Discrete-Time General Nonlinear Robust Control: Stabilization With Closed-Loop Robust DOA Enlargement Based on Interval Analysis

Chaolun Lu, Yongqiang Li, Alexandre Goldsztejn, Zhongsheng Hou, Yu Feng, Yuanjing Feng

Discrete-time general nonlinear robust control: stabilization with closed-loop robust DOA enlargement based on interval analysis

Chaolun Lu, Yongqiang Li, Alexandre Goldsztejn, Zhongsheng Hou, *Fellow, IEEE,* Yu Feng, *Senior Member, IEEE,* and Yuanjing Feng

***Abstract*—For discrete-time nonlinear systems with uncertainty, this paper presents an interval analysis approach to design controller and compute the estimate of the closed-loop robust domain of attraction (RDOA). The dynamics of the system is modelled using difference inclusions. A robust negative-definite and invariant set (RNIS) in the state-control space is proposed. An RNIS is defined by the combination of a robust negative-definite set (RNS) and a robust controlled invariant set (RCIS), which leads to sufficient conditions for Lyapunov stability of the system. The estimate of RDOA can be obtained by projecting an RNIS along the state space. However, the RNIS is hard to obtain by its definition. Drawing inspiration from the RCIS-computation approach, we define a mapping that utilizes the predecessor operator in the state-control space to compute a set limit. Then, the RNIS can be obtained by finding the limit set for an RNS. The computations of RNS and the limit set are based on interval analysis. An algorithm to estimate the RNIS is introduced with rigorous convergence analysis. Finally, we formulate an optimization problem that is solvable, and enlarges the RNIS and the estimate of RDOA. The method is validated on examples of nonlinear systems subject to actuator saturation.**

***Index Terms*—Robust control, robust domain of attraction, invariant set, discrete-time systems, interval analysis, actuator saturation.**

I. Introduction

MODELING physical systems often result in discrete-time state space models with measurement noises [1]. In the early 1970s, the emphasis of research shifted from optimal control to robust control in response to unexpected failures caused by discrepancies between mathematical models and real-world scenarios [2]. At first, control researchers mostly supposed that the models were linear [3], [4]. However, the main weakness of a linear robust controller is that it is very conservative when the nonlinearities are significant. The development of nonlinear robust control methods was strongly motivated by this factor, such as the Lyapunov minimax approach [5], H_∞ optimal control [6], [7], and the robust backstepping approach [8]. Most published nonlinear robust control methods primarily focus on studying nominal models that exhibit affinities concerning the control input. These methods ignore information from nonaffine nonlinearities, which may decrease controller performance. Some works have attempted to solve the control problem using data-driven methods, which design controllers directly from data without the need for modelling [9], [10]. However, the robust control problem considering the general nonlinear nominal model remains a challenge. Therefore, the aim of this study is to explore a control method that can work properly for the general nonlinear discrete-time models.

Recently, the investigation of domain of attraction (DOA) estimation has witnessed growing popularity across various systems [11]–[15]. The DOA plays a significant part in system analysis and system synthesis [16] because it is an asymptotically stabilizable region containing an equilibrium point that reflects the performance of the designed control law [17]. For system analysis, [18] introduces an interval analysis approach to estimate the ROA based on a given Lyapunov candidate function, while [19] presents an approach for evaluating the ROA in non-smooth systems with uncertain parameters. Regarding controller design, several studies have focused on investigating the ROA of closed-loop systems. For general nonlinear discrete-time systems, [20] proposes a data-driven method for configuring robust controllers. The central concept is to filter simulation data points that result in a negative-definite Lyapunov function. However, these proposed methods have a drawback, the invariant sets of the RDOA estimation are limited by the level sets related to the Lyapunov functions, which makes the estimate of RDOA conservative. Also, the data-driven methods involve state-control spatial discretization and random sampling, which means that the set estimation has no quantitative analysis error results.

In this study, the discrete-time general nonlinear systems are modelled by difference inclusions [21]. We propose a robust negative-definite and invariant set in the state-control space (RNIS-SC) to obtain the controller and the estimate of RDOA. RNIS-SC is defined as the combination of a robust

This work was supported in part by the National Natural Science Foundation of China under Grant 62073294, Grant U2341216, Grant U23A20334, and Grant 62373206; in part by the Key Program of Natural Science Foundation of Zhejiang Province under Grant LZ21F030003; in part by China Scholarship Council under Grant 202108330360; and in part by the Major Scientific and Technological Innovation Projects of Hangzhou under Grant 2022AIZD0079.

Chaolun Lu is with the College of Information Engineering, Zhejiang University of Technology, Hangzhou 310000, China, and also with the Department of Computer Science and Information Security, Zhejiang Police College, Hangzhou 310053, China (e-mail: luke@zjut.edu.cn).

Yongqiang Li, Yu Feng and Yuanjing Feng are with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310000 China (e-mail: yqli@zjut.edu.cn, yfeng@zjut.edu.cn, fyjing@zjut.edu.cn).

Alexandre Goldsztejn is with the Laboratoire des Sciences du Numérique de Nantes, Centre National de la Recherche Scientifique, 44321 Nantes, France (e-mail: alexandre.goldsztejn@gmail.com).

Zhongsheng Hou is with School of Automation, Qingdao University, Qingdao, China (e-mail: zshou@qdu.edu.cn).

negative-definite set (RNS-SC) [20] and a robust controlled invariant set (RCIS) [22]. Negative-definiteness and invariance can be guaranteed, respectively, which are sufficient conditions to prove the Lyapunov stability of the system. Consequently, all controllers within the RNIS-SC can stabilize the system asymptotically, and the projections of RNIS-SC onto the state space serve as the estimate of the RDOA. However, obtaining the RNIS-SC by its definition is difficult. Building upon the work of [22], [23], to incorporate the state-control space, an expansion of the predecessor operator is employed, allowing for a set mapping definition between two sets. We utilize multiple mappings to compute the invariant set limit of RNS-SC, which is equivalent to RNIS-SC. Then, based on interval analysis, the inclusion function is extended to difference inclusions, and the convergence can be guaranteed. Next, we propose a numerical algorithm for estimating the RNIS-SC utilizing a given Lyapunov function. This algorithm is based on the set inversion via interval analysis (SIVIA) algorithm, which is a well-established technique in interval arithmetic. The SIVIA algorithm offers a reliable numerical approach to approximate the desired set with a specified tolerance [24]. Additionally, we provide proof of convergence for the proposed algorithm in approximating the RNIS-SC. The RDOA estimate varies depending on the given Lyapunov function, and the volume of the estimate is straightforward to compute because it is composed of non-overlapping hyperrectangles using the interval analysis algorithm. Finally, we formulate a solvable optimization problem aimed at expanding RDOA. We construct a set of parameterized Lyapunov candidate functions, and the problem entails selecting an appropriate Lyapunov function from this set.

Regarding the general nonlinear robust control problem, the main contributions of this study are as follows: 1) In order to stabilize the general nonlinear system with uncertainty, we expanded the RCIS concept to include the state-control space. As a result, our method yields an unstructured controller set RNIS-SC. For a given Lyapunov function, any controller within the proposed RNIS-SC can robustly stabilize the system. Compared to the existing structured robust controllers, RNIS-SC offers greater flexibility in terms of design and implementation. In the field of optimal control for nonlinear systems with uncertainty, most existing methods require a good dataset to train their optimal controllers. The proposed unstructured controller set can serve as the initial control laws and the stability can be guaranteed. 2) The concept of RNIS-SC relies on computing a controlled invariant set, allowing the estimated RDOA to extend beyond the level set of the Lyapunov function. This broader applicability enables a wider range of potential uses. 3) The algorithm for obtaining the estimate of RNIS-SC utilizes the interval analysis approach, which ensures a rigorous convergence analysis. The challenges of this study are as follows: 1) RNIS-SC is defined in state-control space and is estimated based on interval arithmetic. To reduce the conservatism of interval arithmetic, we split the intervals, which leads to an exponential increase in complexity corresponding to the number of state variables. 2) The method for robust DOA enlargement relies on meta-heuristic algorithms, and its convergence time and enlargement results are unknown.

Notation and basic definitions: For given vectors $x, y \in \mathbb{R}^n$, $x < y$ denotes that x is less than y for each component. The function $L(x)$ represents a Lyapunov function in this study, which is supposed to be continuous and non-negative with $L(0) = 0$. For a given compact set $\mathbb{W} \subseteq \mathbb{R}^{n+m}$, $\partial\mathbb{W}$ represents the boundary of the compact set, $\hat{\mathbb{W}}$ represents the internal estimation of $\mathbb{W}$ and consists of non-overlapping hyperrectangles. The image of a set is classically defined by $f(\mathbb{W}) = \{f(w)|w \in \mathbb{W}\} = \bigcup_{w\in\mathbb{W}}\{f(w)\}$ for a real-valued function $f : \mathbb{R}^{n+m} \to \mathbb{R}^n$, and $\mathfrak{F}(\mathbb{W}) = \bigcup_{w\in\mathbb{W}}\mathfrak{F}(w)$ for a set-valued map $\mathfrak{F} : \mathbb{R}^{n+m} \rightrightarrows \mathbb{R}^n$. In particular, $x + \mathbb{W} = \{x + y|y \in \mathbb{W}\}$ is the classical Minkowski addition. The closed ball of radius γ is denoted by B_γ. A set-valued map $\mathfrak{F}$ is upper semi-continuous (u.s.c.) at w if $\forall\gamma > 0, \exists\omega > 0$ such that $\mathfrak{F}(w + B_\omega) \subseteq \mathfrak{F}(w) + B_\gamma$. Furthermore, the set-valued map $\mathfrak{F}$ is upper semi-continuous (u.s.c.) on the set $\mathbb{W}$ if it is so at every $w \in \mathbb{W}$.

II. Robust Stabilization Controller Set

Consider the nonlinear discrete-time system

$$x_{k+1} \in \mathfrak{F}(x_k, u_k), \quad k = 0, 1, 2, ... \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the state of the system at time k, $u_k \in \mathbb{R}^m$ is the control input at time k. The set-valued map $\mathfrak{F} : \mathbb{R}^n \times \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ is the model with unmodelled uncertainty (see Remark 1 below). We furthermore require that $\mathfrak{F}$ is u.s.c. on $\mathbb{R}^n, \mathfrak{F}(0, 0) = 0$ and $\mathfrak{F}(x, u)$ is nonempty and compact. The system (1) around the origin is assumed to be controllable by some linear controllers, this small region around the origin is denoted by $\mathbb{X}_0$. For example, $\forall x \in \mathbb{X}_0$, one can find a controller $u = Kx$, which can asymptotically stabilize the system (1). Our focus is on state-control spaces of interest $\mathbb{W}_{x,u} \subseteq \mathbb{R}^{n+m}$, which we consider to be a compact set. For any nonempty compact subset $\mathbb{W} \subseteq \mathbb{W}_{x,u}$, the projections of $\mathbb{W}$ onto the state space and control space are denoted by $\mathrm{proj}_x(\mathbb{W}) \subseteq \mathbb{R}^n$ and $\mathrm{proj}_u(\mathbb{W}) \subseteq \mathbb{R}^n$, respectively. A set $\mathbb{W}^* \subseteq \mathbb{W}_{x,u}$ is a controller set for system (1) if: any controller $\mu : \mathbb{R}^n \to \mathbb{R}^m$ matches up to $\forall x \in \mathrm{proj}_x(\mathbb{W}^*), (x, \mu(x)) \in \mathbb{W}^*$, is capable of asymptotically stabilizing the system (1), which satisfies

$$\forall(x_0, \mu(x_0)) \in \mathbb{W}^*, \lim_{k\to\infty}(x_k, \mu(x_k)) = (0, 0), \tag{2}$$

and the RDOA for (1) is $\mathrm{proj}_x(\mathbb{W}^*)$.

The main objective of this paper is to compute estimates of the controller set $\mathbb{W}^*$ and RDOA for (1) in the form of a robust controlled invariant set of a given Lyapunov function. Subsequently, we aim to increase the estimates of both the controller set $\mathbb{W}^*$ and RDOA. To achieve this, we select a suitable Lyapunov function from a set of Lyapunov candidate functions.

Remark 1. *If the uncertainty comes from the system and it is bounded, then it can be described by set-valued map $\mathfrak{F}$. Here we introduce two typical forms of set-valued maps: 1) $\mathfrak{F}(x, u) = f(x, u, \Omega)$ where $f(x, u, \Omega) = \{f(x, u, \omega)|\omega \in \Omega\}$, $f(x, u, \omega)$ is a continuous function and differentiable at the*

origin, Ω is a compact set that represents the uncertainty in the parameters ω, and $f(0,0,\omega) = 0, \forall\omega \in \Omega$. 2) $\mathfrak{F}(x,u) = (I+D)f(x,u)$, I is an identity matrix and $D = \mathrm{diag}([-\delta,\delta],...,[-\delta,\delta])$ is an interval diagonal matrix whose diagonal intervals have radius $\delta \geq 0$, a tunable constant that determines the range of the uncertainty. Note that $\mathfrak{F}$ is required to be u.s.c. in both cases.

A. Robust Negative-definite Set in State-Control Space

In this subsection, we briefly introduce the robust negative-definite set in state-control space (RNS-SC) [16]. For the sake of readability, we omit the time instant k in the dynamic system (1). Then x_k, x_{k+1} and u_k are simplified as x, x_+ and u, respectively. For a given Lyapunov function L, state x and control input u, the worst case time difference is defined as

$$\Delta L(x,u) = \max L(\mathfrak{F}(x,u)) - L(x), \tag{3}$$

the maximum is attained because set-valued maps are supposed to have nonempty compact values. Based on the worst case time difference $\Delta L(x,u)$, we give the definition of RNS-SC.

