2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Guaranteed Collision Avoidance for Autonomous Vehicles Fusing Model Predictive Control and Data Driven Reachability Analysis

Tingzhong Fu[1], Hoang Hai Nguyen[1] and Rolf Findeisen[1]

Abstract— **Ensuring collision avoidance is a critical challenge for autonomous vehicles, particularly when faced with uncertain moving obstacles. This work presents a robust collision avoidance framework, integrating data-driven reachability analysis with Model Predictive Control (MPC). The framework is specifically designed to address scenarios where detailed information about the moving obstacles that should be avoided is unavailable. A data-driven approach is employed, which utilizes uncertain measurements corrupted by bounded noise of the obstacle. Based on the measurements, an over-approximation of the reachable sets by moving obstacles represented as zonotopes is constructed. To guarantee security, a safety margin is added to the approximation. The resulting set is employed as a polytopic collision avoidance constraint within the robust MPC scheme, enabling effective control of the autonomous vehicle while guaranteeing avoidance of impacts. The effectiveness of the data-driven collision avoidance scheme is demonstrated through extensive simulations. The presented results outline a promising advancement in collision avoidance for autonomous vehicles operating in uncertain environments.**

I. Introduction

One of the challenging problems for controlling autonomous vehicles, such as cars or robots is to avoid collision with moving obstacles, such as humans, or other vehicles. There exists a large number of literature on this topic; comprehensive reviews on collision avoidance algorithms can be found in [1], [2]. While static collision avoidance has been researched intensively (see, for example, [3]–[7]), avoiding dynamically moving obstacles is often more difficult because of the often inherent motion uncertainty. Frequently, simplified models of the dynamic obstacles are used: the dynamic obstacles are assumed to move with a constant velocity and to move along a straight path [8], [9]. Model Predictive Control (MPC, [10]–[12]) is a popular control scheme for avoiding collisions since it can effectively handle constraints. The basic idea of MPC is to minimize, upon a new measurement of the states, a cost function over a prediction horizon while taking the system dynamics and constraints into account, and implementing the first part of the resulting optimal control input. Once new measurements are available this procedure is repeated in a receding horizon fashion. Collision avoidance constraints can conceptually be incorporated easily in MPC schemes ([9], [13]–[16]). Yet this typically leads to computationally challenging to solve non convex optimization problems.

Furthermore, in practice, the movement and model of dynamical obstacles is often unknown and might only be reconstructed from noisy measurement data. One possible approach is to use machine learning and data-driven techniques to learn the dynamics of the obstacle ([14]–[16]). However, doing so often lacks safety guarantees or does only provide probabilistic guarantees. Moreover, obtaining a sufficiently good model from collected data by system identification may be impossible or time-consuming [17]. This is additionally challenged as the dynamics/path of the moving obstacle might change over time. For example, for a quadrotor or robot carrying cargo, in repeating episodes of operations, the weight of the cargo may vary, which may alter the dynamics of the obstacles ([18]). Thus, it is desirable to adjust the dynamics of the obstacles online based on measurements collected from the obstacles. One way to do this for dynamic obstacles is the use of data-based reachability analysis [19]. Similar to the data-driven control (see, for example [20], [21]), the method proposed in [19] uses matrix zonotopes to over-approximate the reachable sets of a linear system or a Lipschitz-continuous nonlinear system based on sufficiently rich measured data.

In this paper, we utilize data-driven reachability analysis using matrix zonotopes [19] to obtain the reachable sets from previously measured state-input data for the moving obstacle. To guarantee collision avoidance, we use the data-driven reachability sets in robust MPC [22] to control the autonomous vehicle.

The remainder of the paper is organized as follows: Section II introduces the data-driven reachability analysis for dynamic obstacles. Section III presents the collision avoidance MPC that uses reachable sets for the obstacles as constraints. Section IV presents simulation results. Section V concludes the paper.

II. Data-Driven Reachable Set of Moving obstacles

This section recapitulates the data-driven reachability approach for nonlinear systems presented in [19]. First, we introduce zonotopes, which are used to overbound the reachable set of the obstacle in a computationally efficient way exploiting basic operations, such as Minkowski sums and linear transformations [23].

Definition 1 (Zonotope [24]): Given a center $c \in \mathbb{R}^n$ and a number $\kappa \in \mathbb{N}$ of generators $g^{(i)} \in \mathbb{R}^n$, a zonotope $\mathbf{Z}$ is defined as

$$\mathbf{Z} := \left\{ c + \sum_{i=1}^{\kappa} \beta_i g^{(i)} \middle| \beta_i \in [-1, 1] \right\}. \tag{1}$$

[1]Control and Cyber-Physical Systems Laboratory (CCPS), Technical University of Darmstadt, Germany. {hoang.nguyen, rolf.findeisen}@iat.tu-darmstadt.de

For convenience, we denote in the following zonotopes by $\mathbf{Z} =< c, G >$, with the generator matrix $G = [g^{(1)}, g^{(2)}, ..., g^{(\kappa)}]$.
A matrix zonotope is an extension where the center and generators are matrices rather than vectors.

