Examples for separable control Lyapunov functions and their neural network approximation ⋆

Lars Grüne * **Mario Sperl** *

* *Chair of Applied Mathematics, Mathematical Institute, University of Bayreuth, 95440 Bayreuth, Germany*

Abstract: In this paper, we consider nonlinear control systems and discuss the existence of a separable control Lyapunov function. To this end, we assume that the system can be decomposed into subsystems and formulate conditions such that a weighted sum of Lyapunov functions of the subsystems yields a control Lyapunov function of the overall system. Since deep neural networks are capable of approximating separable functions without suffering from the curse of dimensionality, we can thus identify systems where an efficient approximation of a control Lyapunov function via a deep neural network is possible. A corresponding network architecture and training algorithm are proposed. Further, numerical examples illustrate the behavior of the algorithm.

Keywords: deep neural network, curse of dimensionality, separable function, control Lyapunov function, nonlinear control system, small-gain theory

1. INTRODUCTION

Control Lyapunov functions provide a tool for studying stability and controllability issues of nonlinear control systems. They do not only serve as a certificate for asymptotic null-controllability, cf. Sontag (1983), but can also be used to construct stabilizing feedback laws. Since, in general, there is no explicit analytic expression for a control Lyapunov function, we rely on numerical methods. Common numerical approaches suffer from the so-called curse of dimensionality, which describes an exponential growth of the computational effort in the state dimension. More precisely, the number of degrees of freedom for storing a control Lyapunov function grows exponentially with the dimension of the state space. Thus, these approaches are limited to low-dimensional systems. In this paper, we discuss the use of neural networks in order to overcome the curse of dimensionality for computing control Lyapunov functions.

The well-known universal approximation theorem for neural networks states that a neural network with one hidden layer is capable of approximating any continuous function arbitrarily well, see, e.g., Cybenko (1989) and Hornik (1991). For continuously differentiable functions, the function as well as its derivative can be approximated, see, e.g., (Pinkus, 1999, Chapter 4). However, in general, the numerical effort, which is characterized by the number of neurons needed, still grows exponentially in the state dimension (Mhaskar, 1996, Theorem 2.1). Hence, the curse of dimensionality still applies. This situation changes for functions with certain beneficial structures. One example of such functions are so-called Barron functions, cf. Barron (1993). Roughly speaking, these are functions with a high degree of smoothness and suitable form of the Fourier transformation. Such functions were recently exploited for approximating solutions of partial differential equations and optimal value functions, see, e.g., Han et al. (2018), Darbon et al. (2020), and Gonon and Schwab (2021).

Another class of functions with a beneficial structure are compositional functions. These are functions that consist of several component functions that depend only on a number of arguments that is independent of the dimension of the original domain. This compositional structure enables neural networks to approximate such functions with an effort that grows only polynomially, see Poggio et al. (2017), Beneventano et al. (2021), Kang et al. (2021), and Kang and Gong (2022). A particular case of compositional functions are separable functions. In Grüne (2021) it was shown that nonlinear ordinary differential equations satisfying a small-gain condition admit separable Lyapunov functions which can be approximated by neural networks with a polynomial effort. In this paper, we discuss an extension of the result in Grüne (2021) for nonlinear control systems. To this end, we represent the overall system dynamics as an interconnected graph and formulate conditions on the nodes of this graph for existence and non-existence of separable control Lyapunov functions. In particular, the theory about so-called active nodes that is presented in Chen and Astolfi (2020) can be used to establish an existence result.

Moreover, we propose a neural network architecture and a training algorithm that are used to approximate separable control Lyapunov functions. As opposed to other algorithms, this algorithm does not rely on a predefined decomposition of the system into subsystems, but learns an appropriate decomposition during the training process.

⋆ This research has been supported by the German Research Foundation (DFG) under project GR 1569/23-1 within the priority program 2298 "Theoretical Foundations of Deep Learning".

This is a crucial advantage provided by the usage of deep neural networks. Furthermore, we note that while in our numerical examples we use ideas from reinforcement learning, the presented complexity analysis is independent of the concrete learning method. There exist several algorithmically orientated papers that use neural networks for the computation of control Lyapunov functions, see, e.g., Long and Bayoumi (1993); Khansari-Zadeh and Billard (2014); Richards et al. (2018). While our study is inspired by some of the algorithmic ideas therein, we note that these papers do not provide an analytical investigation concerning the curse of dimensionality.

The remainder of this paper is organized as follows. In the next section we introduce the problem. Afterwards, we recall the concept of deep neural networks and the approximation result for separable Lyapunov functions that was presented in Grüne (2021). In Section 4 we discuss the existence of a separable control Lyapunov function of an interconnected control system. A neural network architecture and a corresponding training algorithm together with numerical examples are presented in the last section.

