



HAL
open science

Reachability Analysis of Linear Parameter-Varying Systems with Neural Network Controllers

Arash Sadeghzadeh, Pierre-Loïc Garoche

► **To cite this version:**

Arash Sadeghzadeh, Pierre-Loïc Garoche. Reachability Analysis of Linear Parameter-Varying Systems with Neural Network Controllers. 2022 IEEE Conference on Control Technology and Applications (CCTA), Aug 2022, Trieste, France. pp.1372-1377, 10.1109/CCTA49430.2022.9966104 . hal-04412111

HAL Id: hal-04412111

<https://hal.science/hal-04412111>

Submitted on 23 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Reachability Analysis of Linear Parameter-Varying Systems with Neural Network Controllers

Arash Sadeghzadeh and Pierre-Loic Garoche

Abstract—A method is presented to obtain outer-approximations of forward reachable sets for Linear Parameter-Varying (LPV) systems controlled by neural networks for safety verification. The method relies on abstracting the activation functions by optimization-based sector constraints. The forward reachable sets are obtained in terms of solutions to a set of Linear Matrix Inequalities (LMIs). A numerical example is provided to demonstrate the applicability of the proposed method. A comparison with some available methods is also included.

I. INTRODUCTION

In recent years, there has been a growing interest in employing neural network (NN) controllers to tackle the control problem of complex nonlinear systems. However, the nonlinear and large-scale nature of neural networks makes them hard to analyze. Therefore, developing tools for safety verification of NN controllers is of paramount importance. The goal of this paper is to develop a forward reachability analysis method for the safety verification of Linear Parameter-Varying (LPV) systems controlled by neural networks. LPV framework is particularly promising for modeling of nonlinear (NL) and time-varying (TV) systems, which enables the extension of the available methods for linear time-invariant (LTI) systems to NL/TV systems leveraging the linear proxy representation.

Safety verification or reachability analysis aims to obtain a set of regions in the state space that a dynamical system can evolve to from a set of initial conditions. Having obtained the reachable sets, one can make sure in advance that the closed-loop system remains operating in a safe region in the state space for a finite-time horizon. Safety verification of closed-loop systems with neural network controllers has been investigated in some recent studies [1], [2], [3], [4], [5], [6]. An approach for the exact and over-approximate reachability analysis of deep neural networks with ReLU activation functions is provided in [4]. However, it seems that the extension to other types of activation functions is not straightforward. In [1], a forward reachability analysis method is presented for the closed-loop systems with NN controllers. The approximate reachable sets are obtained using semidefinite programming. The method is suitable for linear time-varying systems. A flow pipe construction scheme to over approximate the reachable sets for continuous-time systems

This work was partially supported by the projects ANR FEANICSES (ANR-17-CE25-0018) and ESA AITIVE.

Pierre-Loic Garoche and Arash Sadeghzadeh are with École Nationale de l'Aviation Civile, Université de Toulouse, France.
{pierre-loic.garoche, arash.sadeghzadeh}
@enac.fr

is presented in [2]. The proposed method is based on generating a polynomial mapping using regression from input-output samples. This approach, however, is only applicable to ReLU activation functions. A reachability analysis approach based on Bernstein polynomials is developed in [3] for NN controlled systems with Lipschitz continuous activation functions.

Motivated by the above methods, in this paper, we propose a semidefinite program (SDP) for reachability analysis of LPV systems in feedback interconnection with NN controllers. Leveraging the LPV methodology to model the nonlinear systems, our proposed method is applicable for NL/TV systems. The method relies on abstracting the activation functions in the NN controller by optimization-based sector constraints. The proposed method can be considered as an extension of the method in [1] for LPV systems. Moreover, considering the approximation-based sector constraints can lead to less conservative reachable sets in comparison with the local sector constraints [7]. The proposed method is able to handle different kinds of activation function.

The notation is fairly standard. In this paper, $0_{n \times m}$ is an $n \times m$ zero matrix and I_n is an $n \times n$ identity matrix. The subscript for the dimension may be dropped if the sizes of matrices are clear from the context. In a symmetric matrix, \star denotes the transpose of an off-diagonal block. For the ease of notation, $\text{diag}(\cdot)$ is employed to represent the block diagonal concatenation of input vector arguments.

II. PROBLEM DEFINITION

Consider the following LPV discrete-time system with n_θ scheduling variables $\theta(k) := [\theta_1(k) \cdots \theta_{n_\theta}(k)]^\top \in \Theta \subset \mathbb{R}^{n_\theta}$ as follows:

$$G: \quad x(k+1) = A(\theta(k))x(k) + B(\theta(k))u(k) \quad (1)$$

where $x: \mathbb{Z}^+ \rightarrow \mathbb{X} \subset \mathbb{R}^{n_x}$ is the state vector and $u: \mathbb{Z}^+ \rightarrow \mathbb{U} \subset \mathbb{R}^{n_u}$ is the input. \mathbb{X} and \mathbb{U} are assumed to be compact polyhedra with known vertices containing the origin. The scheduling variable vector $\theta(k)$ is defined by

$$\theta(k) := \mu(x(k)), \quad (2)$$

where $\mu: \mathbb{X} \rightarrow \Theta \subseteq \mathbb{R}^{n_\theta}$ is a bounded and smooth static real-valued nonlinear function of $x(k)$. Furthermore, it is supposed that $A(\theta)$ and $B(\theta)$, with compatible dimensions, are affine functions with respect to $\theta(k)$, i.e. we have

$$M(\theta(k)) := \sum_{i=1}^{n_\theta} \theta_i(k) M_i,$$

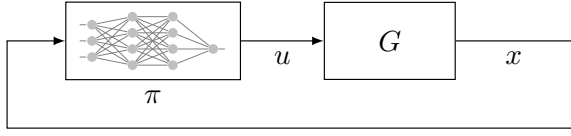


Fig. 1. Feedback system with plant G and NN controller Π .

