2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Neural Exponential Stabilization of Control-affine Nonlinear Systems

Muhammad Zakwan[1], Liang Xu[2], and Giancarlo Ferrari-Trecate[1]

Abstract— **This paper proposes a novel learning-based approach for achieving exponential stabilization of nonlinear control-affine systems. We leverage the Control Contraction Metrics (CCMs) framework to co-synthesize Neural Contraction Metrics (NCMs) and Neural Network (NN) controllers. First, we transform the infinite-dimensional semi-definite program (SDP) for CCM computation into a tractable inequality feasibility problem using element-wise bounds of matrix-valued functions. The terms in the inequality can be efficiently computed by our novel algorithms. Second, we propose a free parametrization of NCMs guaranteeing positive definiteness and the satisfaction of a partial differential equation, regardless of trainable parameters. Third, this parametrization and the inequality condition enable the design of contractivity-enforcing regularizers, which can be incorporated while designing the NN controller for exponential stabilization of the underlying nonlinear systems. Furthermore, when the training loss goes to zero, we provide formal guarantees on verification of the NCM and the exponentional stabilization under the NN controller. Finally, we validate our method through benchmark experiments on set-point stabilization and increasing the region of attraction of a locally pre-stabilized closed-loop system.**

I. INTRODUCTION

Learning-enabled control has demonstrated state-of-the-art empirical performance on various challenging tasks in robotics [1]. However, this performance boost often comes at the cost of safety and stability guarantees, and robustness to disturbances of the learned NN controllers [2]. Traditionally, engineers faced challenges in manually designing certificates such as Lyapunov functions [3], barrier functions [4], and contraction metrics [5], often relying on intuition or experience for specific applications. While numerical methods like sum-of-squares (SoS) have emerged for computing these certificates, many remain impractical [2]. To address these limitations, researchers have employed Neural Networks (NNs) to learn both control policies and safety functions, referred to as neural certificates [6], [7]. These methods have been applied successfully to complex nonlinear control tasks such as stable walking [8], quadrotor flight [2], and safe decentralized control of multi-agent systems [9]. Besides, designing NN controller on dynamical system, several approaches propose modelling the system under control as a NN from data [10], [11], [12] and later ensure the stability of the closed-loop system. However, the main challenge lies in finding tractable and scalable verification techniques for these neural certificates.

This research is supported by the Swiss National Science Foundation under the NCCR Automation (grant agreement 51NF40 180545), and the National Natural Science Foundation of China under Grant 62373239, 62333011.

Muhammad Zakwan and Giancarlo Ferrari-Trecate are with the Institute of Mechanical Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland, email: {muhammad.zakwan, giancarlo.ferraritrecate}@epfl.ch; Liang Xu is with the School of Future Technology, Shanghai University, Shanghai, China, email: liang-xu@shu.edu.cn

In this paper, we train a state feedback controller, parametrized by slope-restricted neural networks (NNs), to achieve exponential stabilization of control-affine nonlinear systems. To this end, we leverage the well-established framework of Control Contraction Metrics (CCMs) [5], [13]. While the existence of CCMs for a dynamical system ensures exponential stabilization, they usually leverage infinite-dimensional intractable inequalities. To tackle this, we derive a tractable inequality guaranteeing exponential stabilizability of the closed-loop system. This condition relies on element-wise bounds of some specific matrix-valued functions. Moreover, we parametrize the CCMs via NNs and guarantee that they are inherently positive definite and satisfy a PDE for all trainable parameters. We call them Neural Contraction Metrics (NCMs). By leveraging both the stabilizability inequality and NCMs, we design regularizers that enforce closed-loop contractivity. These regularizers directly penalize the NN controller's weights based on the Gershgorin disk theorem. Moreover, if the regularizer loss goes to zero, we can formally guarantee the verification of the learned NCM certificate for the NN controller.

Related works: Traditional methods for formulating CCMs, such as SoS programming [14] and Reproducing Kernal Hilbert Space (RKHS) theory [6], suffer from limitations. These include structural restrictions on control input matrices, separate synthesis of controllers and CCMs, assumptions about system dynamics (polynomial or approximated as such), and poor scalability [15]. These drawbacks make these approaches impractical for many real-world applications. Other approximation techniques using gridding methods lack rigorous guarantees [13]. Our proposed method addresses some of the limitations of existing approaches. It provides strong guarantees when the loss function approaches zero and scales well to handle a large number of states. However, there is still room for improvement in terms of conservatism.

