relies on the fact that one has to predict the future motion of other road users, future work will include the investigation and verification of practical and realistic models for the environment prediction and also consider other road users such as cyclists and other vehicles. In addition, since the experimental setup in this article only considered simulated road users, future work will deploy the framework with a sensor suite that enables the online measurements of other road users.

APPENDIX

COMPUTATION OF A STABILIZING TERMINAL SET

In this section, we show how to derive the terminal set and terminal cost for system (6) by decoupling the longitudinal and lateral dynamics.

A. Longitudinal Kinematics

Since the longitudinal kinematics of (6) can be written as

$$\begin{bmatrix} \dot{v} \\ \dot{a} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -t_{acc} \end{bmatrix} \begin{bmatrix} v \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ t_{acc} \end{bmatrix} a^{\text{req}} \quad (23)$$

and the reference trajectory is assumed to satisfy the system dynamics, i.e., v^r and a^r can be generated using (23), we can define the following error states and inputs:

$$e_v := v - v^r, \quad e_a := \dot{e}_v = a - a^r, \quad e_{a^{\text{req}}} := a^{\text{req}} - a^{r,\text{req}}$$

to obtain

$$\begin{bmatrix} \dot{e}_v \\ \dot{e}_a \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -t_{acc} \end{bmatrix} \begin{bmatrix} e_v \\ e_a \end{bmatrix} + \begin{bmatrix} 0 \\ t_{acc} \end{bmatrix} e_{a^{\text{req}}}. \quad (24)$$

Knowing the state and reference bounds

$$0 \leq v \leq \bar{v}, \quad \underline{a} \leq a \leq \bar{a}, \quad \underline{a} \leq a^{\text{req}} \leq \bar{a}, \quad (25)$$

$$0 \leq v^r \leq 50/3.6, \quad |a^r| \leq 1, \quad |a^{r,\text{req}}| \leq 1.05 \quad (26)$$

we can derive the following error state and input bounds:

$$e_v \leq 5/3.6, \quad -4 \leq e_a \leq 1, \quad -3.95 \leq e_{a^{\text{req}}} \leq 0.95. \quad (27)$$

Then, by applying a zero-order hold discretization of (23) and designing an LQR controller, with cost matrices $Q_{\text{lon}}^{\text{LQR}} = \text{diag}(5 \times 10^{-3}, 1)$ and $R_{\text{lon}}^{\text{LQR}} = 1$, we obtain a feedback gain $K_{\text{lon}} = [0.0693 \ 0.4151]$ that stabilizes (24).

Finally, when computing the invariant set using the geometric approach [39], we introduce the additional constraints

$$e_v + e_a \leq 1.4, \quad -32 \leq 2e_v + e_a \quad (28)$$

in order to help reduce the complexity. To perform the actual computation, we use the Multi-Parametric Toolbox (MPT) [35]

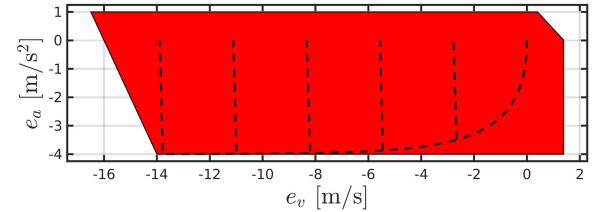


Fig. 19. Longitudinal terminal set X_r^{lon} . The black dashed lines show five trajectories where the vehicle starts to brake and applies a constant braking of $a^{\text{req}} = -4 \text{ m/s}^2$. Each trajectory starts on the reference, i.e., $v = v^r$ and $a = a^r$, where $a^r = 0$ and $v^r \in \{10, 20, 30, 40, 50\} \text{ km/h}$.

and obtain the following invariant set for the longitudinal kinematics:

$$X_r^{\text{lon}}(s) := \left\{ v, a \mid H_{\text{lon}} \begin{bmatrix} v - v^r(s) \\ a - a^r(s) \end{bmatrix} \leq b_{\text{lon}} \right\} \quad (29)$$

where $H_{\text{lon}} \in \mathbb{R}^{6 \times 2}$ and $b_{\text{lon}} \in \mathbb{R}^6$. The resulting longitudinal terminal set is shown in Fig. 19.

