Deterministic Safety Guarantees for Learning-Based Control of Monotone Nonlinear Systems Under Uncertainty

Joshua Adamek, Moritz Heinlein, Lukas Lüken, and Sergio Lucia, *Member, IEEE*

***Abstract*—This letter presents a novel framework to guarantee safety for learning-based control of nonlinear monotone systems under uncertainty. We propose to evaluate online whether a one-step simulation brings a nonlinear system into a robust control invariant (RCI) set. Such evaluation can be very efficiently computed even under the presence of uncertainty for learning-based approximate controllers and monotone systems, which also enable a simple computation of RCI sets. In case the one-step simulation drives the system outside of the RCI set, a fallback strategy is used, which is obtained as a byproduct of the RCI set computation. We also develop a method to calculate an *N*-step RCI set to reduce the conservativeness of the proposed strategy and we illustrate the results with a simulation study of a nonlinear monotone system.**

***Index Terms*—Optimal control, robust control, machine learning.**

I. Introduction

LEARNING-BASED control strategies can provide important advantages when compared to traditional control techniques, for example by using data-based models in model predictive control (MPC) formulations [1], [2] or by approximating complex MPC approaches via simple neural networks [3], [4]. Approximate MPC controllers based on neural networks can lead to significantly faster controller evaluation times compared to standard MPC approaches as well as to potentially improve the closed-loop performance [5]. While there exist probabilistic validation methods for learning-based controllers [6], it can be challenging to apply these controllers in safety-critical nonlinear systems that require deterministic safety guarantees, especially in the presence of approximation errors or uncertainties about the system dynamics. So far, deterministic guarantees under bounded system uncertainties exist for linear systems [7].

In contrast, online optimization-based robust MPC methods can guarantee safety under bounded uncertainties also for nonlinear systems [8]. The drawback of these methods is an increased computational effort, which often becomes impractical for larger systems, thus increasing the appeal of employing learning-based controllers.

An additional advantage of employing a learning-based controller is that the resulting closed-loop state and input trajectories can often be easily simulated online, as it is typically computationally very cheap [9]. As a result, it is possible to detect unwanted closed-loop behavior due to approximation errors or other uncertainties by forward simulation. The remaining problem is the design of a suitable fallback strategy that guarantees safety in case an unwanted behavior is detected. Several approaches deal with the design of such a fallback strategy for nonlinear systems with no uncertainties. These approaches are often referred to as safety filters and an overview of existing methods for nonlinear systems can be found in [2].

The design of a fallback strategy for uncertain nonlinear systems is even more difficult, as one has to ensure safety for all possible uncertainty realizations. This problem is very complex in general and in this letter, we focus on the sub-class of nonlinear monotone systems to efficiently compute safety filters in the uncertain case. Monotone systems arise in different applications ranging from building ventilation [10] to biochemical systems [11]. A large linear monotone system can be found in [12]. In [12], the authors proposed an optimization-based method to calculate a one-step hyperrectangular robust control invariant (RCI) set for monotone systems. This method computes piece-wise constant policies that ensure invariance with respect to the RCI set for any uncertainty realization. These policies are used to design a robust MPC strategy with deterministic guarantees under uncertainty.

This letter extends the approach of [12] to compute N-step robust control invariant sets that overcome some of the conservatism of the one-step hyperrectangular RCI set. In addition, and as the main contribution of this letter, we propose to use the resulting invariance-ensuring policies as a fallback strategy in case a possibly unsafe control action of the learning-based

Manuscript received 8 March 2024; revised 4 May 2024; accepted 25 May 2024. Date of publication 30 May 2024; date of current version 17 June 2024. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Project 423857295. Recommended by Senior Editor J. Daafouz. *(Corresponding author: Joshua Adamek.)*

The authors are with the Department of Biochemical and Chemical Engineering, TU Dortmund University, 44227 Dortmund, Germany (e-mail: joshua.adamek@tu-dortmund.de; moritz.heinlein@tu-dortmund.de; lukas.lueken@tu-dortmund.de; sergio.lucia@tu-dortmund.de).

Digital Object Identifier 10.1109/LCSYS.2024.3407635

controller is predicted. Under this framework, we prove that learning-based controllers operating within the computed RCI set leads to a safe closed-loop behavior for nonlinear monotone systems with bounded uncertainty.

This letter is structured as follows: Section II introduces the concept of monotonicity. In Section III, we firstly recap the optimization problem as proposed in [12] to calculate the one-step RCI set, while building on that to propose the new calculation of an enlarged N-step RCI set. Section IV introduces the proposed safety filter as a new control algorithm while proving the deterministic safety of the method under uncertainty. In Section V, the proposed algorithm is applied to a two-dimensional case study.

