Scalable Robust Model Predictive Control for Linear Sampled-Data Systems

Felix Gruber and Matthias Althoff

Abstract— **We propose a robust reachable-set-based model predictive control method for constrained linear systems. The systems are described by sampled-data models, where a continuous-time physical plant is controlled by a discrete-time digital controller. Thus, the state measurement and the control input are only updated at discrete sampling times, while the constraint satisfaction must be guaranteed not only at, but also between two consecutive time steps. By considering the computation time and using scalable reachability analysis and convex optimization tools, we compute real-time controllers that ensure constraint satisfaction for an infinite time horizon. We demonstrate the applicability of our proposed method using a vehicle platooning benchmark.**

I. Introduction

Model predictive control (MPC) has proven to be a very successful control method for complex systems over the last decades [1]. Its great popularity is mainly due to its simple concept and its ability to deal with state and input constraints. In order to use MPC in safety-critical applications, e.g., autonomous driving or human-robot collaboration, formal robustness guarantees against uncertainties are needed. Thus, robust MPC approaches are required that guarantee constraint satisfaction for all possible realizations originating from uncertainties, e.g., unknown but bounded disturbances.

An ideal robust MPC method performs a min-max optimization over feedback policies [2]. However, this approach is impractical due to its computational complexity. Other robust methods use linear state feedback policies and tighten the state and input constraints so that only the disturbance-free prediction model needs to be considered [3], [4]. A similar idea, which is often referred to as tube-based MPC [5], is to ensure that the state trajectory stays within a tube of fixed size that is ideally given by the minimal robust positively invariant (RPI) set. A variety of extensions of this method exists, such as parameterized or elastic tube-based MPC [6], [7]. Depending on the application, these approaches are successfully applied in real-time. For instance, a system of oscillating masses with sampling times in the order of $1\,\mathrm{ms}$ and the trajectory tracking of a mobile robot with a sampling period of $350\,\mathrm{ms}$ are studied [8], [9].

When considering discrete-time models, as done by most of the literature on robust MPC including all previously mentioned papers, constraint satisfaction can only be guaranteed at discrete time steps. However, in reality, a continuous-time physical plant is controlled by a discrete-time digital controller. Therefore, constraint satisfaction must also be guaranteed between two sampling instants. Due to its practical importance, the consideration of sampled-data robust MPC has gained increasing interest. For instance, the approach in [5] is extended to deal with piecewise constant nominal control inputs while continuous-time state feedback is assumed [10]. In addition to sampled-data nominal inputs, in [11], state measurements are only updated at discrete time instants while the unknown but bounded additive disturbance is assumed to be constant during time intervals.

Due to the high computational complexity of most robust MPC approaches, traditional centralized MPC is often unsuitable to deal with high-dimensional systems. For instance, the computation of the minimal RPI set is a hard task for high-dimensional systems [12]. Typically, these problems can only be solved if the original control problem can be decomposed into simpler ones, as done in distributed MPC [13], [14]. In contrast, our reachability analysis has a complexity of $\mathcal{O}(n_x^3)$ with respect to the state space dimension n_x [15], which allows us to handle comparably larger systems. In particular, we consider a 9-dimensional benchmark problem with a sampling time of $100\,\mathrm{ms}$ without exploiting the sparsity of the system.

In this paper, we propose a scalable robust reachable-set-based MPC approach for constrained linear sampled-data systems. As in [11], [16], our controller receives state measurements and outputs piecewise constant control signals only at discrete time steps. Additionally, we take computation times explicitly into account, which is neglected by most robust MPC approaches, with the exception of, e.g., [17], [18]. We demonstrate the usefulness of our robust reachable-set-based MPC approach using a vehicle platooning benchmark.

The rest of this paper is organized as follows. In Sec. II, the control problem is stated and the applied reachability analysis is summarized. Subsequently, the computation of verified terminal sets is presented in Sec. III. Our reachable-set-based dual-mode MPC approach is stated in Sec. IV, followed by a numerical example in Sec. V. Conclusions and suggestions for future work are provided in Sec. VI.

II. Preliminaries

In this section, we state the control objective. Additionally, we give an overview of our reachability analysis.

A. Problem Statement

In this paper, we consider disturbed, continuous-time, linear time-invariant systems of the form

$$\dot{x}(t) = Ax(t) + Bu(t) + w(t), \tag{1}$$

The authors are with the Department of Informatics, Technical University of Munich, Boltzmannstr. 3, 85748 Garching, Germany.
Email: {felix.gruber, althoff}@tum.de

where $x(t) \in \mathbb{R}^{n_x}$ is the system state and $u(t) \in \mathbb{R}^{n_u}$ is the control input at time $t \in \mathbb{R}_{\geq 0}$. The additive disturbance trajectory $w(\cdot)$ is unknown but bounded by the disturbance set $\mathcal{W} \subset \mathbb{R}^{n_x}$, i.e., $w(t) \in \mathcal{W}$ for all times t. To obtain a more concise notation, we use $w(\cdot) \in \mathcal{W}$.

In contrast to the system dynamics given in continuous-time, the controller receives the measurement of the state and sets the piecewise constant control input u only at discrete sampling times $t_k = k\Delta t$ with $k \in \mathbb{Z}_{\geq 0}$ and $\Delta t \in \mathbb{R}_{>0}$. We employ the simple yet effective sampled-data state feedback control law

$$u(t) = \bar{u}(t_k) + Kx(t_k) \qquad \text{for } t \in [t_k, t_{k+1}), \tag{2a}$$

$$\bar{u}(t) = \bar{u}(t_k) \qquad \text{for } t \in [t_k, t_{k+1}), \tag{2b}$$

where $\bar{u}$ is the piecewise constant correction input that is optimized online, and $K \in \mathbb{R}^{n_u \times n_x}$ denotes a user-defined feedback matrix.

The disturbed system in (1) must satisfy state and input constraints of the form

$$x(\cdot) \in \mathcal{X}, \quad u(\cdot) \in \mathcal{U}, \tag{3}$$

i.e., the state and input trajectories must stay inside the given constraint sets $\mathcal{X} \subset \mathbb{R}^{n_x}$ and $\mathcal{U} \subset \mathbb{R}^{n_u}$, respectively. In this paper, the control objective is to steer the disturbed system to a neighborhood of the origin while minimizing a user-defined cost function and satisfying the constraints in (3) for all times $t \in \mathbb{R}_{\geq 0}$.