2. PROBLEM FORMULATION

We consider a nonlinear control system of the form

$$\dot{x}(t) = f(x(t), u(t)), \tag{1}$$

where $u \in L_\infty(\mathbb{R}, U)$ with $U \subset \mathbb{R}^m$ and $f\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is continuous and Lipschitz continuous in x. We assume that the system (1) has an equilibrium at the origin, i.e., $f(0,0) = 0$.

Definition 1. Let $0 \in D \subset \mathbb{R}^n$ be open. A $\mathcal{C}^1$-function $V : D \to \mathbb{R}$ is called (local) smooth control Lyapunov function (CLF) for (1) if there exist α_1, $\alpha_2 \in \mathcal{K}_\infty$ and $\alpha_3 \in \mathcal{K}$ [1] such that

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|), \tag{2}$$
$$\inf_{u \in U} DV(x) f(x,u) \leq -\alpha_3(\|x\|), \tag{3}$$

for all $x \in D$, where $DV(x)f(x,u)$ is the directional derivative of V at x in direction $f(x,u)$.

As requiring the existence of a smooth CLF is too restrictive in many cases, we need to take into account the following more general definition of a control Lyapunov function in the sense of Dini, see (Braun et al., 2021, Definition 4.4).

Definition 2. Let $0 \in D \subset \mathbb{R}^n$ be open and let $F\colon \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be a set-valued map with $0 \in F(0)$ that takes nonempty and convex values and is upper-semicontinuous. Then a continuous function $V\colon D \to \mathbb{R}$ is called (local) control Lyapunov function for the differential inclusion

$$\dot{x}(t) \in F(x(t))$$

if there exist α_1, $\alpha_2 \in \mathcal{K}_\infty$ and $\alpha_3 \in \mathcal{K}$ such that

$$\alpha_1(\|x\|) \leq V(x) \leq \alpha_2(\|x\|),$$
$$\inf_{w \in F(x)} D_+V(x;w) \leq -\alpha_3(\|x\|),$$

for all $x \in D$, where $D_+V(x;w)$ denotes the lower right Dini derivative of V at x in direction w.

[1] We define $\mathcal{K}$ as the space of all continuous and strictly increasing functions $\alpha\colon [0,\infty) \to [0,\infty)$ with $\alpha(0) = 0$ and $\mathcal{K}_\infty$ as the space of all functions $\alpha \in \mathcal{K}$ with $\lim_{\tau\to\infty} \alpha(\tau) = \infty$.

In order to be able to apply Definition 2 to control systems of the form (1) we define the differential inclusion associated to (1) as

$$\dot{x}(t) \in F(x(t)) := \overline{\text{conv}}\ f(x(t), U). \tag{4}$$

Note that the mapping F defined in (4) satisfies the requirements of Definition 2. We then say that V is a control Lyapunov function in the Dini sense for the control system (1) if V is a control Lyapunov function in the Dini sense for the differential inclusion (4). If V is continuously differentiable this definition matches Definition 1 of a smooth CLF. In the following, we always assume that there exists a control Lyapunov function for the system (1).

Definition 3. Let $V\colon D \subset \mathbb{R}^n \to \mathbb{R}$. We call V a separable function of degree $d_{\max} \in \mathbb{N}$, $d_{\max} < n$, if there exist $s \in \mathbb{N}$, $s < n$, and functions $V_j\colon \mathbb{R}^{d_j} \to \mathbb{R}$ with $d_j \leq d_{\max}$ for $j = 1, \ldots, s$ such that

$$V(x) = \sum\nolimits_{j=1}^{s} V_j(z_j), \quad x \in D,$$

where $z_j = (x_{j_1}, x_{j_2}, \ldots, x_{j_{d_j}})$ for some $j_i \in \{1, \ldots, n\}$.

It is our goal to investigate conditions for the existence and non-existence of a separable CLF with a degree $d_{\max}$ that is independent of the state dimension n.

3. DEEP NEURAL NETWORKS APPROXIMATING LYAPUNOV FUNCTIONS

In this section, we briefly recall the concept of deep neural networks as well as the result presented in Grüne (2021) that states the ability of neural networks to overcome the curse of dimensionality for separable Lyapunov functions.