Remark 2: Since (23) models the longitudinal dynamics of the physical vehicle (the test vehicle in Section V), a negative acceleration means that the vehicle applies the brakes to slow down. This implies that the physics of the systems are such that deceleration is only applied if $v > 0$, i.e., a braking vehicle will not start reversing when applying brakes and reaching a zero velocity. Formally, this yields a hybrid system. However, due to the simplicity of the hybrid system, this issue can be simply tackled by handling the system as if it were a linear system and computing the terminal set as the intersection $X_r^{\text{lon}}(s) \cap \{v, a \mid v \geq 0\}$. To that end, by leaving the lower bound on e_v open, we can construct a low-complexity invariant set and know that $e_v = v - v^r \geq -v^r$ at all times since the physical vehicle may never reverse. We clarify this further using a set of decelerating trajectories in Fig. 19.

As the LQR controller uses a different cost tuning than the MPC controller (4), we compute the associated terminal cost for X_r^{lon} , using YALMIP [45] and the SDPT3 [46] solver to find P_{lon} that satisfies the following LMIs:

$$\begin{aligned} P_{\text{lon}} &:= \min_P \text{trace}(P) \\ \text{s.t. } P &> 0 \\ (A_{\text{lon}} - B_{\text{lon}} K_{\text{lon}})^{\top} P (A_{\text{lon}} - B_{\text{lon}} K_{\text{lon}}) - P &\leq \\ &- (Q_{\text{lon}} + K_{\text{lon}}^{\top} R_{\text{lon}} K_{\text{lon}}) \end{aligned} \quad (30)$$

where A_{lon} and B_{lon} are obtained through zero-order hold discretization and Q_{lon} and R_{lon} correspond to the longitudinal contribution from the cost functions in (21), i.e., $Q_{\text{lon}} = \text{diag}(1, 1)$ and $R_{\text{lon}} = 4$. Upon solving (30), we obtain

$$P_{\text{lon}} = \begin{bmatrix} 210.78 & 80.19 \\ 80.19 & 38.29 \end{bmatrix}. \quad (31)$$

B. Lateral Kinematics

In this section, we derive an invariant set for the lateral kinematics of (6) for a range of velocities $v \in [1, 55/3.6] \text{ m/s}$ and the following terminal state constraints:

$$\begin{aligned} |e_y| &\leq \bar{e}_y = 0.2, \quad |e_{\psi}| \leq \bar{e}_{\psi} = 0.1745, \quad |\delta| \leq \bar{\delta} = 0.5295 \\ |\alpha| &\leq \bar{\alpha} = 0.353, \quad |\delta^{\text{sp}}| \leq \bar{\delta}. \end{aligned} \quad (32)$$

Furthermore, since the reference trajectory is known beforehand, we can easily compute the bounds

$$|\delta^r| \leq \bar{\delta}^r = 0.2079, \quad |\dot{\delta}^r| \leq \bar{\dot{\delta}}^r = 0.1836, \quad |\ddot{\delta}^r| \leq \bar{\ddot{\delta}}^r = 1.68.$$

Now, consider the lateral kinematics in (6) given by

$$\begin{bmatrix} \dot{e}_y \\ \dot{e}_\psi \\ \dot{\delta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} v \sin e_\psi \\ vl^{-1}(\tan \delta - \tan \beta) \\ \alpha \\ w_0^2(\delta^{sp} - \delta) - 2w_0w_1\alpha \end{bmatrix} \quad (33)$$

where we used $\beta := \arctan(\dot{s}/v \tan(\delta^r))$. In order to derive an invariant set for this system, we first need to account for the nonlinear terms. This is done by introducing the parameters

$$\varepsilon_\psi := \frac{\sin e_\psi}{e_\psi}, \quad \varepsilon_\delta := \frac{\tan \delta - \tan \beta}{\delta - \beta} \quad (34)$$

to obtain the following system:

$$\begin{bmatrix} \dot{e}_y \\ \dot{e}_\psi \\ \dot{\delta} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} v\varepsilon_\psi e_\psi \\ vl^{-1}\varepsilon_\delta(\delta - \beta) \\ \alpha \\ w_0^2(\delta^{sp} - \delta) - 2w_0w_1\alpha \end{bmatrix}. \quad (35)$$