II. Reachability of Monotone Systems

We consider nonlinear discrete-time systems of the form

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

where $x \in \mathbb{R}^{n_x}$ denotes the states, $u \in \mathbb{R}^{n_u}$ denotes the inputs and $p \in \mathbb{P} \in \mathbb{R}^{n_p}$ represents uncertain parameters assumed to be in a compact set $\mathbb{P}$ with discrete time index k. The system dynamics $f{:}\mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \mapsto \mathbb{R}^{n_x}$ is assumed to be continuous and differentiable $\forall x \in \mathbb{X}, \forall u \in \mathbb{U}, \forall p \in \mathbb{P}$, where $\mathbb{X}$ and $\mathbb{U}$ denote the compact state and input spaces.

Definition 1 (Monotone Dynamical Systems): A system is called monotone on the sets $\mathbb{X} \in \mathbb{R}^{n_x}$, $\mathbb{U} \in \mathbb{R}^{n_u}$, $\mathbb{P} \in \mathbb{R}^{n_p}$ with respect to the states x (the uncertainties p) if for every pair $\hat{x}$ and $\tilde{x}$ in $\mathbb{X}$ ($\hat{p}$ and $\tilde{p}$ in $\mathbb{P}$) that satisfies the condition $\hat{x} \geq \tilde{x}$ ($\hat{p} \geq \tilde{p}$), the following inequality holds:

$$f\big(\hat{x}, u, p\big) \geq f(\tilde{x}, u, p), \quad \forall u \in \mathbb{U}, \ \forall p \in \mathbb{P} \tag{2a}$$

$$f\big(x, u, \hat{p}\big) \geq f(x, u, \tilde{p}), \quad \forall u \in \mathbb{U}, \forall x \in \mathbb{X}, \tag{2b}$$

where the inequalities are understood elementwise.

The monotonicity conditions in Definition 1 are satisfied if all the elements of the Jacobian of the dynamics with respect to the states (the uncertainties) are non-negative $\forall u \in \mathbb{U}, \forall x \in \mathbb{X}, \forall p \in \mathbb{P}$ [13].

Remark 1: Monotone systems, appear for example in systems modeling the temperature in buildings [10], [12] or biochemical reaction cascades [11], [13] as a consequence of conservation laws. Through a state transformation, some non-monotone systems can be made monotone [11], [14]. Often, non-monotone systems also exhibit partial monotone dynamics, which can be exploited to decompose the system into monotonically increasing and decreasing parts via mixed-monotonicity [15].

Monotonicity enables a direct computation of tight hyperrectangular outer approximations of reachable sets, as stated in the following Proposition. The term hyperrectangle is used to describe the multidimensional interval spanned by two points, the bottom left and top right corners of the hyperrectangle. In the rest of this letter, hyperrectangular sets $\{x | a \leq x \leq b\}$ are denoted as $[a, b]$.

Proposition 1: The one-step reachable set for any fixed input $u \in \mathbb{U}$ of the discrete monotone dynamic system (1) with $x \in [x^-, x^+]$ and $p \in [p^-, p^+]$ is bounded by the multidimensional interval

$$f(x, u, p) \in \big[f\big(x^-, u, p^-\big), f\big(x^+, u, p^+\big)\big]. \tag{3}$$

Proposition 1 can be proven directly by applying (2a) and (2b). Because the top right and bottom left corners of the set in (3) are the corners of the true reachable set, it is the tightest hyperrectangular outer approximation of the reachable set. The repeated application of Proposition 1 can be used to calculate an N-step reachable set, which is the set that can be reached by the system after propagating the dynamics for all possible values of the uncertainty during N-steps with a given sequence of control inputs.

III. Robust Control Invariant Sets for Monotone Systems

The straightforward computation of reachable sets for monotone systems enables the calculation of robust control invariant sets [10], [12].

Definition 2 (Robust Control Invariant Set): A set $\mathbb{X}^{\text{RCI}}$ is robust control invariant if it holds that $\forall x \in \mathbb{X}^{\text{RCI}} \, \exists \ u \in \mathbb{U}$, such that $f(x, u, p) \in \mathbb{X}^{\text{RCI}}, \ \forall p \in \mathbb{P}$.

In the scope of this letter, monotonicity and bounded uncertain parameters are required as specified in the following Assumption to enable the scalable computation of reachable sets for large state and uncertainty dimensions.

Assumption 1: The dynamic system (1) is monotone in the states and the uncertainties, so (2a) and (2b) hold. Additionally, the uncertainties are assumed to take values only within the interval $\mathbb{P} \subseteq [p^-, p^+]$.