Neural networks take an input vector and process it through a certain number of layers in order to produce an output. For our purpose of representing a control Lyapunov function, we use the input vector $x \in \mathbb{R}^n$ and a one-dimensional output $W(x;\theta) \in \mathbb{R}$. This means that the input layer possesses $N_0 = n$ neurons and the output layer consists of only one neuron. The numbers N_l of neurons in the remaining layers (called hidden layers) may vary. The vector $\theta \in \mathbb{R}^p$ represents parameters that determine the output of the neural network and have to be learned during the training process. Denote with $y_k^l \in \mathbb{R}$ the value of the neuron at position k in layer l. It is determined by the values of the neurons at the previous layer via

$$y_k^l = \sigma^l\left(\sum\nolimits_{i=1}^{N_{l-1}} w_{k,i}^l\, y_i^{l-1} + b_k^l\right),$$

where $\sigma^l\colon \mathbb{R} \to \mathbb{R}$ is the so-called activation function of the l-th layer and $w_{k,i}^l$, $b_k^l \in \mathbb{R}$ are parameters that are comprised in θ. The activation function of the output layer is usually chosen to be an affine function. Figure 1 shows a neural network with one hidden layer, where the activation of the neuron y_1^1 is highlighted. Note that in our setting we have $y_i^0 = x_i$ for $i = 1, \ldots, n$ and $y_1^2 = W(x;\theta)$.

In Grüne (2021) it was shown that deep neural networks can overcome the curse of dimensionality for Lyapunov functions. To this end, consider the particular case of system (1), where the right-hand side does not depend on u, i.e., we have a differential equation and the definition of a smooth control Lyapunov function simplifies to the classical definition of a Lyapunov function.

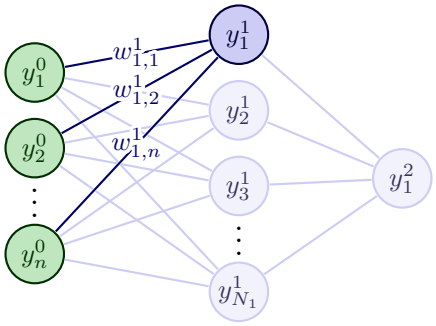

Fig. 1. Neural network with one hidden layer.

Theorem 4. (see Theorem 5.1 in Grüne (2021)). Let b, c, $\varepsilon > 0$ and $d_{\max} \in \mathbb{N}$. Consider a family of compact sets $K_n \subset [-c, c]^n$. Define $\mathcal{F}_n$ to be the set of Lipschitz functions $f\colon K_n \to \mathbb{R}^n$, such that there exists an invertible matrix $T \in \mathbb{R}^{n\times n}$ with $\|T\| \leq b$ such that the differential equation

$$\dot{x}(t) = Tf(T^{-1}x(t)) \tag{5}$$

admits a separable Lyapunov function with degree $d_{\max}$. Then we can construct a family of neural networks W_n with a number of neurons of

$$N_\varepsilon = \mathcal{O}(nd_{\max} + n^{1+d_{\max}}\varepsilon^{-d_{\max}})$$

such that for all $f \in \mathcal{F}_n$

$$\inf_{\theta\in\mathbb{R}^p} \max_{x\in K_n} \|W_n(x;\theta) - V_f(x)\|_\infty \leq \varepsilon,$$

where V_f is a Lyapunov function for (5).

Nonlinear small-gain theory for large-scale systems, cf. Rüffer (2007), Dashkovskiy et al. (2010), Dashkovskiy et al. (2011), and Liu et al. (2011), can be used to identify systems that admit a separable Lyapunov function, i.e., systems with a right-hand side function $f \in \mathcal{F}_n$. To this end, the state space together with the right-hand side function are decomposed into s subsystems. If each of the subsystems possesses an ISS-Lyapunov function (cf. Sontag and Wang (1995)) with suitable assumptions on its gains (see Theorem 4.1 in Dashkovskiy et al. (2011)), we can conclude the existence of a separable Lyapunov function for the original system, see also Section 3 in Grüne (2021). In the next section, we discuss the potential of extending this approach to nonlinear control systems.

4. EXISTENCE OF SEPARABLE CONTROL LYAPUNOV FUNCTIONS

Consider a nonlinear control system of the form (1). We assume that the state space, the control space, and the controlled vector field can be decomposed as

$$x = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_s \end{pmatrix}, \quad u = \begin{pmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_s \end{pmatrix}, \quad f(x) = \begin{pmatrix} f_1(x, \tilde{u}_1) \\ f_2(x, \tilde{u}_2) \\ \vdots \\ f_s(x, \tilde{u}_s) \end{pmatrix},$$