Definition 1. *For a given Lyapunov function L, an interested domain $\mathbb{W}_{x,u} \subseteq \mathbb{R}^{n+m}$, a set $\mathbb{W}_{\mathrm{N}}(L) \subseteq \mathbb{W}_{x,u}$ is RNS-SC for the system* (1) *if*

$$\forall(x,u) \in \mathbb{W}_{\mathrm{N}}(L), \Delta L(x,u) < 0. \tag{4}$$

Lemma 1. *For the system* (1)*, if a Lyapunov function L and a controller $\mu : \mathbb{R}^n \to \mathbb{R}^n$ exists and $\forall x \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N}}(L)), (x,\mu(x)) \in \mathbb{W}_{\mathrm{N}}(L)$, then $\forall x \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N}}(L))$, $L(x_+) - L(x) < 0$.*

Proof. We have $(x,\mu(x)) \in \mathbb{W}_{\mathrm{N}}(L)$, depending on Definition 1, we have $\forall x_+ \in \mathfrak{F}(x,\mu(x)), L(x_+) - L(x) \leq \Delta L(x,\mu(x)) < 0$. Hence, $\forall x \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N}}(L))$, $L(x_+) - L(x) < 0$ is satisfied. □

Since we assume $\mathfrak{F}(0,0) = 0$,then we have $\Delta L(0,0)$=0, hence the origin $(0,0)$ is not in the RNS-SC. Note that the set RNS-SC defined by (4) is not closed and it is hard to estimate. This issue will be handled in Sec. II-D. According to Lemma 1, it might seem that $\mathrm{proj}_x(\mathbb{W}_{\mathrm{N}}(L))$ could be considered as an estimate of RDOA. This is not correct because if $x+$ is out of $\mathrm{proj}_x(\mathbb{W}_{\mathrm{N}}(L))$, it is clear that the time difference of L is no longer negative-definite. One solution is to find a level set that provides an enclosure of the Lyapunov function [13], [18], [19], but this leads to a conservative result. Our goal is to remove the limitations of the level set by constructing a robust controlled invariant set, which we will introduce in the next subsection.

B. Robust controlled invariant set in state-control space (RCIS-SC)

The theory of set invariance has been well established since the seminal paper by Bertsekas [25], and numerous results have been proposed for linear systems [26]. In systems that are prone to uncertainty or noise, one advantage of RCIS is it allows ensuring safety by guaranteeing invariance even in the presence of disturbances. Another advantage of RCIS is its broader range of estimated RDOA results compared to the level set of Lyapunov function. For general nonlinear systems, starting with the pioneering work of Bravo et al. [22] on their interval arithmetic-based computation approach, many other contributions have been made [23], [27]. While these methods ensure invariant sets, the requirement of accuracy introduces additional computational complexity. The accuracy must be sufficiently fine to accurately capture the nonlinear dynamics of the system. Consequently, applying these methods to systems with multiple inputs becomes challenging. Most of the proposed methods focus on the RCIS in the state space. To ensure that the next state x_+ remains negative-definite, it is important to consider the control space as well. Therefore, we extend the RCIS to RCIS-SC as defined below.

Definition 2. *A set $\mathbb{W}_{\mathrm{I}} \subseteq \mathbb{W}_{x,u}$ is an RCIS-SC for system* (1) *if:*

$$\forall(x,u) \in \mathbb{W}_{\mathrm{I}}, \mathfrak{F}(x,u) \subseteq \mathrm{proj}_x(\mathbb{W}_{\mathrm{I}}). \tag{5}$$

For a given controller μ, the set $\mathbb{W}_{\mathrm{I}}$ has the following property.

Lemma 2. *For the system* (1)*, if a controller μ meets the condition that*

$$\forall x \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{I}}), (x,\mu(x)) \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{I}}), \tag{6}$$

then, for any initial state $x_0 \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{I}})$, the orbit $\phi(x_0,k)$ of the closed-loop $x_{k+1} = \mathfrak{F}(x_k,u_k)$ is in $\mathrm{proj}_x(\mathbb{W}_{\mathrm{I}})$ for all $k > 0$.

Proof. We prove Lemma 2 through an induction process. Consider $k = 0$, we have $\phi(x_0,0) = x_0 \in \mathrm{proj}(\mathbb{W}_{\mathrm{I}})$. We also have that $\phi(x_0,k+1) \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{I}})$ when $\phi(x_0,k) \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{I}})$. Because $x_k = \phi(x_0,k) \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{I}})$, from (6), $(x_k,\mu(x_k)) \in \mathbb{W}_{\mathrm{I}}$ is obtained. Based on Definition 2, we can conclude that $x_{k+1} = \phi(x_0,k+1) \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{I}})$. □

C. RNIS-SC

By combining the definitions of RNS-SC $\mathbb{W}_{\mathrm{N}}(L)$ and RCIS-SC $\mathbb{W}_{\mathrm{I}}$, we can define RNIS-SC for a given L.

Definition 3. *A set $\mathbb{W}_{\mathrm{N\&I}}(L) \subseteq \mathbb{W}_{x,u}$ is an RNIS-SC for the system* (1) *if for all state-control pair $(x,u) \in \mathbb{W}_{\mathrm{N\&I}}(L)$, the following conditions are satisfied:*

$$\mathfrak{F}(x,u) \subseteq \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}(L)), \tag{7}$$

$$\Delta L(x,u) < 0. \tag{8}$$

Since $\mathbb{W}_{\mathrm{N\&I}}(L)$ is robust negative-definite, as defined in (4), $\forall(x,u) \in \mathbb{W}_{\mathrm{N\&I}}(L)$, the time difference $\Delta L(x,u)$ is negative-definite. Furthermore, since $\mathbb{W}_{\mathrm{N\&I}}(L)$ is also a robust invariant set, $\forall(x,u) \in \mathbb{W}_{\mathrm{N\&I}}(L)$, the set of possible next states $\mathfrak{F}(x,u)$ is contained in $\mathrm{proj}(\mathbb{W}_{\mathrm{N\&I}}(L))$. This implies the existence of a subsequent control input u^+ such that the time difference of the Lyapunov function remains negative-definite at the next state. The property leads to the following theorem.

Theorem 1. *Given an RNIS-SC* $\mathbb{W}_{\mathrm{N\&I}}(L) \subseteq \mathbb{R}^{n+m}$ *for the system* (1) *and a positive definite function* $L : \mathbb{R}^n \to \mathbb{R}_+$*, if a controller* $\mu : \mathbb{R}^n \to \mathbb{R}^m$ *satisfies* $\mu(0) = 0$*, and*

$$\forall x \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}(L)), (x, \mu(x)) \in \mathbb{W}_{\mathrm{N\&I}}(L), \tag{9}$$

then, for all initial states $x_0 \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}})$*, the closed-loop systems* $x_{k+1} = \mathfrak{F}(x_k, \mu(x_k))$ *are asymptotically stable, indicates that* $\mathbb{W}_{\mathrm{N\&I}} \subseteq \mathbb{W}^*$*.*

Proof. We define $\phi(x_0, k)$ as the orbit of $x_{k+1} = \mathfrak{F}(x_k, \mu(x_k))$, where $k \in \mathbb{N}$. Based on Definition 3, we can conclude that $\mathbb{W}_{\mathrm{N\&I}}(L)$ is a set that is robustly negative-definite. Utilizing (9) and Lemma 1, we obtain the following result:

$$L(\phi(x_0, k+1)) - L(\phi(x_0, k)) < 0, \tag{10}$$

when $\phi(x_0, k) \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}(L))$. Based on Definition 3, $\mathbb{W}_{\mathrm{N\&I}}(L)$ is a robust invariant set. Moreover, from (9) and Lemma 1, we can deduce that $\forall x_0 \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}(L))$, $\forall k \geq 0$,

$$\phi(x_0, k) \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}(L)). \tag{11}$$

From (10) and (11), we have $\forall x_0 \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}})$, the values of Lyapunov function $L(\phi(x_0, k))$ are all monotonically decreasing. Furthermore, L is a positive definite function, which means the limit point of the orbit $L(\phi(x_0, k))$ approaches zero as k approaches infinity. Hence, we have

$$\forall x_0 \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}(L)), \lim_{k\to\infty} L(\phi(x_0, k)) = 0. \tag{12}$$

Based on (12), it can be derived that

$$\forall x_0 \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}(L)), \lim_{k\to\infty} \phi(x_0, k) = 0. \tag{13}$$

The process of (12) to (13) is omitted. One can refer to the proof of Theorem 13.2 in [28], in which the reductio ad absurdum is used. □

With Theorem 1, once the RNIS-SC $\mathbb{W}_{\mathrm{N\&I}}(L)$ has been derived, any controller that satisfies (9) can effectively stabilize the system (1). However, seeking out the solution of $\mathbb{W}_{\mathrm{N\&I}}(L)$ is onerous due to the nonlinearities and uncertainty of $\mathfrak{F}$. It is worth noting that RNIS-SC $\mathbb{W}_{\mathrm{N\&I}}(L)$ is a subset of RNS-SC $\mathbb{W}_{\mathrm{N}}(L)$. In the next subsection, we present an iterative method for obtaining an invariant subset of RNS-SC, which serves as the fundamental basis for the numerical technique proposed in Section III.

D. Finding RNIS-SC by predecessor operator

The predecessor operator, which is also referred to as the one-step backward operator, is a type of preimage, plays a crucial role in computing invariant sets. Predecessor operators have been extensively studied in the literature [25], [29], [30]. Most of the predecessor operators in the existing literature are defined in the state space. Here we broaden the concept of predecessor operators within the state-control space.

$$\mathrm{Pre}(\mathbb{W}) = \{(x, u) \in \mathbb{R}^{n+m} | \mathfrak{F}(x, u) \subseteq \mathrm{proj}_x(\mathbb{W})\}. \tag{14}$$

The predecessor operator $\mathrm{Pre}(\mathbb{W})$ for $\mathbb{W} \subseteq \mathbb{R}^{n+m}$ is a key tool for computing invariant sets in the state-control space. Based on this predecessor operator, we introduce a new mapping denoted as $\mathcal{I}$ that establishes a connection between two sets within the state-control space.

Definition 4. *For a nonempty compact subset* $\mathbb{W} \subseteq \mathbb{R}^{n+m}$*, let mapping* $\mathcal{I}$ *be*

$$\mathcal{I}(\mathbb{W}) = \mathrm{Pre}(\mathbb{W}) \cap \mathbb{W} = \{(x, u) \in \mathbb{W} | \mathfrak{F}(x, u) \subseteq \mathrm{proj}_x(\mathbb{W})\}, \tag{15}$$

The multiple mapping $\mathcal{I}^i$ denotes the composition of the mapping $\mathcal{I}$ with itself i times. A significant property of the mapping $\mathcal{I}$ is shown in the following lemma, see proof in Appendix A.

Lemma 3. *For any* $\mathbb{W} \subseteq \mathbb{R}^{n+m}$*, the set limit* $\mathcal{I}^\infty(\mathbb{W}) := \lim_{i\to\infty} \mathcal{I}^i(\mathbb{W}) := \bigcap_{i=1}^{\infty} \mathcal{I}^i(\mathbb{W})$ *exists and is invariant for system* (1)*.*

Remark 2. *Similar findings to Lemma 3 have been extensively discussed and reported in the literature, starting with the conclusion of Problem 8 in [31]. The lemma is the basis for many iterative algorithms [23], [25], [26]. In the context of infinite-time reachability analysis, a succinct proof in the state space can be found in Proposition 4 of [25]. Additionally, a more comprehensive proof is provided in Theorem 5.2 of [32] (pp. 154). In the case of switched systems, a comparable proof is outlined in Theorem 1 of [23]. All the aforementioned work relates to the state space. In this study, we extend the lemma to difference inclusions and sets in the state-control space. Two challenges that arise when implementing iterative computation are the computation of reachable sets and ensuring finite termination [23]. A branch-and-bound algorithm can be employed to realize this computation [33].*

According to Lemma 3, the existence of $\mathcal{I}^\infty(\mathbb{W})$ is guaranteed, and it serves as the invariant set for a given set $\mathbb{W} \subseteq \mathbb{R}^{n+m}$. However, the RNS-SC $\mathbb{W}_{\mathrm{N}}(L)$ is generally not closed because it is defined by the strict inequality $\Delta L(x, u) < 0$, so the origin can not belong to it. Since the system (1) is assumed to be controllable near the origin, we can safely make a modification to the definitions given in (4) to ensure the closedness of $\mathbb{W}_{\mathrm{N}}(L)$:

$$\mathbb{W}_{\mathrm{N}}(L) = \quad \{(x, u) \in \mathbb{W}_{x,u} | \Delta L(x, u) \leq -\alpha\}, \tag{16}$$

and (8) is modified as follows

$$\Delta L(x, u) \leq -\alpha, \tag{17}$$

where α represents a positive minuscule constant. By adding a constant scalar α to satisfy the Lyapunov function's decreasing condition, a small-scale region $\mathbb{X}_\alpha = [\underline{x}_\alpha, \bar{x}_\alpha]$ around the origin is removed from $\mathbb{W}_{\mathrm{N}}(L)$. In this study, it is assumed that the system (1) is controllable in the origin region $\mathbb{X}_0$. If we choose an appropriate α such that $\mathbb{X}_\alpha \subseteq \mathbb{X}_0$, then $\mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}) \cup \mathbb{X}_\alpha$ is an estimate of RDOA.