Then, by denoting the steering error and steering error rate as

$$e_\delta := \delta - \beta, \quad e_\alpha := \dot{e}_\delta = \dot{\delta} - \dot{\beta} = \alpha - \dot{\beta} \quad (36)$$

we rewrite the first three lines of (35) as

$$\begin{bmatrix} \dot{e}_y \\ \dot{e}_\psi \\ \dot{e}_\delta \end{bmatrix} = \begin{bmatrix} v\varepsilon_\psi e_\psi \\ vl^{-1}\varepsilon_\delta e_\delta \\ e_\alpha \end{bmatrix}. \quad (37)$$

Finally, in order to deal with the last equation in (35), we change the control input from δ^{sp} to

$$\rho := \delta^{sp} - \beta - 2w_0^{-1}w_1\dot{\beta} - w_0^{-2}\ddot{\beta} \quad (38)$$

and define $v_\psi := v\varepsilon_\psi$ and $v_\delta := vl^{-1}\varepsilon_\delta$ to obtain the LPV system

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{e}_y \\ \dot{e}_\psi \\ \dot{e}_\delta \\ \dot{e}_\alpha \end{bmatrix} = \begin{bmatrix} v_\psi e_\psi \\ v_\delta e_\delta \\ e_\alpha \\ w_0^2(\rho - e_\delta) - 2w_0w_1e_\alpha \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} e_y \\ e_\psi \\ e_\delta \\ e_\alpha \end{bmatrix} \quad (39)$$

with state \mathbf{e} , input ρ , and parameters v_ψ and v_δ , which implicitly depend on v , ε_ψ , and ε_δ . Having derived the LPV system (39), our next goal is to construct a polytopic approximation of (39) such that the standard methods from the literature can be used to compute an invariant set [35], [39], [47]. However, in order to do so, we must first understand which possible values the parameters v , ε_ψ , and ε_δ can take.

To characterize which values v_ψ and v_δ can take, we need to first compute the bounds on ε_ψ , ε_δ . By formulating the worst case bound on β as

$$|\beta| = \left| \tan^{-1} \left(\frac{\tan(\bar{\delta}^r)}{1 - \bar{e}_y \tan(\bar{\delta}^r)l^{-1}} \right) \right| \leq \bar{\beta} = 0.2109 \quad (40)$$

we can compute the following parameter bounds:

$$\begin{aligned} 0.995 &\leq \frac{\sin(\bar{e}_\psi)}{\bar{e}_\psi} \leq \varepsilon_\psi \leq \lim_{e_\psi \rightarrow 0} \frac{\sin(e_\psi)}{e_\psi} = 1 \\ 1 &\leq \varepsilon_\delta \leq 1.17 \end{aligned} \quad (41)$$

where the bounds for ε_δ were numerically evaluated by gridding $\delta \in [-\bar{\delta}, \bar{\delta}]$ and $\beta \in [-\bar{\beta}, \bar{\beta}]$. With the bounds from (41), we construct the following polyhedron:

$$\begin{aligned} \mathcal{P} := \{v, v_\psi, v_\delta \mid 1 \leq v \leq 55/3.6, \\ 0.995v \leq v_\psi \leq v, \quad vl^{-1} \leq v_\delta \leq 1.17vl^{-1}\} \end{aligned} \quad (42)$$

which contains all possible realizations of the parameters in (39). Since polyhedron \mathcal{P} now contains eight vertices and (39) is expressed using only v_ψ and v_δ , we realize that we can reduce the number of vertices by projecting \mathcal{P} onto the $v_\psi v_\delta$ -space. In other words, we formulate $\mathcal{P}_{v_\psi v_\delta} = \text{Proj}_{v_\psi v_\delta}(\mathcal{P})$ and denote by $\mathcal{V} = \{v_1, v_2, \dots, v_6\}$ the vertices of $\mathcal{P}_{v_\psi v_\delta}$.