with $z_j \in \mathbb{R}^{d_j}$, $f_j : \mathbb{R}^n \to \mathbb{R}^{d_j}$, and $\tilde{u}_j \in U_j \subset \mathbb{R}^{\beta_j}$ such that $\sum_{j=1}^s d_j = n$, $U = U_1 \times U_2 \times \cdots \times U_s$, and $\sum_{j=1}^s \beta_j = m$. Using this notation we allow the case $\beta_j = 0$ for some j, i.e., there may exist subsystems that do not depend on the control input u. By defining

$$z_{-j} := (z_1, \ldots, z_{j-1}, z_{j+1}, \ldots, z_s)^T$$

we can now write the control system $\dot{x} = f(x, u)$ as s subsystems of the form

$$\Sigma_j : \quad \dot{z_j} = f_j(x, \tilde{u}_j) = f_j(z_j, z_{-j}, \tilde{u}_j), \quad 1 \leq j \leq s. \tag{6}$$

This decomposition can be represented with a directed graph that consists of s vertices representing the respective subsystems. Further, there exists an edge from node i to node j if the subsystem i influences the subsystem j, i.e., if the function f_j depends on the vector z_i. Let us consider the following nonlinear control system that is proposed in Section 4 in Chen and Astolfi (2020):

$$\begin{aligned} \dot{x}_1 &= x_3 + u, \\ \dot{x}_2 &= x_1 - x_2 + x_1^2, \\ \dot{x}_3 &= x_2 - x_3. \end{aligned} \tag{7}$$

The graph corresponding to example (7) is shown in Figure 2. Applying the backstepping procedure (cf. Section

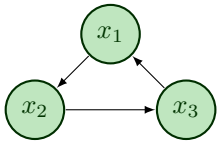

Fig. 2. The graph corresponding to the system (7).

6.1 in Sepulchre et al. (1997)) yields the existence of a control Lyapunov function for the system (7). However, the obtained control Lyapunov function does not have the desired separable form. We now want to use small-gain theory in order to construct a separable control Lyapunov function as sum of Lyapunov functions of the single subsystems. To this end, we assume that for every subsystem Σ_j there exist a function $V_j\colon \mathbb{R}^{d_j} \to \mathbb{R}$, gains $\alpha_j \in \mathcal{K}_\infty$ and $\gamma_{i,j} \in \mathcal{K}_\infty$, $1 \leq i \leq s$, and (provided $\beta_j > 0$) a feedback function $F_j\colon \mathbb{R}^{d_j} \to U_j$ such that

$$DV_j(z_j)f_j(z_j, z_{-j}, F_j(z_j)) \leq -\alpha_j(\|z_j\|) + \sum_{i\neq j} \gamma_{i,j}(\|z_i\|). \tag{8}$$

In other words, V_j is an ISS-Lyapunov function for the system

$$\dot{z_j} = f_j(z_j, z_{-j}, F_j(z_j)), \tag{9}$$

where z_{-j} is seen as the control input. In the following, we discuss whether there exists a vector $r \in \mathbb{R}^s$ such that the function

$$V(x) := \sum\nolimits_{j=1}^s r_j V_j(z_j) \tag{10}$$

is a Lyapunov function for the system

$$\begin{aligned} \dot{z_1} &= f_1(z_1, z_{-1}, F_1(z_1)), \\ &\vdots \\ \dot{z_s} &= f_s(z_s, z_{-s}, F_s(z_s)). \end{aligned} \tag{11}$$

In this case, Theorem 4 yields that a Lyapunov function for (11) can be approximated on a compact domain with a neural network without suffering from the curse of dimensionality provided that (11) is Lipschitz on the chosen domain. Note that a Lyapunov function for (11) satisfies condition (3) with $u = [F_1(z_1) \ \ldots \ F_s(z_s)]^T$ and is thus a control Lyapunov function for the control system given through the subsystems (6).

In Chen and Astolfi (2020), the authors formulate a sufficient condition on the structure of the corresponding graph together with a condition on a subset of the nodes.

To this end, the notion of an active node is defined. The node number j is called active if for a given ISS-Lyapunov function V_j and gains $\gamma_{i,j}$ it holds that for any arbitrary small gain α_j, there exists an appropriate feedback F_j such that the inequality (8) holds. This means the gain α_j of an active node j can be chosen arbitrarily small. For the case of quadratic gain functions, it is then shown that there exists a coefficient vector $r \in \mathbb{R}^s$ such that V defined in (10) is in fact a control Lyapunov function if each directed cycle of the graph describing the system contains at least one active node, see (Chen and Astolfi, 2020, Proposition 3). In the case of non-quadratic gain functions, the authors use augmented nodes in order to reduce this case to the case of quadratic gains. Together with a condition on these augmented states, the property of having at least one active node in every directed cycle of the graph again leads that V is a CLF for an appropriate choice of coefficients $r \in \mathbb{R}^s$, see (Chen and Astolfi, 2020, Chapter 3). Note that the approach in Chen and Astolfi (2020) is constructive provided that the decomposition into subsystems and the corresponding ISS-Lyapunov functions V_j are known.

