Computation of Controlled Invariants for Nonlinear Systems: Application to Safe Neural Networks Approximation and Control

Adnane Saoud ∗ Ricardo G. Sanfelice ∗

∗ Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, USA (e-mail: asaoud, ricardo@ucsc.edu).

Abstract: In this paper, we consider the problem of computing multidimensional interval controlled invariants for nonlinear input-affine systems. We first present sufficient conditions for an interval to be controlled invariant. Then, we introduce the concept of local framers, based on which we present a sound algorithm to compute interval controlled invariants. Finally, we show how the proposed framework makes it possible to provide safety guarantees when using deep neural networks, either as a model or a controller of nonlinear systems. Illustrative examples are provided showing the merits of the proposed approach and its scalability properties.

Keywords: Invariance, nonlinear systems, framers, neural networks.

1. INTRODUCTION

Controlled invariance plays an important role in control theory (Blanchini and Miani [2008]). The notion of controlled invariance reflects the ability to control the system so that all trajectories initialized in a set remain there for all future time. Formulation and definitions of the concept of controlled invariance of a set are summarized in (Aubin [2009], Blanchini and Miani [2008]) for continuous-time and discrete-time systems, and in (Chai and Sanfelice [2020]) for hybrid systems. For continuous-time systems, different approaches have been presented in the literature to compute controlled invariants. In (Blanchini and Miani [2008]) controlled invariants are obtained as sublevel sets of Lyapunov-like functions. Controlled invariants for polynomial systems have been investigated using linear programming in (Korda et al. [2014]). For general nonlinear systems, polytopics controlled invariants are computed in (Cannon et al. [2003]) by embedding the nonlinear dynamics into linear ones. Other approaches have been proposed recently using symbolic control techniques (Saoud [2019], Tabuada [2009]).

In this paper, we are interested in the study of multidimensional interval (or simply interval) controlled invariants for a class of nonlinear input-affine systems. To the best of the authors knowledge, only monotone systems have been considered when dealing with interval controlled invariants. The authors in (Abate et al. [2009]) are dealing with monotone autonomous multi-affine systems and in (Meyer et al. [2016]) the authors present an approach for the computation of robust controlled invariants for monotone systems with inputs. In this paper, a new approach to compute interval controlled invariants that is applicable to a class of nonlinear input-affine systems is proposed. First, we present sufficient conditions for an interval to be controlled invariant. Then, we introduce the concept of local framer and present a sound algorithm to compute interval controlled invariants. Finally, we show how the proposed approach can be used to ensure invariance when using deep neural networks for the purpose of approximation and control of continuous-time nonlinear input-affine systems. To validate the practicality of the proposed we consider the control problem of a Kuramoto oscillator and an agent moving in a 2-D plane.

The remainder of this paper is organized as follows. In Section 2 we introduce the class of systems considered and sufficient conditions for an interval to be controlled invariant. In Section 3, we present applications of the proposed framework to safe deep neural networks approximation and control. Finally, Section 4 presents numerical results validating the merits of the proposed approach. Due to space constraints, the proofs are omitted and will be published elsewhere.

Notation: The symbols N, R, R≥0, and R≤0 denote the set of positive integer, real, non negative real, and non-positive real numbers, respectively. Given vectors \( x^a, x^b \in \mathbb{R}^n \), \( x^a \leq x^b \) stands for \( x^a_i \leq x^b_i \) for all \( i \in \{1, 2, \ldots, n\} \) and \( x^a < x^b \) stands for \( x^a_i < x^b_i \) for all \( i \in \{1, 2, \ldots, n\} \). Using this partial order, we define a multidimensional interval set as follows: for \( \underline{x}, \overline{x} \in \mathbb{R}^n \), with \( \underline{x} \leq \overline{x} \), \( [\underline{x}, \overline{x}] = \{x \in \mathbb{R}^n \mid \underline{x} \leq x \leq \overline{x}\} \) and \( (\underline{x}, \overline{x}] = \{x \in \mathbb{R}^n \mid \underline{x} < x \leq \overline{x}\} \). Given a map \( f : \mathbb{R}^n \to \mathbb{R}^n \) and a compact...
set \( A \subset \mathbb{R}^n \), \( f(A) = \{ y \in \mathbb{R}^m | \exists x \in A \text{ and } f(x) = y \} \) is the reachable set from the set \( A \) under the map \( f \). Given two sets \( A, B \subset \mathbb{R}^n \), the set \( A + B \) represents the Minkowski sum of the sets \( A \) and \( B \), which is defined as \( A + B = \{ a + b | a \in A, b \in B \} \).