Assumption 1: $\mathcal{X}, \mathcal{U}$, and $\mathcal{W}$ are convex, closed, bounded polytopes containing the origin. □

B. Reachability Analysis

To accommodate for the state-dependent control input in our set-based reachability analysis, we consider an augmented state $\tilde{x} \in \mathbb{R}^{n_x+n_u}$ [16]. The corresponding dynamics are

$$\underbrace{\begin{bmatrix}\dot{x}(t)\\ \dot{u}(t)\end{bmatrix}}_{\dot{\tilde{x}}(t)} = \underbrace{\begin{bmatrix}A & B\\ 0 & 0\end{bmatrix}}_{\widetilde{A}} \underbrace{\begin{bmatrix}x(t)\\ u(t)\end{bmatrix}}_{\tilde{x}(t)} + \underbrace{\begin{bmatrix}w(t)\\ 0\end{bmatrix}}_{\widetilde{w}(t)}, \tag{4}$$

where the piecewise constant state-dependent input u is given in (2). Additionally, we define the projection operators

$$\Pi_x\big(\widetilde{\mathcal{S}}\big) = \left\{ x \in \mathbb{R}^{n_x} \,\middle|\, \exists u \in \mathbb{R}^{n_u} \text{ such that } \begin{bmatrix}x\\ u\end{bmatrix} \in \widetilde{\mathcal{S}} \right\},$$

$$\Pi_u\big(\widetilde{\mathcal{S}}\big) = \left\{ u \in \mathbb{R}^{n_u} \,\middle|\, \exists x \in \mathbb{R}^{n_x} \text{ such that } \begin{bmatrix}x\\ u\end{bmatrix} \in \widetilde{\mathcal{S}} \right\},$$

which project a set of augmented states $\widetilde{\mathcal{S}} \subset \mathbb{R}^{n_x+n_u}$ onto the space of nonaugmented states and inputs, respectively.

The solution of (4) at time $t \geq t_0$ based on the nonaugmented initial state $x(t_0)$, piecewise constant correction input trajectory $\bar{u}(\cdot)$, and disturbance trajectory $w(\cdot)$ is denoted by $\tilde{\xi}\big(t, x(t_0), \bar{u}(\cdot), w(\cdot)\big) \in \mathbb{R}^{n_x+n_u}$. By considering an initial state set $\mathcal{X}_0 \subseteq \mathcal{X}$ and the disturbance set $\mathcal{W}$, we obtain the exact reachable set

$$\widetilde{\mathcal{R}}^e\big(t, \mathcal{X}_0, \bar{u}(\cdot), \mathcal{W}\big) = \big\{\tilde{\xi}\big(t, x(t_0), \bar{u}(\cdot), w(\cdot)\big) \;\big|\; x(t_0) \in \mathcal{X}_0, \\ w(\cdot) \in \mathcal{W}\big\}$$

for the system in (4), which is the set of augmented states that are reachable at time t. Similarly, the exact reachable set for the time interval $\tau_k = [t_k, t_{k+1})$ is

$$\widetilde{\mathcal{R}}^e\big(\tau_k, \mathcal{X}_0, \bar{u}(\cdot), \mathcal{W}\big) = \bigcup_{t \in \tau_k} \widetilde{\mathcal{R}}^e\big(t, \mathcal{X}_0, \bar{u}(\cdot), \mathcal{W}\big),$$

which is the union of reachable sets at all times within τ_k. Since, in general, reachable sets cannot be computed exactly [19], we compute tight over-approximations $\widetilde{\mathcal{R}} \supseteq \widetilde{\mathcal{R}}^e$ instead [15].

A crucial aspect of reachability analysis is the choice of the set representation, e.g., polytopes or ellipsoids. In this paper, we represent all reachable sets $\widetilde{\mathcal{R}}$ by zonotopes, since the complexity of our zonotope-based reachability analysis for the augmented system in (4) is only $\mathcal{O}\big((n_x+n_u)^3\big)$, i.e., it scales polynomially with respect to the augmented state space dimension [15]. As a result, reachable sets for systems with several hundred state variables can be computed. Additionally, zonotopes can be stored efficiently as matrices. A zonotope $\widetilde{\mathcal{Z}} \subset \mathbb{R}^{n_x+n_u}$ is defined by

$$\widetilde{\mathcal{Z}} = \left\{ \tilde{x} \in \mathbb{R}^{n_x+n_u} \,\middle|\, \tilde{x} = \tilde{c} + \sum_{i=1}^{n_g} \alpha^{(i)} \tilde{g}^{(i)},\ \alpha^{(i)} \in [-1, 1] \right\},$$

where $\tilde{c} \in \mathbb{R}^{n_x+n_u}$ is the center, and $\tilde{g}^{(i)} \in \mathbb{R}^{n_x+n_u}$ is the i^{th} generator of $\widetilde{\mathcal{Z}}$. To obtain a more compact notation, we use the short form $\widetilde{\mathcal{Z}} = \big\langle \tilde{c}, \tilde{g}^{(1)}, \tilde{g}^{(2)}, \ldots, \tilde{g}^{(n_g)} \big\rangle$.

III. Verified Terminal Set

In this section, we define robust positively invariant (RPI) sets and verified terminal sets. Additionally, we present our scalable algorithm to compute verified terminal sets and introduce a distance between two sets.

A. Set Definitions

As stated in Sec. II-A, our goal is to steer the disturbed system to a neighborhood of the origin while satisfying the constraints in (3) for all times. A common way to ensure constraint satisfaction for an infinite time horizon is to reach a terminal set that is an RPI set. A set $\mathcal{S} \subseteq \mathcal{X}$ is an RPI set for the continuous-time system in (1) if $\Pi_x\big(\widetilde{\mathcal{R}}\big(t, \mathcal{S}, 0, \mathcal{W}\big)\big) \subseteq \mathcal{S}$ holds for all $t \geq t_0$ [20], [21].

A common procedure to compute polytopic RPI sets for a continuous-time system is based on constructing its corresponding discrete-time Euler auxiliary system [22, Def. 4.25]. As a result, methods for computing RPI sets of discrete-time systems can be exploited to compute RPI sets for the original continuous-time ones [22, Lemma 4.26]. However, these methods are unable to deal with high-dimensional systems due to the curse of dimensionality.

A zonotopic RPI set containing the origin can be determined by checking if the condition $\Pi_x\big(\widetilde{\mathcal{R}}\big(\tau_{k+1}, 0, 0, \mathcal{W}\big)\big) \subseteq \Pi_x\big(\widetilde{\mathcal{R}}\big(\tau_k, 0, 0, \mathcal{W}\big)\big)$ holds for some k. However, performing this enclosure check is computationally intractable for high-dimensional systems, since it requires transforming the zonotope $\Pi_x\big(\widetilde{\mathcal{R}}\big(\tau_k, 0, 0, \mathcal{W}\big)\big)$ to a polytope in halfspace-representation [23].