Next, in order to bound the states e_δ , e_α , and input ρ , we compute the worst case bounds

$$|\dot{\beta}| \leq \bar{\beta} = 0.2013, \quad |\ddot{\beta}| \leq \bar{\beta} = 5.9505 \quad (43)$$

numerically by gridding the state space, and compute the following bounds:

$$|e_\delta| \leq \bar{e}_\delta = \bar{\delta} - \bar{\beta} = 0.3186$$

$$|e_\alpha| \leq \bar{e}_\alpha = \bar{\alpha} - \bar{\beta} = 0.1517$$

$$|\rho| \leq \bar{\rho} = \bar{\delta} - \bar{\beta} - 2w_0^{-1}w_1\bar{\beta} - 1/w_0^{-2}\bar{\beta} = 0.2856. \quad (44)$$

Knowing the bounds on $(e_y, e_\psi, e_\delta, e_\alpha)$, we now want to construct an invariant set for (39) that holds for all realizations of the parameters $\mathbf{v} \in \mathcal{V}$. We therefore consider the following polytopic approximation [39] of (39):

$$\Gamma := \{(A, B) \mid A = A(\mathbf{v}), B = B(\mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{V}\} \quad (45)$$

where $A(\mathbf{v})$ and $B(\mathbf{v})$ are obtained through zero-order hold discretization. Using the feedback gain K_{lat} from an LQR controller for system $(A(v_{nom}), B(v_{nom}))$ with $v_{nom} = [13.89, 4.79]^\top$ and tuning $Q_{lat}^{LQR} = \text{diag}([1, 500, 1, 0.1])$ and $R_{lat}^{LQR} = 10^{-4}$, we can stabilize the polytopic system Γ for all $\mathbf{v} \in \mathcal{V}$. Then, following the steps in [47], we obtain an invariant set for the lateral dynamics of the form:

$$\tilde{\mathcal{X}}_{\mathbf{r}}^{\text{lat}}(s) = \{e_y, e_\psi, e_\delta, e_\alpha \mid H_{\text{lat}}[e_y, e_\psi, e_\delta, e_\alpha]^\top \leq b_{\text{lat}}\} \quad (46)$$

where $\tilde{\mathcal{X}}_{\mathbf{r}}^{\text{lat}}$ consists of 16 hyperplanes, i.e., $H_{\text{lat}} \in \mathbb{R}^{16 \times 4}$ and $b_{\text{lat}} \in \mathbb{R}^{16}$. The projection of $\tilde{\mathcal{X}}_{\mathbf{r}}^{\text{lat}}$ in the $e_y e_\psi e_\delta$ -plane is shown in Fig. 20. Since the lateral kinematics are expressed through states $\mathbf{x}_{\text{lat}} = [e_y, e_\psi, \delta, \alpha]^\top$ for Problem (4), the considered lateral terminal set hence becomes

$$\mathcal{X}_{\mathbf{r}}^{\text{lat}}(s) = \{e_y, e_\psi, \delta, \alpha \mid H_{\text{lat}}[e_y, e_\psi, \delta - \delta^r, \alpha - \alpha^r]^\top \leq b_{\text{lat}}\}.$$

To find the corresponding terminal cost for $\mathcal{X}_{\mathbf{r}}^{\text{lat}}$, we can use Finsler's lemma [48] to derive the following LMIs:

$$\begin{aligned} P_{\text{lat}} &:= \min_P \text{trace}(P) \\ \text{s.t. } P &> 0 \\ \begin{bmatrix} P - F & A_{\text{lat}, \text{cl}}(\mathbf{v})^\top P^\top \\ PA_{\text{lat}, \text{cl}}(\mathbf{v}) & P \end{bmatrix} &> 0 \quad \forall \mathbf{v} \in \mathcal{V} \end{aligned} \quad (47)$$

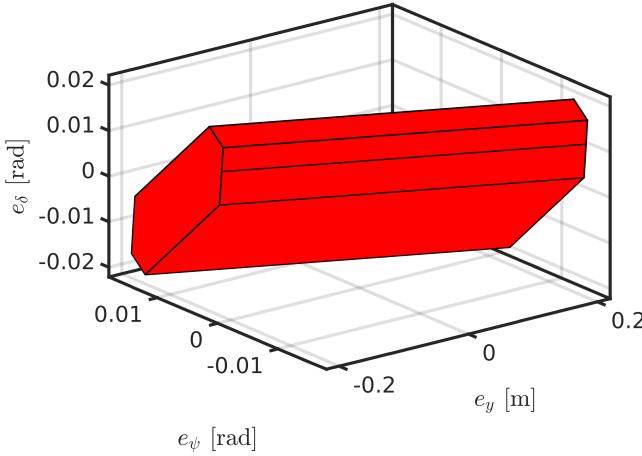


Fig. 20. Projection of the lateral terminal set $\tilde{\mathcal{X}}_r^{\text{lat}}$ in the $e_y e_\psi e_\delta$ -frame.