For the system in (3), \( u_i \in U_i, i \in \{1,2,\ldots,n\} \), represents all input components with a direct influence on \( x_i \). With some abuse of notation, we rewrite the dynamics of each \( x_i, i \in \{1,2,\ldots,n\} \), in (3) as follows:

\[
\dot{x}_i = f_i(x_i, z_i) + g_i(x_i, z_i)u_i
\]

with \( z_i = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{n-1} \), \( i \in \{1,2,\ldots,n\} \). For \( i \in \{1,2,\ldots,n\} \), we define the linear map \( \pi_i, 1 : \mathbb{R}^n \to \mathbb{R} \) and \( \pi_i, 2 : \mathbb{R}^n \to \mathbb{R}^{n-1} \) such that for all \( x = (x_1, x_2, \ldots, x_n) \), \( \pi_i, 1(x) = x_i \) and \( \pi_i, 2(x) = z_i \).

In the following, we will denote \( \mathcal{X}_i = \pi_{i,1}(\mathcal{X}) \) and \( \mathcal{Z}_i = \pi_{i,2}(\mathcal{X}) \).

**Remark 1.** Let us mention that in some cases, even if Assumption 2 is not satisfied by the original system, one can find a change of coordinates ensuring Assumption 2 for the transformed system.

### 2.2 Sufficient conditions for controlled invariance

In this section, we provide a result for the computation of controlled invariants for the considered class of systems. First, we recall the concept of controlled invariant (Blanchini and Miani [2008]). In simple words, a controlled invariant set is a set for which there exists a controller such that if the state of the system is initialized in this set then its solutions remain there for all time.

**Definition 1.** The set \( K \subset \mathcal{X} \) is said to be a controlled invariant for the system \( \Sigma \) in (1) if there exists a controller \( \kappa : K \to U \) such that, for all \( x_0 \in K \), each solution \( \Phi(x_0,t) : \mathbb{R}_{\geq 0} \to \mathcal{X} \) satisfies \( \Phi(x_0,t) \in K \) for all \( t \in \mathbb{R}_{\geq 0} \). When this property holds, \( \kappa \) is said to be an invariant controller for the system \( \Sigma \) and the set \( K \).

The following result provides a characterization of controlled invariance of a set \( K \) given by an interval. The invariance controller is designed by exploiting the sign of the nonlinear functions \( f \) and \( g \) in the boundary of \( K \).

**Theorem 1.** Suppose that the nonlinear system \( \Sigma \) defined in (1) satisfies Assumption 2. Let \( K \subset \mathcal{X} \) be an interval defined as \( K = [\underline{x}, \overline{x}] \), where \( \underline{x} < \overline{x} \). The set \( K \) is a controlled invariant for the system \( \Sigma \) if, for each \( i \in \{1,2,\ldots,n\} \), there exist \( \underline{u}_i, \overline{u}_i \in U_i \) such that

\[
\begin{align*}
&f_i(\underline{x}, [\underline{z}_i; \overline{z}_i]) + g_i(\underline{x}, [\underline{z}_i; \overline{z}_i])\underline{u}_i \subset \mathbb{R}_{\geq 0} \\
&f_i(\overline{x}, [\underline{z}_i; \overline{z}_i]) + g_i(\overline{x}, [\underline{z}_i; \overline{z}_i])\overline{u}_i \subset \mathbb{R}_{\leq 0}
\end{align*}
\]

where \( f_i \) and \( g_i \) are given as in (4).

A graphical representation of the conditions in Theorem 1 is provided in Figure 1.