Definition 2 (Interval Matrix [24]): An interval matrix $\mathbf{I}$ is a special case of a matrix zonotope, which defines the interval of all possible values for each matrix element between the lower bound $\underline{\mathbf{I}}$ and the upper bound $\overline{\mathbf{I}}$

$$\mathbf{I} := [\underline{\mathbf{I}}, \overline{\mathbf{I}}], \quad \forall i, j : \underline{\mathbf{I}}_{ij} < \overline{\mathbf{I}}_{ij}, \quad \underline{\mathbf{I}}, \overline{\mathbf{I}} \in \mathbb{R}^{m \times n}. \tag{2}$$

We consider nonlinear discrete-time systems

$$z_{k+1} = h(z_k, v_k) + d_k, \tag{3}$$

where $z_k \in \mathbb{R}^n$, $v_k \in \mathbb{R}^{n_u}$ are states and inputs and d_k is a bounded additive noise. The function $h(\cdot)$ is assumed to be twice differentiable and Lipschitz continuous, i.e., there exists a real constant $L \geq 0$ such that, for all α, $\tilde{\alpha} \in \mathbb{R}^{n+n_u}$

$$||h(\alpha) - h(\tilde{\alpha})|| \leq L||\alpha - \tilde{\alpha}||. \tag{4}$$

If the inputs and additive noise of system (3) are bounded by the sets $v_k \in \mathbf{V}$, $d_k \in \mathbf{Z}_d$, the reachable set of the system after N steps is defined as follows.

Definition 3 (Reachable Set): For the system (3), the reachable set $\mathbf{R}_N$ after N time steps, starting from the initial set $\mathbf{R}_0$, subject to the inputs sequences $v_k \in \mathbf{V}_k, \forall k \in \{0, \ldots, N-1\}$ and the noise sequences $d_k \in \mathbf{Z}_d$, is the set of all states encountered along all possible solution trajectories

$$\begin{aligned} \mathbf{R}_N := \{ & z_N \in \mathbb{R}^n | z_{k+1} = h(z_k, v_k) + d_k, \\ & z_0 \in \mathbf{R}_0,\ v_k \in \mathbf{V}_k,\ d_k \in \mathbf{Z}_d : \forall k \in \{0, \ldots, N-1\}\}. \end{aligned} \tag{5}$$

Since in the case of uncertain obstacles the model of the discrete-time nonlinear system (3) is unknown, we construct the reachable sets of the system directly from the collected input-state trajectories. This data-driven approach leverages the property of Lipschtz continuity. Assume that we are able to collect the data set

$$\begin{aligned} \mathcal{V}_- &= [v_0, \ldots, v_{T-1}], \\ \mathcal{Z}_- &= [z_0, \ldots, z_{T-1}], \mathcal{Z}_+ = [z_1, \ldots, z_T]. \end{aligned} \tag{6}$$

Based on the collected data set $\mathcal{D} = (\mathcal{Z}_-, \mathcal{V}_-, \mathcal{Z}_+)$, the data-driven reachable set is given by the following two steps [19]

1) Obtaining a linearized model by the least-squares method around the linearization point $p^* = (z^*, v^*)$:

$$\tilde{M} = (\mathcal{Z}_+ - C_{\mathbf{M}_d})H, \text{ with } H = \begin{bmatrix} 1_{1\times T} \\ \mathcal{Z}_- - 1_{1\times T} \otimes z^* \\ \mathcal{V}_- - 1_{1\times T} \otimes v^* \end{bmatrix}^{\dagger}, \tag{7}$$

where $\dagger$ denotes the right inverse, $\otimes$ denotes the Kronecker product, $C_{\mathbf{M}_d}$ is the so-called center of noise matrix zonotope $\mathbf{M}_d =< C_{\mathbf{M}_d}, G_{\mathbf{M}_d} >$, whose construction is provided in [19].

2) Over-approximating the Lagrange remainder and the model mismatch by a zonotope: After obtaining the approximate linearized model, i.e., the matrix $\tilde{M}$, one over-approximates the model mismatch and the Lagrange remainder. This is based on the assumption that the system dynamic $h(\cdot)$ is Lipschitz continuous for all (z, v).