Overall we conclude that provided there exists an ISS-Lyapunov function for every subsystem (9), we can use nonlinear small-gain theory to establish the existence of a separable CLF, i.e., to establish the existence of a CLF that can be approximated without suffering from the curse of dimensionality. Let us come back to example (7). For each subsystem Σ_j, $1 \leq j \leq 3$, the function $V_j(x_j) = x_j^2$ is an ISS-Lyapunov function. As there is only one directed cycle in the corresponding graph (cf. Figure 2) and since x_1 is an active node, cf. (Chen and Astolfi, 2020, Chapter 4), we can conclude that $V(x) = \sum_{j=1}^{3} r_j x_j^2$ is a CLF for an appropriate choice of the coefficients r_j. For example, those can be chosen as $r_j = 1$, $1 \leq j \leq 3$. The use of a neural network in order to compute such a CLF is shown in Section 5.

Let us now consider a variation of the system (7):

$$\begin{aligned} \dot{x}_1 &= x_3 + u, \\ \dot{x}_2 &= x_1 - x_2 + x_2^2, \\ \dot{x}_3 &= x_2 + x_3. \end{aligned} \tag{12}$$

Again, we are concerned with the existence of a separable CLF of the form

$$V(x) = \sum\nolimits_{j=1}^{3} V_j(x_j). \tag{13}$$

However, the third subsystem $\dot{x}_3 = x_2 + x_3$ is not ISS, whence we cannot apply the same construction as in example (7). The fact that the third subsystem is not asymptotically stable if all influences from other subsystems are neglected even implies that there cannot exist a CLF of the form (13). This is proven in the following lemma.

Lemma 5. Consider a control system $\dot{x} = f(x, u)$ of the form

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2), \\ \dot{x}_2 &= f_2(x_1, x_2, u), \end{aligned} \tag{14}$$

which has an asymptotically controllable equilibrium at the origin. Let V be a possibly nonsmooth, Lipschitz continuous control Lyapunov function in the Dini sense for the system. Assume that for the ODE $\dot{x}_1 = f(x_1, 0)$ the origin is not asymptotically stable. Then V is not of the form

$$V(x) = V_1(x_1) + V_2(x_2).$$

Proof. Assume that there exists a CLF $V(x_1, x_2) = V_1(x_1) + V_2(x_2)$ for (14). Since we assumed that the first system with $x_2 = 0$ is not asymptotically stable, there exists an $x_1 \neq 0$ in the domain of V_1 such that

$$D_+(V_1(x_1), f_1(x_1, 0)) \geq 0,$$

because otherwise V_1 would be a Lyapunov function in the Dini sense for $\dot{x}_1 = f(x_1, 0)$, implying asymptotic stability of the origin by, e.g., (Braun et al., 2021, Theorem 4.5), contrary to the assumption.

Since V is assumed to be a Lyapunov function we can conclude that V_2 has a minimum at the origin. Thus, the Dini derivative of V_2 in direction $f_2(x_1, 0, u)$ satisfies

$$D_+(V_2(0), f_2(x_1, 0, u)) \geq 0$$

for all $u \in U$. This implies that $D_+(V_2(0), w_2) \geq 0$ for all $w_2 \in \overline{\text{conv}}\, f_2(x_1, 0, U)$. Since all $w \in \overline{\text{conv}}\, f(x_1, 0, U)$ are of the form $w^T = (w_1^T, w_2^T)$ with $w_1 = f_1(x_1, 0)$ and w_2 as above, this yields

$$\begin{aligned} D_+(V(x_1, 0), w) \geq & D_+(V_1(x_1, 0), w_1) \\ & + D_+(V_2(x_1, 0), w_2) \geq 0 \end{aligned}$$

for all $w \in \overline{\text{conv}}\, f(x_1, 0, U)$. This is a contradiction to V being a CLF for (14). □

In this section we have discussed a sufficient as well as a necessary condition for the existence of a separable CLF. These conditions allow us to identify systems where a curse-of-dimensionality-free computation of a CLF is possible. However, the conditions do not need to be known in order to be able to apply the neural network algorithm that is presented in the following section.