**Remark 2.** At times, we may have that, for some \( i \in \{1,2,\ldots,n\} \), \( g_i(x) = 0 \) for all \( x \in \mathcal{X} \). In such a case, conditions (5) and (6) reduce to, respectively, \( f_i(\underline{x}, [\underline{z}_i; \overline{z}_i]) \subset \mathbb{R}_{\geq 0} \) and \( f_i(\overline{x}, [\underline{z}_i; \overline{z}_i]) \subset \mathbb{R}_{\leq 0} \), whose satisfaction solely depends on the properties of \( f \) alone.

Theorem 1 provides theoretical guidelines to compute interval controlled invariants. The main difficulty is to provide an efficient algorithmic procedure to check conditions (5) and (6). Indeed, the difficulty lies in the fact that the sets \( f_i(\underline{x}, [\underline{z}_i; \overline{z}_i]), g_i(\underline{x}, [\underline{z}_i; \overline{z}_i]), f_i(\overline{x}, [\underline{z}_i; \overline{z}_i]), \).
The result in Proposition 1 leads to the following algorithm.

**Algorithm 1: Controlled invariance**

**Input:** Functions \( f \) and \( g \) of a system \( \Sigma \) as in (1) and an interval \( K = [\mathbf{x}, \mathbf{z}] \) with dimension \( n \).

**Output:** True, if \( K \) is a controlled invariant for \( \Sigma \).

**begin**

for \( i = 1 : n \)

Compute local framers:

\[
\begin{align*}
\int_{\mathbf{x}_i} (x_i, \mathbf{z}_i, \mathbf{z}_i), \quad \mathbf{f}_i (x_i, \mathbf{z}_i, \mathbf{z}_i), \\
\mathbf{g}_i (x_i, \mathbf{z}_i, \mathbf{z}_i), \\
\mathbf{f}_i (x_i, \mathbf{z}_i, \mathbf{z}_i) + \mathbf{g}_i (x_i, \mathbf{z}_i, \mathbf{z}_i)
\end{align*}
\]

Set \( b_i = 1 \) if conditions (8) and (9) of Proposition 1 are satisfied, otherwise set \( b_i = 0 \)

**end for**

\( b = \bigwedge_{i=1}^n b_i \)

**return** True if \( b = 1 \).

**Remark 3.** Algorithm 1 is linear with respect to the state space dimension. Indeed, the computational complexity is given by \( \mathcal{O}((\alpha + \beta)n) \), where \( n \) represents the state-space dimension, \( \alpha \) represents the complexity to compute the local framer, and \( \beta \) represents the complexity to check (8)-(9) in Algorithm 1.

The following result characterizes the class of possible invariance controllers enforcing conditions (8) and (9) of Proposition 1.

**Corollary 1.** Suppose that the system \( \Sigma \) defined in (1) satisfies Assumption 2. Let \( \int_{\mathbf{x}_i} \mathbf{f}_i, \mathbf{g}_i, i \in \{1, 2, \ldots, n\} \), be a local framer of \( f \) (respectively, \( g \)). Set \( K \subseteq \mathcal{X} \) be an interval defined as \( K = [\mathbf{z}; \mathbf{z}] \), where \( \mathbf{z} < \mathbf{x} \). If the set \( K \) is a controlled invariant for the system \( \Sigma \), then any continuous function \( \kappa : \mathcal{X} \rightarrow U \) satisfying Assumption 1 with \( \kappa(x) = (\kappa_1(x), \kappa_2(x), \ldots, \kappa_n(x)) \) and \( \kappa_i : \mathcal{X} \rightarrow U_i, i \in \{1, 2, \ldots, n\} \), satisfying

\[
\begin{align*}
\{ u_i \in U_i | \text{ condition (8) holds } \} & \quad \text{if } x_i = \mathbf{z}_i \\
\{ u_i \in U_i | \text{ condition (9) holds } \} & \quad \text{if } x_i = \mathbf{z}_i
\end{align*}
\]

is an invariance controller for the system \( \Sigma \) and the set \( K \).

**Remark 4.** The continuity of the controller \( \kappa \) in Corollary 1 can be dropped, at the cost of potentially more restrictive conditions. Indeed, in view of ([Aubin, 2009, Theorem 52.1.1]), conditions (8) and (9) do not need to be satisfied only at the boundary of the set of interest \( K \), but also on the external part of a sufficiently small neighborhood of the set \( K \).