where $M(\theta)$ represents either $A(\theta)$ or $B(\theta)$. M_i $i = 1, \dots, n_\theta$ are constant real matrices with compatible dimensions. The scheduling variable vector $\theta(k)$ is assumed to lie in a hyperrectangle Θ defined as follows:

$$\underline{\theta}_i \leq \theta_i(k) \leq \bar{\theta}_i, \quad i = 1, \dots, n_\theta \quad (3)$$

with the a priori known values of $\underline{\theta}_i, \bar{\theta}_i$ for $i = 1, \dots, n_\theta$. Suppose that G is controlled by a NN controller $\Pi(x(k)) : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$, which is a L -layer feed-forward fully connected NN given as follows:

$$\begin{aligned} a^0(k) &= x(k), \\ v^{[l]}(k) &= W^{[l]}a^{[l-1]}(k) + b^{[l]}, \quad l = 1, \dots, L \\ a^{[l]}(k) &= \Phi^{[l]}(v^{[l]}(k)), \quad l = 1, \dots, L \\ u(k) &= W^{[L+1]}a^{[L]}(k) + b^{[L+1]}, \end{aligned} \quad (4)$$

where $a^{[l]} \in \mathbb{R}^{n_l}$ $l = 1, \dots, L$ is the output of l th layer having n_l neurons. $W^{[l]} \in \mathbb{R}^{n_l \times n_{l-1}}$ and $b^{[l]}$ are weight matrices and bias vectors of the l th layer, respectively. The closed-loop system is shown in Fig. 1. It is supposed that the activation function $\Phi^{[l]}$ is applied element-wise. Thus, considering

$$v^{[l]} := \begin{bmatrix} v_1^{[l]} & v_2^{[l]} & \dots & v_{n_l}^{[l]} \end{bmatrix}^\top \in \mathbb{R}^{n_l},$$

we have

$$\Phi^{[l]}(v^{[l]}) := \begin{bmatrix} \phi^{[l]}(v_1^{[l]}) & \phi^{[l]}(v_2^{[l]}) & \dots & \phi^{[l]}(v_{n_l}^{[l]}) \end{bmatrix}^\top, \quad (5)$$

where $\phi^{[l]} : \mathbb{R} \rightarrow \mathbb{R}$ represents the activation function of l th layer. In the sequel, without loss of generality, we assume that $\phi^{[l]}(v) = \tanh(v)$. However, different kinds of activation functions can also be considered in a similar manner.

Let us define the closed-loop system as:

$$x(k+1) = \mathcal{H}(x(k)), \quad x(0) = x_0 \quad (6)$$

where the initial state x_0 lies in an initially known ellipsoid $\mathcal{E}_0 \subseteq \mathbb{X}$. Thus, we have

$$\mathcal{E}_0 : \begin{bmatrix} x_0 \\ 1 \end{bmatrix}^\top \Upsilon_0 \begin{bmatrix} x_0 \\ 1 \end{bmatrix} \leq 0, \quad \Upsilon_0 \in \mathbb{S}^{n_x+1}. \quad (7)$$

To proceed, let us first define the forward reachable set $\mathcal{R}_{k+1}(\mathcal{E}_k)$ at instant $k+1$ for a given set of current states \mathcal{E}_k as follows:

$$\mathcal{R}_{k+1} := \{x(k+1) \mid x(k+1) = \mathcal{H}(x(k)), x(k) \in \mathcal{E}_k\}. \quad (8)$$

It is worth mentioning that the reachable set \mathcal{R}_{k+1} can be of any shape convex or non-convex.

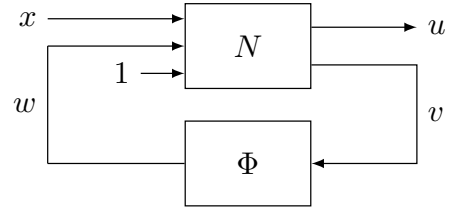


Fig. 2. LFT representation for the NN controller $\Pi(x(k))$.

The problem we tackle in this paper is to iteratively compute a sequence of minimum-volume ellipsoids \mathcal{E}_k , $k = 1, \dots, N$ for an initially given set \mathcal{E}_0 so that \mathcal{E}_k contains the reachable set \mathcal{R}_k , i.e. $\mathcal{R}_k \subseteq \mathcal{E}_k$, for the finite-time horizon $N \geq 0$. The problem in hand can be formulated as follows:

$$\begin{aligned} \min \quad & \text{volume } \mathcal{E}_k \\ \text{s.t.} \quad & \mathcal{R}_k(\mathcal{E}_{k-1}) \subseteq \mathcal{E}_k \end{aligned} \quad (9)$$

for $k = 1, \dots, N$. In the subsequent sections, we derive an LMI relaxation of the optimization problem (9). This way, the sequence of outer-approximation ellipsoids of the reachable sets can be computed iteratively starting from given \mathcal{E}_0 by solving a sequence of semidefinite programming (SDP) problems.