Corollary 1. *Given RNIS-SC* $\mathbb{W}_{\mathrm{N\&I}}(L)$ *that satisfies* (17) *for the system* (1) *and a Lyapunov function L, if the controller* μ *satisfies* (9)*, there exists a constant* α *such that, for all* $x_0 \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}) \cup \mathbb{X}_\alpha$*, the closed-loop systems* $x_{k+1} = \mathfrak{F}(x_k, \mu(x_k))$ *are asymptotically stable.*

Proof. According to (1), we still have (13) because the orbit $L(\phi(x_0,k))$ continues to exhibit a monotonically decreasing trend. Then, we aim to show the existence of a parameter α such that $\mathbb{X}_\alpha = [\underline{x}_\alpha, \bar{x}_\alpha] \subseteq \mathbb{X}_0$. Since α is an arbitrarily small constant, we have $\lim_{\alpha\to 0, x\to 0} L(x_{k+1}) - L(x_k) = 0$, which implies $\lim_{\alpha\to 0} \underline{x}_\alpha = 0, \lim_{\alpha\to 0} \bar{x}_\alpha = 0 \Rightarrow \lim_{\alpha\to 0} \mathbb{X}_\alpha = \{0\}$. The set $\mathbb{X}_0$ is a fixed nonempty region around the origin, therefore there must exist a small constant α such that $\mathbb{X}_\alpha = [\underline{x}_\alpha, \bar{x}_\alpha] \subseteq \mathbb{X}_0$. It can be obtained that

$$\mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}) \cup \mathbb{X}_\alpha = \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}) \cup \mathbb{X}_0. \tag{18}$$

The system (1) is assumed to be controllable in the origin region $\mathbb{X}_0$. Hence, $\forall x_0 \in \mathrm{proj}_x(\mathbb{W}_{\mathrm{N\&I}}) \cup \mathbb{X}_\alpha, \lim_{k\to\infty} \phi(x_0,k) = 0$. □

Remark 3. *In this work, we assume that the system around the origin is controllable and that the state in the small-scale region can be stabilized using existing linear controllers. The range of the controllable set $\mathbb{X}_0$ can be computed by using some specific techniques such as interval Lyapunov equation [19]. Under this assumption, RNIS-SC $\mathbb{W}_{\mathrm{N\&I}}(L)$ with new condition* (17) *is compact and the above corollary proves the stability for $\mathbb{W}_{\mathrm{N\&I}}(L)$.*

With the fulfillment of the closedness condition of $\mathbb{W}_{\mathrm{N}}(L)$, as defined in (16), and the compactness of the set $\mathbb{W}_{\mathrm{N}}(L)$, the following theorem enables us to obtain the RNIS-SC.

Theorem 2. *The RNIS-SC $\mathbb{W}_{\mathrm{N\&I}}(L)$, as stated in Theorem 1, complies with the following condition:*

$$\mathbb{W}_{\mathrm{N\&I}}(L) = \mathcal{I}^\infty(\mathbb{W}_{\mathrm{N}}(L)). \tag{19}$$

Proof. Lemma 3 asserts the existence of the set limit $\mathcal{I}^\infty(\mathbb{W}_{\mathrm{N}}(L)) = \lim_{i\to\infty} \mathcal{I}^i(\mathbb{W}_{\mathrm{N}}(L))$. Moreover, this set limit is invariant under system(1), which confirms the satisfaction of (7).

Given the definition of $\mathcal{I}$, it can be observed that

$$\mathcal{I}^\infty(\mathbb{W}_{\mathrm{N}}(L)) \subseteq \cdots \subseteq \mathcal{I}^2(\mathbb{W}_{\mathrm{N}}(L)) \subseteq \mathcal{I}(\mathbb{W}_{\mathrm{N}}(L)) \subseteq \mathbb{W}_{\mathrm{N}}(L).$$

Hence, $\mathbb{W}_{\mathrm{N\&I}}(L)$ belongs to $\mathbb{W}_{\mathrm{N}}(L)$. It can be concluded that $\mathcal{I}^\infty(\mathbb{W})$ exhibits negative-definiteness for system (1), thus verifying the satisfaction (17). Consequently, we obtain $\mathbb{W}_{\mathrm{N\&I}}(L) = \mathcal{I}^\infty(\mathbb{W}_{\mathrm{N}}(L))$. □

Remark 4. *The theorem presented in this paper is akin to Proposition 2 of [16], with the main distinction being that we take into account a system with uncertainty, which is not addressed in [16].*

III. Estimating RNIS-SC Via Interval Analysis

In this section, we propose a computational method for estimating the RNIS-SC. Firstly, we introduce the SEVIA algorithm, an interval analysis method, for estimating a specific set. Secondly, we introduce an algorithm that utilizes the SEVIA algorithm to estimate the RNIS-SC $\mathbb{W}_{\mathrm{N\&I}}(L)$. Lastly, we establish the convergence of the proposed algorithm.

A. Introduction of SEVIA

Interval analysis is a methodology used to approximate sets when the actual solution of the problem is known. This methodology guarantees an expected level of tolerance for the approximations of sets [24]. The foundational concepts of interval analysis include the construction of interval vectors and the formulation of inclusion functions. These concepts are introduced as follows (for more details, refer to [24]). An interval is denoted by $[x_i] = [\underline{x_i}, \bar{x}_i]$. $[x_i]$ is a compact set of real numbers within boundaries so that $\underline{x_i} \le x_i \le \bar{x}_i$, where $\underline{x_i} \in \mathbb{R}$ denotes the lower boundary and the $\bar{x}_i \in \mathbb{R}$ denotes the upper boundary of $[x_i]$. An interval vector (or box) $[x]$ belongs to $\mathbb{R}^{n_1}$. Similar to the definition of vector, $[x] = [x_{(1)}] \times [x_{(2)}] \times \cdots \times [x_{(n_1)}]$, where the i-th interval $[x_{(i)}]$ is defined as previously, the lower boundary of $[x]$ is $\underline{x} = (\underline{x_1}, \underline{x_2}, ..., \underline{x_{n_1}})$ and the upper boundary is $\bar{x} = (\bar{x_1}, \bar{x_2}, ..., \bar{x_{n_1}})$. The interval union of two intervals, denoted as $[a] \sqcup [b]$, is represented by $[\min \underline{a}, \underline{b}, \max \bar{a}, \bar{b}]$. Let $\mathbb{IR}^1$ represent the set of all interval numbers and let $\mathbb{IR}^{n_1}$ represents the set of hyperrectangles in n_1-dimension. If a function is defined as $f : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ and $\forall [x] \in \mathbb{IR}^{n_1}, f([x]) \subseteq [f]([x])$, then the inclusion function for f is defined as $[f] : \mathbb{IR}^{n_1} \to \mathbb{IR}^{n_2}$. Let $d([x]) = \max_{1\le i\le n_1}(\bar{x}_{(i)} - \underline{x}_{(i)})$ represent the width of interval vector $[x]$, if $\forall [x] \in \mathbb{IR}^{n_1}, \lim_{d([x])\to 0} d([f]([x])) = 0$, the inclusion function $[f]$ is convergent. Note that there exist different inclusion function forms for a given function.

Suppose the estimated set $\mathbb{W} \subseteq \mathbb{W}_{\mathrm{cons}}$ is defined in the form

$$\mathbb{W} = \{w \in \mathbb{W}_{\mathrm{cons}} | w \quad \text{satisfies} \quad \mathrm{CONDITIONS}\}, \tag{20}$$

where CONDITIONS refers to the constraints imposed on w, such as $||w|| < 1, w \neq \emptyset$, etc. The SEVIA algorithm is similar to the SIVIA algorithm in [24], but this study focuses more on set estimation than set inversion. The approach to estimate the set $\mathbb{W} \subseteq \mathbb{W}_{\mathrm{cons}}$ using the SEVIA algorithm can be outlined as follows: 1) Convert the CONDITIONS into an interval inclusion function expression, which allows us to verify whether a given hyperrectangle $[w] \subseteq \mathbb{R}^{n+m}$ is completely within or outside of $\mathbb{W}$. 2) Begin with an initial set $\hat{\mathbb{W}}_{\mathrm{init}}$ of hyperrectangles. 3) Utilize the SEVIA algorithm to recursively explore different cases and determine the suitable actions for each case. (1) Inner condition tests: If $[w]$ is inside the set $\mathbb{W}$, then it will be saved in the hyperrectangle set $\hat{\mathbb{W}}_{\mathrm{in}}$, which is executed in Lines 7 and 8 of Algorithm 1. (2) Outer condition tests: If $[w]$ is out of $\mathbb{W}$, then $[w]$ will be saved in the set $\hat{\mathbb{W}}_{\mathrm{out}}$, which is executed in Lines 9 and 10 of Algorithm 1. (3) If $[w]$ fails to meet both the inner and outer tests, it indicates that $[w]$ is bounded by $\mathbb{W}$ and is called an indeterminate box. The width of $[w]$ is denoted by $d[w]$. If $d[w]$ is smaller than a tolerance parameter $\epsilon > 0$, then $[w]$ is deemed sufficiently small to curve the bound and is saved in the set $\hat{\mathbb{W}}_{\mathrm{bou}}$. This process is shown in Lines 11 and 12 of Algorithm 1. (4) If $[w]$ is indeterminate but $d[w]$ is larger than ϵ, it will be split into two new hyperrectangles through a bisection process. This results in the creation of two distinct hyperrectangles. These hyperrectangles should be explored further, so they are saved in the set $\hat{\mathbb{W}}_{\mathrm{do}}$, as shown

in Lines 13-15 of Algorithm 1.

The algorithm should execute the entire exploration procedure recursively if the set $\hat{\mathbb{W}}_{\rm do}$ is not empty.

Algorithm 1 SEVIA

1: **Input** Inner test, Outer test, $\hat{\mathbb{W}}_{x,u}, \epsilon$
2: Initialize the sets $\hat{\mathbb{W}}_{\rm in}, \hat{\mathbb{W}}_{\rm out}, \hat{\mathbb{W}}_{\rm bou}$ as empty sets
3: $\hat{\mathbb{W}}_{\rm do} := \hat{\mathbb{W}}_{x,u}$
4: **while** $\hat{\mathbb{W}}_{\rm do}$ not empty **do**
5: Retrieve a hyperrectangle $[w]$ from $\hat{\mathbb{W}}_{\rm do}$
6: Delete $[w]$ from $\hat{\mathbb{W}}_{\rm do}$
7: **if** $[w]$ meets Inner test **then**
8: Append $[w]$ to $\hat{\mathbb{W}}_{\rm in}$
9: **else if** $[w]$ meets Outer test **then**
10: Append $[w]$ to $\hat{\mathbb{W}}_{\rm out}$
11: **else if** $\epsilon \geq d([w])$ **then**
12: Append $[w]$ to $\hat{\mathbb{W}}_{\rm bou}$
13: **else**
14: Bisect the hyperrectangle $[w]$ and append the newly generated hyperrectangles to $\hat{\mathbb{W}}_{\rm do}$.
15: **end if**
16: **end while**
17: **Output** $\hat{\mathbb{W}}_{\rm in}$

B. Estimating RNIS-SC via interval analysis

This subsection introduces the RNISEVIA algorithm (RNIS-SC estimation via interval analysis) for estimating the RNIS-SC $\mathbb{W}_{\rm N\&I}(L)$ through the utilization of the SEVIA algorithm.

The inclusion function, which is well-defined and extensively studied, is applicable to functions defined from Euclidean space to Euclidean space [24]. However, the existing literature does not provide a definition for the inclusion function of the difference inclusion $\mathfrak{F}$. Let $w = (x, u)$, the functions $\underline{f}, \bar{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ are defined as

$$\underline{f}(w) = (\underline{f_1}(w), \underline{f_2}(w), ..., \underline{f_i}(w), ..., \underline{f_n}(w)), \tag{21}$$
$$\bar{f}(w) = (\bar{f}_1(w), \bar{f}_2(w), ..., \bar{f}_i(w), ..., \bar{f}_n(w)), \tag{22}$$

where $\underline{f_i}(w) = \min\{x_i \in \mathbb{R} : x \in \mathfrak{F}(w)\}$ and $\bar{f}_i(w) = \max\{x_i \in \mathbb{R} : x \in \mathfrak{F}(w)\}$. The interval function $[\mathfrak{F}] : \mathbb{IR}^{n+m} \to \mathbb{IR}^n$ is defined as

$$[\mathfrak{F}]([w]) = [\underline{f}]([w]) \sqcup [\bar{f}]([w]). \tag{23}$$

Lemma 4. *The interval function* $[\mathfrak{F}]$ *defined by* (23) *is the inclusion function of* $\mathfrak{F}$*, namely* $\forall [w] \in \mathbb{IR}^{n+m}, \mathfrak{F}([w]) \subseteq [\mathfrak{F}]([w])$.