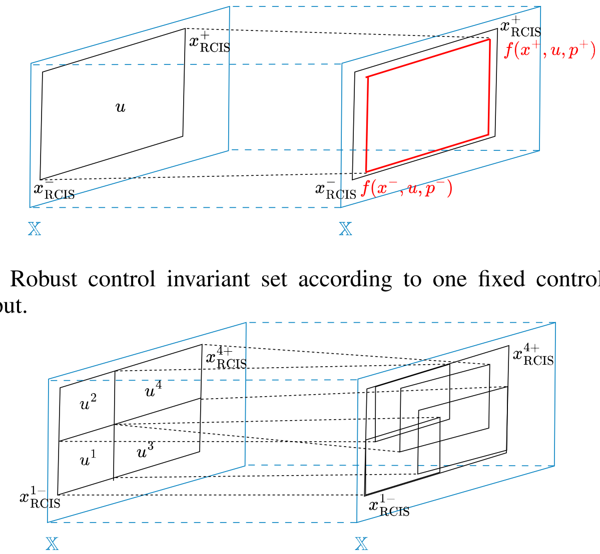
For systems satisfying Assumption 1, it is possible to compute an RCI set as defined in Definition 2 by finding a hyperrectangular RCI set $\mathbb{X}^{\text{RCI}} = [x^-_{\text{RCI}}, x^+_{\text{RCI}}]$ for which for a fixed control input u_{RCI} it holds that:

$$x^-_{\text{RCI}} \leq f\big(x^-, u_{\text{RCI}}, p^-\big) \leq f\big(x^+, u_{\text{RCI}}, p^+\big) \leq x^+_{\text{RCI}}. \tag{4}$$

However, in reality, constraint (4) is often infeasible, as there does not exist one single u_{RCI} ensuring invariance for both extreme realizations of the uncertainty. To enable different control inputs for different states and hence relax said constraint, a feedback policy is necessary. For locally controllable systems that are additionally monotone in u, [10] developed an interpolating control policy to ensure invariance of a hyperrectangular RCI set. In [12], the hyperrectangular RCI set is divided into N_s smaller hyperrectangles to improve performance without the need for monotonicity in u. Each of the smaller hyperrectangles, described by the top right x^{s+} and bottom left x^{s-} corners have one associated control input u^s, resulting in a piece-wise constant control policy that can be defined as:

$$\phi(x) = u^s, \ \text{ if } x \in \big[x^{s-}, \ x^{s+}\big], \forall s \in \mathbb{N}_{N_s}. \tag{5}$$

The RCI set requirement is fulfilled if the reachable sets of all smaller hyperrectangles lie inside the hyperrectangular RCI set. Figure 1 visualizes the additional flexibility achieved by the piece-wise constant control policy on smaller subsets.



(a) Robust control invariant set according to one fixed control input.

(b) Robust control invariant set according to hyperrectangular division.

Fig. 1. Visualization of the assumptions on the set $\mathbb{X}^{\text{RCI}}$. In the bottom plot, the black rectangle is divided and each subregion is propagated with an individual input. The propagations need to lie inside $\mathbb{X}^{\text{RCI}}$.

We denote with N_s the number of smaller hyperrectangles, $\mathbb{X}^{\text{RCI}}$ is divided into, and with $\mathbb{N}_{N_s} = \{1, \dots, N_s\}$ the set of all integers from 1 to N_s. The optimization problem formulated in [12] to calculate such a hyperrectangular RCI set is:

$$\max_{x^{s+}, x^{s-}, u^s,\ \forall s \in \mathbb{N}_{N_s}} \quad V\left(x^{[1:N_s]\pm}\right) \tag{6a}$$

$$\text{s.t:} \quad \left[x^{1-}, x^{N_s+}\right] \in \mathbb{X}, \tag{6b}$$

$$u^s \in \mathbb{U}, \quad \forall s \in \mathbb{N}_{N_s}, \tag{6c}$$

$$h\left(x^{[1:N_s]\pm}\right) \leq 0, \tag{6d}$$

$$x^{N_s+} \geq f\left(x^{s\pm}, u^s, p^{\pm}\right) \geq x^{1-}, \quad \forall s \in \mathbb{N}_{N_s}. \tag{6e}$$

The function $V(x^{[1:N_s]\pm})$ is an arbitrary measure for the size of the RCI set. Here, we consider the maximization of the set volume

$$V\left(x^{[1:N_s]\pm}\right) = \prod_{i=1}^{n_x} \left(x_i^{N_s+} - x_i^{1-}\right). \tag{7}$$

The state constraint satisfaction (6b) can be checked easily for box constraints, as well as polyhedral constraints via Farkas Lemma [16]. The constraint (6d) requires that the smaller hyperrectangles fill the RCI set spanned by $[x^{1-}, x^{N_s+}]$ without holes and (6e) enforces the invariance of the set.

The requirement of a hyperrectangular RCI set results in unnecessarily conservative sets, as often the RCI set can be described as an ellipsoid or a polytope [16]. To alleviate this problem, we propose in this letter the computation of N-step robust control invariant sets, which are defined in the following.