III. OPTIMIZATION-BASED SECTOR CONSTRAINT ON THE ACTIVATION FUNCTIONS

The nonlinear activation functions in the NN controller $\Pi(x(k))$ are the major stumbling block to formulate the optimization problem (9) as an LMI problem. To cope with the nonlinearity of the activation functions, one can resort to sector constraints. Contrary to the available methods in the literature, the sector constraints are obtained by solving a set of convex optimization problems in this paper.

First of all, leveraging the *linear fractional transformation* (LFT) representation of the NN controller shown in Fig. 2, one can represent the NN controller $\Pi(x(k))$ as follows:

$$\begin{bmatrix} u(k) \\ v(k) \end{bmatrix} := N \begin{bmatrix} x(k) \\ w(k) \\ 1 \end{bmatrix}, \quad (10)$$

$$w(k) = \Phi(v(k)), \quad (11)$$

where

$$\begin{aligned} N &:= \begin{bmatrix} N_{ux} & N_{uw} & N_{ub} \\ N_{vx} & N_{vw} & N_{vb} \end{bmatrix} \\ &= \left[\begin{array}{c|ccc|c|c} 0 & 0 & 0 & \dots & W^{[L+1]} & b^{[L+1]} \\ \hline W^{[1]} & 0 & \dots & 0 & 0 & b^{[1]} \\ 0 & W^{[2]} & \dots & 0 & 0 & b^{[2]} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & W^{[L]} & 0 & b^{[L]} \end{array} \right] \end{aligned} \quad (12)$$

with

$$v(k) := \begin{bmatrix} v^{[1]}(k) \\ \vdots \\ v^{[L]}(k) \end{bmatrix}, \quad w(k) := \begin{bmatrix} w^{[1]}(k) \\ \vdots \\ w^{[L]}(k) \end{bmatrix},$$

$$\Phi(v) := \begin{bmatrix} \Phi^{[1]}(v^{[1]}) \\ \vdots \\ \Phi^{[L]}(v^{[L]}) \end{bmatrix} \in \mathbb{R}^{n_\Phi}, \quad n_\Phi := n_1 + \dots + n_L.$$

The proposed sector constraint for the activation functions is defined below.

Definition 1: Suppose that $\underline{m}, \underline{h}, \underline{v}, \overline{m}, \overline{h}, \overline{v}$ are given. The nonlinear function $\phi : \mathbb{R} \rightarrow \mathbb{R}$ fulfills the sector constraint if both

$$\underline{L}(v) := \underline{m}v + \underline{h} \leq \phi(v), \quad (13)$$

$$\overline{L}(v) := \overline{m}v + \overline{h} \geq \phi(v), \quad (14)$$

hold for all $\underline{v} \leq v \leq \overline{v}$.

To obtain the tightest sector constraint, one can consider the following optimization problem:

$$\min_{\overline{m}, \overline{h}, \underline{m}, \underline{h}} \left\| (\overline{m} - \underline{m})v + (\overline{h} - \underline{h}) \right\|_2 \quad (15)$$

s.t. (13), (14) $\forall v \in [\underline{v}, \overline{v}]$

which is obviously a non-convex one since $\phi(v) = \tanh(v)$ is a non-convex function. To overcome this difficulty, one can resort to a fine grid for v in the desired interval $\underline{v} \leq v \leq \overline{v}$, which leads to a convex optimization problem. After finding $\underline{m}, \underline{h}, \overline{m}, \overline{h}$ from the optimization problem (15), we can rewrite the sector constraints (13) and (14) as a quadratic constraint as follows [7]:

$$(\overline{L}(v) - \phi(v))(\phi(v) - \underline{L}(v)) \geq 0. \quad (16)$$

As an example, sector constraint for the tanh function is illustrated in Fig. 3 considering $\underline{v} = -1.5$ and $\overline{v} = 0.9$. The obtained upper and lower lines are $\overline{L}(v) = 0.687v + 0.125$ and $\underline{L}(v) = 0.783v - 0.071$, respectively.

To be able to obtain the sector constraints for all neurons in the NN controller, we need first to compute the upper and lower bounds \underline{v} and \overline{v} for each neuron. Let us first define

$$\overline{v}^{[l]} := \begin{bmatrix} \overline{v}_1^{[l]} & \dots & \overline{v}_{n_l}^{[l]} \end{bmatrix}^\top \in \mathbb{R}^{n_l},$$

$$\underline{v}^{[l]} := \begin{bmatrix} \underline{v}_1^{[l]} & \dots & \underline{v}_{n_l}^{[l]} \end{bmatrix}^\top \in \mathbb{R}^{n_l},$$

where $\overline{v}_i^{[l]}$ and $\underline{v}_i^{[l]}$ for $i = 1, \dots, n_l$ represent the upper and lower values related to the i th neuron of the l th layer.