Instead of computing an RPI set, we propose a scalable method to obtain a verified terminal set. We call a set $\Omega \subseteq \mathcal{X}$ a verified terminal set for the disturbed system in (1) and the terminal controller $u(t) = Kx(t_k)$ for $t \in \tau_k$, if there exists some finite time $t_\Omega > t_0$ such that

$$\Pi_x\big(\widetilde{\mathcal{R}}(t_\Omega, \Omega, 0, \mathcal{W})\big) \subseteq \Omega \tag{5a}$$

$$\Pi_x\big(\widetilde{\mathcal{R}}(t, \Omega, 0, \mathcal{W})\big) \subseteq \mathcal{X} \qquad \text{for } t \in [t_0, t_\Omega) \tag{5b}$$

$$\Pi_u\big(\widetilde{\mathcal{R}}(t, \Omega, 0, \mathcal{W})\big) \subseteq \mathcal{U} \qquad \text{for } t \in [t_0, t_\Omega) \tag{5c}$$

holds. Hence, we allow the state trajectory $x(\cdot)$ to leave Ω for some time $t \in (t_0, t_\Omega)$ in contrast to RPI sets. Nevertheless, (5b) and (5c) ensure that the state and input constraints are satisfied for $t \in [t_0, t_\Omega)$ while employing the terminal controller. Thus, the constraints in (3) are fulfilled for an infinite time horizon after the state trajectory enters Ω at t_0.

B. Algorithm

Before we propose an algorithm for computing a verified terminal set Ω satisfying the conditions in (5), we introduce the distance

$$d(\mathcal{S}_1, \mathcal{S}_2) = \beta \tag{6}$$

between two sets $\mathcal{S}_1$ and $\mathcal{S}_2$, where $\beta \in \mathbb{R}_{\geq 0}$ is the smallest nonnegative value that satisfies $\mathcal{S}_1 \subseteq (1+\beta)\mathcal{S}_2$ [24]. Thus, $d(\mathcal{S}_1, \mathcal{S}_2) = 0$ if and only if $\mathcal{S}_1 \subseteq \mathcal{S}_2$.

We present Alg. 1 to efficiently compute a verified terminal set Ω for the case that Ω is an axis-aligned box, which significantly simplifies the enclosure check in (5a). To customize our scalable algorithm, we introduce $\beta_{\text{max}} \in \mathbb{R}_{>0}$ and $l_{\text{max}} \in \mathbb{R}_{>0}$ as the user-defined maximum distance and interval length, respectively. Moreover, the function $\mathsf{box}(\mathcal{S})$ returns the smallest axis-aligned box enclosing the set $\mathcal{S}$.

Algorithm 1 Verified terminal set

```
Input: $\mathcal{X}, \mathcal{U}, \mathcal{W}, \beta_{\text{max}}, l_{\text{max}}$
Output: $\Omega$
 1: $\mathcal{B}_{\text{min}} \leftarrow \mathsf{computeMinimalBox}(\mathcal{X}, \mathcal{W}, \beta_{\text{max}})$
 2: if $\mathsf{safeUntilEnclosed}(\mathcal{B}_{\text{min}}, \mathcal{X}, \mathcal{U}, \mathcal{W})$ then
 3:     $\mathcal{I} \leftarrow \big[1,\ 1 + d\big(\mathsf{box}(\mathcal{X}), \mathcal{B}_{\text{min}}\big)\big]$
 4:     while $l_{\text{max}} < \mathsf{length}(\mathcal{I})$ do
 5:         $\mathcal{B}_{\text{scaled}} \leftarrow \mathsf{center}(\mathcal{I})\, \mathcal{B}_{\text{min}}$
 6:         if $\mathsf{safeUntilEnclosed}(\mathcal{B}_{\text{scaled}}, \mathcal{X}, \mathcal{U}, \mathcal{W})$ then
 7:             $\mathcal{I} \leftarrow [\mathsf{center}(\mathcal{I}),\ \mathsf{max}(\mathcal{I})]$
 8:         else
 9:             $\mathcal{I} \leftarrow [\mathsf{min}(\mathcal{I}),\ \mathsf{center}(\mathcal{I})]$
10:         end if
11:     end while
12:     $\Omega \leftarrow \mathsf{min}(\mathcal{I})\, \mathcal{B}_{\text{min}}$
13: else
14:     $\Omega \leftarrow \{\}$
15: end if
```

The function $\mathsf{computeMinimalBox}$ is called in line 1 of Alg. 1 and is presented in Alg. 2. This function computes reachable sets for consecutive time intervals corresponding to the following two initial state sets: the origin and the state constraint set $\mathcal{X}$. These reachable sets are denoted by $\widetilde{\mathcal{S}}_0$ and $\widetilde{\mathcal{S}}_\mathcal{X}$. Intuitively, the sets $\Pi_x(\widetilde{\mathcal{S}}_0)$ and $\Pi_x(\widetilde{\mathcal{S}}_\mathcal{X})$ would converge to the minimal RPI set as $t \to \infty$ [21], if all eigenvalues of $A + BK$ had negative real part, $\Delta t = 0$, and reachable sets could be computed exactly. Since these assumptions cannot be fulfilled in practice, we use an easily computable termination criterion in line 4 of Alg. 2. If the distance between the box enclosures of $\Pi_x(\widetilde{\mathcal{S}}_\mathcal{X})$ and $\Pi_x(\widetilde{\mathcal{S}}_0)$ is smaller than β_{max}, we stop computing reachable sets for further time intervals. Finally, the minimal box $\mathcal{B}_{\text{min}}$, which is illustrated in Fig. 1, is obtained in line 9.

Algorithm 2 Function $\mathsf{computeMinimalBox}$

```
Input: $\mathcal{X}, \mathcal{W}, \beta_{\text{max}}$
Output: $\mathcal{B}_{\text{min}}$
1: $k \leftarrow 0$
2: $\widetilde{\mathcal{S}}_0 \leftarrow \widetilde{\mathcal{R}}(\tau_k, 0, 0, \mathcal{W})$
3: $\widetilde{\mathcal{S}}_\mathcal{X} \leftarrow \widetilde{\mathcal{R}}(\tau_k, \mathcal{X}, 0, \mathcal{W})$
4: while $\beta_{\text{max}} \leq d\Big(\mathsf{box}\big(\Pi_x(\widetilde{\mathcal{S}}_\mathcal{X})\big),\ \mathsf{box}\big(\Pi_x(\widetilde{\mathcal{S}}_0)\big)\Big)$ do
5:     $k \leftarrow k + 1$
6:     $\widetilde{\mathcal{S}}_0 \leftarrow \widetilde{\mathcal{R}}(\tau_k, 0, 0, \mathcal{W})$
7:     $\widetilde{\mathcal{S}}_\mathcal{X} \leftarrow \widetilde{\mathcal{R}}(\tau_k, \mathcal{X}, 0, \mathcal{W})$
8: end while
9: $\mathcal{B}_{\text{min}} \leftarrow (1 + \beta_{\text{max}})\, \mathsf{box}\big(\Pi_x(\widetilde{\mathcal{S}}_0)\big)$
```