Definition 3 (N-Step Robust Control Invariant Set): A set $\mathbb{X}^{N\text{-RCI}}$ is N-step robust control invariant if it holds that for each state $x_1 \in \mathbb{X}^{N\text{-RCI}}$ there exists a sequence of control policies $\phi_i(x_i) \in \mathbb{U}$ such that the N-step reachable set of $\mathbb{X}^{N\text{-RCI}}$ lies within $\mathbb{X}^{N\text{-RCI}}$ $\forall p_i \in \mathbb{P}$, with $i = 1, \dots, N$.

To check for the above condition, we propose N hyperrectangles $[x_i^{1-},\ x_i^{N_s+}] = \mathbb{X}_i^{N\text{-RCI}}$, which each are propagated with a piece-wise constant policy

$$\phi_i(x_i) = u_i^s, \text{ if } x_i \in \left[x_i^{s-},\ x_i^{s+}\right], \forall s \in \mathbb{N}_{N_s}, \ \forall i \in \mathbb{N}_N. \tag{8}$$

We enforce that the reachable set of each hyperrectangle is bounded by the next hyperrectangle. The N-step robust control invariance is enforced by constraining the propagation of the last hyperrectangle to lie again in the first hyperrectangle. All hyperrectangles should satisfy the state constraints.

We propose to compute the N-step robust reachable sets by solving the following optimization problem:

$$\max_{x_i^{s+}, x_i^{s-}, u_i^s,\ \forall s \in \mathbb{N}_{N_s}, \forall i \in \mathbb{N}_N} \quad V\left(x_{[1:N]}^{[1:N_s]\pm}\right) \tag{9a}$$

$$\text{s.t:} \quad \left[x_i^{1-}, x_i^{N_s+}\right] \in \mathbb{X}, \quad \forall i \in \mathbb{N}_N, \tag{9b}$$

$$u_i^s \in \mathbb{U}, \quad \forall s \in \mathbb{N}_{N_s}, \forall i \in \mathbb{N}_N, \tag{9c}$$

$$h\left(x_i^{[1:N_s]\pm}\right) \leq 0, \forall i \in \mathbb{N}_N \tag{9d}$$

$$x_{i+1}^{N_s+} \geq f\left(x_i^{s\pm}, u_i^s, p^{\pm}\right) \geq x_{i+1}^{1-}, \quad \forall s \in \mathbb{N}_{N_s}, \quad \forall i \in \mathbb{N}_{N-1}, \tag{9e}$$

$$x_1^{N_s+} \geq f\left(x_N^{s\pm}, u_N^s, p^{\pm}\right) \geq x_1^{1-}, \quad \forall s \in \mathbb{N}_{N_s}. \tag{9f}$$

We denote the union of all $\mathbb{X}_i^{N\text{-RCI}}$ as $\mathbb{X}_{[1:N]}^{N\text{-RCI}}$.

Proposition 2: Suppose Assumption 1 holds, then the hyperrectangles $\mathbb{X}_i^{N\text{-RCI}}$, resulting from (9), are all N-step robust control invariant according to Definition 3.

Proposition 2 can be proven by extending the proof from [12] over multiple time steps and is omitted here for brevity.

If the cost function (9a) is chosen as $V(x_1^{[1:N_s]\pm})$ in (7), this will lead to a hyperrectangle $\mathbb{X}_1^{N\text{-RCI}}$ with at least the size of the one-step RCI set resulting from (6). The number of hyperrectangles N_s can be chosen as a trade-off between complexity and conservatism. The proposed approach enables to calculate N-step RCI sets for monotone systems with a complexity scaling linearly in N and N_s. Each hyperrectangle can be defined over its corner points $x_i^{s\pm}$ and its safe input u_i^s (linear in N_s). The propagation of every hyperrectangle to the next RCI set is again defined with two points $f(x_i^{s+}, u_i^s, p^+)$ and $f(x_i^{s-}, u_i^s, p^-)$ and are bounded by the next RCI set, which again is defined by a constant N_s hyperrectangles (linear in N).

IV. Robust Control Invariant Set as a Safety Filter for Monotone Systems

We propose to use the N-step RCI set defined by (9) for a safety filter that guarantees a safe closed-loop behavior inside the N-step RCI set when any control law $\Pi(x_k)$ is used for the system. We will use $\Pi(x_k)$ as notation for the control law throughout the following Section for simplicity, without loss of generality for any other possible control law Π, especially $\Pi(x_k, u_{k-1})$, which is a typical control law for tracking problems with desired input smoothing as seen in Section V. The proposed strategy checks online whether the control law $\Pi(x_k)$ leads the system into the N-step RCI set,

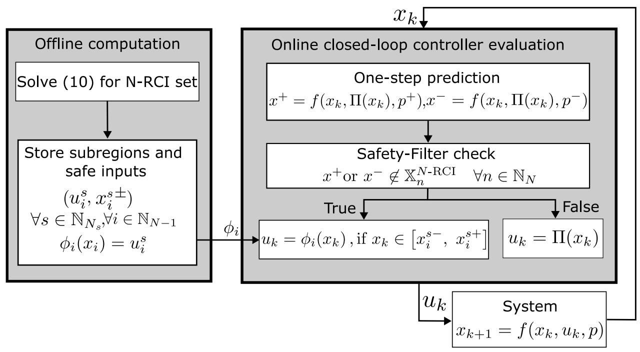


Fig. 2. Algorithmic overview of the proposed control law $\kappa(x_k, \Pi)$ The offline computation of the safe fallback inputs as well as the respective subregions is used in the online application as a safe alternative input in case one of the predicted x^+ or x^- are leaving the N-step RCI set.