For the first layer in the NN controller, $\overline{v}^{[1]}$ and $\underline{v}^{[1]}$ can be computed using the following optimization problems:

$$\underline{v}_i^{[1]} := \min_{x(k)} W_i^{[1]}x(k) + b_i^{[1]} \quad (17)$$

s.t. $x(k) \in \mathcal{E}_k$,

$$\overline{v}_i^{[1]} := \max_{x(k)} W_i^{[1]}x(k) + b_i^{[1]} \quad (18)$$

s.t. $x(k) \in \mathcal{E}_k$

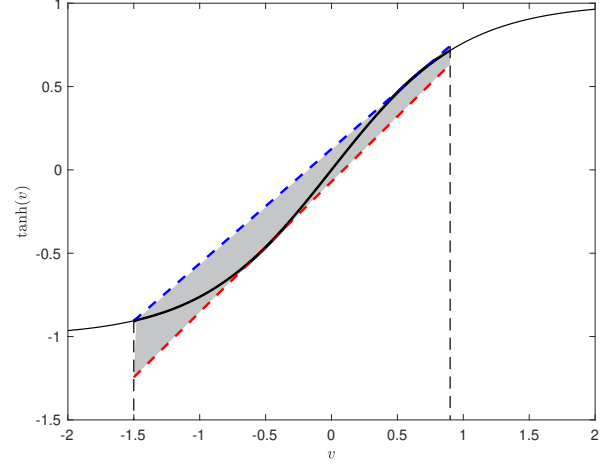


Fig. 3. Sector constraint on the tanh activation function.

for $i = 1, \dots, n_1$. Subsequently, since tanh is a nondecreasing function, we can obtain the upper and lower bounds on $a^{[1]}(k)$ as follows:

$$\overline{a}^{[1]} = \tanh(\overline{v}^{[1]}), \quad \underline{a}^{[1]} = \tanh(\underline{v}^{[1]}).$$

Now that we obtain $\overline{a}^{[1]}$ and $\underline{a}^{[1]}$, the upper and lower bounds $\overline{v}^{[l]}$ and $\underline{v}^{[l]}$ can be computed using propagation through all the layers of the NN controller. Using classical affine arithmetic operators, we can define the center

$$v_c^{[l]} := \frac{1}{2}W^{[l]}(\overline{a}^{[l-1]} + \underline{a}^{[l-1]}),$$

and the radius

$$v_r^{[l]} := \frac{1}{2}\text{abs}\left(W^{[l]}(\overline{a}^{[l-1]} - \underline{a}^{[l-1]})\right).$$

We have then

$$\underline{v}^{[l]} = v_c^{[l]} - v_r^{[l]}, \quad \overline{v}^{[l]} = v_c^{[l]} + v_r^{[l]},$$

for $l = 2, \dots, L$. Note that

$$\underline{a}^{[l]} = \tanh(\underline{v}^{[l]}), \quad \overline{a}^{[l]} = \tanh(\overline{v}^{[l]}).$$

After finding $\underline{v}^{[l]}$ and $\overline{a}^{[l]}$ for $l = 1, \dots, L$, we are able to obtain the values $\underline{m}_i^{[l]}, \underline{h}_i^{[l]}, \overline{m}_i^{[l]}, \overline{h}_i^{[l]}$ corresponding to the neuron i in the layer l , using the optimization problem (15). Afterwards, the quadratic constraints

$$\left[\overline{L}_i^{[l]}(v_i^{[l]}) - \phi^{[l]}(v_i^{[l]}) \right] \left[\phi^{[l]}(v_i^{[l]}) - \underline{L}_i^{[l]}(v_i^{[l]}) \right] \geq 0, \quad (19)$$

where

$$\overline{L}_i^{[l]}(v_i^{[l]}) = \overline{m}_i^{[l]}v_i^{[l]} + \overline{h}_i^{[l]}, \quad \underline{L}_i^{[l]}(v_i^{[l]}) = \underline{m}_i^{[l]}v_i^{[l]} + \underline{h}_i^{[l]},$$

can be reformulated as a unique quadratic constraint using the following lemma:

Lemma 1: Suppose that $\underline{m}_i^{[l]}, \underline{h}_i^{[l]}, \underline{v}_i^{[l]}, \overline{m}_i^{[l]}, \overline{h}_i^{[l]}, \overline{v}_i^{[l]}$ are given for all $l = 1, \dots, L$ and $i = 1, \dots, n_l$. Assume that

the sector constraints (19) hold for all $l = 1, \dots, L$ and $i = 1, \dots, n_l$. Then there exists $\lambda \in \mathbb{R}_+^{n_\Phi}$ such that

$$0 \leq \begin{bmatrix} v \\ w \\ 1 \end{bmatrix}^\top \Upsilon_a \begin{bmatrix} v \\ w \\ 1 \end{bmatrix},$$

holds, where

$$\Upsilon_a := \begin{bmatrix} -\overline{\mathcal{M}}\Lambda\mathcal{M} & \frac{1}{2}\Lambda(\overline{\mathcal{M}} + \mathcal{M}) & -\frac{1}{2}\Lambda(\overline{\mathcal{M}}\underline{h} + \mathcal{M}\overline{h}) \\ \star & -\Lambda & \frac{1}{2}\Lambda(\overline{h} + \underline{h}) \\ \star & \star & -\overline{h}^\top \Lambda \underline{h} \end{bmatrix},$$

$$\overline{m} := \begin{bmatrix} \overline{m}_1^{[1]} & \dots & \overline{m}_{n_1}^{[1]} & \dots & \overline{m}_1^{[L]} & \dots & \overline{m}_{n_L}^{[L]} \end{bmatrix}^\top,$$

$$\underline{m} := \begin{bmatrix} \underline{m}_1^{[1]} & \dots & \underline{m}_{n_1}^{[1]} & \dots & \underline{m}_1^{[L]} & \dots & \underline{m}_{n_L}^{[L]} \end{bmatrix}^\top,$$

$$\overline{h} := \begin{bmatrix} \overline{h}_1^{[1]} & \dots & \overline{h}_{n_1}^{[1]} & \dots & \overline{h}_1^{[L]} & \dots & \overline{h}_{n_L}^{[L]} \end{bmatrix}^\top,$$

$$\underline{h} := \begin{bmatrix} \underline{h}_1^{[1]} & \dots & \underline{h}_{n_1}^{[1]} & \dots & \underline{h}_1^{[L]} & \dots & \underline{h}_{n_L}^{[L]} \end{bmatrix}^\top,$$

$$\overline{\mathcal{M}} = \text{diag}(\overline{m}), \quad \mathcal{M} = \text{diag}(m), \quad \Lambda = \text{diag}(\lambda).$$