Theorem 1 ([19]): Given the collected data set $\mathcal{D} = (\mathcal{Z}_-, \mathcal{V}_-, \mathcal{Z}_+)$, the over-approximation $\tilde{\mathbf{R}}_k \supset \mathbf{R}_k$ of the reachable sets defined as (5) for the system dynamic $h(\cdot)$ can be computed by

$$\tilde{\mathbf{R}}_{k+1} = \tilde{M}(1 \times \tilde{\mathbf{R}}_k \times \mathbf{V}_k) + \mathbf{Z}_d + \mathbf{Z}_L + \mathbf{Z}_\epsilon, \tag{8}$$

with

$$\mathbf{Z}_L = \text{zonotope}(\underline{\mathbf{Z}}_L, \overline{\mathbf{Z}}_L), \tag{9a}$$

$$(\underline{\mathbf{Z}}_L)_i = \arg\min_j (\underline{\mathbf{M}}_L)_{i,j}, (\overline{\mathbf{Z}}_L)_i = \arg\max_j (\overline{\mathbf{M}}_L)_{i,j}, \tag{9b}$$

$$[\underline{\mathbf{M}}_L, \overline{\mathbf{M}}_L] = \text{intervalMatrix}(\mathbf{M}_L), \tag{9c}$$

$$\mathbf{M}_L = \mathcal{Z}_+ - \mathbf{M}_d - \tilde{M} \begin{bmatrix} 1_{1\times T} \\ \mathcal{Z}_- - 1_{1\times T} \otimes z^* \\ \mathcal{V}_- - 1_{1\times T} \otimes v^* \end{bmatrix}, \tag{9d}$$

$$\mathbf{Z}_\epsilon =< 0, \text{diag}(L\delta, ..., L\delta) > . \tag{9e}$$

Here $\tilde{M}$ is given by (7). It is computed based on a chosen linearization point p^*. The operator $\text{zonotope}(\cdot)$ converts an interval to a zonotope and $\text{intervalMatrix}(\cdot)$ converts a matrix zonotope to an interval matrix. $\mathbf{Z}_d$ is the noise zonotope, and δ is the so-called covering radius, see [19]. Note that this is an over-approximation of the reachable set which can be, however, obtained purely based on data.

III. Collision Avoidance Using Robust MPC and Data Driven Reachability Sets

In this section, we first provide the descriptions of the systems and illustrate how the collision avoidance problem can be formulated. Then we present the robust model predictive control scheme that guarantees collision avoidance.

A. Autonomous Systems Model

The autonomous system that we control, e.g. a car, robot or drone, is described by the known nonlinear discrete-time system dynamics

$$x_{k+1} = f(x_k, u_k) + e_k. \tag{10}$$

Here, $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^{n_u}$ are states and inputs and $f(\cdot)$ is the known nominal model of the autonomous system. $e_k \in \mathbf{Z}_e$ is assumed to be an additive bounded disturbance, where the upper-bound is assumed to be known. The system is subject to state and input constraints

$$\ell_j(x_k, u_k) \leq 0,\ j = 1, ..., p. \tag{11}$$

The reachable set of the controlled vehicle at the k-th step is denoted as $\mathbf{R}_k^{(\text{obj})}$ and can be obtained exploiting the vehicle model. Moreover, we assume that the vehicle has a physical shape $\mathbb{B}^{(\text{obj})}$. Given those sets the reachable occupied space of the controlled autonomous vehicle can be described by a polytope of the form

$$\mathbb{E}(\mathbf{R}_k^{(\text{obj})}, \mathbb{B}^{(\text{obj})}) = \{y \in \mathbb{R}^{n_p} : G_k y \leq g_k\}. \tag{12}$$

Here $G : \mathbb{R}^n \to \mathbb{R}^{l\times n_p}$, $g : \mathbb{R}^n \to \mathbb{R}^l$, where l is the number of half-spaces that define the object polytope $\mathbb{E}$.

B. Obstacles

We assume that the obstacle dynamics is given by a nonlinear model similar to (3) subject to the same assumptions on $h(\cdot)$. However, the model is not known by the controller. Instead, the autonomous vehicle (10) can only access the past data (6) generated by the obstacle. Moreover, we assume that the obstacle has a physical shape $\mathbb{B}$. Similar to the autonomous vehicle, based on the data-driven reachable set $\tilde{\mathbf{R}}_k$, its reachable occupied space can be over-approximated by

$$\mathbb{O}(\tilde{\mathbf{R}}_k, \mathbb{B}) = \{y \in \mathbb{R}^{n_p} : A_k y \leq b_k\}, \tag{13}$$

where $A : \mathbb{R}^n \to \mathbb{R}^{\xi\times n_p}$, $b : \mathbb{R}^n \to \mathbb{R}^\xi$, and ξ is the number of half-spaces that define the obstacle polytope $\mathbb{O}$.