Proof. According to the (22), we know that $\mathfrak{F}(w) \subseteq [\underline{f}(w), \bar{f}(w)]$, which indicates that

$$\mathfrak{F}([w]) \subseteq [\inf\{a \in \mathbb{R}^n | \forall x \in \underline{f}([w]), a \leq x\}, \sup\{b \in \mathbb{R}^n | \forall x \in \bar{f}([w]), x \leq b\}]. \tag{24}$$

Based on the properties of the inclusion function, we have

$$\forall [w] \in \mathbb{IR}^{n+m}, \underline{f}([w]) \subseteq [\underline{f}]([w]),$$
$$\forall [w] \in \mathbb{IR}^{n+m}, \bar{f}([w]) \subseteq [\bar{f}]([w]).$$

We have

$$\forall [w] \in \mathbb{IR}^{n+m}, \inf\{a \in \mathbb{R}^n | \forall x \in \underline{f}([w]), a \leq x\} \in [\underline{f}]([w]),$$
$$\forall [w] \in \mathbb{IR}^{n+m}, \sup\{b \in \mathbb{R}^n | \forall x \in \bar{f}([w]), x \leq b\} \in [\bar{f}]([w]).$$

which means$[\inf\{a \in \mathbb{R}^n | \forall x \in \underline{f}([w]), a \leq x\}, \sup\{b \in \mathbb{R}^n | \forall x \in \bar{f}([w]), x \leq b\}] \subseteq [\underline{f}]([w]) \sqcup [\bar{f}]([w])$. Hence, we have $\forall [w] \in \mathbb{IR}^{n+m}, \mathfrak{F}([w]) \subseteq [\mathfrak{F}]([w])$. □

Based on the inclusion function $[\mathfrak{F}]([w])$ of $\mathfrak{F}(w)$, the RNS-SC $\mathbb{W}_{\rm N}(L)$ can be computed by utilizing the SEVIA algorithm. By (16), $\forall w \in \mathbb{W}_{\rm N}(L), \Delta L(w) \leq -\alpha$. Hence, if $[w] = [x] \times [u] \subseteq \mathbb{W}_{\rm N}(L)$, $[w]$ satisfies

$$L(\mathfrak{F}([w])) - L([x]) \subset (-\infty, -\alpha]. \tag{25}$$

By utilizing the properties of interval operations, we can derive

$$L(\mathfrak{F}([w])) - L([x]) \subset [L]([\mathfrak{F}]([w])) - [L]([x]). \tag{26}$$

By (26), if

$$[L]([\mathfrak{F}]([w])) - [L]([x]) \subset (-\infty, -\alpha], \tag{27}$$

then (25) holds true so that $[w] \subseteq \mathbb{W}_{\rm N}(L)$. Similarly, if

$$[L]([\mathfrak{F}]([w])) - [L]([x]) \cap (-\infty, -\alpha] = \emptyset, \tag{28}$$

then $L(\mathfrak{F}([w])) - L([x])$ is outside $(-\infty, -\alpha]$ and $[w]$ is outside $\mathbb{W}_{\rm N}(L)$. With the internal validation expressions (27) and external validation expressions (28), the estimated set $\hat{\mathbb{W}}_{\rm N}(L)$ of RNS-SC $\mathbb{W}_{\rm N}(L)$ can be obtained by

$$\hat{\mathbb{W}}_{\rm N}(L) := {\rm SEVIA}((27), (28), \{[w_{x,u}]\}, \epsilon), \tag{29}$$

where $[w_{x,u}] \subseteq \mathbb{IR}^{n+m}$ is a hyperrectangle, representing the interested region.

Now that we have the initial set $\hat{\mathbb{W}}_{\rm N}(L)$, the approximation of mapping $\mathcal{I}$ can be obtained by executing the SEVIA algorithm. According to Theorem 2, the internal estimation of RNIS-SC $\mathbb{W}_{\rm N\&I}(L)$ can be acquired. The steps outlined in Lines 4-10 of the Algorithm 2 demonstrate the process.

For any hyperrectangle set $\hat{\mathbb{W}}_1 \subseteq \hat{\mathbb{W}}_{\rm N}(L)$, from (15), we know that for any $[w] \subseteq \mathcal{I}(\hat{\mathbb{W}}_1)$, $[w]$ satisfies that

$$\mathfrak{F}([w]) \subseteq {\rm proj}_x(\hat{\mathbb{W}}_1). \tag{30}$$

Since solving for the function range can be challenging, the interval inclusion function is employed as a substitute to verify the aforementioned formula. Since $\mathfrak{F}([w]) \subseteq [\mathfrak{F}]([w])$, we have

$$[\mathfrak{F}]([w]) \subseteq {\rm proj}_x(\hat{\mathbb{W}}_1), \tag{31}$$

then (30) can be satisfied, which implies that $[w] \subseteq \mathcal{I}(\hat{\mathbb{W}}_1)$. Similarly, if

$$[\mathfrak{F}]([w]) \cap {\rm proj}_x(\hat{\mathbb{W}}_1) = \emptyset, \tag{32}$$

then $[\mathfrak{F}]([w])$ must be outside ${\rm proj}(\hat{\mathbb{W}}_1)$, therefore $[w]$ is outside $\mathcal{I}(\hat{\mathbb{W}}_1)$. Using the internal validation expressions (31) and the external validation expressions (32), the estimated $\hat{\mathbb{W}}_2$ of $\mathcal{I}(\hat{\mathbb{W}}_1)$ can be derived using the following formula:

$$\hat{\mathbb{W}}_2 := {\rm SEVIA}((31), (32), \hat{\mathbb{W}}_1, \epsilon). \tag{33}$$

The full implementations are introduced in the Algorithm 2.

Algorithm 2 RNISEVIA

1: **Input** $\mathfrak{F}, L, [w_{\text{init}}], \epsilon$
2: Obtaining the internal validation expressions (27) and external validation expressions (28) by utilizing $\mathfrak{F}$ and L.
3: $\hat{\mathbb{W}}_{\text{N}}(L) := \text{SEVIA}((27), (28), \{[w_{\text{init}}]\}, \epsilon)$
4: Obtaining the internal validation expressions (31) and external validation expressions (32) by utilizing $\mathfrak{F}$ and L.
5: $\hat{\mathbb{W}}_1 := \emptyset$
6: $\hat{\mathbb{W}}_2 := \hat{\mathbb{W}}_{\text{N}}(L)$
7: **while** $\hat{\mathbb{W}}_1 \neq \hat{\mathbb{W}}_2$ **do**
8: $\quad \hat{\mathbb{W}}_1 := \hat{\mathbb{W}}_2$
9: $\quad \hat{\mathbb{W}}_2 := \text{SEVIA}((31), (32), \hat{\mathbb{W}}_1, \epsilon)$
10: **end while**
11: $\hat{\mathbb{W}}_{\text{N\&I}}(L) := \hat{\mathbb{W}}_1$
12: **Output** $\hat{\mathbb{W}}_{\text{N\&I}}(L)$

Lemma 5. *RNIDEVIA is a finite algorithm that terminates after executing SEVIA less than* $(d([w_{init}])/\epsilon)^{n+m}$ *times.*

Proof. Firstly, SEVIA algorithm is a finite algorithm, the maximum number of generated hyperrectangles is $(d([w_{init}])/\epsilon)^{n+m}$. These hyperrectangles are called unit hyperrectangles $[w_u]$, which satisfy $d([w_u]) = \epsilon$. Secondly, the SEVIA algorithm in (33) causes the number of $[w_u]$ inside $\hat{\mathbb{W}}_1$ to decrease monotonically. Hence RNIDEVIA is a finite algorithm and the worst case is executing $(d([w_{init}])/\epsilon)^{n+m}$ SEVIA times. □

Remark 5. *If the algorithm reaches the maximum execution times, it means the whole state-control space has been partitioned into unit hyperrectangles with width of* ϵ*. Each unit hyperrectangle has been analyzed by SEVIA algorithm. Since the SEVIA algorithm in (32) causes the number of* $[w_u]$ *inside* $\hat{\mathbb{W}}_1$ *to decrease monotonically. The worst case scenario is that the output set* $\hat{\mathbb{W}}_1 = \emptyset$*, then the stopping criterion* $\hat{\mathbb{W}}_1 = \hat{\mathbb{W}}_2$ *is finally satisfied. However, the induced set is useless in this case because the tolerance* ϵ *is too large. If* ϵ *is small enough, the induced set can present the required properties. However, Lemma 5 indicates that the computational complexity of system* (1) *is* $O(2^{n+m})$*. This means that the procedure RNIDEVIA is computationally expensive if we choose small* ϵ*, particularly for high-dimensional systems. In fact, most set estimation methods have exponential complexity [18]–[20], [22], and designing more efficient algorithms is a focus of our future work.*

It is important to note that the output of Algorithm 2 can be $\emptyset$. The convergence of Algorithm 2 depends on the choice of tolerance ϵ, and if $\epsilon \to 0$, its convergence can be guaranteed. This will be discussed in the next subsection.

C. Convergence of RNISEVIA algorithm

Based on the (23), $[\mathfrak{F}]([w])$ represents the interval union of two inclusion functions in Euclidean space. For a comprehensive analysis of various types of inclusion functions, including convergent forms such as natural, center, and Taylor, please refer to [24] (pp. 27-38). Extending the convergence property to the interval union $[g] \sqcup [h]$ of two inclusion functions $[g]$ and $[h]$, we have:

Definition 5. *For* $[\underline{f}]([w]) : \mathbb{IR}^{n+m} \to \mathbb{IR}^n$ *and* $[\bar{f}]([w]) : \mathbb{IR}^{n+m} \to \mathbb{IR}^n$*, the inclusion function* $[\mathfrak{F}]([w]) = [\underline{f}]([w]) \sqcup [\bar{f}]([w]) : \mathbb{IR}^{n+m} \to \mathbb{IR}^n$ *is convergent if, for all sequences of hyperrectangles* $[w](k)$*, the following condition holds:*

$$\lim_{k\to\infty} d([w](k)) = 0 \Rightarrow \lim_{k\to\infty} d([\mathfrak{F}]([w](k))) = D, \qquad (34)$$

where $D = \max_{1\leq i\leq n+m}(\bar{f}(w)_i - \underline{f}(w)_i)$.

Lemma 6. *If both inclusion functions* $[\underline{f}]([w]), [\bar{f}]([w])$ *are convergent, then* $[\mathfrak{F}]([w])$ *that satisfies* (23) *is also convergent.*

Proof. Since $[\mathfrak{F}]([w]) = [\underline{f}]([w]) \sqcup [\bar{f}]([w])$, it follows that for any sequence of hyperrectangles $[w](k)$

$$\lim_{k\to\infty} d([w](k)) = 0 \Rightarrow \lim_{k\to\infty} d([\mathfrak{F}]([w](k))) = \max_{1\leq i\leq n+m} (\sup(x \in [\bar{f}]([w](k))_i) - \inf(x \in [\underline{f}]([w](k))_i)). \qquad (35)$$

The inclusion functions $[\underline{f}]([w]), [\bar{f}]([w])$ are convergent, this implies that

$$\lim_{k\to\infty} d([w](k)) = 0 \Rightarrow \lim_{k\to\infty} d([\underline{f}]([w(k)])) = 0,$$
$$\lim_{k\to\infty} d([w](k)) = 0 \Rightarrow \lim_{k\to\infty} d([\bar{f}]([w](k))) = 0.$$

Hence,

$$\lim_{k\to\infty} d([w](k)) = 0 \Rightarrow \lim_{k\to\infty} [\underline{f}]([w](k)) = \underline{f}(w),$$
$$\lim_{k\to\infty} d([w](k)) = 0 \Rightarrow \lim_{k\to\infty} [\bar{f}]([w](k)) = \bar{f}(w).$$

We have $D = \max_{1\leq i\leq n+m}(\bar{f}(w)_i - \underline{f}(w)_i)$, then we can rewrite (35) as

$$\lim_{k\to\infty} d([w](k)) = 0 \Rightarrow \lim_{k\to\infty} d([\mathfrak{F}]([w](k))) = D,$$

this completes the proof. □

Definition 6. *Let* $\text{clo}(\mathbb{A})$ *represents the closure of the set* $\mathbb{A}$ *and* $\text{int}(\mathbb{A})$ *represents the interior of the set* $\mathbb{A}$*, the compact set* $\mathbb{A}$ *is said to be full if* $\text{clo}(\text{int}(\mathbb{A}))$ *is equal to* $\mathbb{A}$ *[34].*

Theorem 3. *For a full set* $\mathbb{W}_{\text{N\&I}}(L)$*, the set* $\hat{\mathbb{W}}_{\text{N\&I}}(L)$ *evaluated by RNIDEVIA (*$\mathfrak{F}, L, [w_{\text{init}}], \epsilon$*) satisfies the following property:*

$$\lim_{\epsilon\to 0} \hat{\mathbb{W}}_{\text{N\&I}}(L) = \mathbb{W}_{\text{N\&I}}(L).$$

Proof. To establish convergence, we need to demonstrate that as ϵ approaches 0, the $\hat{\mathbb{W}}_{\text{N}}(L)$ obtained at step 3 converges to $\mathbb{W}_{\text{N}}(L)$. In (29), there are two functions involved: $[L]$ and $[\mathfrak{F}]$. The convergence of $[\mathfrak{F}]([w])$ is established in Lemma 6, which states that $\mathfrak{F}([w])$ is a subset of $[\mathfrak{F}]([w])$ and as $d([w])$ approaches 0, $[\mathfrak{F}]([w])$ converges to $\mathfrak{F}([w])$. The function L is in Euclidean space. Then, the inclusion function $[L] : \mathbb{IR}^n \to \mathbb{IR}_+$ exhibits the properties such that $[L](\mathfrak{F}([w])) \subseteq [L]([\mathfrak{F}]([w]))$ and $\lim_{[\mathfrak{F}]([w])\to\mathfrak{F}([w])} [L]([\mathfrak{F}]([w])) = [L](\mathfrak{F}([w]))$. Consequently, we have $\lim_{d([w])\to 0} [L]([\mathfrak{F}]([w])) = L(\mathfrak{F}([w])), \lim_{d([w])\to 0} [L]([x]) = L([x])$.