**Remark 5.** (On the computation of local framers). The construction of local framers can be done for general nonlinear functions using tools from interval arithmetic. If a function \( f : \mathbb{R}^n \rightarrow \mathbb{R}^n \) can be expressed as a finite composition of the basic operators \( \{+,-,x,\} \) and elementary functions \( \{\sin, \cos, \exp, \log, \ldots\} \), a local framer for \( f \) can be constructed using the natural inclusion approach in ([Jaulin et al., 2001, Section 2.1]) by replacing each variable and each operator or function by their interval counterparts.

1 The same concept has been used in interval arithmetic under the name of interval enclosure function (Jaulin et al. [2001])
3. APPLICATIONS

In this section, we show how the proposed approach can be used to ensure safety when using deep neural networks for the purpose of approximation and control for the considered class of systems.

3.1 Safe deep neural networks approximation of continuous time dynamical systems

It is well known that deep neural networks (DNNs) can be used as universal approximators of static functions (Hornik [1991]). Indeed, any continuous function can be approximated over a compact set up to a desired level of accuracy by the selection of suitable activation functions and an adequate number of hidden layer neurons. Motivated by this property, DNNs have been extensively used in the control community as models for complex systems, where the unknown dynamics are generally learned using a DNN to be used for control purposes, ([Lewis et al., 2020, Section 6-2], Bansal et al. [2016]). Since DNNs approximation results are valid as long as the state of the system belongs to the compact set on which the data have been collected, one need to ensure, a priori, that the trajectories of the closed-loop controlled system will always remain in that compact set. In this section, we first start by presenting some preliminaries on DNNs, then we formulate the safe DNNs approximation problem and finally propose a solution using the tools presented in the previous section.

Preliminaries on deep neural networks: In this paper we assume that the DNN is a feedforward neural network. A feedforward neural network has $N^0 = n$ input neurons, $L$ hidden layers, with $N^l$, $l \in \{1, 2, \ldots, L\}$, neurons per hidden layer, and one output layer with $N^{L+1} = m$ output neurons. The neural network $N : \mathbb{R}^n \to \mathbb{R}^m$ can be formally defined for $x \in \mathbb{R}^n$ as follows:

\[ N(x) = a^{L+1} \circ f^{L+1} \circ \ldots \circ a^1 \circ f^1(x) \]

where, for each $l \in \{1, 2, \ldots, L + 1\}$, $a^l$ is the activation function and each $f^l$ is an affine transformation of the output of the previous layer given by

\[ f^l(x) = W^l x + b^l \]

for each $l \in \{1, 2, \ldots, L + 1\}$, the matrix $W^l$ and the offset vector $b^l$ have the following size:

\[ W^l \in \begin{cases} \mathbb{R}^{N_l \times N_l} & \text{if } l = 1 \\ \mathbb{R}^{N_l \times N^{-1}} & \text{if } l \in \{2, 3, \ldots, L\} \\ \mathbb{R}^{m \times N_l} & \text{if } l = L + 1 \end{cases} \]

\[ b^l \in \begin{cases} \mathbb{R}^{N_l} & \text{if } l \in \{1, 2, \ldots, L\} \\ \mathbb{R}^m & \text{if } l = L + 1 \end{cases} \]

The activation function of the output layer, $a^{L+1}$, is the identity map from $\mathbb{R}^m$ to $\mathbb{R}^m$.

Problem statement: Given the system $\Sigma$ defined in (1) with the functions $f$ and $g$ modeled as DNNs $N_f$ and $N_g$, where the DNNs $N_f$ and $N_g$ have been learned using the data collected on a compact interval $K \subseteq \mathbb{R}^n$, provide a collection of controllers $K : K \to \mathcal{U}$, ensuring that the set $K$ is an invariant for the closed-loop system $\Sigma_N$ defined by

\[ \dot{x} = N_f(x) + N_g(x)k(x) \]

Solution strategy: To provide a solution to our problem, we will follow the approach proposed in Section 2. The main ingredient used to compute controlled invariants are the local framers.