which has been precomputed offline. this is the case, it is applied to the system. If this is not the case, a safe fallback strategy has to be chosen instead. As a direct result of (2b), for a monotone system under uncertainty it is sufficient to check whether for the current state x_k, $x^- = f(x_k, \Pi(x_k), p^-)$ and $x^+ = f(x_k, \Pi(x_k), p^+)$ are within the N-step RCI set. Since this check is computationally very simple, it can be easily performed online, by calculating the error value e_k as

$$e_k = \begin{cases} 1 \text{ if } x^+ \text{or } x^- \notin \mathbb{X}_n^{N\text{-RCI}} & \forall n \in \mathbb{N}_N \\ 0 \text{ else.} \end{cases} \tag{10}$$

Note that while $\mathbb{X}_{[1:N]}^{N\text{-RCI}}$ is the union of the N hyperrectangles, there has to be one rectangle $\mathbb{X}_n^{N\text{-RCI}}$, $n \in \mathbb{N}_N$ that includes both x^+ and x^-.

Solving (9) generates inputs as defined by (8) that ensure the N-step invariance of each point within the N-step RCI set. This enables the use of a straightforward fallback strategy

$$u^f(x_k) = \phi_i(x_k) \quad \text{if } x_k \in \left[x_i^{s-},\ x_i^{s+}\right] \tag{11}$$

that is used in case the check in (10) fails. In general, as there can be more than one hyperrectangle that includes x_k, $u^f(x_k)$ is non-unique. In those cases the first found hyperrectangle that fulfills said condition is chosen for $u^f(x_k)$, as this concludes the search as early as possible. Other variants like choosing the input that is the closest to $\Pi(x_k)$ are possible. This framework is presented in Figure 2. With this, the main Theorem can be stated as follows:

Theorem 1: If Assumption 1 holds, then for any controller $\Pi(x_k)$ and the fallback strategy (11), the control strategy

$$\kappa(x_k, \Pi) = \begin{cases} u^f(x_k), \text{ if } e_k = 1 \\ \Pi(x_k), \text{ if } e_k = 0 \end{cases} \tag{12}$$

makes the set $\mathbb{X}_{[1:N]}^{N\text{-RCI}}$ computed as in (9) a robust control invariant set and therefore guarantees safe operation of the controller.

Proof: The proof is conducted using case distinction:

Case 1: $\exists n \quad x^-$ and $x^+ \in \mathbb{X}_n^{N\text{-RCI}} \rightarrow e_k = 0$.

A rectangular set $\mathbb{X}_n^{N\text{-RCI}}$ that is part of the N-step RCI set is found such that x^+ and x^- are within this hyperrectangle. Applying Proposition (1), shows that

$$x_{k+1} = f(x_k, u_k, p) \in \left[x^-,\ x^+\right] \tag{13}$$

for an input $u_k = \Pi(x_k)$ and any $p \in [p^-, p^+]$. Therefore, by Assumption 1 there exists an n, such that $x_{k+1} \in \mathbb{X}_n^{N\text{-RCI}} \subset \mathbb{X}_{[1:N]}^{N\text{-RCI}}$ must hold.

Case 2: $\forall n \quad x^-$ or $x^+ \notin \mathbb{X}_n^{N\text{-RCI}} \rightarrow e_k = 1$

There is no rectangular set $\mathbb{X}_n^{N\text{-RCI}}$ that is part of the RCI set such that x^+ and x^- are within this hyperrectangle for the input $u_k = \Pi(x_k)$. Therefore, the control law will switch to the fallback law $u_k = u^f(x_k) = \phi_i(x_k)$. Per definition in (8), $\phi_i(x_k)$ is the input policy in x_k that guarantees that $x_{k+1} \in \mathbb{X}_{[1:N]}^{N\text{-RCI}} \quad \forall p \in [p^-, p^+]$.

In both cases, x_{k+1} remains in the N-step RCI set $\mathbb{X}_{[1:N]}^{N\text{-RCI}}$ under the given control law. Since $\mathbb{X}_{[1:N]}^{N\text{-RCI}} \subseteq \mathbb{X}$ ■

V. Illustrative Example: Nonlinear Monotone System

A. Nonlinear Double Integrator

We consider the control of a two-dimensional discrete-time nonlinear double integrator of the following form:

$$x_{k+1} = Ax_k + Bu_k + F\sqrt{x_k^T x_k} \tag{14}$$

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \ B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ F = \begin{pmatrix} p \\ p \end{pmatrix}$$

which can be controlled by a two-dimensional input vector $u_k \in [(-10\ -5)^T,\ (10\ 5)^T]$. The states space is constrained to $x_k \in [(0\ 0)^T,\ (10\ 10)^T]$. The system is subject to uncertainty with the parameter being uncertain in the range $p^- = 0.0 \leq p \leq 0.3 = p^+$.