5. TRAINING ALGORITHM AND NUMERICAL EXAMPLES

We start with proposing a suitable network architecture. Our neural network consists of 2 hidden layers, see Figure 3. The input layer matches the state space. The first hidden layer is used to compute a linear transformation that decomposes the system into subsystems, cf. (6). Thus, we construct this layer as s sublayers with a chosen number of $d_{\max}$ nodes each and use the identity as activation function. The node z_j^i represents the i-th entry of the vector z_j. The second hidden layer is then designed to compute the ISS Lyapunov functions for each of the subsystems. To this end, we have s sublayers $Y_1, \ldots, Y_s$, whose inputs are the nodes $z_1, \ldots, z_s$, respectively. We set the activation function to be the softplus function $\sigma(\cdot) = \ln(e^{(\cdot)}+1)$. Each sublayer has M nodes $y_j^1, \ldots, y_j^M$. A linear combination of the outputs of these sublayers then yields the output value of the neural network $W(x; \theta)$. Note that a crucial point of the network design is that the decomposition into subsystems is not given a priori, but computed by the first hidden layer. Moreover, we want to stress that it is possible to merge the two hidden layers into one hidden layer.

In order to be able to train the neural network, we need to define a suitable loss function. To this end, we make use of the characterization of a CLF via the inequalities in (2) and the partial differential inequality (3). This is inspired by the literature on solving PDEs with neural networks, see Section 1 for some references. We first state a Lemma that simplifies the inequality (3) for systems with an affine linear control input.

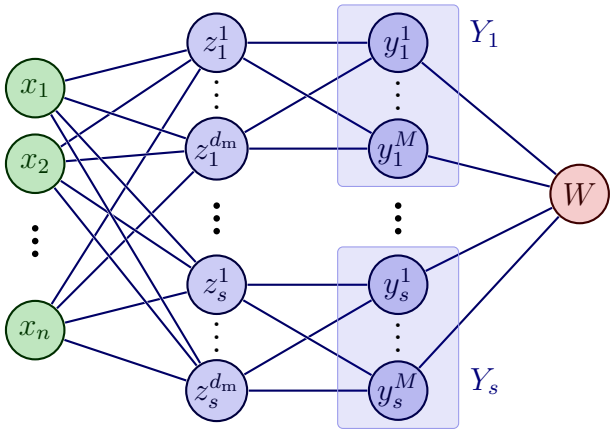

Fig. 3. Architecture of the neural network with the abbreviations $d_m := d_{max}$ and $W := W(x;\theta)$.

Lemma 6. Assume we have given a system of the form

$$\dot{x} = f(x(t), u(t)) = h(x(t)) + g(x(t))u,$$

where $h\colon \mathbb{R}^n \to \mathbb{R}^n$ and $g\colon \mathbb{R}^n \to \mathbb{R}^{n\times m}$. Let $V \in \mathcal{C}^1(D,\mathbb{R})$ for some $D \subset \mathbb{R}^n$ and let $\alpha_3 \in \mathcal{K}$.

(i) Let $U = [-C,C]^m$ for some $C > 0$. Then the inequality (3) holds if and only if

$$\alpha_3(\|x\|) + DV(x)h(x) - C\|DV(x)g(x)\|_1 \le 0. \quad (15)$$

(ii) Let $U = \mathbb{R}^m$. Then the inequality (3) holds if and only if the following implication holds:

$$DV(x)g(x) = 0 \implies \alpha_3(\|x\|) + DV(x)h(x) \le 0.$$

Proof. Case (i) holds since we have

$$\begin{aligned}
&\inf_{u\in U} DV(x)f(x,u)\\
=&DV(x)h(x) + \inf_{u\in U}\sum\nolimits_{i=1}^{m}(DV(x)g(x))_i\, u_i\\
=&DV(x)h(x) - C\sum\nolimits_{i=1}^{m}|(DV(x)g(x))_i|\\
=&DV(x)h(x) - C\|DV(x)g(x)\|_1.
\end{aligned}$$

The second case follows from the first case as C can be chosen arbitrarily large. □

Since the implication in (ii) is difficult to be implemented numerically, we use a mapping $\nu\colon \mathbb{R}^n \to \mathbb{R}_{\ge 0}$ to circumvent it. Provided the values of ν are large enough such that for all $x \in D$ with $DV(x)g(x) \neq 0$ it holds that

$$\nu(x) \ge (\alpha_3(\|x\|) + DV(x)h(x))(\|DV(x)g(x)\|_1)^{-1},$$

then the implication in (ii) holds if and only if

$$\alpha_3(\|x\|) + DV(x)h(x) - \nu(x)\|DV(x)g(x)\|_1 \le 0. \quad (16)$$

Hence, we use the inequality (16) in our implementation. By setting $\nu(x) \equiv C$, we obtain the inequality (15) from case (i). Overall, we arrive at the following cost function L that depends on the state x as well as on the neural network output $W(x;\theta)$ and its derivative:

$$\begin{aligned}
&L(x, W(x;\theta), DW(x;\theta)) := ([W(x;\theta) - \alpha_1(\|x\|)]_-)^2\\
&+ ([W(x;\theta) - \alpha_2(\|x\|)]_+)^2 + \mu\Big(\Big[\alpha_3(\|x\|) + DW(x;\theta)h(x)\\
&-\nu(x)\|DW(x;\theta)g(x)\|_1\Big]_+\Big)^2,
\end{aligned}$$

where $[\cdot]_+ := \max(\cdot, 0)$, $[\cdot]_- := \min(\cdot, 0)$ and $\mu > 0$ is a weighting factor. Our computations were performed with Python 3.8.10 and TensorFlow 2.9.1 (see Abadi et al. (2015)) on Ubuntu 20.04 with the following CPU: Intel(R) Xeon(R) E-2278G (3.40GHz). At first, we consider example (7) and want to approximate a CLF on the cube $[-1,1]^3$. The network displayed in Figure 3 is used with $s = 3$ sublayers, $d_{max} = 1$ and $M = 256$. The training is performed using the Adam optimizer from TensorFlow and 400 000 training points in $[-1,1]^3$. Further, we use the comparison functions $\alpha_1(r) = 0.01r^2$, $\alpha_2(r) = 100r^2$, and $\alpha_3(r) = 0.1r^2$. The training was stopped after 30 epochs with an error of

$$\max_{i=1,\dots,N} L(x_i, W(x_i;\theta), DW(x_i;\theta)f(x_i)) \approx 4.56 * 10^{-4},$$

where $x_1, \dots, x_N \in [-1,1]^3$ are 400 000 validation points. Note that we thus know that $W(x;\theta)$ is an approximation of a CLF, but it is not guaranteed that it is actually a CLF on the whole domain, i.e., that the conditions from Definition 1 are satisfied at all points $x \in [-1,1]^3$. The computation time was 328 seconds. Figure 4 shows the computed CLF on the (x_1,x_2)-plane. The mesh plot depicts the directional derivative resulting from the control value that we use in (16), i.e.,

$$DW(x;\theta)h(x) - \nu(x)\|DW(x;\theta)g(x)\|_1.$$

We can extend example (7) to a higher dimensional state space by extending the cycle of the graph in Figure 2. A corresponding 6-dimensional example is then given by

$$\begin{aligned}
\dot{x}_1 &= x_6 + u,\\
\dot{x}_2 &= x_1 - x_2 + x_1^2, \qquad (17)\\
\dot{x}_j &= x_{j-1} - x_j, \qquad 3 \le j \le 6.
\end{aligned}$$

Performing the training algorithm with the same settings as for the 3-dimensional example, but tighter bounds $\alpha_1(r) = 0.1r^2$ and $\alpha_2(r) = 10r^2$ yields a maximal error in the validation data less than 10^{-3}. The computation time was 820 seconds. Figure 5 shows the computed approximate CLF on the (x_2,x_3)-plane.

Furthermore, for example (12) the training process does not yield a satisfying result if we set the transformation of the first hidden layer to be the identity, cf. Lemma 6. However, the situation changes if we allow the neural network to compute a suitable state transformation. This effect as well as different graph structures and control systems with a non-affine control input are part of current research.

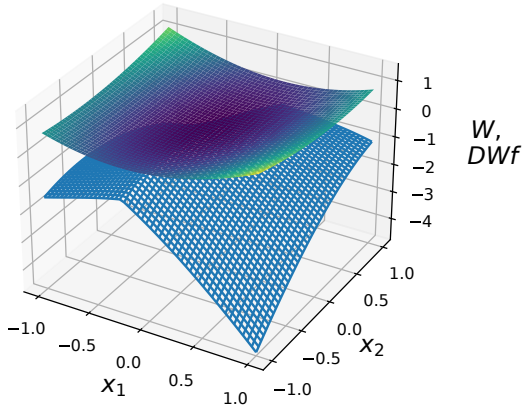

Fig. 4. Approximate CLF (solid) and its corresponding orbital derivative (mesh) for the system (7) on the (x_1,x_2)-plane.