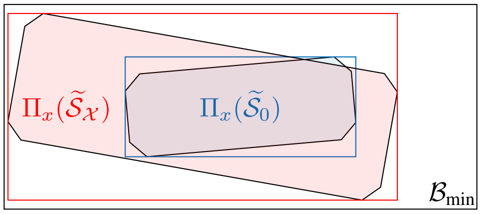

$\Pi_x(\widetilde{\mathcal{S}}_\mathcal{X})$
$\Pi_x(\widetilde{\mathcal{S}}_0)$
$\mathcal{B}_{\text{min}}$

Fig. 1. The minimal box $\mathcal{B}_{\text{min}}$, which is obtained by executing Alg. 2, is shown in black. The zonotopes $\Pi_x(\widetilde{\mathcal{S}}_0)$ and $\Pi_x(\widetilde{\mathcal{S}}_\mathcal{X})$ and their smallest axis-aligned box enclosures are shown in blue and red, respectively. It is assumed that the center of $\Pi_x(\widetilde{\mathcal{S}}_0)$ is the origin.

By calling the function $\mathsf{computeMinimalBox}$ in line 1 of Alg. 1, the box $\mathcal{B}_{\text{min}}$ is obtained. By construction, we know that $\Pi_x\big(\widetilde{\mathcal{R}}(\tau_k, \mathcal{X}, 0, \mathcal{W})\big) \subseteq \mathcal{B}_{\text{min}}$ holds for some finite k. Thus, if $\mathcal{B}_{\text{min}} \subseteq \mathcal{X}$ holds, the condition in (5a) is guaranteed to be satisfied for the verified terminal set candidate $\mathcal{B}_{\text{min}}$. To ensure (5b) and (5c) are also fulfilled, we use the function $\mathsf{safeUntilEnclosed}(\mathcal{B}, \mathcal{X}, \mathcal{U}, \mathcal{W})$. This function returns true for a box $\mathcal{B}$ if some $j \in \mathbb{Z}_{>0}$ exists such that

$$\Pi_x\big(\widetilde{\mathcal{R}}(t_j, \mathcal{B}, 0, \mathcal{W})\big) \subseteq \mathcal{B}$$

$$\Pi_x\big(\widetilde{\mathcal{R}}(\tau_k, \mathcal{B}, 0, \mathcal{W})\big) \subseteq \mathcal{X} \qquad \text{for } k \in \{0, 1, 2, \dots, j-1\}$$

$$\Pi_u\big(\widetilde{\mathcal{R}}(\tau_k, \mathcal{B}, 0, \mathcal{W})\big) \subseteq \mathcal{U} \qquad \text{for } k \in \{0, 1, 2, \dots, j-1\}$$

holds, otherwise it returns false. Thus, if the hard constraints in (3) are not satisfied for a reachable set of $\mathcal{B}_{\text{min}}$ in line 2 of Alg. 1, the procedure returns an empty terminal set in line 14. Otherwise, in line 3, we initialize an admissible one-dimensional interval $\mathcal{I}$ corresponding to scaling factors of $\mathcal{B}_{\min}$ such that $\mathsf{safeUntilEnclosed}(\min(\mathcal{I})\,\mathcal{B}_{\min}, \mathcal{X}, \mathcal{U}, \mathcal{W})$ is true whereas $\mathsf{safeUntilEnclosed}(\max(\mathcal{I})\,\mathcal{B}_{\min}, \mathcal{X}, \mathcal{U}, \mathcal{W})$ is, in general, false. The functions min, max, center, and length determine the minimum, maximum, center, and length of an interval, respectively. In lines 4 to 11, we perform a binary search to find an admissible scaling interval of maximum length $l_{\max}$ in order to increase the size of the verified terminal set. In line 12 of Alg. 1, the final verified terminal set Ω, which is guaranteed to satisfy the conditions in (5), is obtained.

IV. Robust Reachable-Set-Based MPC

In this section, we present our robust reachable-set-based dual-mode MPC approach considering a prediction horizon of $N \in \mathbb{Z}_{>0}$. Before we state the optimal control problem and propose our algorithm, we take the computation time into consideration and describe our constraint tightening procedure.

A. Computation Time Consideration

In order to guarantee safety of the system, i.e., satisfaction of the constraints in (3) for all times, we need to consider the nonzero computation time for solving the optimal control problem [17], [18]. The correction input trajectory that is optimized during the time interval τ_k is piecewise constant, subsequently denoted by $\bar{u}_k(\cdot)$, and illustrated in Fig. 2.

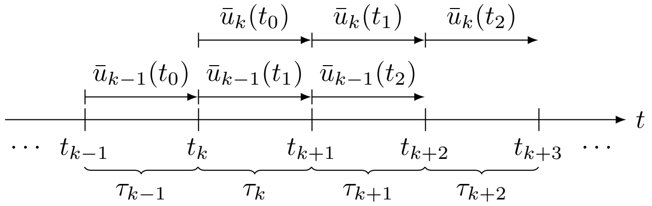


Fig. 2. Piecewise constant correction input trajectories $\bar{u}_{k-1}(\cdot)$ and $\bar{u}_k(\cdot)$ for the prediction horizon $N = 3$.

At t_k, the available computation time for the optimization of $\bar{u}_{k-1}(\cdot)$ is over. Then, we set

$$\bar{u}_k(t) = \bar{u}_{k-1}(t + t_1) \qquad \text{for } t \in [t_0, t_1), \tag{7}$$

and apply the input $\bar{u}_{k-1}(t_1) + Kx(t_k)$ to the system during τ_k while we optimize $\bar{u}_k(t)$ for $t \in [t_1, t_N)$. In case the optimization needs more time to finish than the length of one sampling time interval Δt, we abort it prematurely. If the obtained correction input trajectory $\bar{u}_k(\cdot)$ is not feasible, we use the solution of the previous optimization as backup, i.e., we set

$$\bar{u}_k(t) = \begin{cases} \bar{u}_{k-1}(t + t_1) & \text{for } t \in [t_0, t_{N-1}) \\ 0 & \text{for } t \in [t_{N-1}, t_N) \end{cases}. \tag{8}$$

This backup solution approach is valid, since we assume that an initial feasible solution for the optimal control problem exists, which is a common assumption in robust MPC.