In the following, we will briefly describe how the approach proposed in the toolbox ReluVal (Wang et al. [2018]) can be adapted to efficiently compute accurate local framers for DNNs. Let us mention that ReluVal works only with ReLU activation functions. The approach of ReluVal is made of three main ingredients:

- Symbolic interval propagation: it consists of a layer-by-layer application of the natural inclusion approach to the linear map (11). Indeed, if we consider the layer $l$ of the DNN, the local framer of the linear map $f^l$ in (11) is given for $i \in \{1, 2, \ldots, n\}$, $x_i^{l-1} \in X_l$ and $z_i^{l-1} \in Z_l$ as follows:

\[ f^l_i (x_i^{l-1}, z_i^{l-1}) = W^l_{i,1} x_i^{l-1} + b^l_{i,1} \]

\[ f^l_i (x_i^{l-1}, z_i^{l-1}) = W^l_{i,2} x_i^{l-1} + b^l_{i,2} \]

where for the matrix $A \in \mathbb{R}^{n \times m}$ the matrices $A_+$ and $A_-$ are defined for $i, j \in \{1, 2, \ldots, n\}$ as $(A_+)^{ij} = \max\{A_{ij}, 0\}$, and $\min\{A_{ij}, 0\}$.

- Concretization: the ReLU function is then applied to the symbolic equations in (15) and we distinguish three cases: if $f^l_i (x_i^{l-1}, z_i^{l-1}) > 0$, we keep the symbolic dependency on the input variables. If $f^l_i (x_i^{l-1}, z_i^{l-1}) \leq 0$ we concretize to 0. Finally, if $f^l_i (x_i^{l-1}, z_i^{l-1}) < 0$ and $f^l_i (x_i^{l-1}, z_i^{l-1}) \geq 0$ then while passing the local framers through the ReLU, we can no longer keep the symbolic representation. Therefore, we concretize the values of the local framers.

- Refinement: to improve the precision of the local framer, an iterative refinement is performed, through the use of bisections.

Once the local framers are computed, we rely on the result of Corollary 1 to provide a collection of controllers, ensuring that the set of interest $K$ is an invariant for the closed-loop system.

3.2 Safety verification of deep neural network controllers for continuous-time dynamical systems

DNNs have been extensively used as controllers in the control community when dealing with complex and challenging dynamical systems (Koopman and Wagner [2017], Sallab et al. [2017]). Motivated by the lack of formal guarantees when using DNNs (as discussed in the previous subsection), in this section, we present an approach to verify invariance properties of dynamical systems under control of continuous-time DNN controllers with ReLU activation functions.

\[ K \in \mathbb{R} \]

\[ \text{ReLU}(x) = \max\{0, x\} \]
**Problem statement:** Given the system $\Sigma$ defined in (1), an interval $K \subseteq \mathbb{R}^n$, a DNN continuous-time controller $N : X \rightarrow U$, which has already been trained, verify whether the set $K$ is an invariant of the closed-loop system given by

$$\dot{x} = f(x) + g(x)N(x) \quad (16)$$

**Solution strategy:** To provide a solution to this control problem, we follow the approach proposed in Section 2 and use local framers both for the system dynamics and the controller. The following corollary of Proposition 1 summarizes the main idea.

**Corollary 2.** Suppose that the system $\Sigma$ defined in (1) satisfies Assumption 2. Let $f_i, \overline{f}_i$ (respectively, $g_i, \overline{g}_i$), $i \in \{1, 2, \ldots, n\}$, be a local framer of $f$ (respectively, $g$). Let $K \subseteq X$ be an interval defined as $K = [x; \overline{x}]$, where $x < \overline{x}$. Let $N$ be the already trained DNN controller and let $\overline{N}_i, \overline{N}_i$, $i \in \{1, 2, \ldots, n\}$, be a local framer for $N$ (which can be constructed, as discussed in the previous subsection). The set $K$ is a controlled invariant for the closed-loop controlled system in (16) if the following conditions are satisfied:

$$f_i + \min\{g_i, \overline{g}_i, \overline{g}_i, \overline{g}_i\} \geq 0$$

$$f_i - \max\{g_i, \overline{g}_i, \overline{g}_i, \overline{g}_i\} \leq 0$$

4. **EXAMPLES**

4.1 Kuramoto oscillator

The Kuramoto oscillator has many applications ranging from automated vehicle coordination to power networks (Dorfler and Bullo [2012]). The fully interconnected Kuramoto model consists of $n$ phase oscillators with state $\theta_i$, $i \in \{1, 2, \ldots, n\}$, evolving according to

$$\dot{\theta} = f(\theta) + g(\theta)u$$

with $f(\theta) = w + \frac{K}{n} \phi(\theta)$ and $g(\theta) = 1$, where $\theta = (\theta_1, \theta_2, \ldots, \theta_n) \in X = [0; 2\pi]^n$ and $w = (w_1, w_2, \ldots, w_n)$ is the natural frequency of the oscillators. In the simulations, $w_i, i \in \{1, 2, \ldots, n\}$ is randomly chosen in the set $[0, \pi]$. The map $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is given by

$$\phi(\theta) = \left( \sum_{j=1}^{n} \sin(\theta_j - \theta_1) - \sum_{j=1}^{n} \sin(\theta_j - \theta_2) - \ldots - \sum_{j=1}^{n} \sin(\theta_j - \theta_n) \right).$$

The constant $K = 2$ is the coupling strength, and control input $u = (u_1, u_2, \ldots, u_n)$ with $u_i \in [-3.5; 2]$, $i \in \{1, 2, \ldots, n\}$. The proposed model satisfies Assumption 2 with $f_i$ and $g_i$, $i \in \{1, 2, \ldots, n\}$, given by $f_i(\theta) = w_i + \frac{K}{n} \sum_{j=1}^{n} \sin(\theta_j - \theta_i)$ and $g_i(\theta) = 1$.

The objective is to construct a controller allowing to keep the trajectories of the closed loop system in the safe set $K = [\overline{x}_i, \overline{z}_i]^n$. Safety of Kuramoto oscillators has been addressed recently in (Jagtap et al. [2020]), where the authors use compositional barrier certificates, which can be constructed only under a small-gain like condition. In this paper we do not rely on such type of conditions.

4.2 Safe deep neural network approximation

We consider the nonlinear system $\Sigma$ described by:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = f(x) + g(x)u := \begin{bmatrix} x_2 \sin(k_1 x_1) \\ x_1 \cos(k_2 x_2^2) \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

This system represents an agent moving in the 2-D plane with position $x = (x_1, x_2)$. The authors in (Hou et al. [2009]) assume that the dynamics of $\Sigma$ are unknown and use a single hidden layer neural network to learn the model. In this paper, we approximate the system using a DNN

---

4 The dependence of the functions $f_i, \overline{f}_i, \overline{g}_i$, and $\overline{N}_i, \overline{N}_i$ on $x_i, \overline{x}_i$, and $\overline{z}_i$ is omitted to improve readability.
model, i.e., the function $f$ is approximated by a DNN $N_f$. The data is collected on the state space of interest $X = [-2; 2] \times [-2; 2]$, where the input space is given by $U = [-2; 2] \times [-2; 2]$. In view of Problem 3.1.2, the objective is to ensure the safety of the DNN approximation result. More precisely, the objective is to provide a collection of state-feedback controllers $\kappa : X \to U$ ensuring that the set of interest $X$, on which the data has been collected is an invariant for the closed loop system

$$\dot{x} = N_f(x) + \kappa(x). \quad (18)$$

For the simulations, the function $f$ is approximated by a 2 hidden layers NN ($L = 2$). The number of neurons each hidden layer is 4 ($N^1 = N^2 = 8$). The ReLU activation function is used in the NN hidden layers with a randomly initialized input weights. The training has been done using the Matlab Deep Learning toolbox. We used a feedforward DNN with Levenberg-Marquardt training algorithm and Mean Squared Error as a loss function. Once the approximative DNN $N_f$ is obtained, we follow the ReluVal toolbox (Wang et al. [2018]) to compute the local framers. Then, we rely on Algorithm 1 and Corollary 1 to provide a collection of controllers, ensuring that the set of interest $X$ is an invariant for the closed-loop system in (18). The collection of possible controllers is given by $\kappa : X \to U$ with $\kappa(x) = (\kappa_1(x), \kappa_2(x))$, where $\kappa_i : X \to U$, $i \in \{1, 2\}$ satisfy