C. Optimization-Based Obstacle Avoidance Formulation

Given the reachable occupied spaces of the controlled autonomous system and the obstacle as described in (12) and (13), we formulate the collision avoidance problem by defining a required distance ([25])

$$\operatorname{dist}(\mathbb{E}(\mathbf{R}_k^{(\mathrm{obj})}, \mathbb{B}^{(\mathrm{obj})}), \mathbb{O}(\tilde{\mathbf{R}}_k, \mathbb{B})) > d_{\min}, \tag{14}$$

where $d_{\min} \geq 0$ denotes an additional safety margin between the autonomous system and the obstacle. Here the notion of *distance* is defined as

$$\operatorname{dist}(\mathbb{E}(\cdot), \mathbb{O}(\cdot)) := \min_\gamma \{||\gamma|| : \ (\mathbb{E}(\cdot)+\gamma)\cap\mathbb{O}(\cdot) \neq \emptyset\}. \tag{15}$$

Note that, condition (14) is equivalent to

$$\begin{aligned} &\exists \lambda_k \geq 0, \ \mu_k \geq 0 : -g_k^\top \mu_k - b_k^\top \lambda_k > d_{\min}, \\ &G_k^\top \mu_k + A_k^\top \lambda_k = 0, ||A_k^\top \lambda_k|| \leq 1, \end{aligned} \tag{16}$$

where $\lambda \in \mathbb{R}^\xi$, $\mu \in \mathbb{R}^l$. The equivalent formulation (16) is smooth and thus can be handled by general-purpose optimizers. We do not elaborate on this here further.

Remark 1: For simplicity of presentation, we consider only one obstacle. However, one can easily extend the approach to multiple obstacles by considering multiple realizations of (16) at the same time.

D. Robust Model Predictive Control for Collision Avoidance

To account for the disturbance acting on the autonomous vehicle, we exploit tube-based robust MPC, cf. e.g. [22]. The basic idea is that a tube size $s_{k|t}$ characterizing the sublevel sets of the incremental Lyapunov function $V_\delta(\tilde{x}, x_{k|t}, u_{k|t}) \leq s_{k|t}^2$ is online computed to tighten the state and input constraints. Together with the data-driven reachability analysis (8) and the smooth formulation of collision avoidance (16), we propose the following scheme

$$\begin{aligned} \min_{\substack{x_{\cdot|t}, u_{\cdot|t}, s_{\cdot|t}, \\ w_{\cdot|t}, \lambda_{\cdot|t}, \mu_{\cdot|t}}} \ & \sum_{k=0}^{N-1} (\|x_{k|t} - \hat{x}_{r,k|t}\|_Q^2 + \|u_{k|t} - \hat{u}_{r,k|t}\|_R^2) \\ & + \|x_{N|t} - \hat{x}_{r,N|t}\|_P^2 && \text{(17a)} \\ \text{s.t.} \quad & x_{0|t} = x_t, \ s_{0|t} = 0, && \text{(17b)} \\ & x_{k+1|t} = f(x_{k|t}, u_{k|t}), && \text{(17c)} \\ & s_{k+1|t} = \rho s_{k|t} + w_{k|t}, && \text{(17d)} \\ & w_{k|t} \geq \tilde{w}_\delta(x_{k|t}, u_{k|t}, s_{k|t}), && \text{(17e)} \\ & \ell_j(x_{k|t}, u_{k|t}) + \eta_j s_{k|t} \leq 0, && \text{(17f)} \\ & s_{k|t} \leq \overline{s}, \ w_{k|t} \leq \overline{w}, && \text{(17g)} \\ & (x_{N|t}, s_{N|t}) \in \mathcal{X}_f, && \text{(17h)} \\ & \tilde{\mathbf{R}}_{k+1|t} = \tilde{M}_t(1 \times \tilde{\mathbf{R}}_{k|t} \times \mathbf{V}_{k|t}) + \mathbf{Z}_d + \mathbf{Z}_{L,t} + \mathbf{Z}_{\epsilon,t}, && \text{(17i)} \\ & \tilde{\mathbf{R}}_{0|t} = z_t, && \text{(17j)} \\ & \lambda_{k|t} \geq 0, \ \mu_{k|t} \geq 0, && \text{(17k)} \\ & -g_{k|t}^\top \mu_{k|t} - b_{k|t}^\top \lambda_{k|t} > d_{\min}, && \text{(17l)} \\ & G_{k|t}^\top \mu_{k|t} + A_{k|t}^\top \lambda_{k|t} = 0, ||A_{k|t}^\top \lambda_{k|t}|| \leq 1, && \text{(17m)} \\ & k = 0, ..., N-1, \ j = 1, ..., p. \end{aligned}$$

Here N is the prediction horizon, $\hat{x}_r$, $\hat{u}_r$ denote a desired reference trajectory, Q, R are state and input weight matrix of the stage cost, P is the terminal weight matrix of the terminal cost, $\mathcal{X}_f$ is the terminal region. $s_{k|t}$ characterizes the tube size, $w_{k|t}$ characterizes the uncertainty bound, ρ is the so-called contraction constant. In the disturbance dynamic constraint (17e), $\tilde{w}_\delta$ is determined from the additive bounded disturbance e_k in system dynamic (10) [22]. As outlined in [26], the state and input constraints $g_j(x_{k|t}, u_{k|t})$ are tightened by the tube size $s_{\cdot|t}$. Summarizing, the conditions (17b)-(17h) ensure the robustness despite of disturbances, conditions (17i)-(17j) represent the data-driven reachable sets of the moving obstacle, while conditions (17k)-(17m) ensure collision avoidance.