After the completion of the SEVIA algorithm at step 3, if $[w] \in \hat{\mathbb{W}}_{\rm bou}$, then it implies that $d([w]) < \epsilon$. As ϵ approaches 0, $d([w])$ tends to 0 as well. To rephrase, the inner test (27) and the outer test (28) can be reformulated as follows:

$$L(\mathfrak{F}([w])) - L([x]) \subseteq (-\infty, -\alpha], \tag{36}$$

$$L(\mathfrak{F}([w])) - L([x]) \cap (-\infty, -\alpha] = \emptyset. \tag{37}$$

If $[w] \in \hat{\mathbb{W}}_{\rm bou}$ does not satisfy (36) or (37), it follows that $L(\mathfrak{F}([w])) - L([x]) = 0$ as ϵ approaches 0. Hence,

$$\begin{aligned} \forall [w] \in \hat{\mathbb{W}}_{\rm bou}, \quad & \lim_{\epsilon \to 0} h_\infty([w], \partial \mathbb{W}_{\rm N}(L)) = 0 \\ \Rightarrow \quad & \lim_{\epsilon \to 0} h_\infty(\hat{\mathbb{W}}_{\rm bou}, \partial \mathbb{W}_{\rm N}(L)) = 0, \end{aligned} \tag{38}$$

where $h_\infty(\hat{\mathbb{W}}_{\rm bou}, \partial \mathbb{W}_{\rm N}(L))$ denotes the Hausdorff distance between $\hat{\mathbb{W}}_{\rm bou}$ and $\partial \mathbb{W}_{\rm N}(L))$. The Hausdorff distance, defined using the infinity norm, serves as a metric between the sets of compact subsets of $\mathbb{R}^{n+m}$ (*e.g.*, Definition 10 in [34]). It is straightforward that $\partial \mathbb{W}_{\rm N}(L) \subseteq \hat{\mathbb{W}}_{\rm bou}$. When $\epsilon = 0$, it follows that $h_\infty(\hat{\mathbb{W}}_{\rm bou}, \partial \mathbb{W}_{\rm N}(L)) = 0$, implying $h_\infty(\hat{\mathbb{W}}_{\rm bou} = \partial \mathbb{W}_{\rm N}(L))$. Therefore, we can conclude that $\lim_{\epsilon \to 0} \hat{\mathbb{W}}_{\rm bou} = \partial \mathbb{W}_{\rm N}(L)$.

Now that we have identified the boundary $\partial \mathbb{W}_{\rm N}(L)$, the rest of the hyperrectangles are either outside (36) or inside (37). The set $\mathbb{W}_{\rm N\&I}(L)$ is full and $\mathbb{W}_{\rm N\&I}(L) \subseteq \mathbb{W}_{\rm N}$. Hence the set $\mathbb{W}_{\rm N}$ is full, indicating the existence of a finite hyperrectangles $\hat{\mathbb{W}}_{\rm in}$ that satisfy (36). We have $\lim_{\epsilon \to 0} \hat{\mathbb{W}}_{\rm bou} = \partial \mathbb{W}_{\rm N}(L)$, the rest hyperrectangles are either inner or outer. We can finally get $\lim_{\epsilon \to 0} \hat{\mathbb{W}}_{\rm N}(L) := \hat{\mathbb{W}}_{\rm in} = \mathbb{W}_{\rm N}(L)$. Based on the convergent inclusion function $[\mathfrak{F}]([w])$, SEVIA ensures the convergence of the output as ϵ approaches zero.

Similarly, regardless of the number of times the SEVIA algorithm is executed in the loop from Step 7-10 of Algorithm 2, the inclusion function $[\mathfrak{F}]([w])$ in (32) is convergent, SEVIA iterations satisfy that

$$\lim_{\epsilon \to 0} h_\infty(\hat{\mathbb{W}}_{\rm bou}, \partial \mathbb{W}) = 0.$$

This implies that $h_\infty(\hat{\mathbb{W}}_2, \mathbb{W}_2)$ approaches zero as ϵ approaches zero. Therefore, the output set $\hat{\mathbb{W}}_{\rm N\&I}(L) := \hat{\mathbb{W}}_1 = \hat{\mathbb{W}}_2$ satisfies that

$$\lim_{\epsilon \to 0} \ h_\infty(\hat{\mathbb{W}}_{\rm N\&I}(L), \mathbb{W}_{\rm N\&I}(L)) = 0.$$

Since $\hat{\mathbb{W}}_{\rm N\&I}(L) \subseteq \mathbb{W}_{\rm N\&I}(L)$. When $\epsilon = 0$, we have $h_\infty(\hat{\mathbb{W}}_{\rm N\&I}(L), \ \mathbb{W}_{\rm N\&I}(L)) = 0$, which implies $\hat{\mathbb{W}}_{\rm N\&I}(L) = \mathbb{W}_{\rm N\&I}(L)$. Consequently, we can conclude that $\lim_{\epsilon \to 0} \hat{\mathbb{W}}_{\rm N\&I}(L) = \mathbb{W}_{\rm N\&I}(L)$. □

Based on Theorem 3, when ϵ approaches zero, the inner approximation $\hat{\mathbb{W}}_{\rm N\&I}(L)$ converges to the set $\mathbb{W}_{\rm N\&I}(L)$. This indicates that RNIDEVIA can approximate RNIS-SC $\mathbb{W}_{\rm N\&I}(L)$ with arbitrary precision. However, it is worth noting that the complexity of the resulting description imposes a limitation on the achievable tolerance [35]. By using the RNIS-SC estimation method, the estimate of RDOA can be obtained. In the next section, the RDOA enlargement approach is introduced.

IV. Closed-loop RDOA Enlargement and Controller Design

A. Closed-loop RDOA Enlargement

According to Algorithm (2), the estimate set $\hat{\mathbb{W}}_{\rm N\&I}(L)$ can be attained and the estimate of RDOA is given by $\text{proj}_x(\hat{\mathbb{W}}_{\rm N\&I}(L))$. Since various Lyapunov functions can lead to completely different RNIS-SCs $\hat{\mathbb{W}}_{\rm N\&I}(L)$ and their projections $\text{proj}_x(\hat{\mathbb{W}}_{\rm N\&I}(L))$, we aim to select the best Lyapunov function for which the related projection $\text{proj}_x(\hat{\mathbb{W}}_{\rm N\&I}(L))$ is the largest. To achieve this aim, we construct a set of Lyapunov candidate functions that are parameterized by sum-of-squares (SOS) polynomials [36]. Then, we formulate an optimization problem to select a proper Lyapunov function from the set. The estimate of RDOA can be obtained using the algorithm proposed in the previous subsection, and its volume is easy to calculate and compare for a given set of Lyapunov candidate functions [37]. Let $\mathfrak{m}(\text{proj}_x(\hat{\mathbb{W}}_{\rm N\&I}(L)))$ represent the Lebesgue measure, such as volume if the space is 3D Euclidean space. Let $\mathfrak{L}_{n,2d}$ represent a subset of SOS polynomials $\mathfrak{R}_{n,2d}$ [38], n is the number of variables, and its degree is less than $2d$. $\mathfrak{L}_{n,2d}$ is in the form of

$$\mathfrak{L}_{n,2d} = \left\{ L \in \mathfrak{R}_{n,2d} \middle| L(x) = S_d^T(x) P^T P S_d(x), x \in \mathbb{R}^n \right\},$$

where $P \in \mathbb{R}^{r \times r}$ is a matrix,

$$S_d(x) = (x_{(1)}, \cdots, x_{(n)}, x_{(1)}x_{(2)}, \cdots, x_{(n)}^d) \in \mathbb{R}^r,$$

$r = \binom{n+d}{d} - 1$ denotes the dimension of matrix P.

To search for a proper function from a set of Lyapunov candidate functions that maximizes the estimate of RDOA, we define the optimization problem as follows

$$\max_{L \in \mathfrak{L}_{n,2d}} \mathfrak{m}(\text{proj}_x(\hat{\mathbb{W}}_{\rm N\&I}(L))). \tag{39}$$

Remark 6. *For a given function L, according to Algorithm 2, we know that* $\text{proj}_x(\hat{\mathbb{W}}_{\rm N\&I}(L))$ *is a set of hyperrectangles, and its volume is easy to calculate because each hyperrectangle is known.*

The function L is chosen from the set of Lyapunov candidate functions, and the parameter $P \in \mathbb{R}^{r \times r}$ is our primary focus since $L(x) = S_d^T(x) P^T P S_d(x)$. Applying Algorithm 2 yields the estimation of $\hat{\mathbb{W}}_{\rm N\&I}(L)$. Let $m : \mathbb{R}^{r \times r} \to \mathbb{R}$ be defined as

$$m(P) = \mathfrak{m}(\text{proj}_x(\hat{\mathbb{W}}_{\rm N\&I}(L))). \tag{40}$$

By utilizing the function defined in (40), the optimization expression (39) can be reformulated as an equivalent expression:

$$\max_{P \in \mathbb{R}^{r \times r}} m(P). \tag{41}$$

Due to the difficulty of obtaining the analytical expression of (40), (41) cannot be solved using traditional optimization approaches such as gradient descent algorithms. We observe that evaluating $m(P)$ is easier. Therefore, we use meta-heuristic optimization approaches to handle (41). The advantage of meta-heuristic optimization approaches is that the objective functions only need to be evaluable. Some popular meta-heuristic optimization approaches include the

particle swarm optimization (PSO) method [39], evolution strategies [40], and differential evolution algorithm [41]. Since there are many publications on meta-heuristic optimizers [42], we omit a more detailed explanation of these approaches. The convergence time of the meta-heuristic algorithm is not guaranteed. In practice, we set termination conditions to limit the algorithm's running time, such as convergence to a specific threshold or reaching the maximum iteration count. Therefore, its actual convergence time and enlargement results are unknown. However, for a given system, our approach first determines the Lyapunov function and then designs the controller, as shown in the Algorithm 3. Therefore, the convergence time of the meta-heuristic algorithm does not affect the time it takes to stabilize the system through the controller.

B. Controller Design

As stated in Corollary 1, there exists a small-scale domain $\mathbb{X}_\alpha$ around $0 \in \mathbb{R}^n$ that is not a subset of $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L))$, but there must be some hyperrectangles near the origin $0 \in \mathbb{R}^{n+m}$ that are part of $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$. Actually, the volume of the small-scale domain $\mathbb{X}_\alpha$ depends on the tolerance ϵ. Since the system(1) is assumed to be controllable around the origin $0 \in \mathbb{R}^{n+m}$, all states in the small-scale domain $\mathbb{X}_0$ can be stabilized by some existing linear controllers, and $\mathbb{X}_\alpha \subseteq \mathbb{X}_0$. Consequently, the estimate of RDOA is in the form of $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)) \cup \mathbb{X}_0$.

Let $Kx, K \in \mathbb{R}^{m\times n}$ represent the linear controller that can be obtained by linearizing of system (1). According to Theorem 1, any controller which belongs to the $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$ can robustly stabilize the system (1). Let $\tilde{\mu}: \mathbb{R}^n \to \mathbb{R}^m$ represent the controller when $x \in \mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L))$. $\tilde{\mu}$ matches up to

$$\forall x \in \mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)), (x, \tilde{\mu}(x)) \in \hat{\mathbb{W}}_{\mathrm{N\&I}}(L).$$

By combining these two closed-loop controllers, the unified controller is in the form

$$\mu(x) = \begin{cases} Kx, & \text{if } x \in \mathbb{X}_0 \\ \tilde{\mu}(x), & \text{if } x \in \mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)) \end{cases}. \tag{42}$$

Note that we assume the linear feedback controller $\mu = Kx$ is given. The process of estimating RNIS-SC $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$ is already a controller design process. The difference from other methods is that we obtain a set of controllers. To examine the effectiveness of our methods, we propose the following specific continuous controller design approach for a given $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$.

First, for a given RNIS-SC $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$, a training set $\mathbb{W}_{\mathrm{tran}} = \{(x^i, u^i), i = 1, ..., N\} \subseteq \hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$ is selected by uniformly sampling in the state-control space. For the training set $\mathbb{W}_{\mathrm{tran}}$, some function estimation methods can be used to obtain the specific controller $\tilde{\mu}$, such as interpolation, Gaussian processes regression. The output of the controller $\tilde{\mu}$ is restricted in RNIS-SC. Note that in the process of adding constraints to the controller, the restriction $\mu(0) = 0$ is also included. Then, the controller $\tilde{\mu}$ acquired from the function estimator can be assured to satisfy $\forall x \in \mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)), (x, \tilde{\mu}(x)) \in \hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$, see also [20]. For the linear feedback controller, we assume the closed-loop gain K is given. If K is unknown, for a given system, we can linearize the system via the Taylor series expansion of functions around the equilibrium point. Then the closed-loop gain K can be obtained by solving the discrete-time algebraic Riccati equation (DARE) equation.

The global algorithm that includes a controller design and an optimization routine is introduced in the Algorithm 3.