Recent research explores synthesizing safety certificates through NNs. Both the controller and certificates are usually parameterized via NNs, followed by verification [15]. A popular technique, the learner-verifier approach (also called counter-example guided inductive synthesis), involves training a certificate network while a verifier (usually using a satisfiability modulo theory (SMT) solver) assesses its feasibility [15]. If valid, the verifier halts training; if invalid, it provides counterexamples for training data enrichment. Verification can be implemented in various ways. For instance, if given piecewise affine dynamics and NNs (for controllers and certificates) are implemented with ReLU activations, verification can be formulated as a Mixed-Integer Linear Program (MILP) solved with standard solvers [16], [17]. These methods have gained traction and have been applied to learning Lyapunov functions for general nonlinear systems [6], [18] and hybrid systems [19]. However, both SMT- (NP-complete [15]) and MILP-based methods are computationally expensive and their complexity grows exponentially with the number of neurons in the NN certificate, limiting their applicability [15]. Additionally, Lyapunov-based methods are restricted to non-autonomous systems. On the contrary, CCMs, being more general, can handle time-varying systems and are the focus of this paper.

Some recent works explore learning CCMs with NNs. For example, the work [20] uses recurrent NNs to parametrize the contraction metric before constructing the controller. However, this approach assumes specific dynamics and is limited to convex combinations of state-dependent coefficients. In contrast, under mild assumptions, our method can simultaneously synthesize the controller and the associated CCM certificate for general control-affine systems. The work [2] also proposes an NN-based framework for co-synthesizing CCMs and controllers for control-affine systems. However, there are several key differences with our work. First, [2] addresses tracking problems, assuming access to desired trajectories and control policies. We focus instead on the exponential stabilization of equilibrium, a different problem formulation. Second, we provide a free-parametrization of NCMs that are positive definite by design and satisfy a PDE constraint characteristic of CCMs for all trainable parameters. On the other hand, the authors of [2] try to penalize this constraint empirically, and do not provide rigorous guarantees. Finally, in specific cases, for instance, when the underlying system is global Lipschitz, our method provides global stability guarantees, which is not the case in [2].

Contributions: The contributions of our paper can be summarized as follows:

- We propose an NN training problem that simultaneously learns both the controller and the CCM for achieving exponential stabilization of control-affine nonlinear systems. We provide formal guarantees for the verification of the learned NCM under the condition that the training loss converges to zero. Furthermore, our approach is scalable w.r.t both the dimension of state and the number of neurons in the NN controller.
- We introduce a novel free parametrization for NCMs. These NCMs are inherently positive definite, and their Jacobians satisfy a PDE constraint regardless of the trainable parameters. This eliminates the need for explicit penalization of the constraint violation during training, simplifying the training process.
- We develop an algorithm that computes element-wise upper and lower bounds of the Jacobian of the NN controller. These bounds are then leveraged to transform the verification of CCMs from an infinite-dimensional SDP into a more tractable finite-dimensional inequality (see Theorem 2).
- We validate the efficacy of our method through benchmark experiments on two key tasks: set-point stabilization and expanding the region of attraction of a locally pre-stabilized closed-loop system.

Organization: Section II provides a brief overview of contraction theory and CCMs. Section III presents our core contributions: designing regularizers that enforce closed-loop contractivity, introducing a free parametrization of NCMs, and outlining Algorithm 1 to facilitate implementation of these regularizers. Section IV validates the proposed methods through benchmark experiments. Finally, Section V draws some conclusions.

Notation: We denote by $\mathbb{R}$ and $\mathbb{R}_{\geq 0}$ the set of real and non-negative real numbers respectively. For a symmetric matrix $A \in \mathbb{R}^{n\times n}$, the notation $A \succ 0 (A \prec 0)$ means A is positive (negative) definite. The set of positive definite $n \times n$ matrices is denoted by $\mathbb{S}_{\succ 0}$. For a matrix-valued function $M(x) : \mathbb{R}^n \mapsto \mathbb{R}^{n\times n}$, its element-wise Lie derivative along a vector $v \in \mathbb{R}^n$ is $\partial_v M := \sum_i v_i \frac{\partial M}{\partial x_i}$. Unless otherwise stated, x_i denotes the i-th element of vector x. For a matrix $M \in \mathbb{R}^{n\times n}$, we denote $M + M^\top$ by Sym$[M]$. For two matrices A and B of the same dimensions, the notation $A < (\leq) B$ means that all the entries of A are element-wise less (less-equal) than the entries of B. The maximum eigenvalue of a symmetric matrix A is given by $\bar{\lambda}(A)$. The kernel of a linear map g is defined by $\mathbf{Ker}(g)$.