Remark 2: It is clear that, if the over-approximation of reachable set can be computed from (8) and if the optimization problem (17) can be solved at each time step, then the outlined approach guarantees collision avoidance between the autonomous system and the obstacle(s). An in-depth analysis will be performed in forthcoming work.

E. Implementation Considerations

To employ the outlined scheme, one needs to determine the incremental Lyapunov function, the contraction constant ρ, the upper bound for the uncertainty $\tilde{w}_\delta$ and the scalars η_j based on [22], [26]. Furthermore, the collected data of the obstacle $\mathcal{D} = (\mathcal{Z}_-, \mathcal{V}_-, \mathcal{Z}_+)$ needs to be split into q overlapping parts $\mathcal{D}_i = (\mathcal{Z}_{-,i}, \mathcal{V}_{-,i}, \mathcal{Z}_{+,i})$, so that in the online implementation, the system matrix of obstacle $\tilde{M}_t$ and zonotopes $\mathbf{Z}_{L,t}$, $\mathbf{Z}_{\epsilon,t}$ in (17i) can be determined by a part $\mathcal{D}_i$ that is based on the current state z_t. A possible selection criterion is the shortest distance between the current state z_t and the centroid c_i, which is the mean of points in $\mathcal{Z}_{-,i}$. We make the assumption that the selected divided data $\mathcal{D}_i$ based on this selection criterion covers the space the obstacle can reach for a given horizon N.

Remark 3: If the obstacle dynamics is linear, and provided that it is sufficiently persistently excited, we only need a single sufficiently long data set to compute the reachable set (see [20]). In case the obstacle is nonlinear and the reachable set

computation is over-approximated at each point, collecting more data at different states will reduce conservatism.

The resulting approach is summarized in Algorithm 1.

Algorithm 1 MPC with Collision Avoidance and Data-driven Reachability Analysis

Preparation 1: determine the contraction constant ρ, upper bound of uncertainty $\tilde{w}_\delta$, scalars η_j based on [22], [26].
Preparation 2: divide $\mathcal{D} = (\mathcal{Z}_-, \mathcal{V}_-, \mathcal{Z}_+)$ into q parts $\mathcal{D}_i = (\mathcal{Z}_{-,i}, \mathcal{V}_{-,i}, \mathcal{Z}_{+,i})$, determine their centroid c_i from $\mathcal{Z}_{-,i}$.
Input: contraction constant ρ, uncertainty bound $\tilde{w}_\delta$, scalars η_j, divided input-state data $\mathcal{D}_i$ and its centroid c_i, physical shape $\mathbb{B}^{(\mathrm{obj})}$ of controlled object and that of obstacle $\mathbb{B}$, reference point $(\hat{x}_r, \hat{u}_r)$, input constraint $\mathbf{U}_{k|t}$, state constraint $\mathbf{X}_{k|t}$, cost matrices (Q, R, P), prediction horizon N, initial measured states x_t, z_t.

1: Set $t \leftarrow 0$.
2: Select the part of data $\mathcal{D}_i$ based on the closest distance between its centroid c_i and the current state z_t.
3: Compute the reachable sets of the obstacle $\tilde{\mathbf{R}}_{k+1}$ by (7) and (8) using the center of current state zonotope and input zonotope as the linearziation points ([19], [27]) compute the reachable occupied space of the obstacle $\mathbb{O}(\tilde{\mathbf{R}}_k, \mathbb{B})$, compute A_k, b_k.
4: Compute the reachable occupied space of the object $\mathbb{E}(\mathbf{R}_k^{(\mathrm{obj})}, \mathbb{B}^{(\mathrm{obj})})$, compute G_k, g_k.
5: Solve (17) to obtain the optimal solution $x^* = (x^*_{0|t}, \cdots, x^*_{N|t})$, $u^* = (u^*_{0|t}, \cdots, u^*_{N-1|t})$ based on the current state of object x_t.
6: Apply the first input $u^*_{0|t}$ to the autonomous system.
7: Set $t \leftarrow t + 1$.
8: Return to step 2.

IV. Simulation Results

We apply the collision avoidance MPC scheme exploiting data-based reachability (17) to a simple example.