Algorithm 3 Full implementation for obtaining controller

1: **Input** $\mathfrak{F}, [w_{\mathrm{init}}], \epsilon, \mathbb{X}_0, d, K$
2: Constructing the Lyapunov function set $\mathfrak{L}_{n,2d}$, n is the number of variables of $\mathfrak{F}$.
3: Obtaining the P^* by solving (41) based on a meta-heuristic optimization approach, e.g., PSO function in MATLAB: $P^* = particalswarm(fun, nvars, lb, ub)$, where fun is (40), $nvars$ is the number of the variables of P and lb, ub is the boundary of P. Note that $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$ in (40) is obtained via Algorithm 2.
4: Obtaining $L^*(x) = S_d^T(x) P^{*T} P^* S_d(x)$
5: Obtaining $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L^*) := \mathrm{RNISEVIA}(\mathfrak{F}, L^*, [w_{\mathrm{init}}], \epsilon)$
6: Obtaining the unified controller $\mu(x)$ with Kx and $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L^*))$ by (42).
7: **Output** $\mu(x)$

V. Illustrative Examples

A. Nonlinear system with uncertainty

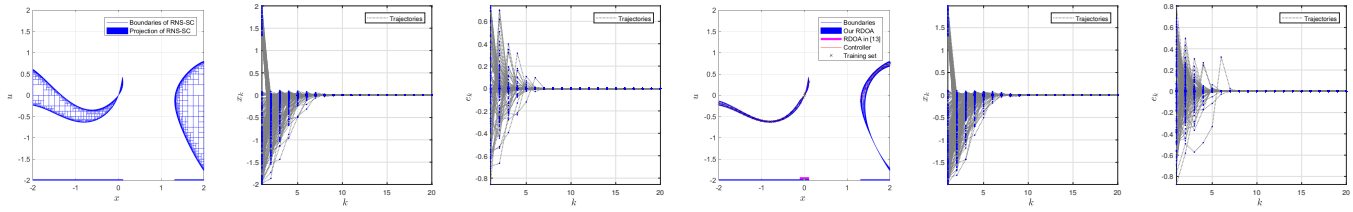
Consider $\mathfrak{F}(x,u) = f(x,u,\Omega), \Omega = [-\delta(x,u), \delta(x,u)]$ where:

$$\begin{aligned} f(x,u,\Omega) &= -\sin(2x) - xu - 0.2x - u^2 + u + \Omega, \\ \delta(x,u) &= 1 - \exp\left(-0.5(x^2+u^2)\right), \end{aligned}$$

the function $\delta: \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ denotes the modelling error bound, the interested domain is $\mathbb{W}_{x,u} = [-2,2] \times [-2,2]$.

The system under study is identical to that presented in [10], [13], and the modelling error bound Ω is also the same as that used in [20] for comparison purposes. Various interval software packages that implement interval operations, such as IBEX-LIB [43] and Rump's INTLAB [44], are available, we utilized the former for this work. To reduce the conservatism of interval arithmetic, we split the intervals, which leads to an exponential increase in complexity corresponding to the number of state variables. In Appendix B, we provide the computation code of our method for reference to other systems.

1) Stabilization: Let the Lyapunov function be $L(x) = x^2$. In Algorithm 2, the tolerance ϵ is set to $\epsilon = 1e{-}4$. By applying step 3 of Algorithm 2, we obtain the internal estimation $\hat{\mathbb{W}}_{\mathrm{N}}(L)$ of the RNS-SC $\mathbb{W}_{\mathrm{N}}(L)$, which is shown in Figure 1 (a) and represented by blue rectangles. The projection of $\hat{\mathbb{W}}_{\mathrm{N}}(L)$ onto the control space, denoted as $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N}}(L)) = [-2, -1.25 \times 10^{-4}] \cup [1.25 \times 10^{-4}, 0.105] \cup [1.315, 2]$ is represented by blue line segments in state space. By applying Algorithm 2, The internal estimation $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$ of RNIS-SC $\mathbb{W}_{\mathrm{N\&I}}(L)$ is obtained which is shown in Figure 1 (d) represented by blue rectangles. The projection of $\mathbb{W}_{\mathrm{N\&I}}(L)$



(a) (b) (c) (d) (e) (f)

Fig. 1: (a) Internal estimation $\hat{\mathbb{W}}_{\mathrm{N}}(L)$ and $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N}}(L))$. (b, c) The 200 state trajectories of x_k and actual model error trajectories e_k. (d) RNIS-SC $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$, controller $\tilde{\mu}$ and $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L))$. (e, f) The 200 state trajectories of x_k of $\tilde{\mu}$ and actual model error trajectories e_k of $\tilde{\mu}$.

along the control space $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)) = [-2, -1.25 \times 10^{-4}] \cup [1.25 \times 10^{-4}, 0.105] \cup [1.315, 2]$ is also represented by blue line segment in state space. The small-scale domain $\mathbb{X}_0$ in (42), which is represented by a green line segment in state space, is $[-0.03, 0.03]$. Consequently, the RDOA estimate we acquire is $[-2, 0.105] \cup [1.315, 2]$. The RDOA estimates obtained from other methods in [13], [18] are constrained by the level set of $\mathbb{X}_{\mathrm{ls}}(L, 1.17 \times 10^{-2})$. Their RDOA estimation is $[-0.108, 0.108]$, illustrated in Figure 1 (d) by a magenta line segment in the state space. Compared to these methods, our result is broader. We set $K = 1.8357$ for the linear controller in (42). The linear controller is represented by the green straight line between the boundaries of the small-scale domain $\mathbb{X}_0$ in Figure 1 (d). To confirm that the system (1) can be stabilized by any controller belonging to $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$, we randomly select 200 initial states from $[-2, 0.105] \cup [1.315, 2]$. For each state, we randomly select the control input from $\mathbb{U}(x)$, where $\mathbb{U}(x) = \{u \in \mathbb{R} | (x, u) \in \hat{\mathbb{W}}_{\mathrm{N\&I}}(L)\}$ is the controller set. The 200 state trajectories of the simulation are shown in Figure 1 (b). The 200 model error trajectories $e_k \in [-\delta(x_k, u_k), \delta(x_k, u_k)]$ are shown in Figure 1 (c). To obtain the controller $\tilde{\mu}$ in (42), we choose a training dataset represented by black $'\times'$s in Figure 1 (d). We utilize the Gaussian processes regression method to obtain the nonlinear controller $\tilde{\mu}$, which is represented by a red line in Figure 1 (d). We choose the same 200 initial states to examine the controller $\tilde{\mu}$. The state and actual model error trajectories of the simulation are shown in Figure 1 (e,f). All trajectories of both controllers converge to the origin.

As can be observed from the Figure 1 (b,e), no state belongs to $(0.105, 1.315) \subseteq \mathbb{R}$. This confirms that the uniting set $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L) \cup \{(x, u) | u = 1.8357x, x \in \mathbb{X}_0\}$ is an invariant set.

2) Enlarge the closed-loop RDOA: Based on the given system and the number of state-control space dimensions, we have $n = 1$, $d = 2$, and $P \in \mathbb{R}^{2\times 2}$. Therefore, the set of Lyapunov candidate functions in (39) is

$$\mathfrak{L}_{1,4} = \left\{ L \in \mathfrak{R}_{1,4} \middle| L(x) = (x, x^2)^T P^T P (x, x^2), x \in \mathbb{R} \right\}$$

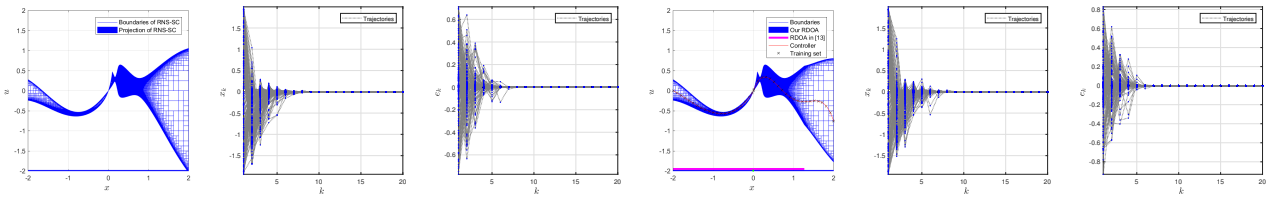
We use the PSO method to solve the optimization problem (41). The appropriate Lyapunov function acquired by the PSO method is $L^*(x) = 1.1285x^2 + 2.3112x^3 + 1.5328x^4$. By applying step 3 of Algorithm 2, we obtain the internal estimation $\hat{\mathbb{W}}_{\mathrm{N}}(L^*)$ of the RNS-SC $\mathbb{W}_{\mathrm{N}}(L^*)$, which is represented by blue rectangles in Figure 2 (a). By applying Algorithm2, we also obtain the internal estimation $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L^*)$ of the RNIS-SC $\mathbb{W}_{\mathrm{N\&I}}(L^*)$, which is represented by blue rectangles in Figure 2 (d). The projection $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L^*)) = [-2, -1.25\times 10^{-4}] \cup [1.25\times 10^{-4}, 2]$, is represented by the blue line segment in state space. The small-scale domain (42) is $[-0.03, 0.03]$, which is represented by the green line segment in state space. Therefore, we derive an RDOA estimate of $[-2, 2]$. The RDOA estimation in the approach introduced in [13] is limited by the level set $\mathbb{X}_{\mathrm{ls}}(L, 10.5408)$ and the RDOA result is $[-2, 1.27]$, depicted as a magenta line segment in the state space of Figure 2 (d). Our findings offer a wider range in comparison to this method. The value of K for the linear controller in (42) is also $K = 1.8357$. The linear controller is represented by the green dashed line between the boundaries of the small-scale domain $\mathbb{X}_0$ in Figure 2 (d). To confirm that the system (1) can be stabilized by any controller belonging to $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L^*)$, we randomly select 200 initial states from $[-2, 2]$. For each state, we choose the control input in the same way as in the previous subsection. The 200 state trajectories and actual model error e_k trajectories of the closed-loop system are shown in Figure 2 (b,c). To obtain the controller $\tilde{\mu}$ in (42), we choose a training dataset represented by black $'\times'$s in Figure 2 (d). We utilize the Gaussian processes regression method to obtain the nonlinear controller $\tilde{\mu}$, which is represented by a red line in Figure 2 (d). We choose the same 200 initial states to examine the controller $\tilde{\mu}$. The 200 state trajectories and actual model error e_k of the simulation are shown in Figure 2 (e,f). All trajectories of both controllers converge to the origin.

B. System subject to actuator saturation

In this subsection, we are interested in the control of the example subject to actuator saturation where

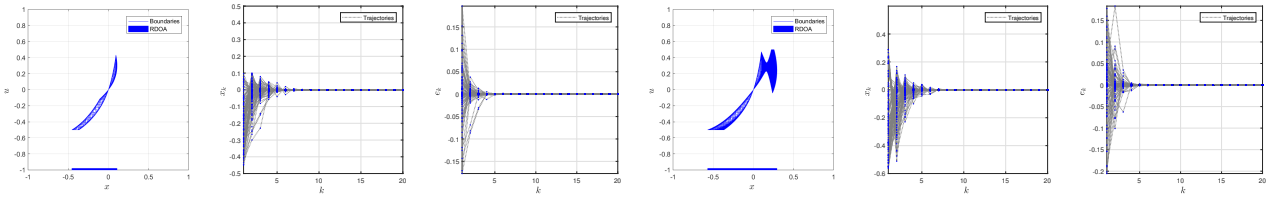
$$f(x, u, \Omega) = -\sin(2x) - x\mathrm{sat}(u) - 0.2x - \mathrm{sat}(u)^2 + \mathrm{sat}(u) + \Omega, \delta(x, u) = 1 - \exp\left(-0.5(x^2 + \mathrm{sat}(u)^2)\right).$$

The function $\mathrm{sat}()$ is the standard saturation function, here we set $\mathrm{sat}(u) = \min\{0.5, \max\{-0.5, u\}\}$ for this example. For the Lyapunov function $L(x) = x^2$, we use the same tolerance $\epsilon = 1e-4$. By applying Algorithm 2, The internal estimation $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$ of RNIS-SC $\mathbb{W}_{\mathrm{N\&I}}(L)$ is obtained which is shown in Figure 3 (a) represented by blue rectangles. As a result,



(a) (b) (c) (d) (e) (f)

Fig. 2: (a) Internal estimation $\hat{\mathbb{W}}_{\mathrm{N}}(L^*)$ and $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N}}(L^*))$. (b, c) The 200 state trajectories of x_k and actual model error trajectories e_k. (d) RNIS-SC $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L^*)$, controller $\tilde{\mu}$ and $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L^*))$. (e, f) The 200 state trajectories of x_k of $\tilde{\mu}$ and actual model error trajectories e_k of $\tilde{\mu}$.



(a) (b) (c) (d) (e) (f)

Fig. 3: For system subject to actuator saturation: (a) $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$ and the estimate of RDOA $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L))$. (b, c) The 100 state trajectories of x_k and 100 actural model error trajectories e_k. (d) $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L^*)$ and the estimate of RDOA $\mathrm{proj}_x(\hat{\mathbb{W}}_{\mathrm{N\&I}}(L^*))$. (e, f) The 100 state trajectories of x_k and 100 model error trajectories e_k.

we obtain an estimate of RDOA, which is $[-0.455, 0.105]$. The estimates of RDOA from the other methods in [13], [18] are limited by the level set of $\mathbb{X}_{\mathrm{ls}}(L, 1.17 \times 10^{-2})$. Their RDOA result is $[-0.108, 0.108]$. To confirm that the system (1) can be stabilized by any controller belonging to $\hat{\mathbb{W}}_{\mathrm{N\&I}}(L)$, we randomly select 100 initial states from $[-0.455, 0.105]$. For each state, we randomly select the control input from $\mathbb{U}(x) \subseteq \mathbb{R}$. The 100 state trajectories and actual model error e_k of the simulation are shown in Figure 3 (b,c). Figures 3 (d,e,f) show the same result for the Lyapunov function $L^*(x) = 1.1285x^2 + 2.3112x^3 + 1.5328x^4$. The estimate of RDOA is $[-0.57, 0.293]$. In the other method which is limited by the level set of $\mathbb{X}_{\mathrm{ls}}(L, 0.1004)$, the RDOA results are $[-0.57, 0.238]$. All trajectories of both controllers converge to the origin, and for the system subject to actuator saturation, our RDOA results are also broader. It is also observed that the larger the range of inputs, the more effective our method becomes.