Proof: For any $v \in \mathbb{R}^{n_\Phi}$ and $w = \Phi(v)$, we have

$$\begin{bmatrix} v \\ w \\ 1 \end{bmatrix}^\top \Upsilon_a \begin{bmatrix} v \\ w \\ 1 \end{bmatrix} = \sum_{l=1}^L \sum_{i=1}^{n_l} \lambda_j \left[\overline{L}_i^{[l]}(v_i^{[l]}) - \phi^{[l]}(v_i^{[l]}) \right] \left[\phi^{[l]}(v_i^{[l]}) - \underline{L}_i^{[l]}(v_i^{[l]}) \right],$$

where $j = i + \sum_{m=1}^{l-1} n_m$. If (19) holds for all $l = 1, \dots, L$ and $i = 1, \dots, n_l$, then each term in the aforementioned sum is non-negative, thus, (20) holds. ■

IV. OUTER-APPROXIMATION OF ONE-STEP AHEAD REACHABLE SET

In this section, using the quadratic constraint (20) on the activation functions, we formulate the problem of computing one-step ahead reachable set. To this aim, we describe first the outer-approximation of the reachable set at instant $k+1$ as an ellipsoid. Then, the proposed optimization problem to find the outer-approximation of the reachable set is provided.

A. Representing reachable set as a quadratic constraint

One can describe the outer-approximation of the reachable set at instant $k+1$ as follows:

$$\|M_r x(k+1) + F_r\|_2 \leq 1. \quad (20)$$

This represents an ellipsoid in which $M_r \in \mathbb{R}^{n_x \times n_x}$ and $F_r \in \mathbb{R}^{n_x}$ determine the orientation, center, and the size of the ellipsoid. Note that (20) can alternatively be represented as

$$\mathcal{E}_r: \quad 0 \geq \quad (21)$$

$$\begin{bmatrix} x(k+1) \\ 1 \end{bmatrix}^\top \underbrace{\begin{bmatrix} M_r^\top M_r & M_r^\top F_r \\ F_r^\top M_r & F_r^\top F_r - 1 \end{bmatrix}}_{\Upsilon_r} \begin{bmatrix} x(k+1) \\ 1 \end{bmatrix}. \quad (22)$$

Suppose that $x(k)$ is known, then, the state vector $x(k+1)$ can be obtained by (1) when $u(k)$ is given by (10)-(12). We get

$$x(k+1) = \begin{bmatrix} A(\theta(k)) + B(\theta(k))N_{ux} & B(\theta(k))N_{uw} \\ B(\theta(k))N_{ub} \end{bmatrix} \check{x}(k), \quad (23)$$

where $\check{x}(k) := [x(k)^\top \quad w(k)^\top \quad 1]^\top$. This way, (20) can be described as

$$\check{x}(k)^\top \Gamma_r^\top(\theta) \Upsilon_r \Gamma_r(\theta) \check{x}(k) \leq 0, \quad (24)$$

which represents a quadratic constraint, where

$$\Gamma_r(\theta) := \begin{bmatrix} A(\theta(k)) + B(\theta(k))N_{ux} & B(\theta(k))N_{uw} & B(\theta(k))N_{ub} \\ 0 & 0 & 1 \end{bmatrix}.$$

B. Outer-approximation of reachable set

It is supposed that the state vector of the closed-loop system at instant k lies in \mathcal{E}_k given by

$$\mathcal{E}_k: \quad \begin{bmatrix} x(k) \\ 1 \end{bmatrix}^\top \Upsilon_k \begin{bmatrix} x(k) \\ 1 \end{bmatrix} \leq 0, \quad (25)$$

which can be rewritten as follows:

$$\check{x}(k)^\top \Gamma_k^\top \Upsilon_k \Gamma_k \check{x}(k) \leq 0, \quad (26)$$

where

$$\Gamma_k = \begin{bmatrix} I & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Similarly, the quadratic constraint (20) on the activation functions can also be reformulated as:

$$\check{x}(k)^\top \Gamma_a^\top \Upsilon_a \Gamma_a \check{x}(k) \geq 0, \quad (27)$$

since we have

$$\begin{bmatrix} v(k) \\ w(k) \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} N_{vx} & N_{vw} & N_{vb} \\ 0 & I & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\Gamma_a} \check{x}(k).$$