A. Controlled Autonomous Vehicle

We consider an autonomous car described by the following simplified dynamics

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} u_1 \cos(x_3) \\ u_1 \sin(x_3) \\ u_2 \end{bmatrix} + e. \tag{18}$$

Here x_1 and x_2 are the Cartesian coordinates of the center of mass of the car, x_3 is the angle with respect to the Cartesian coordinate x_1. It is assumed that the states can be accurately measured. The input u_1 is the longitudinal velocity of the car and u_2 is the angular velocity. The system is subject to additive disturbance e given by

$$e \in \mathbf{Z}_e = [-0.05, 0.05] \times [-0.05, 0.05] \times [-0.05, 0.05]. \tag{19}$$

The constraints for the inputs are

$$\mathbf{U} = [0, 2] \times [-\frac{2}{5}\pi, \frac{2}{5}\pi]. \tag{20}$$

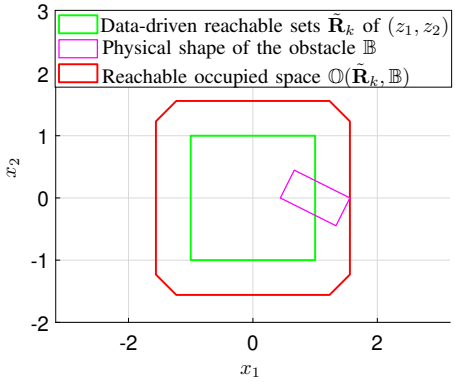


Fig. 1: Over-approximation of the reachable occupied space $\mathbb{O}(\tilde{\mathbf{R}}_k, \mathbb{B})$ for the car example by expanding reachable sets of (z_1, z_2) by $\sqrt{(\frac{\text{length}}{2})^2 + (\frac{\text{width}}{2})^2}$.

We consider that the car has a rectangular shape with length 1 and width 0.5. Since the reachable sets (states) of the car is fully described by the tubes $\mathbf{R}^{(\mathrm{obj})}_{k|t} = V_\delta(\tilde{x}, x_{k|t}, u_{k|t}) \leq s^2_{k|t}$, its reachable occupied spaces can be represented by ([28])

$$\mathbb{E}(s_{k|t}, x_{k|t}) = p_x(x_{k|t}) + R(x_{k|t})(1 + Ls_{k|t})\mathbb{B}^{(\mathrm{obj})}, \tag{21}$$

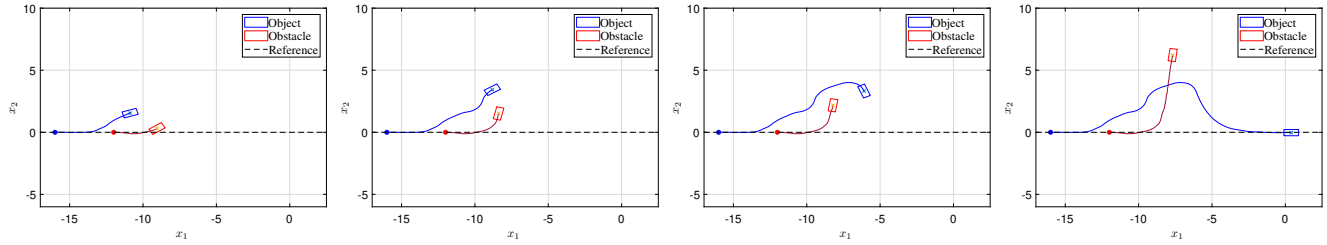
where $p_x(x_k)$ denotes the position of the object depending the current state x_k, $R(x_k)$ denotes the rotation of the object. $\mathbb{B}^{(\mathrm{obj})}$ is the physical shape of the controlled object, and its physical shape is expanded by a factor $(1+Ls_{k|t})$ to account for disturbance. The $L = 0.0754$ is determined by numerical verification.

B. Description of the Dynamic Obstacle

The dynamic obstacle is also a car with the same dynamics, i.e., has the same dynamic equation (18), subject to additive disturbances (19), and the same physical shape. In comparison to the controlled car, we assume that the input is constrained to $[0, 1] \times [-\frac{1}{5}\pi, \frac{1}{5}\pi]$. Note that we use the model of the obstacle car only to generate data $\mathcal{D}$, the dynamic model of the obstacle is not used in the controller. Since the reachable occupied space of the obstacle $\mathbb{O}(\tilde{\mathbf{R}}_k, \mathbb{B})$ is complicated, we over-approximate it by expanding the reachable sets by the maximum distance that the obstacle can reach as shown in Fig. 1.

C. MPC Setup

The control goal for the controlled car is to drive with a constant velocity of $1\,\mathrm{m/s}$ along $x_2 = 0$ towards a positive direction of the x_1-axis, while avoiding the other car on its way. The corresponding reference trajectory $[\hat{x}_{r,k|t}, \hat{u}_{r,k|t}]$ is given by $\hat{x}_{r,k|t} = [0, 0, 0]^\top, \hat{u}_{r,k|t} = [1, 0]^\top$. The weight matrices are chosen to be $Q = \mathrm{diag}([0, 1, 1]), R = \mathrm{diag}([100, 1])$ and the terminal weight P is computed based on the Linear-quadratic Regulator (LQR) for the linearized dynamics at the reference point ([10], [29]). The dynamics is discretized by a fourth-order Runge-Kutta (RK4) method with sampling time $T_s = 0.2\,\mathrm{s}$. The prediction horizon is set



(a) $t = 30$ (b) $t = 45$ (c) $t = 60$ (d) $t = 100$

Fig. 2: Collision avoidance simulation example depicting four time steps. (a) $t = 30$. (b) $t = 45$. (c) $t = 60$. (d) $t = 100$.