C. Discretized continuous system

For continuous systems, a discretizing procedure must be applied in order to implement our method. The selection of the sampling time can significantly impact the overall performance of a controlled system. To conduct an analysis, a benchmarking nonlinear continuous system is employed. Consider the nonlinear continuous system from [47]:

$$\dot{x} = f(x, u) = x(1 - x^2) + u, \tag{43}$$

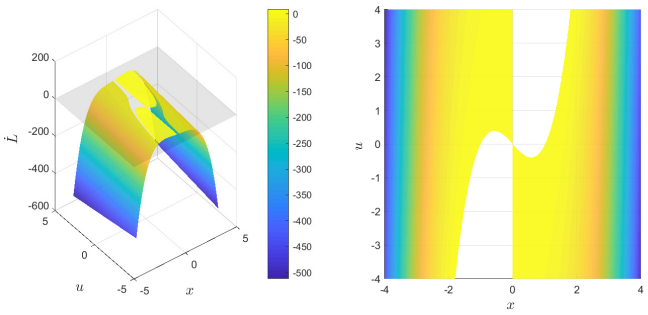
where x and u denote the state and the control input, respectively. Firstly, we discretize the continuous model (43). We consider two classic discretization methods, the Euler method and the fourth-order Runge-Kutta method with a sampling time Δt. Then, the discretized system via the Euler method is

$$x_{k+1} = x_k + \Delta t f(x_k, u_k), \tag{44}$$

where $f(x_k, u_k) = x_k(1 - x_k^2) + u_k$. The discretized system via the fourth-order Runge-Kutta method is

$$x_{k+1} = x_k + \frac{\Delta t}{6}(h_1 + 2h_2 + 2h_3 + h_4), \tag{45}$$

where $h_1 = f(x_k, u_k), h_2 = f(x_k + \frac{\Delta t}{2}h_1, u_k), h_3 = f(x_k + \frac{\Delta t}{2}h_2, u_k), h_4 = f(x_k + \Delta t h_3, u_k)$. Consider the Lyapunov function $L = x^2$, $\mathbb{W}_{x,u} = [-4, 4] \times [-4, 4]$, $\epsilon = 0.01$ and $\mathbb{X}_0 = [-0.2, 0.2]$. The value of the derivative of the Lyapunov function $\dot{L} = \frac{\partial L}{\partial x} f(x, u)$ is shown in Figure 4 (a) and the negative region $\dot{L} < 0$ is shown in Figure 4 (b). By applying our method, we obtain six different results with respect to the variation of Δt, which is shown in Figure 5 (a-c) for model (44), (d-f) for model (45). The set composed of blue grids is RNIS-SC, and the projection of the set on the x-axis represents the estimated DOA denoted as the blue line segment. For each RNIS-SC, we randomly select the control input from RNIS-SC, and the selected control input is held constant during the sampling time. We use the $ode45$ function in MATLAB to simulate the continuous system (43) and the 100 state trajectories of the simulation are shown in Figure 5 (g-l). From Figure 5, we can see that different discretization methods lead to different estimation results. The estimated RNIS-SC via the Runge-Kutta discretization method has a wider range. However, from Figure 5 (d, e), we can see that due to a too large sampling time, some parts of the estimated RNIS-SC are in the region $\dot{L} \geq 0$ for system (43), these



(a) (b)

Fig. 4: (a) For continuous system (43), the value of the derivative of the Lyapunov function $\dot{L}$, the gray plane represents $\dot{L} = 0$. (b) The colorful region denotes $\dot{L} < 0$.

parts lead to bad control performance as shown in Figure 5 (j, k). As the sampling time decreases, the estimation results of the two discretization models both tend towards the negative definite region of the continuous system. We believe the proper sampling time should be less than 0.1 for the continuous system (43). For different continuous systems, the selection of sampling time and sampling methods have varying impacts on performance. It is generally believed that a shorter sampling time is better. Our method can serve as a tool for analysis in this regard.

VI. Conclusion

We have introduced an interval analysis approach to obtain a set of unstructured robust controllers for difference inclusions. The presented results show the usefulness of our approach in the following ways: 1) By utilizing a specific Lyapunov function, we have successfully obtained a wider estimate of RDOA in comparison to the Lyapunov function constrained by the level set. This achievement was made with respect to the RNIS-SC. 2) The estimate of RDOA is totally different for different Lyapunov functions. After formulating the resolvable optimization problem, we expanded the estimate of RDOA by selecting an appropriate Lyapunov function. The effectiveness of the proposed method is validated by the improved suitability of the enlarged RDOAs. For difference inclusions, the next state x_{k+1} can be considered in a bounded interval vector. Hence, the proposed RNIDEVIA algorithm, which is built upon interval arithmetic, exhibits great potential in tackling this problem effectively. Furthermore, the utilization of RNIS-SC and the representation of the estimated RDOA in hyper-rectangle form ensure the rigor and reliability of the computed result. The controller can be designed directly in RNIS-SC, which might be helpful in the field of optimal control.

Appendix A
Proof of Lemma 3

Depending on the mapping $\mathcal{I}$ definition (15), it is obvious that

$$\mathcal{I}^{i+1}(\mathbb{W}) = \left\{(x,u) \in \mathcal{I}^i(\mathbb{W}) \middle| \mathfrak{F}(x,u) \in \mathrm{proj}_x(\mathcal{I}^i(\mathbb{W}))\right\}. \quad (46)$$

Rely on (46), we have $\mathcal{I}^{i+1}(\mathbb{W}) \subseteq \mathcal{I}^i(\mathbb{W}), i \geq 1$. It can be concluded that sets sequence $\{\mathcal{I}^i(\mathbb{W})\}$ are monotonically decreasing. Furthermore, $\mathcal{I}^i(\mathbb{W}), i \geq 1$ is closed, there exists a set limit of $\mathcal{I}^i(\mathbb{W})$ (refer to [45], pp. 111). First, we want to prove the compactness of $\mathcal{I}^\infty(\mathbb{W})$. The set $\mathbb{W}$ is a compact set, meaning it is both bounded and closed in the metric space [46] (pp.77). Hence the projection sets $\mathrm{proj}_x(\mathbb{W})$ and $\mathrm{proj}_u(\mathbb{W})$ are both bounded. The difference inclusions $\mathfrak{F}$ is u.s.c. over $\mathbb{R}^n \times \mathbb{R}^m$, this implies that the image of $\mathbb{W}$ under the mapping $\mathcal{I}$ is bounded. Next, we aim to show that $\mathcal{I}(\mathbb{W})$ is closed. To begin, consider a sequence (x_i, u_i) that converges within $\mathcal{I}(\mathbb{W})$. As both $\mathrm{proj}_x(\mathbb{W})$ and $\mathrm{proj}_u(\mathbb{W})$ are compact sets, we can conclude that the limit points $\bar{x} = \lim_{i\to\infty} x_i \in \mathrm{proj}_x(\mathbb{W})$ and $\bar{u} = \lim_{i\to\infty} u_i \in \mathrm{proj}_u(\mathbb{W})$. Let $y_i = \mathfrak{F}(x_i, u_i)$, we have $y_i \in \mathrm{proj}_x(\mathcal{I}(\mathbb{W}))$. By utilizing the upper semi-continuity of $\mathfrak{F}$, it can be obtained that $\lim_{i\to\infty} y_i \subseteq \lim_{i\to\infty} \mathfrak{F}(x_i, u_i) \subseteq \mathfrak{F}(\lim_{i\to\infty} x_i, \lim_{i\to\infty} u_i) \subseteq \mathfrak{F}(\bar{x}, \bar{u}) \subseteq \mathrm{proj}_x(\mathcal{I}(\mathbb{W}))$. Consequently, $(\bar{x}, \bar{u}) \in \mathcal{I}(\mathbb{W})$, it follows that $\mathcal{I}(\mathbb{W})$ is closed. Hence, $\mathcal{I}(\mathbb{W})$ is a compact set. Furthermore, it can be inferred that $\mathcal{I}^\infty(\mathbb{W})$ is a compact set as well. To prove that $\mathcal{I}^\infty(\mathbb{W})$ is invariant (Definition 2), we need to demonstrate that $\forall (x,u) \in \mathcal{I}^\infty(\mathbb{W}) = \bigcap_{i=0}^{\infty} \mathcal{I}^{i+1}(\mathbb{W}), \mathfrak{F}(x,u) \subseteq \mathrm{proj}_x(\mathcal{I}^\infty(\mathbb{W})) = \mathrm{proj}_x(\bigcap_{i=0}^{\infty} \mathcal{I}^{i+1}(\mathbb{W}))$.

For all $i \geq 1$, by Definintion 4, we have $\forall (x,u) \in \mathcal{I}^{i+1}(\mathbb{W}), \mathfrak{F}(x,u) \in \mathrm{proj}_x(\mathcal{I}^i(\mathbb{W}))$, which implies $\forall (x,u) \in \bigcap_{i=0}^{\infty} \mathcal{I}^{i+1}(\mathbb{W}), \mathfrak{F}(x,u) \in \bigcap_{i=0}^{\infty} \mathrm{proj}_x(\mathcal{I}^i(\mathbb{W}))$. Next, we aim to demonstrate that $\bigcap_{i=0}^{\infty} \mathrm{proj}_x(\mathcal{I}^i(\mathbb{W})) = \mathrm{proj}_x(\bigcap_{i=0}^{\infty} \mathcal{I}^i(\mathbb{W}))$. Since $\mathrm{proj}_x(\bigcap_{i=0}^{\infty} \mathcal{I}^i(\mathbb{W})) \subseteq \mathrm{proj}_x(\mathcal{I}^i(\mathbb{W})), \forall i \geq 0$, in general that $\mathrm{proj}_x(\bigcap_{i=0}^{\infty} \mathcal{I}^i(\mathbb{W})) \subseteq \bigcap_{i=0}^{\infty} \mathrm{proj}_x(\mathcal{I}^i(\mathbb{W}))$. Then we aim to prove the reverse inclusion.

Suppose $x \in \bigcap_{i=0}^{\infty} \mathrm{proj}_x(\mathcal{I}^i(\mathbb{W}))$. This implies that there exists a sequence $\mathfrak{U}_i = \{u \in \mathbb{R}^m | \exists x \in \mathrm{proj}_x(\mathcal{I}^i(\mathbb{W})), (x,u) \in \mathcal{I}^i(\mathbb{W})\}$ so that $(x, u_j) \in \mathcal{I}^i(\mathbb{W}), u_j \in \mathfrak{U}_i$. Given that $\mathcal{I}(\mathbb{W})$ and $\mathrm{proj}_u(\mathbb{W})$ are compact, according to the compactness definition [46], we can find a convergent subsequence $\mathfrak{U}_\infty \subseteq \mathfrak{U}_i$. This subsequence $\mathfrak{U}_\infty$ contains at least one limit point $\bar{u} = \lim_{j\to\infty} u_j$, $\bar{u} \in \mathfrak{U}_\infty$. As a consequence, we have $(x, \bar{u}) \in \mathcal{I}^\infty(\mathbb{W}) = \bigcap_{i=0}^{\infty}(\mathcal{I}^i(\mathbb{W}))$, implying that $x \in \mathrm{proj}(\bigcap_{i=0}^{\infty}(\mathcal{I}^i(\mathbb{W})))$. Consequently, $\bigcap_{i=0}^{\infty} \mathrm{proj}_x(\mathcal{I}^i(\mathbb{W})) \subseteq \mathrm{proj}_x(\bigcap_{i=0}^{\infty} \mathcal{I}^i(\mathbb{W}))$, this completes the proof.

Appendix B

The example is computed by Python binding of IBEX-LIB [43]. The entire code and computation time are available at https : //github.com/CharlieLuuke/IDRCS.

References

[1] L. Yin and Y. Shen, "Robust filtering of discrete-time linear systems with correlated process and measurement noises," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 3, pp. 1008–1020, Mar. 2020.

[2] M. G. Safonov, "Origins of robust control: Early history and future speculations," *Annu. Rev. in Control*, vol. 36, no. 2, pp. 173–181, Oct. 2012.