Note that at instant k , the scheduling variable vector $\theta(k)$ lies in a much smaller hyperrectangle than Θ considering the current possible values for $x(k)$. To be able to obtain a tighter reachable set, we need to obtain a momentary hyperrectangle based on the current possible values of states. Since it is assumed that $x(k) \in \mathcal{E}_k$ at instant k , so one is able to compute the related momentary hyperrectangle considering $\theta(k) = \mu(x(k))$ by defining $\overline{\theta}(k)$ and $\underline{\theta}(k)$ as follows:

$$\overline{\theta}_i(k) := \max \theta_i(k) \quad (28)$$

$$s.t. \quad \theta(k) = \mu(x(k)), \quad x(k) \in \mathcal{E}_k$$

and

$$\underline{\theta}_i(k) := \min \theta_i(k) \quad (29)$$

$$s.t. \quad \theta(k) = \mu(x(k)), \quad x(k) \in \mathcal{E}_k$$

for all $i = 1, \dots, n_\theta$. Therefore, $\theta(k)$ lies in a hyperrectangle Θ_k (with 2^{n_θ} vertices) described as follows:

$$\underline{\theta}_i(k) \leq \theta_i(k) \leq \overline{\theta}_i(k), \quad i = 1, \dots, n_\theta.$$

Note that $\Theta_k \subseteq \Theta$. If $\mu(x(k))$ is a convex function on \mathcal{E}_k , then one can easily solve the convex optimization problems (28) and (29); otherwise, one may resort to a very fine gridding of \mathbb{X} to obtain $\underline{\theta}_i(k)$ and $\bar{\theta}_i(k)$.

Next theorem states our proposed method regarding the computation of the outer-approximation of one-step ahead reachable set.

Theorem 1: Consider the closed-loop system (6). Assume that $\underline{m}_i^{[l]}$, $\underline{h}_i^{[l]}$, $\underline{v}_i^{[l]}$, $\bar{m}_i^{[l]}$, $\bar{h}_i^{[l]}$, $\bar{v}_i^{[l]}$ are given for the NN controller for all $l = 1, \dots, L$ and $i = 1, \dots, n_l$. Moreover, suppose that the quadratic constraint (20) holds for the activation functions. Furthermore, assume that \mathcal{E}_k is known and $x(k) \in \mathcal{E}_k$ at instant k . If there exists a scalar $\tau > 0$ and $\lambda \in \mathbb{R}_+^{n_\Phi}$ such that

$$\Gamma_r^\top(\theta)\Upsilon_r\Gamma_r(\theta) + \Gamma_a^\top\Upsilon_a\Gamma_a - \tau\Gamma_k^\top\Upsilon_k\Gamma_k \leq 0 \quad (30)$$

holds for all $\theta \in \Theta_k$, then $\mathcal{E}_r \supseteq \mathcal{R}_{k+1}(\mathcal{E}_k)$.

Proof: Post- and pre-multiplying (30) by $\check{x}(k)$ and its transpose, respectively, and considering (26) and (27), one obtains

$$\check{x}(k)^\top \Gamma_r^\top(\theta)\Upsilon_r\Gamma_r(\theta)\check{x}(k) \leq 0,$$

which implies (21) considering the fact that (24) is equivalent to (21). ■

In order to find the outer-approximation of the reachable set, one should solve (30); nonetheless, condition (30) is not an LMI with respect to the decision variables M_r , F_r , λ and τ due to the multiplication terms of M_r and F_r in Υ_r . On top of that, condition (30) is an infinite-dimensional inequality since θ may get any value in Θ_k . In the next section, a finite-dimensional LMI relaxation for (30) is provided, using the polytopic structure of Θ_k .

V. LMI RELAXATION

Let us define

$$e := \begin{bmatrix} 0_{1 \times n_x} & 1 \end{bmatrix}, \quad L_r := \begin{bmatrix} M_r & F_r \end{bmatrix},$$

which immediately implies

$$\Upsilon_r = L_r^\top L_r - e^\top e. \quad (31)$$

Now, using Schur complement formula [8] and taking into account (31), inequality (30) can be equivalently rewritten as

$$\left[\begin{array}{c|c} \Gamma_a^\top \Upsilon_a \Gamma_a - \tau \Gamma_k^\top \Upsilon_k \Gamma_k - \Gamma_r(\theta)^\top e^\top e \Gamma_r(\theta) & \star \\ \hline L_r \Gamma_r(\theta) & -I \end{array} \right] \leq 0, \quad (32)$$

which is an LMI with respect to the decision variables M_r , F_r (L_r), λ , τ . However, it is still an infinite-dimensional LMI condition. In the following lemma, leveraging the same method as in [9], a finite-dimensional relaxation for (32) is provided.

Lemma 2: If the following conditions

$$\begin{cases} \Xi^{(i,i)} \leq 0, & i = 1, \dots, 2^{n_\theta}, \\ \Xi^{(i,j)} + \Xi^{(j,i)} \leq 0, & i = 1, \dots, 2^{n_\theta} - 1, j = i, \dots, 2^{n_\theta} \end{cases} \quad (33)$$

hold, then (32) holds for all $\theta \in \Theta_k$, where

$$\Xi^{(i,j)} := \left[\begin{array}{c|c} \Gamma_a^\top \Upsilon_a \Gamma_a - \tau \Gamma_k^\top \Upsilon_k \Gamma_k - \Gamma_r^\top(\theta^{(i)}) e^\top e \Gamma_r(\theta^{(j)}) & \star \\ \hline L_r \Gamma_r(\theta^{(i)}) & -I \end{array} \right].$$

in which $\theta^{(i)}$ and $\theta^{(j)}$ represents the i th and j th vertices of the hyperrectangle Θ_k .