B. Constraint Tightening

Since the system in (1) and the disturbance set $\mathcal{W}$ are time-invariant, the reachable set $\widetilde{\mathcal{R}}\big(t, 0, 0, \mathcal{W}\big)$ can be computed offline for any $t \in \mathbb{R}_{\geq 0}$. Thus, we can reduce the online computational effort by tightening the constraints in (3) based on the superposition principle [3], [4]. This allows us to ensure constraint satisfaction despite disturbances based on the disturbance-free system.

First, we introduce the Minkowski addition $\mathcal{S}_1 \oplus \mathcal{S}_2 = \{s_1 + s_2 \mid s_1 \in \mathcal{S}_1,\ s_2 \in \mathcal{S}_2\}$, and the Minkowski difference $\mathcal{S}_1 \ominus \mathcal{S}_2 = \{s_3 \mid s_3 \oplus \mathcal{S}_2 \subseteq \mathcal{S}_1\}$ of two sets $\mathcal{S}_1$ and $\mathcal{S}_2$. Then, the tightened state, input, and terminal constraint sets are

$$\bar{\mathcal{X}}(\tau_i) = \mathcal{X} \ominus \Pi_x\big(\widetilde{\mathcal{R}}(\tau_i, 0, 0, \mathcal{W})\big), \tag{9a}$$

$$\bar{\mathcal{U}}(\tau_i) = \mathcal{U} \ominus \Pi_u\big(\widetilde{\mathcal{R}}(\tau_i, 0, 0, \mathcal{W})\big), \tag{9b}$$

$$\bar{\Omega} = \frac{1}{1+\lambda}\Omega \ominus \Pi_x\big(\widetilde{\mathcal{R}}\big(t_N, 0, 0, \mathcal{W}\big)\big), \tag{9c}$$

where $i \in \{0, 1, 2, \ldots, N-1\}$ and $\lambda \in \mathbb{R}_{>0}$ is a user-defined contraction parameter. In (9c), before the verified terminal set Ω is tightened, it is shrank by $\frac{1}{1+\lambda}$. Based on this shrank set, convergence of the state trajectory to Ω in finite time can be guaranteed by introducing a contraction constraint [18], [24].

Assumption 2: The tightened state, input, and terminal constraint sets in (9) are nonempty. □

C. Optimal Control Problem

Before we state the optimal control problem, we define the contraction cost for time step t_k and $j \in \mathbb{Z}_{>0}$ by

$$J(k, j) = \sum_{i=1}^{j-1} d\bigg(\Pi_x\big(\widetilde{\mathcal{R}}\big(t_i, x(t_k), \bar{u}_k(\cdot), \mathcal{W}\big)\big),\ \frac{1}{1+\lambda}\Omega\bigg).$$

In general, for three nonempty sets $\widetilde{\mathcal{S}}_1$, $\widetilde{\mathcal{S}}_2$, and $\mathcal{S}_3$, the overall distance $d\big(\Pi_x(\widetilde{\mathcal{S}}_1) \oplus \Pi_x(\widetilde{\mathcal{S}}_2), \mathcal{S}_3\big)$ is unequal the sum $d\big(\Pi_x(\widetilde{\mathcal{S}}_1), \mathcal{S}_3\big) + d\big(\Pi_x(\widetilde{\mathcal{S}}_2), \mathcal{S}_3\big)$. Thus, the contraction cost cannot be easily decomposed based on the superposition principle in contrast to the constraint tightening in Sec. IV-B. Nevertheless, we can significantly reduce the online computational effort by computing the minimum and maximum values of $\Pi_x\big(\widetilde{\mathcal{R}}\big(t_i, 0, 0, \mathcal{W}\big)\big)$ for $i \in \{1, 2, \ldots, N-1\}$ in each dimension offline, i.e., by computing $\mathsf{box}\big(\Pi_x\big(\widetilde{\mathcal{R}}\big(t_i, 0, 0, \mathcal{W}\big)\big)\big)$.

The optimization problem solved during the time interval τ_k is

$$\min_{\bar{u}_k(\cdot)} \sum_{i=1}^{N-1} L\big(\bar{x}(t_i), \bar{u}_k(t_i)\big) + V\big(\bar{x}(t_N)\big), \tag{10}$$

where $\bar{x}(t) = \Pi_x\big(\widetilde{\mathcal{R}}\big(t, x(t_k), \bar{u}_k(\cdot), 0\big)\big)$, L denotes a positive definite stage cost, and V is a positive definite terminal cost, subject to (2), (7), and the tightened state, input, and terminal constraints

$$\Pi_x\big(\widetilde{\mathcal{R}}\big(\tau_i, x(t_k), \bar{u}_k(\cdot), 0\big)\big) \subseteq \bar{\mathcal{X}}(\tau_i), \tag{11a}$$

$$\Pi_u\big(\widetilde{\mathcal{R}}\big(\tau_i, x(t_k), \bar{u}_k(\cdot), 0\big)\big) \subseteq \bar{\mathcal{U}}(\tau_i), \tag{11b}$$

$$\Pi_x\big(\widetilde{\mathcal{R}}\big(t_N, x(t_k), \bar{u}_k(\cdot), 0\big)\big) \subseteq \bar{\Omega}, \tag{11c}$$

where $i \in \{0, 1, 2, \dots, N-1\}$. Additionally, the minimization in (10) is subject to the contraction constraint

$$J\big(k, N\big) - J\big(k-1, m_{k-1}\big) < -\lambda, \tag{12}$$

where λ is the contraction parameter used in (9c), and $m_k \in \mathbb{Z}_{>0}$ is the smallest positive integer that verifies $d\big(\Pi_x\big(\widetilde{\mathcal{R}}\big(t_{m_k}, x(t_k), \bar{u}_k(\cdot), \mathcal{W}\big)\big), \frac{1}{1+\lambda}\Omega\big) = 0$.

In order to use an off-the-shelf solver for the optimization problem (10), the set-based constraints need to be transformed into a system of inequalities. To determine if a zonotope $\widetilde{\mathcal{Z}} = \big\langle \tilde{c}, \tilde{g}^{(1)}, \tilde{g}^{(2)}, \dots, \tilde{g}^{(n_g)} \big\rangle$ is enclosed by a convex, closed, bounded polytope $\widetilde{\mathcal{P}} = \{\tilde{x} \in \mathbb{R}^{n_x+n_u} \mid \widetilde{H}\tilde{x} \leq \tilde{h}\}$, we have to check if

$$\widetilde{H}\tilde{c} + \sum_{i=1}^{n_g} \big|\widetilde{H}\tilde{g}^{(i)}\big| \leq \tilde{h}$$

is fulfilled [18], where the absolute value and the inequality are applied elementwise.