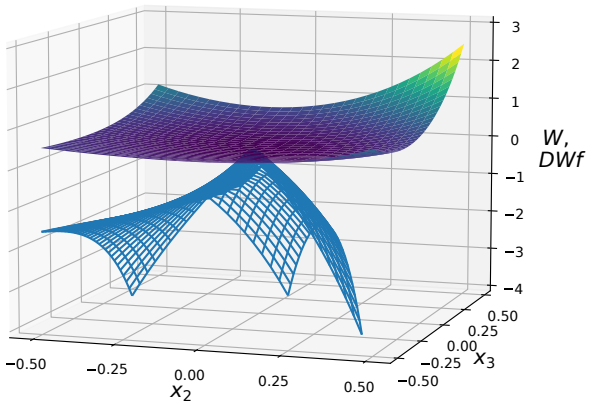

Fig. 5. Approximate CLF on $[-0.5, 0.5]^2$ on the (x_2, x_3)-plane for the system (17) as well as its corresponding orbital derivative on $[-0.5, 0.25]^2$.

6. CONCLUSION AND OUTLOOK

In this paper, we discussed the use of neural networks for computing control Lyapunov functions without suffering from the curse of dimensionality. To this end, we gave examples on the existence and non-existence of separable control Lyapunov functions. These examples motivate future research on conditions for the existence of a separable CLF and on examples where this can be beneficial for control design. Moreover, since Lyapunov functions can be viewed as solutions of particular kinds of Hamilton-Jacobi Bellman PDEs, we are interested in the generalization to optimal control problems and a curse-of-dimensionality-free computation of optimal value functions.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Barron, A. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3), 930–945.

Beneventano, P., Cheridito, P., Graeber, R., Jentzen, A., and Kuckuck, B. (2021). Deep neural network approximation theory for high-dimensional functions. *Preprint, arXiv: 2112.14523*.

Braun, P., Grüne, L., and Kellett, C. (2021). *(In-)Stability of Differential Inclusions. Notions, Equivalences, and Lyapunov-like Characterizations*. Springer, Cham.

Chen, K. and Astolfi, A. (2020). On the active nodes of network systems. In *59th IEEE Conference on Decision and Control (CDC)*, 5561–5566. IEEE.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4), 303–314.

Darbon, J., Langlois, G., and Meng, T. (2020). Overcoming the curse of dimensionality for some Hamilton-Jacobi partial differential equations via neural network architectures. *Research in the Mathematical Sciences*, 7.

Dashkovskiy, S., Ito, H., and Wirth, F. (2011). On a small gain theorem for ISS networks in dissipative lyapunov form. *European Journal of Control*, 17(4), 357–365.

Dashkovskiy, S., Rüffer, B., and Wirth, F. (2010). Small gain theorems for large scale systems and construction of ISS Lyapunov functions. *SIAM Journal on Control and Optimization*, 48(6), 4089–4118.

Gonon, L. and Schwab, C. (2021). Deep relu neural networks overcome the curse of dimensionality for partial integrodifferential equations. *Preprint, arXiv: 2102.11707*.

Grüne, L. (2021). Computing Lyapunov functions using deep neural networks. *Journal of Computational Dynamics*, 8(2), 131–152.

Han, J., Jentzen, A., and E, W. (2018). Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34), 8505–8510.

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

Kang, W. and Gong, Q. (2022). Feedforward neural networks and compositional functions with applications to dynamical systems. *SIAM Journal on Control and Optimization*, 60(2), 786–813.

Kang, W., Sun, K., and Xu, L. (2021). Data-driven computational methods for the domain of attraction and Zubov's equation. *Preprint, arXiv: 2112.14415*.

Khansari-Zadeh, S. and Billard, A. (2014). Learning control Lyapunov function to ensure stability of dynamical system-based robot reaching motions. *Robotics and Autonomous Systems*, 62(6), 752–765.

Liu, T., Hill, D., and Jiang, Z. (2011). Lyapunov formulation of ISS cyclic-small-gain in continuous-time dynamical networks. *Automatica*, 47(9), 2088–2093.

Long, Y. and Bayoumi, M. (1993). Feedback stabilization: Control Lyapunov functions modelled by neural networks. In *Proceedings of 32nd IEEE Conference on Decision and Control*, 2812–2814. IEEE.

Mhaskar, H. (1996). Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation*, 8(1), 164–177.

Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8, 143–195.

Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., and Liao, Q. (2017). Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14(5), 503–519.

Richards, S., Berkenkamp, F., and Krause, A. (2018). The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Proceedings of the 2nd Conference on Robot Learning*, 466–476. PMLR.

Rüffer, B. (2007). *Monotone dynamical systems, graphs, and stability of large-scale interconnected systems*. Ph.D. thesis, Universität Bremen.

Sepulchre, R., Janković, M., and Kokotović, P. (1997). *Constructive Nonlinear Control*. Springer, London.

Sontag, E. and Wang, Y. (1995). On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24, 351–359.

Sontag, E. (1983). A Lyapunov-like characterization of asymptotic controllability. *SIAM Journal on Control and Optimization*, 21(3), 462–471.