Proof: For any value $\theta(k)$ in Θ_k , we have

$$\theta(k) = \sum_{i=1}^{2^{n_\theta}} \alpha_i \theta^{(i)}, \quad \alpha_i \geq 0, \quad \sum_{i=1}^{2^{n_\theta}} \alpha_i = 1.$$

Considering the fact that $A(\theta)$ and $B(\theta)$ appear linearly in $\Gamma_r(\theta)$, and they are also linear function of θ , one can describe $\Gamma_r(\theta)^\top e^\top e \Gamma_r(\theta)$ and $L_r \Gamma_r(\theta)$ as follows:

$$\Gamma_r(\theta)^\top e^\top e \Gamma_r(\theta) = \sum_{i=1}^{2^{n_\theta}} \sum_{j=1}^{2^{n_\theta}} \alpha_i \alpha_j \Gamma_r(\theta^{(i)})^\top e^\top e \Gamma_r(\theta^{(j)}) \quad (34)$$

$$L_r \Gamma_r(\theta) = \sum_{i=1}^{2^{n_\theta}} \sum_{j=1}^{2^{n_\theta}} \alpha_i \alpha_j L_r \Gamma_r(\theta^{(i)}), \quad (35)$$

since $\sum_{i=1}^{2^{n_\theta}} \alpha_i = \sum_{j=1}^{2^{n_\theta}} \alpha_j = 1$. ■

Now, considering (34) and (35), we get

$$\left[\begin{array}{c|c} \Gamma_a^\top \Upsilon_a \Gamma_a - \tau \Gamma_k^\top \Upsilon_k \Gamma_k - \Gamma_r(\theta)^\top e^\top e \Gamma_r(\theta) & \star \\ \hline L_r \Gamma_r(\theta) & -I \end{array} \right] = \sum_{i=1}^{2^{n_\theta}} \alpha_i \alpha_i \Xi^{(i,i)} + \sum_{i=1}^{2^{n_\theta}-1} \sum_{j=i}^{2^{n_\theta}} \alpha_i \alpha_j \left(\Xi^{(i,j)} + \Xi^{(j,i)} \right),$$

which immediately implies that when (33) holds then (32) holds for all $\theta \in \Theta_k$ since α_i and α_j are not negative.

Applying Theorem 1 and Lemma 2, we obtain a minimum-volume ellipsoid that encloses $\mathcal{R}_{k+1}(\mathcal{E}_k)$, solving the following optimization problem:

$$\begin{aligned} \min_{M_r, F_r, \tau, \lambda} \quad & -\log \det(M_r) \\ \text{s.t.} \quad & (33) \end{aligned} \quad (36)$$

which is a SDP problem. Since \mathcal{E}_0 is known, the outer-approximation of one-step ahead reachable sets can be iteratively computed for a finite-time horizon $k = 1, \dots, N$.

VI. NUMERICAL ILLUSTRATION

In this section, the applicability of the proposed method is evaluated using a numerical example. To solve the LMI problem (36), YALMIP [10] interface with the LMI solver MOSEK [11] are utilized. Furthermore, CVX [12] a package for specifying and solving convex programs, is used to solve the convex optimization problems.

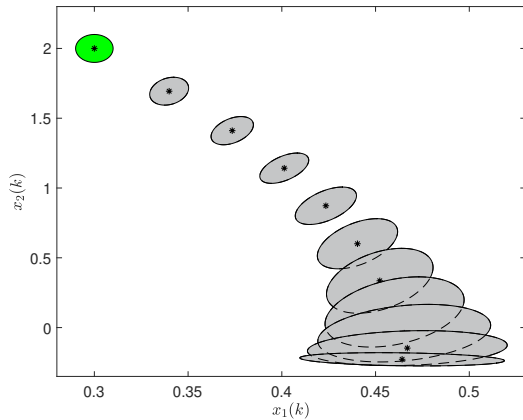


Fig. 4. Gray ellipsoids: Outer-approximations of the reachable sets for finite-time horizon $N = 10$. Green ellipsoid: the initial set \mathcal{E}_0 .

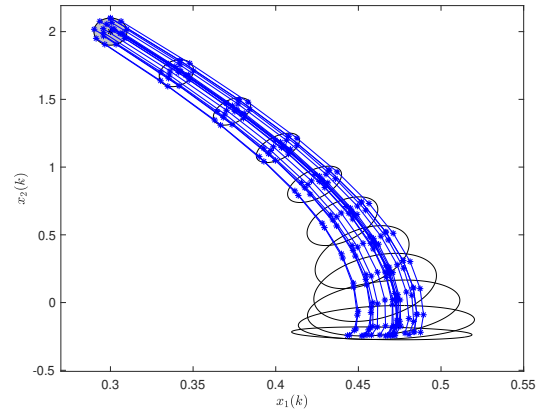


Fig. 5. System operation trajectories starting from some random initial states in \mathcal{E}_0 .

A. Inverted Pendulum

Consider the problem of balancing an inverted pendulum on a cart. The inverted pendulum can be modeled by the following discrete-time LPV model:

$$x(k+1) = \begin{bmatrix} 1 & T \\ \frac{g}{l}T\theta & 1 - \frac{\kappa}{ml^2}T \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \frac{T}{ml^2} \end{bmatrix} u(k), \quad (37)$$

$x = [\vartheta \ \dot{\vartheta}]^\top$ is the state vector where ϑ denotes the angular position (rad) and u is the control input (Nm). Let us consider gravity constant $g = 9.8$ m/s², mass $m = 0.15$ Kg, length $l = 0.5$ m, friction coefficient $\kappa = 0.05$ Nms/rad, and the sampling time $T = 0.02$.