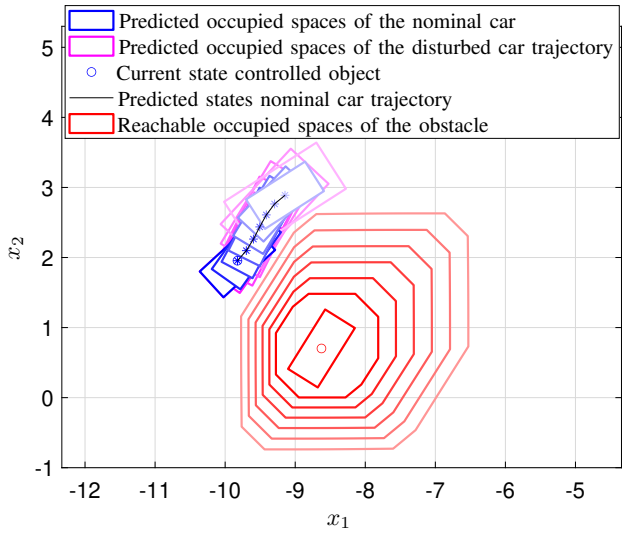


Fig. 3: Predicted reachable occupied spaces of the controlled car $\mathbb{E}(\mathbf{R}_k^{(\mathrm{obj})}, \mathbb{B}^{(\mathrm{obj})})$ and the dynamic obstacle $\mathbb{O}(\tilde{\mathbf{R}}_k, \mathbb{B})$ for $k = 0, ..., N$ at time step $t = 36$ for an exemplary simulation.

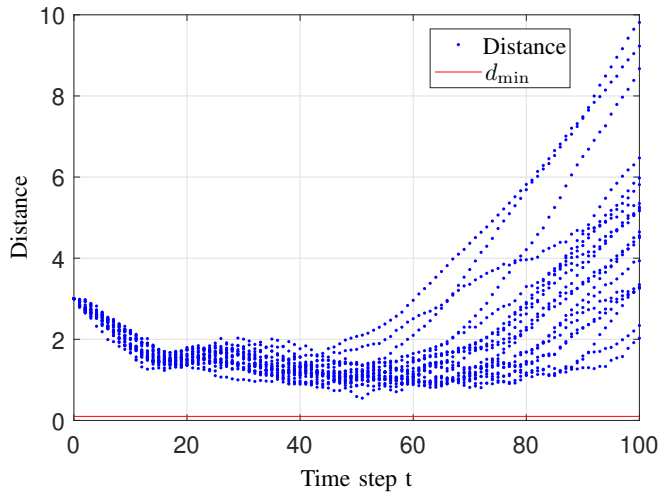


Fig. 4: Distance between object and obstacle for 20 exemplary simulations, starting from the same initial states. For each simulation and for each time step, the obstacle moves "randomly".

to $N = 6$. We apply the method as in [26] to obtain the incremental Lyapunov function. Based on the incremental Lyapunov function, we obtain the contraction constant $\rho = 0.9998$ and the uncertainty bound $\tilde{w}_\delta = 1.2480$.

D. Simulation Results

The results of an exemplary simulation are shown in Fig. 2 and 3, where the inputs of the dynamic obstacle at each moment are randomly generated among its input constraint. Fig. 2 shows four moments of the process of collision avoidance, where we can see that the controlled object succeeds in avoiding the dynamic obstacle. The planned motion to avoid the dynamic obstacle at moment $t = 36$ is shown in Fig. 3.

Moreover, we repeated the simulations for 20 times under the same initial states for both object and obstacle. For each simulation and each step, the obstacle drives randomly, which means that the real trajectory of the dynamic obstacle is random and unpredictable. The distance between the car and the obstacle is shown in Fig. 4, where the distance is always larger than the safety margin $d_{\min}$ despite of the disturbances.

V. CONCLUSIONS

We presented an obstacle avoidance framework that fuses data-based reachability analysis and robust MPC. As the model of the dynamic obstacle is unknown, the reachable set for collision avoidance is over-approximated using collected data. The resulting obstacle avoidance conditions are reformulated smoothly as obstacle avoidance constraints which are incorporated into a robust tube-based MPC. The simulation results underline the effectiveness of the proposed framework. Future research will consider uncertain dynamics of the autonomous systems and the obstacle as well. Furthermore, we are interested in reducing the conservatism.

REFERENCES

[1] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006.

[2] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[3] S. Singh, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust online motion planning via contraction theory and convex optimization," in *IEEE Int. Conf. Robot. Automat. (ICRA)*, 2017, pp. 5883–5890.

[4] M. Nolte, M. Rose, T. Stolte, and M. Maurer, "Model predictive control based trajectory generation for autonomous vehicles—an architectural approach," in *IEEE Intell. Vehicles Symp. (IV)*, 2017, pp. 798–805.