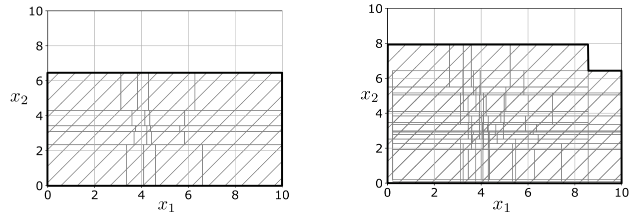
The task for the controller is a tracking problem, with the desired setpoint defined as $\bar{x} = (5\ 2)^T$.

B. Evaluated Controllers

We investigate the proposed method on two different controllers, a nonlinear MPC and a neural network controller imitating a nonlinear MPC (approximate MPC). Note that any other learning-based controller such as reinforcement learning methods would be suitable as well. The nonlinear MPC solves the following optimization problem at every sampling time:

$$\begin{aligned} \min_u \quad & \sum_{j=0}^{N_{\text{pred}}-1} \Delta u_j^T R \Delta u_j + \sum_{j=0}^{N_{\text{pred}}} (x_j - \bar{x}_j)^T Q (x_j - \bar{x}_j) \\ \text{s.t.} \quad & x_{j+1} = f(x_k, u_k, p) \\ & \Delta u_j = u_j - u_{j-1} \\ & 0 \leq x_j \leq c_x \\ & -c_u \leq u_j \leq c_u \\ & x_0 = x_k \\ & u_{-1} = u_{k-1} \\ & \forall j = 0, \ldots, N_{\text{pred}} - 1. \end{aligned} \tag{15}$$

The optimal sequence of inputs is denoted as $u^\star = [u_0^\star \ldots u_{N_{\text{pred}}-1}^\star]$, while $\Pi_{\text{MPC}}(x_k, u_{k-1}) = u_0^\star \in \mathbb{U}$ is applied to the system in each step. The initial state of the optimized trajectories in each time step is $x_k \in \mathbb{X}$. N_{pred} is the prediction horizon of the MPC. The MPC is evaluated with the nominal value of $p = 0.15$, a prediction horizon of $N_{\text{pred}} = 10$, state



(a) One-step RCI set. (b) N-step RCI set.

Fig. 3. Robust control invariant sets for the system as proposed in [12] (left) and the increased N-step RCI set as defined in (9) under the proposed cost function in (7) (right).

and input constraints as stated in Section V-A. The cost function is defined with $Q = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Secondly, we define the approximate MPC controller as a feedforward neural network, $\mathcal{N}(x, \theta, d, m)$, parameterized by a number of layers d, the number of neurons m, a nonlinear activation function, the inputs x in the network, and its network parameters θ. The approximate control law is therefore $\Pi_{\text{approx}}(x_k, u_{k-1}) = \mathcal{N}(x_k, u_{k-1})$. For the given system in Section V-A we train a neural network using $n_{\text{train}} = 10000$ sampling points, $d = 6$, $m = 100$, a ReLU activation function and the ADAM solver [17] , and use this network as an approximated MPC Π_{approx}. Finally, the proposed framework is compared against the robust MPC approach Π_{robust} in [12], which instead of relying on a safety filter to ensure constraints satisfaction under uncertainty, incorporates the robustness directly in the underlying MPC formulation.

C. Results

All the results presented in this letter are openly available.[1] The MPC is implemented with the do-mpc framework [18] using the NLP solver IPOPT [19], which is also used for the offline computation of (9). All of the performance metrics are computed on an i7-CPU/16 GB RAM computer.

1) Improved N-Step Robust Control Invariant Set: One of the contributions of this letter is to enlarge the set in which the controller can be operated by calculating an N-step robust control invariant set. With the cost function as proposed in (7), we ensure to have at least the same area as for the RCI set proposed in (6). For the given system, $N = 3$ with $N_s = 25$ subregions yields the best results, as no gain in the overall area for both the one-step RCI set and N-step RCI set is achieved by further increasing N and N_s. The area in which the framework can be applied safely increases by 20.03 % by using the N-step RCI set compared to the one-step RCI set, and is depicted in Figure 3. By increasing the RCI set size the overall area is not a hyperrectangular anymore, as it is the union of all optimized sets. In theory, these sets can even be disjunct. The computation time of the offline N-step RCI set calculation is 0.15 s.