In (10), we evaluate $\bar{x}$ only at discrete time points, since we obtain these evaluations as a byproduct of our reachability analysis for the sets in (11) and (12) resulting in small computation times. Clearly, other cost functions are possible. In case the chosen cost function is convex, the optimal control problem is a convex optimization problem and can be solved efficiently by existing convex optimization algorithms [25].

D. Algorithm

Our robust reachable-set-based dual-mode MPC approach is presented in Alg. 3. As mentioned in Sec. IV-A, we assume that an initial feasible solution $\bar{u}_{-1}(\cdot)$ for the optimization problem (10) with $J\big(-2, m_{-2}\big) = \infty$ is given at t_0.

Algorithm 3 Robust reachable-set-based dual-mode MPC

```
Input: $\bar{u}_{k-1}(\cdot), x(t_k)$
Output: $\bar{u}_k(\cdot)$
 1: $u(t) \leftarrow \bar{u}_{k-1}(t_1) + Kx(t_k)$ for $t \in \tau_k$
 2: if $x(t_k) \in \Omega$ then
 3:     $\bar{u}_k(\cdot) \leftarrow 0$
 4: else
 5:     $\big(r_{\bar{u}_k}(\cdot),\ r_{\text{feasible}}\big) \leftarrow$ solve (10)
 6:     if $r_{\text{feasible}}$ then
 7:         $\bar{u}_k(\cdot) \leftarrow r_{\bar{u}_k}(\cdot)$
 8:     else
 9:         $\bar{u}_k(\cdot) \leftarrow$ apply $\bar{u}_{k-1}(\cdot)$ to (8)
10:     end if
11: end if
```

In line 1 of Alg. 3, the input u that is applied to the system during the time interval τ_k is updated based on Sec. IV-A. In line 2, we check if the measured state $x(t_k)$ lies inside the verified terminal set Ω. If this is the case, the stabilizing terminal controller from Sec. III is used, i.e., $\bar{u}_k(\cdot)$ is set to 0. Otherwise, the optimal control problem (10), which provides two return values, is solved until t_{k+1} in line 5. The first return value $r_{\bar{u}_k}(\cdot)$ corresponds to the optimized correction input trajectory and the second return value r_{feasible} is a Boolean flag that is true if the optimization problem was solved successfully, otherwise it is false. In case of an infeasible solution, we compute a correction input trajectory according to (8) in line 9 of Alg. 3. As mentioned in Sec. IV-A, the optimization is aborted prematurely when the computation time reaches Δt.

Proposition 1: If Assumptions 1 and 2 are fulfilled and there exists an initial feasible solution at t_0, then Alg. 3 steers the disturbed system in (1) to a neighborhood of the origin while minimizing the cost function in (10) and satisfying the constraints in (3) for all times. □

The proof follows the same line of thoughts as [18, Thm. 1] with a minor modification, which is explained subsequently. Based on [18], a sufficient condition for guaranteeing convergence of the state trajectory to the verified terminal set Ω is given by replacing N with m_k in (12). This leads to a mixed-integer optimization problem that is much harder to solve. Nevertheless, we know that $m_k \leq N$ holds due to the tightened terminal constraint in (11c). Since the distance introduced in (6) is nonnegative, it follows that $0 \leq J(k, m_k) \leq J(k, N)$ holds. Thus, convergence to the verified terminal set Ω is assured.

V. Numerical Example

In this section, we show the applicability of our approach using a vehicle platooning benchmark [26]. The dynamics corresponding to the relative motion of the i^{th} following vehicle with $i \in \mathbb{Z}_{>0}$ and its vehicle ahead are

$$\ddot{e}_i = a_{i-1} - a_i,$$

where the relative position error e_i denotes the difference between the two vehicles and a given safe reference distance. Moreover, a_i corresponds to the effective acceleration described by the drivetrain dynamics

$$\dot{a}_i = -\frac{1}{T_i}a_i + \frac{1}{T_i}u_i,$$

where u_i is a control input and T_i is a time constant, which is assumed to be 0.5 s for all i. The acceleration of the leading vehicle a_0 is considered as a disturbance. By assuming three following vehicles, the nonaugmented state of the platoon is given by $x = \begin{bmatrix} e_1 & \dot{e}_1 & a_1 & e_2 & \dot{e}_2 & a_2 & e_3 & \dot{e}_3 & a_3 \end{bmatrix}^T$, the control input is $u = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T$, and the disturbance is $w = \begin{bmatrix} 0 & a_0 & 0 & \dots & 0 \end{bmatrix}^T$.

Since vehicle-to-vehicle communication is assumed, a central controller can be designed. The used state feedback matrix K in (2a) is obtained by an LMI-based control approach [26]. We also want to mention that some sparsity exists in the system matrix that could be exploited to speed up the optimization, as done in distributed MPC [13], [14]. However, since we want to show the applicability of our approach to large systems, we ignore the underlying structure and consider the system as a black box.

We assume the acceleration of the leading vehicle a_0, which acts as a disturbance, to be guaranteed to lie within the interval $[-1, 1]\,\frac{\text{m}}{\text{s}^2}$. For all $i \in \{1, 2, 3\}$, we require the relative position error $e_i(\cdot)$ to be bounded by $[-10, 10]\,\text{m}$.

Thus, by setting the safe reference distance to $10\,\mathrm{m}$, the distance between two following vehicles is guaranteed to be within $[0, 20]\,\mathrm{m}$. As a result, compared to a nonplatooning scenario, the air resistance of the following vehicles is reduced and the road capacity is increased. In addition to the relative position error constraints, we restrict the velocity $\dot{e}_i$ and acceleration a_i of all following vehicles to be bounded by $[5, 5]\,\frac{\mathrm{m}}{\mathrm{s}}$ and $[-8, 8]\,\frac{\mathrm{m}}{\mathrm{s}^2}$, respectively. Additionally, the control inputs u_i are constrained to lie within $[-8, 8]\,\frac{\mathrm{m}}{\mathrm{s}^2}$ for all i.

In (10), we use the stage cost $L\big(\bar{x}, \bar{u}_k\big) = \bar{x}^T\bar{x} + 10\bar{u}_k^T\bar{u}_k$ and terminal cost $V(\bar{x}) = \bar{x}^T\bar{x}$. The prediction horizon is $N = 20$ and the sampling time is $\Delta t = 100\,\mathrm{ms}$. The maximum distance and interval length required for computing the verified terminal set Ω are $\beta_{\text{max}} = 10^{-3}$ and $l_{\text{max}} = 10^{-3}$, respectively. The contraction parameter used in (9c) is $\lambda = 0.2$ and $\bar{u}_{-1}(t_0)$ is 0. The initial state is $x(t_{-1}) = \begin{bmatrix}-7\,\mathrm{m} & 3\,\frac{\mathrm{m}}{\mathrm{s}} & 3\,\frac{\mathrm{m}}{\mathrm{s}^2} & 7\,\mathrm{m} & -4\,\frac{\mathrm{m}}{\mathrm{s}} & 4\,\frac{\mathrm{m}}{\mathrm{s}^2} & 1\,\mathrm{m} & 2\,\frac{\mathrm{m}}{\mathrm{s}} & 0\end{bmatrix}^T$.