[5] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Automat. Mag.*, vol. 4, no. 1, pp. 23–33, 1997.

[6] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *Int. J. Robot. Research*, vol. 28, no. 8, pp. 933–945, 2009.

[7] G. Schildbach and F. Borrelli, "A dynamic programming approach for nonholonomic vehicle maneuvering in tight environments," in *IEEE Intell. Vehicles Symp. (IV)*, 2016, pp. 151–156.

[8] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: A collision cone approach," *IEEE Trans. Syst. Man Cybern.*, vol. 28, no. 5, pp. 562–574, 1998.

[9] M. Ammour, R. Orjuela, and M. Basset, "Collision avoidance for autonomous vehicle using MPC and time varying sigmoid safety constraints," *6th IFAC Conf. Engine Powertrain Control Simulation Modeling E-COSM*, vol. 54, no. 10, pp. 39–44, 2021.

[10] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing, 2017, vol. 2.

[11] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux Meeting Syst. Contr.*, vol. 11, 2002, pp. 119–141.

[12] S. Lucia, M. Kögel, P. Zometa, D. E. Quevedo, and R. Findeisen, "Predictive control, embedded cyberphysical systems and systems of systems–a perspective," *Annu. Rev. Contr.*, vol. 41, pp. 193–207, 2016.

[13] S. Taherian, K. Halder, S. Dixit, and S. Fallah, "Autonomous collision avoidance using MPC with LQR-based weight transformation," *Sensors*, vol. 21, no. 13, p. 4296, 2021.

[14] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for collision avoidance and control of UAVs with dynamic obstacles," *IEEE Robot. Automat. Let.*, vol. 5, no. 4, pp. 6001–6008, 2020.

[15] C. Pan, Z. Peng, L. Liu, and D. Wang, "Data-driven distributed formation control of under-actuated unmanned surface vehicles with collision avoidance via model-based deep reinforcement learning," *Ocean Eng.*, vol. 267, p. 113 166, 2023.

[16] S. X. Wei, A. Dixit, S. Tomar, and J. W. Burdick, "Moving obstacle avoidance: A data-driven risk-aware approach," *IEEE Contr. Syst. Let.*, vol. 7, pp. 289–294, 2023.

[17] A. Simpkins, "System identification: Theory for the user," *IEEE Robot. Automat. Mag.*, vol. 19, no. 2, pp. 95–96, 2012.

[18] J. Bethge, B. Morabito, J. Matschek, and R. Findeisen, "Multi-mode learning supported model predictive control with guarantees," *6th IFAC Conf. Nonlinear Model Predictive Control NMPC*, vol. 51, no. 20, pp. 517–522, 2018.

[19] A. Alanwar, A. Koch, F. Allgöwer, and K. H. Johansson, "Data-driven reachability analysis using matrix zonotopes," in *Learning for Dynamics and Control*, PMLR, 2021, pp. 163–175.

[20] J. Willems, I. Markovsky, P. Rapisarda, and B. De Moor, "A note on persistency of excitation," in *43rd IEEE Conf. Decision Contr. (CDC)*, vol. 3, 2004, pp. 2630–2631.

[21] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annu. Rev. Contr.*, vol. 52, pp. 42–64, 2021.

[22] J. Köhler, R. Soloperto, M. A. Müller, and F. Allgöwer, "A computationally efficient robust model predictive control framework for uncertain nonlinear systems," *IEEE Trans. Autom. Contr.*, vol. 66, no. 2, pp. 794–801, 2020.

[23] M. Althoff and B. H. Krogh, "Zonotope bundles for the efficient computation of reachable sets," in *50th IEEE Conf. Decision Contr. Europ. Contr. Conf.*, 2011, pp. 6814–6821.

[24] M. Althoff, "An introduction to CORA 2015," *Proc. of the workshop on applied verification for continuous and hybrid systems*, vol. 34, pp. 120–151, 2015.

[25] X. Zhang, A. Liniger, and F. Borrelli, "Optimization-based collision avoidance," *IEEE Trans. Contr. Syst. Technol.*, vol. 29, no. 3, pp. 972–983, 2020.

[26] J. Köhler, M. A. Müller, and F. Allgöwer, "A nonlinear model predictive control framework using reference generic terminal ingredients," *IEEE Trans. Autom. Contr.*, vol. 65, no. 8, pp. 3576–3583, 2019.

[27] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technische Universität München, 2010.

[28] R. Soloperto, J. Köhler, F. Allgöwer, and M. A. Müller, "Collision avoidance for uncertain nonlinear systems with moving obstacles using robust model predictive control," in *18th Europ. Contr. Conf. (ECC)*, 2019, pp. 811–817.

[29] H. Chen and F. Allgöwer, "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," in *Europ. Contr. Conf. (ECC)*, 1997, pp. 1421–1426.