2) Performance Metrics: To further analyze the performance of the proposed method, the tracking task described in Section V-A is performed with and without the safety filter for both the exact nominal MPC Π_{MPC} and the approximate MPC Π_{approx}. The respective controllers with the proposed safety filter are denoted as $\kappa(x_k, \Pi_{\text{MPC}})$ and $\kappa(x_k, \Pi_{\text{approx}})$. Furthermore, these controllers are compared against the exact robust MPC (Π_{robust}) method proposed in [12]. We set the initial state to $(x_0 = 8.57)^T$ and $u_{-1} = (-5\ \ -10)^T$, which would have been an infeasible state value given just the one-step RCI set, and compute a trajectory of 20 s. We compare the stage cost, the evaluation time, and the constraint violations for $M = 1000$ different uncertainty realizations. The mean values of stage costs, evaluation time, percentage of violations as well as percentage of steps where the safety filter is applied over the entire time can be found in Table I. The results are shown for simplicity for a fixed initial condition. We obtained similar conclusions for varying initial conditions. Both the nominal MPC and approximate MPC achieve the best overall stage costs stage costs without the filter. When employing the proposed safety filter, the stage cost increases due to the usage of the filter inputs. The robust method is more conservative in tracking the setpoint, achieving an overall higher stage cost compared to nominal or approximate MPC. As expected, the computation time required for evaluating the nominal MPC is smaller than the required for the evaluation of the robust MPC, but it can be significantly reduced if approximate MPC is used. In addition, the nominal and approximate controllers lead to constraint violations for 0.72 % and 1.44 % of the time steps. If the safety filter is employed, it is active in 5.00 % of the conducted time steps while achieving its goal that no constraint violations are encountered.

TABLE I

PERFORMANCE METRICS (PERCENTAGE OF SAFETY FILTER APPLICATION ABBREVIATED WITH u^f) FOR THE EVALUATED CONTROLLERS WITH AND WITHOUT A SAFETY FILTER IN COMPARISON TO THE ROBUST MPC METHOD PROPOSED IN [12]

	Cost [-]	Ev. Time [ms]	Viol. [%]	u^f [%]
Π_{robust}	115.81	641	0	–
Π_{MPC}	106.91	8.58	0.72	–
$\kappa(x_k, \Pi_{\text{MPC}})$	117.32	8.69	0	5.00
Π_{approx}	108.35	0.55	1.44	–
$\kappa(x_k, \Pi_{\text{approx}})$	116.96	0.63	0	5.00

Figure 4 shows one state trajectory of the approximate MPC with and without a safety filter as well as the robust MPC approach for one specific uncertainty realization. For $t = 1$ s the approximate MPC without a safety filter registers a constraint violation, which is detected in the proposed safety framework and corrected by the safety filter.

3) Benefits of the Proposed Safety Filter: With the results in Table I, Theorem 1 is validated as the safety filter method ensures deterministic guarantees even for the approximate MPC controller which, as a standalone controller, would be unsafe. As both MPC and approximate MPC are designed on the nominal model and therefore not optimal with uncertainty, their usage can lead to constraint violations. Since the fallback inputs are not optimal but safe, the performance will degrade if the fallback inputs are applied while ensuring the safety for these controllers. As the safety filter approach guarantees safety through an offline computation of the N-step RCI

[1] https://github.com/JoshuaAda/CDC_DGAMUNMS

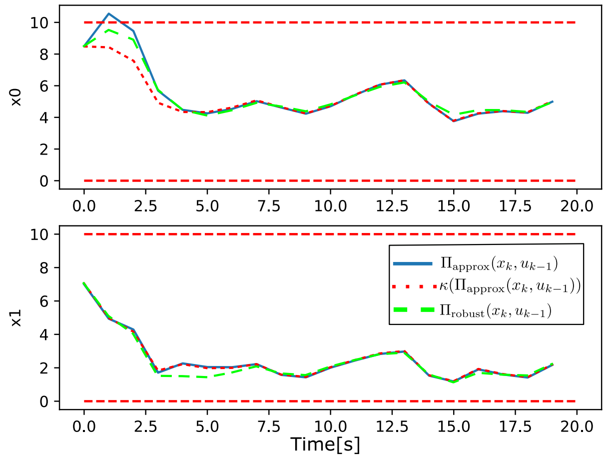


Fig. 4. State trajectory over time of the approximate MPC with and without the safety filter as well as the robust MPC approach. Due to the applied noise, there is a control violation at $t = 1\,\text{s}$ for the approximate MPC, which is rescued by the applied fallback strategy in the proposed control scheme.

set, it drastically reduces evaluation time and computational effort during the online execution against the method proposed in [12]. The online evaluation scales at most linearly with the number of subregions as the check in (11) can be efficiently evaluated via a search tree, that exploits the sequencing of the partition of the state space so that checking iteratively for each dimension if the current state is within the respective interval narrows down the search space proportional to the number of partitions in that dimension. Therefore, the proposed method scales better than the approach in [12], therefore increasing its applicability to larger nonlinear monotone systems. Finally, the main advantage of the method is its general applicability to all controller types. Especially learning-based controllers are suitable as they typically come with fast online evaluation times, as shown in Table I, and could be used to further improve the closed-loop performance.