where $F := Q_{\text{lat}} + K_{\text{lat}}^\top R_{\text{lat}} K_{\text{lat}}$, $A_{\text{lat},\text{cl}}(\mathbf{v}) := A(\mathbf{v}) - B(\mathbf{v})K_{\text{lat}}$ and Q_{lat} and R_{lat} correspond to the lateral components of (21), i.e., $Q_{\text{lat}} = \text{diag}(1, 1, 10, 1)$ and $R_{\text{lat}} = 10$. Note that this corresponds to imposing

$$A_{\text{lat},\text{cl}}(\mathbf{v})^\top P A_{\text{lat},\text{cl}}(\mathbf{v}) - P \preceq -F$$

which is quadratic in the parameter \mathbf{v} . However, this is the Schur complement of the LMI condition in (47), which explains the equivalence. Finally, solving (47) yields

$$P_{\text{lat}} = \begin{bmatrix} 325.51 & 593.13 & 97.32 & 1.46 \\ 593.13 & 6091.11 & 1979.43 & 29.75 \\ 97.32 & 1979.43 & 1159.47 & 17.15 \\ 1.46 & 29.75 & 17.15 & 1.28 \end{bmatrix}. \quad (48)$$

C. Resulting Terminal Set and Terminal cost

Following the steps and computing the terminal sets and costs in Appendixes A and B, we can express the stabilizing terminal set for system (6) and Problem (4) as

$$\mathcal{X}_r^s(s) := \left\{ \mathbf{x} \mid [\mathbf{v}, \mathbf{a}]^\top \in \mathcal{X}_r^{\text{lon}}(s), [e_y, e_\psi, \delta, \alpha]^\top \in \mathcal{X}_r^{\text{lat}}(s) \right\}$$