II. Preliminaries

In this paper, we consider control-affine systems as

$$\dot{x} = f(x) + gu(t) \ , \qquad (1)$$

where $x \in \mathcal{X} \subseteq \mathbb{R}^n$, $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ for all $t \in \mathbb{R}_{\geq 0}$ are states, and inputs respectively. Here, $\mathcal{X}$, and $\mathcal{U}$ are compact state and input sets, respectively. We assume that $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ is a smooth map, the control input $u : \mathbb{R}_{\geq 0} \mapsto \mathcal{U}$ is a piece-wise continuous function. The goal of this paper is to design a NN state-feedback controller $u_\theta(x)$, where θ are the trainable parameters, such that the controlled trajectory $x(t)$ can reach the desired equilibrium point $x^\star$ whenever $x(0)$ is in a neighborhood of $x^\star$. In this work, we leverage contraction theory to achieve our goal.

Contraction theory [21], [5] analyzes the incremental stability of systems by examining the evolution of the distance between neighboring trajectories. For a time-invariant autonomous system $\dot{x} = f(x)$, given a pair of neighboring trajectories denote the infinitesimal displacement between them by δx (also called a virtual displacement). The evolution of δx can be represented by a linear time-varying (LTV) system: $\dot{\delta x} = \partial_x f(x)\delta x$. Then, the squared distance between these trajectories $\delta x^\top \delta x$, evolves according to $\frac{d}{dt}(\delta x^\top \delta x) = 2\delta x \dot{\delta x} = 2\delta x^\top \partial_x f(x)\delta x$. If the symmetric part of the Jacobian $\partial_x f$ is uniformly negative definite, i.e., $\frac{1}{2}(\partial_x f + \partial_x f^\top) \preceq -\rho I$ for some $\rho > 0$, then the system is contracting. This condition ensures that $\delta x^\top \delta x$ converges exponentially to zero at a rate of 2ρ. Consequently, all trajectories of the system converge to a common equilibrium trajectory [21].

The concept of contraction can be generalized using a contraction metric $M : \mathbb{R}^n \mapsto \mathbb{S}_{\succ 0}$, which is a smooth matrix-valued function. Since $M(x)$ is always positive definite, if $\delta x^\top M(x)\delta x$ converges exponentially to zero, the system is contracting. The converse is also true [13]. Contraction theory can be further extended to control-affine systems. First, the corresponding differential dynamics of (1) is given by $\dot{\delta x} = F(x)\delta x + g\delta u$, where $F(x) := \partial_x f$, and δu is the infinitesimal displacement between two neighboring inputs. A fundamental theorem in Control Contraction Metric (CCM) theory [13] says that if there exists a metric $M(x)$ such that the following conditions hold for all x and some $\rho > 0$

$$\dot{M}(x) + \text{Sym}[M(x)\partial_x(f(x) + gu)] \prec -2\rho M(x) \ .$$

Then, the closed-loop system is contracting with rate ρ under metric $M(x)$ [13]. However, as pointed out in [13], finding a suitable metric or designing a controller based on a predetermined metric can be challenging. To address this limitation, we propose a method that leverages NNs to learn both the metric and the controller simultaneously, as explored in [2].

III. Main Results

This section leverages contraction theory to design NN controllers for exponential stabilization. We begin by introducing a class of NNs with slope-restricted activation functions, well-suited for parameterizing the control policy for the system (1). Next, we establish a sufficient condition to guarantee the exponential stabilization of the system (1). This condition relies on element-wise upper and lower bounds of terms crucial for CCM synthesis. We also provide algorithms for computing these bounds efficiently. Furthermore, we propose a novel free parametrization of NCMs that ensures they are inherently positive definite and their Jacobians satisfy CCM-related PDE constraint by design. By leveraging our proposed condition, and NCMs, we design regularizers that enforce contractivity in the closed-loop system.

NN architecture: We consider the following class of NN controllers

$$\begin{aligned} z_1 &= \sigma_1\left(W_1 x + b_1\right) \\ z_\ell &= \sigma_\ell\left(W_\ell z_{\ell-1} + b_\ell\right), \ \ell = 2, \cdots, N \ , \\ u_\theta &= W_o z_N \ , \end{aligned} \tag{2}$$

where $N \geq 1$ represents the depth of the NN, and the set $\theta := \{W_o, W_N, \cdots, W_1, b_1, \cdots, b_N\}$ contains all the trainable parameters. We consider a class of slope-restricted activation functions $\sigma(\cdot)$ such that their derivative w.r.t. the input satisfies $0 < a \leq \sigma'(\cdot) < b$. A typical activation function that satisfies this assumption is the smooth LeakyReLu

$$\sigma(x) = \alpha x + (1-\alpha)\log(1 + e^x) \ , \tag{3}$$

where α controls the angle of the negative slope. The proposed parametrization exhibits universal approximation properties [22].