VI. Conclusion and Future Work

We propose a framework to safely use learning-based controllers for nonlinear monotone systems under uncertainty. An efficient approach to calculate an N-step RCI set has been developed which, compared to a one-step RCI set, has a significantly larger volume and therefore leads to less conservative behavior. The resulting offline solution is used as a safety filter making the framework generalizable to all nonlinear controllers. Since learning-based approximate controllers come with fast evaluation times, they are suited best for the proposed framework, which enables better scalability and reduced computation time in comparison to [12]. We demonstrate the advantages of the proposed approach in a simulation case study by showing decreased conservatism and computational effort as well as a favorable scaling to larger systems. Future research will focus on extending the findings from monotone systems to the larger set of nonlinear mixed-monotone systems [20].

References

[1] A. Mesbah et al., “Fusion of machine learning and MPC under uncertainty: What advances are on the horizon?” in *Proc. Amer. Control Conf. (ACC)*, Atlanta, GA, USA, Jun. 2022, pp. 342–357.

[2] K. P. Wabersich et al., “Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems,” *IEEE Control Syst.*, vol. 43, no. 5, pp. 137–177, Oct. 2023.

[3] B. Karg and S. Lucia, “Efficient representation and approximation of model predictive control laws via deep learning,” *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3866–3878, Sep. 2020.

[4] S. Chen et al., “Approximating explicit model predictive control using constrained neural networks,” in *Proc. Annu. Amer. Control Conf. (ACC)*, Milwaukee, WI, USA, Jun. 2018, pp. 1520–1527.

[5] B. Karg and S. Lucia, “Reinforced approximate robust nonlinear model predictive control,” in *Proc. 23rd Int. Conf. Process Control (PC)*, Jun. 2021, pp. 149–156.

[6] M. Hertneck, J. Kohler, S. Trimpe, and F. Allgower, “Learning an approximate model predictive controller with guarantees,” *IEEE Control Syst. Lett.*, vol. 2, no. 3, pp. 543–548, Jul. 2018.

[7] J. A. Paulson and A. Mesbah, “Approximate closed-loop robust model predictive control with guaranteed stability and constraint satisfaction,” *IEEE Control Syst. Lett.*, vol. 4, no. 3, pp. 719–724, Jul. 2020.

[8] D. Q. Mayne and E. C. Kerrigan, “Tube-based robust nonlinear model predictive control,” *IFAC Proc. Vol.*, vol. 40, no. 12, pp. 36–41, 2007.

[9] H. Hose, J. Köhler, M. N. Zeilinger, and S. Trimpe, “Approximate nonlinear model predictive control with safety-augmented neural networks,” 2023, *arXiv:2304.09575*.

[10] P.-J. Meyer, A. Girard, and E. Witrant, “Robust controlled invariance for monotone systems: Application to ventilation regulation in buildings,” *Automatica*, vol. 70, pp. 14–20, Aug. 2016.

[11] C. Kallies, M. Schliemann-Bullinger, R. Findeisen, S. Lucia, and E. Bullinger, “Monotonicity of kinetic proofreading,” *IFAC-PapersOnLine*, vol. 49, no. 26, pp. 306–311, 2016.

[12] M. Heinlein, S. Subramanian, M. Molnar, and S. Lucia, “Robust MPC approaches for monotone systems,” in *Proc. IEEE 61st Conf. Decis. Control (CDC)*, Dec. 2022, pp. 2354–2360.

[13] D. Angeli and E. D. Sontag, “Monotone control systems,” *IEEE Trans. Autom. Control*, vol. 48, no. 10, pp. 1684–1698, Oct. 2003.

[14] M. Abate and S. Coogan, “Improving the fidelity of mixed-monotone reachable set approximations via state transformations,” in *Proc. Amer. Control Conf. (ACC)*, New Orleans, LA, USA, May 2021, pp. 4674–4679.

[15] S. Coogan, “Mixed monotonicity for reachability and safety in dynamical systems,” in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, Jeju, South Korea, Dec. 2020, pp. 5074–5085.

[16] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic* (Advanced textbooks in control and signal processing). Cham, Switzerland: Springer, 2016.

[17] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.

[18] F. Fiedler et al., “DO-MPC: Towards FAIR nonlinear and robust model predictive control,” *Control Eng. Pract.*, vol. 140, Nov. 2023, Art. no. 105676.

[19] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Math. Program.*, vol. 106, pp. 25–57, Mar. 2006.

[20] A. Makdesi, A. Girard, and L. Fribourg, “Safe learning-based model predictive control using the compatible models approach,” *Eur. J. Control*, vol. 74, Nov. 2023, Art. no. 100849.