The projections of sets and trajectories onto relative position and velocity error dimensions are shown in Fig. 3. The actual state, input, and disturbance trajectories over time, which correspond to the red trajectories in Fig. 3, are shown in Fig. 4. As illustrated in Fig. 4b, the additive disturbance a_0 is chosen randomly from the admissible set $[-1, 1]\,\frac{\mathrm{m}}{\mathrm{s}^2}$. Although a_0 is assumed piecewise constant in this simulation, our reachability analysis takes all possible realizations within a time interval into account.

Our robust reachable-set-based MPC method is integrated into our MATLAB reachability tool CORA [27]. The convex optimization problem (10) is modeled by YALMIP [28] with parameter 'allownonconvex' set to 0, and solved by MOSEK [29] with default parameters. All computations are run on a laptop with an Intel Core i7-7820HQ and $32\,\mathrm{GB}$ memory. The time needed to compute the verified terminal set Ω by executing Alg. 1 is $0.7\,\mathrm{s}$ (offline), and the maximum computation time for a single optimization is $84\,\mathrm{ms}$ (online). Since it is smaller than the sample time Δt, we never have to abort the optimization prematurely.

VI. Conclusions and Future Work

We have proposed a scalable robust reachable-set-based MPC approach for constrained linear sampled-data systems. By taking the computation time into consideration and using verified terminal sets, we ensure constraint satisfaction for an infinite time horizon while steering the system to a neighborhood of the origin. Moreover, we have shown that the online computations of our approach can be performed in real-time. In the future, we plan to include state measurement uncertainty by means of set-based observers. Additionally, we want to apply our robust MPC approach to even higher-dimensional problems.

Acknowledgments

We gratefully acknowledge partial financial support by the European Research Council (ERC) project justITSELF under grant agreement No 817629 and the project interACT under grant agreement No 723395; both projects are part of the EU Horizon 2020 program.

References

[1] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[2] M. Lazar, D. Muñoz de la Peña, W. P. M. H. Heemels, and T. Alamo, "On input-to-state stability of minmax nonlinear model predictive control," *Systems & Control Letters*, vol. 57, no. 1, pp. 39–48, 2008.

[3] L. Chisci, J. A. Rossiter, and G. Zappa, "Systems with persistent disturbances: Predictive control with restricted constraints," *Automatica*, vol. 37, no. 7, pp. 1019–1028, 2001.

[4] A. Richards and J. How, "Robust stable model predictive control with constraint tightening," in *American Control Conference*, 2006, pp. 1557–1562.

[5] D. Q. Mayne, M. M. Seron, and S. V. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, no. 2, pp. 219–224, 2005.

[6] S. V. Raković, B. Kouvaritakis, M. Cannon, C. Panos, and R. Findeisen, "Parameterized tube model predictive control," *IEEE Transactions on Automatic Control*, vol. 57, no. 11, pp. 2746–2761, 2012.

[7] S. V. Raković, W. S. Levine, and B. Açikmese, "Elastic tube model predictive control," in *American Control Conference*, 2016, pp. 3594–3599.

[8] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones, "On real-time robust model predictive control," *Automatica*, vol. 50, no. 3, pp. 683–694, 2014.

[9] R. Gonzalez, M. Fiacchini, T. Alamo, J. L. Guzman, and F. Rodríguez, "Online robust tube-based MPC for time-varying systems: A practical approach," *International Journal of Control*, vol. 84, no. 6, pp. 1157–1170, 2011.

[10] M. Farina and R. Scattolini, "Tube-based robust sampled-data MPC for linear continuous-time systems," *Automatica*, vol. 48, no. 7, pp. 1473–1476, 2012.

[11] F. A. C. C. Fontes, S. V. Raković, and I. V. Kolmanovsky, "Rigid tube model predictive control for linear sampled-data systems," in *IFAC World Congress*, 2017, pp. 9840–9845.

[12] I. Alvarado, D. Limon, D. Muñoz de la Peña, T. Alamo, and E. F. Camacho, "Enhanced ISS nominal MPC based on constraint tightening for constrained linear systems," in *UKACC International Conference on Control*, 2010, pp. 1–6.

[13] P. D. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers and Chemical Engineering*, vol. 51, pp. 21–41, 2013.

[14] J. M. Maestre and R. R. Negenborn, *Distributed model predictive control made easy*, 1st ed., ser. Intelligent Systems, Control and Automation: Science and Engineering. Springer, 2014, vol. 69.

[15] M. Althoff, "Reachability analysis and its application to the safety assessment of autonomous cars," Ph.D. dissertation, Technical University of Munich, 2010.

[16] I. M. Mitchell, J. Yeh, F. J. Laine, and C. J. Tomlin, "Ensuring safety for sampled data systems: An efficient algorithm for filtering potentially unsafe input signals," in *IEEE Conference on Decision and Control*, 2016, pp. 7431–7438.

[17] V. M. Zavala and L. T. Biegler, "The advanced-step NMPC controller: Optimality, stability and robustness," *Automatica*, vol. 45, no. 1, pp. 86–93, 2009.

[18] B. Schürmann, N. Kochdumper, and M. Althoff, "Reachset model predictive control for disturbed nonlinear systems," in *IEEE Conference on Decision and Control*, 2018, pp. 3463–3470.

[19] G. Lafferriere, G. J. Pappas, and S. Yovine, "Symbolic reachability computation for families of linear vector fields," *Journal of Symbolic Computation*, vol. 32, no. 3, pp. 231–253, 2001.

[20] S. Tarbouriech and C. Burgat, "Positively invariant sets for constrained continuous-time systems with cone properties," *IEEE Transactions on Automatic Control*, vol. 39, no. 2, pp. 401–405, 1994.

[21] S. V. Raković and K. I. Kouramas, "Invariant approximations of the minimal robust positively invariant set via finite time Aumann integrals," in *IEEE Conference on Decision and Control*, 2007, pp. 194–199.

[22] F. Blanchini and S. Miani, *Set-theoretic methods in control*, 2nd ed. Birkhäuser, 2015.