We use the same NN controller as in [7] to stabilize the inverted pendulum. It is a 2-layer NN with $n_1 = n_2 = 32$ and tanh as the activation function for both layers. The control input is supposed to be restricted to $-0.7 \leq u(t) \leq 0.7$ due to practical limitations. To cope with the input saturation, we consider a third layer for the NN controller whose activation function is the saturation function.

Using the optimization problem (36), we iteratively compute a sequence of ellipsoidal outer-approximations of the reachable sets for a finite-time horizon $N = 10$. To do so, the initial state vector is supposed to lie in the following given ellipsoid:

$$\mathcal{E}_0 : \left(\frac{x_1(0) - 0.3}{0.01} \right)^2 + \left(\frac{x_2(0) - 2}{0.1} \right)^2 \leq 1$$

The outer-approximations of the reachable sets are illustrated in Fig. 4. To have an insight on the conservativeness of the obtained reachable sets, time-domain simulation of the closed-loop system considering 10 randomly selected initial states from \mathcal{E}_0 is performed. The results are given in Fig. 5. One can clearly see that all the trajectories remain in the computed reachable sets.

For the sake of comparison, we have applied the presented method in [1] on the inverted pendulum. To do so, we have slightly modified the method of [1] in order to cope with

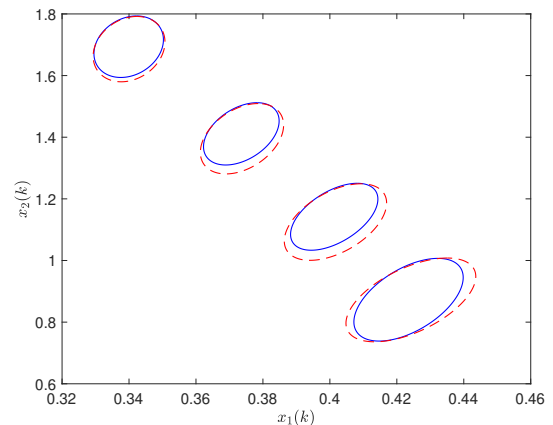


Fig. 6. Blue ellipsoids: the reachable set obtained by the proposed method. Red ellipsoids: the reachable set obtained by the method in [1].

the tanh activation functions instead of the ReLU ones. To have a fair comparison, we use the local sector constraints to bound the activation functions as QCs [7] for the method of [1]. In Fig. 6, the ellipsoids obtained by our method and that of [1] are depicted. Obviously, our proposed method results in tighter outer-approximations of the reachable sets, which implies that the proposed method is capable of providing less conservative results.

VII. CONCLUSION

In this paper, we have developed a new method to address the reachability analysis problem of LPV systems in feedback interconnection with NN controllers. Leveraging the LPV methodology to model the nonlinear systems, the approach can successfully be employed for nonlinear/time-varying systems. The method relies on abstracting the activation functions in the NN controller by optimization-based sector constraints. We have shown that considering the approximation-based sector constraints can lead to less conservative reachable sets in comparison with some of the

available methods in the literature. The proposed method is capable of handling different kinds of activation function.

REFERENCES

- [1] H. Hu, M. Fazlyab, M. Morari, and G. J. Pappas, "Reach-SDP: Reachability analysis of closed-loop systems with neural network controllers via semidefinite programming," 2020.
- [2] S. Dutta, X. Chen, and S. Sankaranarayanan, "Reachability analysis for neural feedback systems using regressive polynomial rule inference," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 157–168.
- [3] C. Huang, J. Fan, W. Li, X. Chen, and Q. Zhu, "Reachnn: Reachability analysis of neural-network controlled systems," *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, Oct. 2019.
- [4] H.-D. Tran, D. Manzananas Lopez, P. Musau, X. Yang, L. V. Nguyen, W. Xiang, and T. T. Johnson, "Star-based reachability analysis of deep neural networks," in *Formal Methods – The Next 30 Years*, M. H. ter Beek, A. McIver, and J. N. Oliveira, Eds. Cham: Springer International Publishing, 2019, pp. 670–686.
- [5] R. Ivanov, J. Weimer, R. Alur, G. J. Pappas, and I. Lee, "Verisig: Verifying safety properties of hybrid systems with neural network controllers," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 169–178. [Online]. Available: <https://doi.org/10.1145/3302504.3311806>
- [6] M. Fazlyab, M. Morari, and G. J. Pappas, "Safety verification and robustness analysis of neural networks via quadratic constraints and semidefinite programming," *ArXiv*, vol. abs/1903.01287, 2019.
- [7] H. Yin, P. Seiler, and M. Arcak, "Stability analysis using quadratic constraints for systems with neural network controllers," *IEEE Transactions on Automatic Control*, 2021.
- [8] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: SIAM, 1994.
- [9] R. C. L. F. Oliveira and P. L. D. Peres, "Parameter-Dependent LMIs in Robust Analysis: Characterization of Homogeneous Polynomially Parameter-Dependent Solutions Via LMI Relaxations," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1334–1340, 2007.
- [10] J. Löfberg, "YALMIP: A Toolbox for Modeling and Optimization in MATLAB," in *Proc. of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004, pp. 284–289.
- [11] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28)*, 2015. [Online]. Available: <http://docs.mosek.com/7.1/toolbox/index.html>
- [12] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <http://cvxr.com/cvx>, Mar. 2014.