and the terminal cost as

$$P = \text{blockdiag}(P_{\text{lat}}, P_{\text{lon}}). \quad (49)$$

REFERENCES

- [1] N. Lubbe, H. Jeppsson, A. Ranjbar, J. Fredriksson, J. Bärgman, and M. Östling, "Predicted road traffic fatalities in Germany: The potential and limitations of vehicle safety technologies from passive safety to highly automated driving," in *Proc. IRCOBI Conf.*, Athena, Greece, 2018, pp. 1–36.
- [2] J. Ziegler et al., "Making Bertha drive—An autonomous journey on a historic route," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 2, pp. 8–20, Summer 2014.
- [3] I. Batkovic, U. Rosolia, M. Zanon, and P. Falcone, "A robust scenario MPC approach for uncertain multi-modal obstacles," *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 947–952, Jul. 2021.
- [4] O. Andersson, O. Ljungqvist, M. Tiger, D. Axehill, and F. Heintz, "Receding-horizon lattice-based motion planning with dynamic obstacle avoidance," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 4467–4474.
- [5] I. Batkovic, M. Zanon, M. Ali, and P. Falcone, "Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 256–262.
- [6] G. Cesari, G. Schildbach, A. Carvalho, and F. Borrelli, "Scenario model predictive control for lane change assistance and autonomous driving on highways," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 3, pp. 23–35, Fall 2017.
- [7] Y. Chen, U. Rosolia, W. Ubellacker, N. Csomay-Shanklin, and A. D. Ames, "Interactive multi-modal motion planning with branch model predictive control," 2021, *arXiv:2109.05128*.
- [8] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear MPC: Bridging the gap via the real-time iteration," *Int. J. Control.*, vol. 93, no. 1, pp. 62–80, Jan. 2020.
- [9] P. F. Lima, G. C. Pereira, J. Mårtensson, and B. Wahlberg, "Experimental validation of model predictive control stability for autonomous driving," *Control Eng. Pract.*, vol. 81, pp. 244–255, Dec. 2018.
- [10] S. H. Nair, V. Govindarajan, T. Lin, C. Meissen, H. E. Tseng, and F. Borrelli, "Stochastic MPC with multi-modal predictions for traffic intersections," 2021, *arXiv:2109.09792*.
- [11] U. Rosolia and F. Borrelli, "Learning model predictive control for iterative tasks. A data-driven control framework," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 1883–1896, Jul. 2018.
- [12] O. Ljungqvist, D. Axehill, and J. Löfberg, "On stability for state-lattice trajectory tracking control," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 5868–5875.
- [13] R. Oliveira, P. F. Lima, G. C. Pereira, J. Mårtensson, and B. Wahlberg, "Path planning for autonomous bus driving in highly constrained environments," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2743–2749.
- [14] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Energy-optimal coordination of autonomous vehicles at intersections," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 602–607.
- [15] S. Uebel, N. Murgovski, B. Bäker, and J. Sjöberg, "A two-level MPC for energy management including velocity control of hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5494–5505, Jun. 2019.
- [16] M. Zanon, "A Gauss–Newton-like Hessian approximation for economic NMPC," *IEEE Trans. Autom. Control*, vol. 66, no. 9, pp. 4206–4213, Sep. 2021.
- [17] S. Uebel, N. Murgovski, C. Tempelhahn, and B. Bäker, "Optimal energy management and velocity control of hybrid electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 327–337, Jan. 2018.
- [18] G. R. D. Campos, P. Falcone, R. Hult, H. Wymeersch, and J. Sjöberg, "Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics," *IEEE Intell. Transp. Syst. Mag.*, vol. 9, no. 1, pp. 8–21, Spring 2017.
- [19] G. R. Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjöberg, "Cooperative receding horizon conflict resolution at traffic intersections," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 2932–2937.
- [20] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections: Theory and experiments," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 6, pp. 2510–2525, Nov. 2019.
- [21] N. Evestedt, O. Ljungqvist, and D. Axehill, "Motion planning for a reversing general 2-trailer configuration using closed-loop RRT," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 3690–3697.
- [22] O. Ljungqvist, N. Evestedt, M. Cirillo, D. Axehill, and O. Holmer, "Lattice-based motion planning for a general 2-trailer system," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 819–824.
- [23] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.
- [24] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, vol. 2. Madison, WI, USA: Nob Hill Publishing, 2017.
- [25] J. P. Alsterda, M. Brown, and J. C. Gerdes, "Contingency model predictive control for automated vehicles," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 717–722.
- [26] J. P. Alsterda and J. C. Gerdes, "Contingency model predictive control for linear time-varying systems," 2021, *arXiv:2102.12045*.
- [27] T. Brüdigam, M. Olbrich, D. Wollherr, and M. Leibold, "Stochastic model predictive control with a safety guarantee for automated driving," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 22–36, Jan. 2023.
- [28] T. Brüdigam, R. Jacumet, D. Wollherr, and M. Leibold, "Safe stochastic model predictive control," in *Proc. IEEE 61st Conf. Decis. Control (CDC)*, Dec. 2022, pp. 1796–1802.
- [29] I. Batkovic, M. Ali, P. Falcone, and M. Zanon, "Safe trajectory tracking in uncertain environments," *IEEE Trans. Autom. Control*, vol. 68, no. 7, pp. 4204–4217, Jul. 2023.