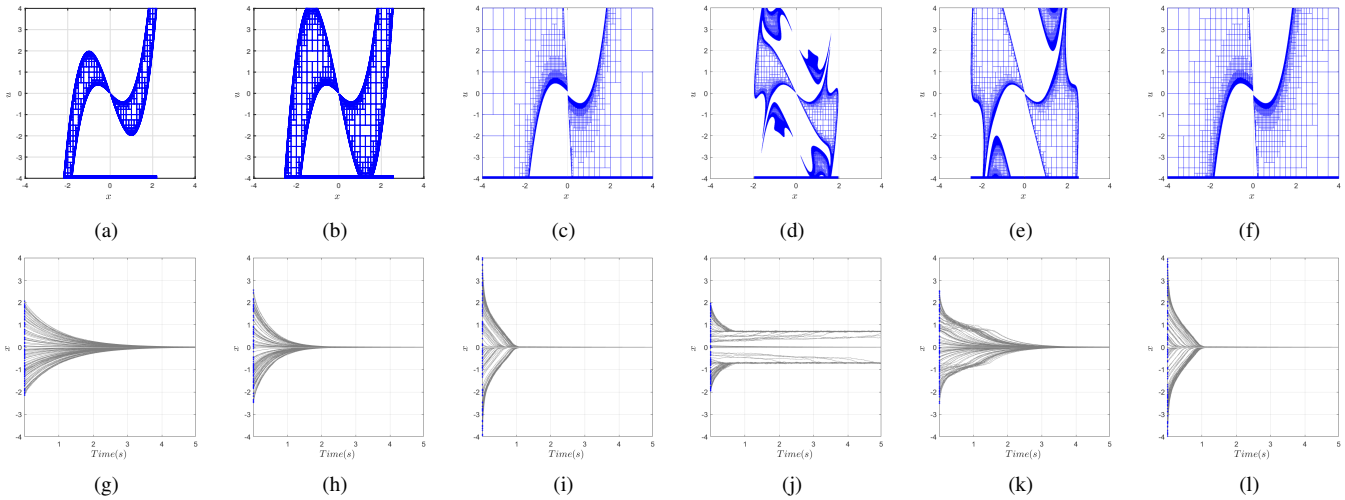


Fig. 5: (a-c) For discretized system (44), the estimated RNIS-SC with $\Delta t = 1, 0.5, 0.1$, respectively. (d-f) For discretized system (45), the estimated RNIS-SC with $\Delta t = 1, 0.5, 0.1$, respectively. (g-l) The 100 state trajectories of x_k for the controller were randomly selected from the obtained RNIS-SC corresponding to (a-f).

[3] I. R. Petersen and R. Tempo, "Robust control of uncertain systems: Classical results and recent developments," *Automatica*, vol. 50, no. 5, pp. 1315–1335, Mar. 2014.

[4] S. P. Bhattacharyya, "Robust control under parametric uncertainty: An overview and recent results," *Annu. Rev. in Control*, vol. 44, pp. 45–77, May 2017.

[5] M. J. Corless, *Robust Stability Analysis and Controller Design with Quadratic Lyapunov Functions*. Berlin, Germany: Springer-Verlag, 1993.

[6] T. Basar and J. W. Helton, H_∞ *Optimal Control and Related Minimax Design Problems*. Boston, USA: Birkhauser, 1995.

[7] S. Xu, J. Lam, and X. Mao, "Delay-dependent H_∞ control and filtering for uncertain markovian jump systems with time-varying delays," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 9, pp. 2070–2077, Sep. 2007.

[8] R. A. Freeman and P. Kokotovic, *Robust Nonlinear Control Design: State-space and Lyapunov Techniques*. Boston, USA: Birkhauser, 2008.

[9] H. Zhang, L. Cui, X. Zhang, and Y. Luo, "Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2226–2236, Oct. 2011.

[10] Y. Li, C. Yang, Z. Hou, Y. Feng, and C. Yin, "Data-driven approximate Q-learning stabilization with optimality error bound analysis," *Automatica*, vol. 103, pp. 435–442, Feb. 2019.

[11] Y. Chen, M. Tanaka, K. Tanaka, and H. Wang, "Stability analysis and region-of-attraction estimation using piecewise polynomial Lyapunov functions: Polynomial fuzzy model approach," *IEEE Trans. on Fuzzy Syst.*, vol. 23, no. 4, pp. 1314–1322, Aug. 2015.

[12] S. Gering, L. Eciolaza, J. Adamy, and M. Sugeno, "A piecewise approximation approach to nonlinear systems: Stability and region of attraction," *IEEE Trans. on Fuzzy Syst.*, vol. 23, no. 6, pp. 2231–2244, Dec. 2015.

[13] Y. Li and Z. Hou, "Data-driven asymptotic stabilization for discrete-time nonlinear systems," *Syst. Control Lett.*, vol. 64, pp. 79–85, Feb. 2014.

[14] D. Han and M. Althoff, "Control synthesis for non-polynomial systems: A domain of attraction perspective," in *Proc. 54th IEEE Conf. Decision and Control,* Osaka, Japan, Dec. 2015, pp. 1160–1167.

[15] S. Y. Tsai, C. Wang, and C. Wu, "Stability analysis of autonomous ratio-memory cellular nonlinear networks for pattern recognition," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 2156–2167, Aug. 2010.

[16] X. Qiu, Z. Feng, C. Lu, and Y. Li, "Stabilization with closed-loop DOA enlargement: An interval analysis approach," 2021, *arXiv:1912.11775*.

[17] M. A. Davo, C. Prieur, M. Fiacchini, and D. Nesic, "Enlarging the basin of attraction by a uniting output feedback controller," *Automatica*, vol. 90, pp. 73–80, Apr. 2018.

[18] R. Swiatlak, B. Tibken, T. Paradowski, and R. Dehnert, "An interval arithmetic approach for the estimation of the robust domain of attraction for nonlinear autonomous systems with nonlinear uncertainties," in *Proc. Amer. Control Conf.,* Chicago, IL, USA, Jul. 2015, pp. 2679–2684.

[19] A. Goldsztejn and G. Chabert, "Estimating the robust domain of attraction for non-smooth systems using an interval Lyapunov equation," *Automatica*, vol. 100, pp. 371–377, Feb. 2019.

[20] Y. Li, C. Lu, Z. Hou, and Y. Feng, "Data-driven robust stabilization with robust domain of attraction estimate for nonlinear discrete-time systems," *Automatica*, vol. 119, p. 109031, Sep. 2020.

[21] S. Hui and S. H. Żak, "On the Lyapunov stability of discrete-time processes modeled by difference inclusions," *Syst. Control Lett.*, vol. 10, no. 3, pp. 207–209, Mar. 1988.

[22] J. M. Bravo, D. Limon, T. Alamo, and E. F. Camacho, "On the computation of invariant sets for constrained nonlinear systems: An interval arithmetic approach," *Automatica*, vol. 41, no. 9, pp. 1583–1589, Sep. 2005.

[23] Y. Li and J. Liu, "Invariance control synthesis for switched nonlinear systems: an interval analysis approach," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 2206–2211, Jul. 2018.

[24] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Berlin, Germany: Springer-Verlag, 2001.

[25] D. P. Bertsekas, "Infinite time reachability of state-space regions by using feedback control," *IEEE Trans. Autom. Control*, vol. 17, no. 5, pp. 604–613, Oct. 1972.

[26] M. Rungger and P. Tabuada, "Computing robust controlled invariant sets of linear systems," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3665–3670, Feb. 2017.

[27] L. Schäfer, F. Gruber, and M. Althoff, "Scalable computation of robust control invariant sets of nonlinear systems," *IEEE Trans. Autom. Control*, vol. early access, pp. 1–15, May. 2023.

[28] W. M. Haddad and V. Chellaboina, *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*. New Jersey, USA: Princeton University Press, 2008.

[29] S. V. Rakovic, E. C. Kerrigan, D. Q. Mayne, and J. Lygeros, "Reachability analysis of discrete-time systems with disturbances," *IEEE Trans. Autom. Control*, vol. 51, no. 4, pp. 546–561, Apr. 2006.

[30] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, Nov. 1999.

[31] J. Dugundji, *Topology*. Boston, USA: Allyn and Bacon, 1968.

[32] B. Franco and M. Stefano, *Set-Theoretic Methods in Control*. Boston, USA: Birkhäuser, 2008.

[33] S. Ocloo and W. Edmonson, "IIR filter adaptation using branch-and-bound: A novel approach," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 11, pp. 3393–3403, Dec. 2008.

[34] L. Jaulin and E. Walter, "Set inversion via interval-analysis for nonlinear bounded-error estimation," *Automatica*, vol. 29, no. 4, pp. 1053–1064, Jul. 1993.

[35] M. Kieffer, L. Jaulin, and E. Walter, "Guaranteed recursive non-linear state bounding using interval analysis," *Int. J. Adaptive Control Signal Process.*, vol. 16, no. 3, pp. 193–218, Jan. 2002.

[36] V. Powers and T. Wormann, "An algorithm for sums of squares of real polynomials," *J. Pure Applied Algebra*, vol. 127, no. 1, pp. 99–104, May. 1998.

[37] U. Topcu, A. Packard, and P. Seiler, "Local stability analysis using simulations and sum-of-squares programming," *Automatica*, vol. 44, no. 10, pp. 2669–2675, Oct. 2008.

[38] T. Ufuk, P. Andy, S. Peter, and B. Gary, "Help on SOS," *IEEE Control Syst. Mag.*, vol. 30, no. 4, pp. 18–23, Oct. 2010.

[39] K. Parsaei, R. Keshavarz, R. M. Boroujeni, and N. Shariati, "Compact pixelated microstrip forward broadside coupler using binary particle swarm optimization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 70, no. 12, pp. 5265–5274, Dec. 2023.

[40] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[41] K. R. Opara and J. Arabas, "Differential evolution: A survey of theoretical analyses," *Swarm Evol. Comput.*, vol. 44, pp. 546–558, Feb. 2019.

[42] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Struct. Multidisciplinary Optim.*, vol. 26, no. 6, pp. 369–395, Apr. 2004.

[43] C. Gilles. *Ibex: A C++ library for constraint processing over real numbers*. [Online]. Available: https://github.com/ibex-team/ibex-lib.

[44] S. Rump, "INTLAB - INTerval LABoratory," in *Developments in Reliable Computing*, Dordrecht, Netherlands: Kluwer Academic Publishers, 1999, pp. 77–104.

[45] R. T. Rockafellar and R. J. Wets, *Variational Analysis*. Berlin, Germany: Springer-Verlag, 2009.

[46] E. Kreyszig, *Introductory Functional Analysis with Applications*. Canada: John Wiley & Sons, 1978.

[47] E. Clemente, M. C. Rodríguez-Liñán, M. Meza-Sánchez, L. Monay-Arredondo, and L. Herrera, "A class of bounded and partially bounded nonlinear controllers for first and second order dynamical systems," *IEEE Control Syst. Lett.*, vol. 6, pp. 1028–1033, Jun. 2022.

Chaolun Lu received the Bachelor's degree from Zhejiang University of Technology in 2017. He was a joint Ph.D. student with the Laboratoire des Sciences du Numérique de Nantes, Nantes, France, from 2022 to 2023. Currently, he is pursuing the Doctor's degree at Zhejiang University of Technology, Hangzhou, China. His research interests are data-driven control, robust control and reinforcement learning.

Yongqiang Li received the Ph.D. degree in control theory and control engineering from Beijing Jiaotong University, Beijing, China, in 2014. Currently, he is working as an associate professor with the College of Information Engineering in Zhejiang University of Technology, Hangzhou, China. His research interests include non-linear control,optimal control, robotic control and reinforcement learning.

Alexandre Goldsztejn received the Engineer degree in computer science and mathematics from the Institut Supérieur d'Electronique et du Numérique, Lille, France, in 2001, and the Ph.D. degree in computer science from the University of Nice Sophia Antipolis, Nice, France, in 2005. He has spent one year as a Postdoctoral Fellow with the University of Central Arkansas, Conway, AR, USA, and the University of California, Irvine, CA, USA. Since 2007, he has been a full-time CNRS Researcher with the Laboratoire des Sciences du Numérique de Nantes, Nantes, France. His research interests include interval analysis and its applications to constraint satisfaction, nonlinear global optimization, robotics, and control.

Zhongsheng Hou (Fellow, IEEE) received the B.S. and M.S. degrees from the Jilin University of Technology, China, in 1983 and 1988, respectively, and the Ph.D. degree from Northeastern University, China, in 1994. He was a Post-Doctoral Fellow with the Harbin Institute of Technology, China, from 1995 to 1997, and a Visiting Scholar with Yale University, CT, USA, from 2002 to 2003. In 1997, he joined Beijing Jiaotong University, China, where he had been a distinguished Professor and the founding Director of the Advanced Control Systems Laboratory and the Head of the Department of Automatic Control. Currently, he is a Chair Professor with the School of Automation, Qingdao University, Qingdao, China. He is also the founding Director of the Technical Committee on Data Driven Control, Learning and Optimization (DDCLO), Chinese Association of Automation (CAA), and is a fellow of CAA. He is an IFAC Technical Committee Member on both "Adaptive and Learning Systems" and "Transportation Systems."

Yu Feng (Senior Member, IEEE) received the B.E. degree from Nanjing University of Science and Technology, Nanjing, China, in 2005, the M.S. degree from École Centrale de Nantes, Nantes, France, in 2008, and the diplôme d'Ingénieur and the Ph.D. degrees from École des Mines de Nantes (currently IMT Atlantique), Nantes, France, in 2007 and 2011, respectively, all in automatic control. From 2012 to 2013, he held the postdoctoral position at University of Windsor, Canada. Since 2013, he has been a faculty member in the Information Engineering College, Zhejiang University of Technology, Hangzhou, China, where he is a Professor. His current research interests include networked control systems, robust analysis and control for uncertainty systems, and applications of game theory and machine learning in decision-making. Dr. Feng has served as an Associate Editor for IEEE CSS Conference Editorial Board, and is member of IEEE CSS Technical Committee on Systems With Uncertainty.

Yuanjing Feng holds a Ph.D in control science and engineering from Xi'an Jiaotong University, M.S. in Mechanical design and theory from Northwest A&F University. Currently, he is working as a professor at Zhejiang University of Technology. His interests include data-driven modeling and optimization in the field of intelligence transportation system, medical image analysis.