[23] M. Althoff, O. Stursberg, and M. Buss, "Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes," *Nonlinear Analysis: Hybrid Systems*, vol. 4, no. 2, pp. 233–249, 2010.

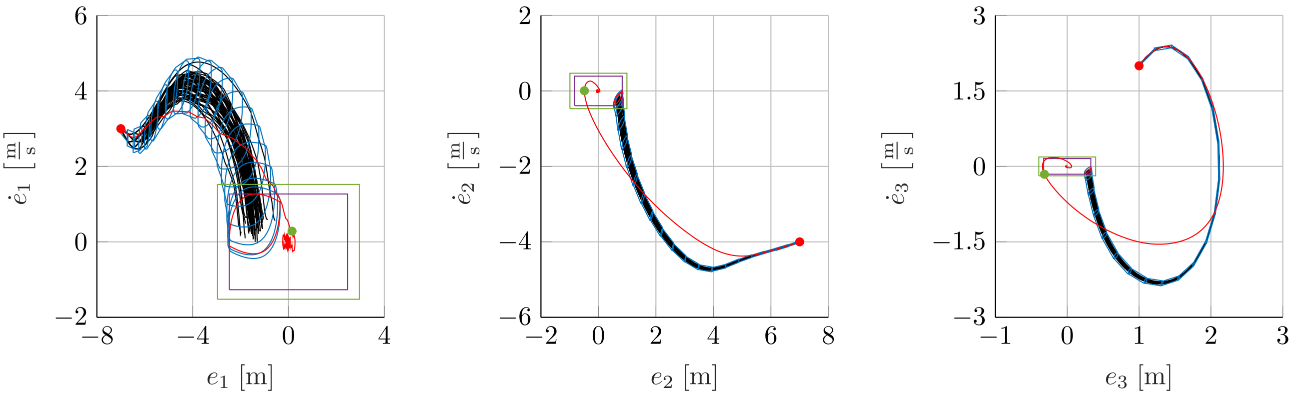
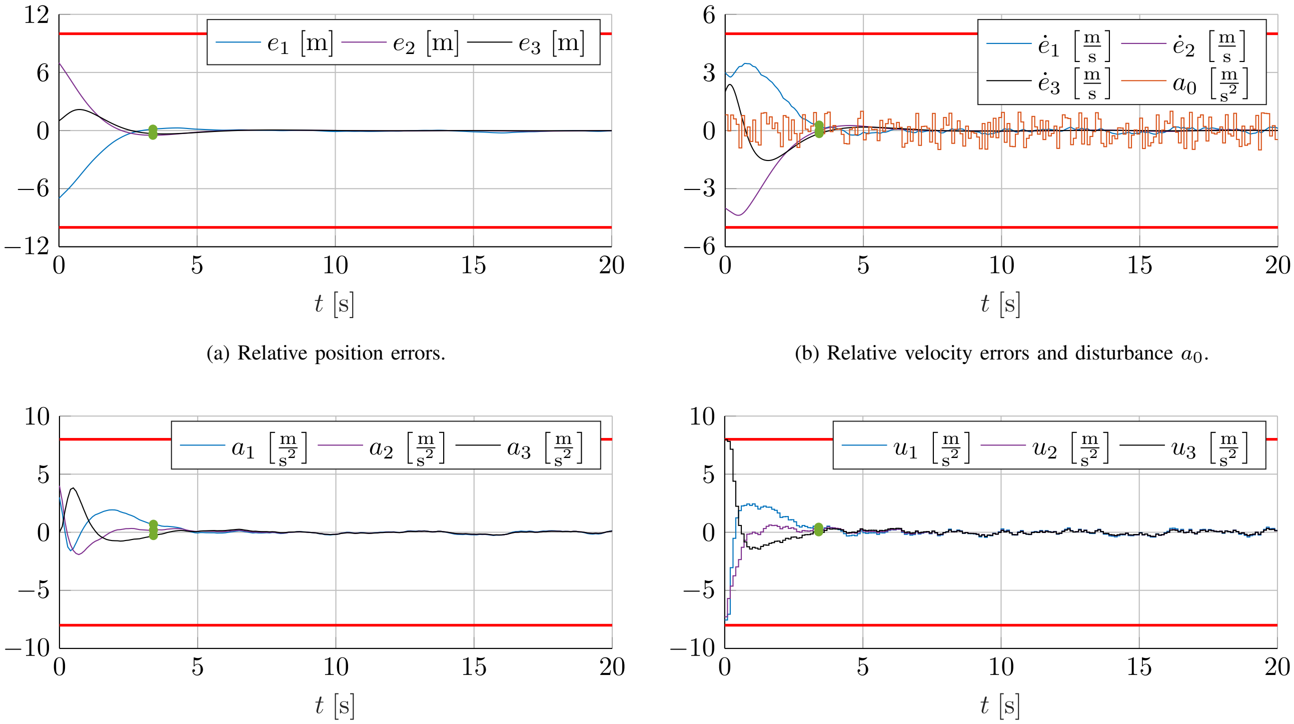


Fig. 3. Projections of sets and trajectories onto relative position and velocity error dimensions. The red and green dots correspond to $x(t_{-1})$ and the first time the state trajectory enters Ω, respectively. Ω and $\frac{1}{1+\lambda}\Omega$ are shown as green and purple rectangles, respectively. The initial reachable sets for time intervals corresponding to $\bar{u}_{-1}(\cdot)$ are shown in blue and for the last prediction time point in red. 50 random trajectories for the initial solution are shown in black, and the actual trajectories for $t \in [0, 20]$ s are illustrated in red.



(a) Relative position errors.

(b) Relative velocity errors and disturbance a_0.

(c) Effective accelerations.

(d) Control inputs.

Fig. 4. State, input, and disturbance trajectories, which correspond to the red trajectories in Fig. 3. The green dots correspond to the first time the state trajectory enters Ω, and the red lines represent the state and input bounds, respectively.

[24] J. M. Bravo, T. Alamo, and E. F. Camacho, "Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets," *Automatica*, vol. 42, no. 10, pp. 1745–1751, 2006.

[25] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[26] I. Ben Makhlouf and S. Kowalewski, "Networked cooperative platoon of vehicles for testing methods and verification tools," in *Workshop on Applied Verification for Continuous and Hybrid Systems*, 2014, pp. 37–42.

[27] M. Althoff, "An introduction to CORA 2015," in *Workshop on Applied Verification for Continuous and Hybrid Systems*, 2015, pp. 120–151.

[28] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *IEEE International Symposium on Computer Aided Control Systems Design*, 2004, pp. 284–289.

[29] MOSEK Aps, "The MOSEK optimization toolbox for MATLAB manual. Version 8.1," 2019. [Online]. Available: https://docs.mosek.com/8.1/toolbox/index.html