- [30] L. Grüne and J. Pannek, *Nonlinear Model Predictive Control*. London, U.K.: Springer, 2011.
- [31] P. F. Lima, J. Mårtensson, and B. Wahlberg, "Stability conditions for linear time-varying model predictive control in autonomous driving," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Dec. 2017, pp. 2775–2782.
- [32] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Dept. Elect. Eng. Inf. Technol., Technische Universität München, Munich, Germany, 2010.
- [33] S. K. Mulagaleti, A. Bemporad, and M. Zanon, "Computation of input disturbance sets for constrained output reachability," 2021, *arXiv:2111.03009*.
- [34] I. Kolmanovsky and E. G. Gilbert, "Theory and computation of disturbance invariant sets for discrete-time linear systems," *Math. Problems Eng.*, vol. 4, no. 4, pp. 317–367, 1998.
- [35] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-parametric toolbox 3.0," in *Proc. Eur. Control Conf.*, Zürich, Switzerland, Jul. 2013, pp. 502–510. [Online]. Available: <http://control.ee.ethz.ch/~mpt>
- [36] I. Batkovic, M. Zanon, N. Lubbe, and P. Falcone, "A computationally efficient model for pedestrian motion prediction," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2018, pp. 374–379.
- [37] M. Koschi, C. Pek, M. Beikirch, and M. Althoff, "Set-based prediction of pedestrians in urban environments considering formalized traffic rules," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2704–2711.
- [38] M. Koschi and M. Althoff, "Set-based prediction of traffic participants considering occlusions and traffic rules," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 2, pp. 249–265, Jun. 2021.
- [39] F. Blanchini and S. Miani, *Set-Theoretic Methods in Control*. Springer, 2008.
- [40] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA, USA: SIAM, 1994.
- [41] R. Verschueren et al., "acados: A modular open-source framework for fast embedded optimal control," Tech. Rep., 2020.
- [42] G. Frison and M. Diehl, "HPIPM: A high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.
- [43] M. Diehl, H. G. Bock, and J. P. Schlöder, "A real-time iteration scheme for nonlinear optimization in optimal feedback control," *SIAM J. Control Optim.*, vol. 43, no. 5, pp. 1714–1736, Jan. 2005.
- [44] P. O. M. Scokaert and J. B. Rawlings, "Feasibility issues in linear model predictive control," *AIChE J.*, vol. 45, no. 8, pp. 1649–1659, Aug. 1999.
- [45] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2004, pp. 284–289.
- [46] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "SDPT3—A MATLAB software package for semidefinite programming, version 1.3," *Optim. Methods Softw.*, vol. 11, nos. 1–4, pp. 545–581, 1999.
- [47] M. Kvasnica, B. Takács, J. Holaza, and D. Ingole, "Reachability analysis and control synthesis for uncertain linear systems in MPT," *IFAC-PapersOnLine*, vol. 48, no. 14, pp. 302–307, 2015.
- [48] P. Pinsler, "Über das vorkommen definiter und semidefiniter formen in scharen quadratischer formen," *Commentarii Mathematici Helveticae*, vol. 9, no. 1, pp. 188–192, Dec. 1936. [Online]. Available: <http://eudml.org/doc/138679>



Ankit Gupta received the bachelor's degree in electronics and telecommunications engineering and the master's degree in control engineering from the University of Mumbai, Mumbai, India, in 2010 and 2013, respectively, and the Ph.D. degree from the Chalmers University of Technology, Gothenburg, Sweden, in 2021.

He is currently working as a Control System Engineer at Zenseact AB, Gothenburg. His research interests include invariant sets, reachability analysis, model predictive control, and automotive control.



Mario Zanon (Member, IEEE) received the master's degree in mechatronics from the University of Trento, Trento, Italy, in 2010, the Diplôme d'Ingénieur degree from the École Centrale Paris, Gif-sur-Yvette, France, in 2010, and the Ph.D. degree in electrical engineering from KU Leuven, Leuven, Belgium, in November 2015.

He conducted research at KU Leuven; the University of Bayreuth, Bayreuth, Germany; the Chalmers University of Technology, Gothenburg, Sweden; and the University of Freiburg, Freiburg im Breisgau, Germany. He was a Post-Doctoral Researcher with the Chalmers University of Technology until 2017. In 2018, he became an Assistant Professor at the IMT School for Advanced Studies Lucca, Lucca, Italy, where he became an Associate Professor in 2021. His research interests include numerical methods for optimization, economic model predictive control (MPC), reinforcement learning, and the optimal control and estimation of nonlinear dynamic systems, in particular for aerospace and automotive applications.



Ivo Batkovic received the M.Sc. degree in engineering physics and the Ph.D. degree in electrical engineering from the Chalmers University of Technology, Gothenburg, Sweden, in 2016 and 2022, respectively.

He is currently a Research Engineer with Zenseact, Gothenburg, working with decision-making and control for autonomous driving. His research interests include model predictive control for constrained autonomous systems in combination with prediction models for safe decision-making in autonomous driving applications.



Paolo Falcone received the M.Sc. ("Laurea degree") degree from the University of Naples "Federico II," Naples, Italy, in 2003, and the Ph.D. degree in information technology from the University of Sannio, Benevento, Italy, in 2007.

He is currently a Professor with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. His research focuses on constrained optimal control applied to autonomous and semiautonomous mobile systems, cooperative driving, and intelligent vehicles, in cooperation with the Swedish automotive industry, with a focus on autonomous driving, cooperative driving, and vehicle dynamics control.