$$\kappa_1(x) \in \begin{cases} [1.9; 2] & \text{if } x_1 = -2 \\ [-2; 2] & \text{if } x_1 \in (-2; 2) \\ [-2; 1.9] & \text{if } x_1 = 2 \end{cases}$$

$$\kappa_2(x) \in \begin{cases} [0.6; 2] & \text{if } x_2 = -2 \\ [-2; 2] & \text{if } x_2 \in (-2; 2) \\ [-2; -0.6] & \text{if } x_2 = 2 \end{cases}$$

5. CONCLUSION

In this paper, we have presented an approach to the computation of interval controlled invariants for nonlinear input-affine systems using the concept of local framers. Moreover, we have shown how the proposed approach makes it possible to ensure safety properties when using DNN representations for the purpose of modelling and control. Illustrative examples are presented showing the merits of the proposed approach and its scalability properties. In future works, we will address the question of the computation of tight local framers, which will make it possible to construct a complete algorithm. Moreover, we will generalize the approach to deal with other set structures beyond intervals used in this paper.

REFERENCES

Abate, A., Tiwari, A., and Sastry, S. (2009). Box invariance in biologically-inspired dynamical systems. *Automatica*, 45(7), 1601–1610.

Aubin, J.-P. (2009). *Viability theory*. Springer Science & Business Media.

Bansal, S., Akametalu, A.K., Jiang, F.J., Laine, F., and Tomlin, C.J. (2016). Learning quadrotor dynamics using neural network for flight control. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, 4653–4660.

Blanchini, F. and Miani, S. (2008). *Set-theoretic methods in control*. Springer.

Cannon, M., Deshmukh, V., and Kouvaritakis, B. (2003). Nonlinear model predictive control with polytopic invariant sets. *Automatica*, 39(8), 1487–1494.

Chai, J. and Sanfelice, R.G. (2020). Forward invariance of sets for hybrid dynamical systems (part ii). *IEEE Transactions on Automatic Control*.

Dorfler, F. and Bullo, F. (2012). Synchronization and transient stability in power networks and nonuniform kuramoto oscillators. *SIAM Journal on Control and Optimization*, 50(3), 1616–1642.

Horvitz, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2), 251–257.

Hou, Z.G., Cheng, L., and Tan, M. (2009). Decentralized robust adaptive control for the multiagent system consensus problem using neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(3), 636–647.

Jagtap, P., Swikir, A., and Zamani, M. (2020). Compositional construction of control barrier functions for interconnected control systems. In *Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control*, 1–11.

Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. (2001). Interval analysis. In *Applied interval analysis*, 11–43. Springer.

Koopman, P. and Wagner, M. (2017). Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1), 90–96.

Korda, M., Henrion, D., and Jones, C.N. (2014). Convex computation of the maximum controlled invariant set for polynomial control systems. *SIAM Journal on Control and Optimization*, 52(5), 2944–2969.

Lewis, F., Jagannathan, S., and Yesildirak, A. (2020). *Neural network control of robot manipulators and nonlinear systems*. CRC press.

Meyer, P.J., Girard, A., and Witrant, E. (2016). Robust controlled invariance for monotone systems: application to ventilation regulation in buildings. *Automatica*, 70, 14–20.

Moisan, M., Bernard, O., and Gouzé, J.L. (2007). Near optimal interval observers bundle for uncertain bioreactors. In *2007 European Control Conference (ECC)*, 5115–5122. IEEE.

Sallab, A.E., Abdou, M., Perot, E., and Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19), 70–76.

Saoud, A. (2019). *Compositional and Efficient Controller Synthesis for Cyber-Physical Systems*. Ph.D. thesis, Université Paris Saclay.

Tabuada, P. (2009). *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media.

Wang, S., Pei, K., Whitehouse, J., Yang, J., and Jana, S. (2018). Formal security analysis of neural networks using symbolic intervals. In *27th {USENIX} Security Symposium* ({USENIX} Security 18), 1599–1614.