Neural contractive closed-loop system: Before presenting our main results, let us define the set containing the desired equilibrium point $x^\star$ for the system (1) as

$$\mathcal{E} := \left\{ x \in \mathcal{X} | f(x^\star) + gu_\theta(x^\star) = 0 \right\} .$$

Then, the following result holds, adapted from [13]

Theorem 1: Suppose the set $\mathcal{E}$ is non-empty, and there exists a NN $M_\phi : \mathcal{X} \mapsto \mathbb{S}_{\succ 0}$ endowed with some trainable parameters ϕ, such that

$$\dot{M}_\phi + \text{Sym}\left[M_\phi \partial_x(f(x) + gu_\theta(x))\right] \prec -2\rho M_\phi \tag{4}$$

holds for all $x \in \mathcal{X}$ and for some $\rho > 0$. Then, the equilibrium point $x^\star$ is exponentially stable. ■

Note that the inequality (4) in Theorem 1 is an infinite-dimensional SDP and can be cumbersome to verify for all $x \in \mathcal{X}$ using standard convex optimization solvers [13]. To tackle this issue, we provide a sufficient condition for ensuring the exponential stability of the closed-loop system. Specifically, this condition is based on the Gershgorin disc theorem and the proof is provided in Appendix of the accompanying technical report [23].

Theorem 2: Suppose $x^\star \in \mathcal{E}$ is the desired equilibrium point of the system (1) and there exists a metric NCM $M_\phi : \mathcal{X} \mapsto \mathbb{S}_{\succ 0}$ endowed with some trainable parameters ϕ. Then, $x^\star$ is exponentially stable if the following inequality is satisfied

$$2U_{ii} + \eta \leq -\sum_{j \neq i} \max\left\{|L_{ij} + L_{ji}|, |U_{ij} + U_{ji}|\right\} \ , \tag{5}$$

where U_{ii} is the ith diagonal entry of the matrix $U = U_\theta + U_\phi$, and L_{ij} is the (i,j) entry of the matrix $L = L_\theta + L_\phi$ and $L_\theta, L_\phi, U_\theta, U_\phi$ verify

$$L_\theta \leq M_\phi(x) g \partial_x u_\theta(x) \leq U_\theta, \ L_\phi \leq \dot{M}_\phi(x) \leq U_\phi \ .$$

Moreover, $\eta = -(c_1 + c_2)$ is a scalar and the constants c_1 and c_2 satisfy

$$c_1 \geq 2\rho \sup_{s \in \mathcal{X}} \bar{\lambda}(M_\phi(s)), c_2 \geq \sup_{s \in \mathcal{X}} \bar{\lambda}\left(\text{Sym}[M_\phi(s)\partial_s f(s)]\right)$$

for all $x \in \mathcal{X}$. ■

Once the constants c_1 and c_2, and the element-wise bounds L and U have been determined, condition (5) becomes a finite-dimensional scalar inequality, making its verification significantly easier. While Theorem 2 leverages the Gershgorin disc theorem, resulting in a more conservative condition compared to Theorem 1, this trade-off is necessary for tractability. Nevertheless, as shown in Section IV, good performance can still be achieved despite this conservatism.

Neural Contraction Metric (NCM): An integral part of Theorem 2 is the parametrization of the CCM by a NN. In this paper, motivated by [2], we parametrize NCM as $M_\phi(x) = \Gamma_\phi(x)^\top \Gamma_\phi(x) + \epsilon I$, where $\Gamma_\phi : \mathcal{X} \mapsto \mathbb{R}^{n \times n}$ is a smooth matrix-valued function depending on some trainable weights ϕ, and $\epsilon > 0$ is a small positive constant. Since $M_\phi(x)$ is positive definite irrespective of the weight matrices ϕ and for all values of $x \in \mathcal{X}$, as desired, we call it a free parametrization. Besides the positive definiteness of $M_\phi(x)$, in inequality (4), $\dot{M}_\phi(x)$ is a matrix with (i,j) elements given by $\partial_x \left(M_{\phi,ij}^\top(x)\right)(f(x)+gu_\theta(x))$, which are affine in the control signal u. Therefore, for inequality (4) to hold for all u, it is crucial that

$$\partial_x \left(M_{\phi,ij}^\top(x)\right) g = 0 \tag{6}$$

for all $x \in \mathcal{X}$. The constraint (6) is a PDE and is cumbersome to solve. Motivated by [24], the next result provides a parametrization of the NCM $M_\phi(x)$ such that the state-dependent equality (6) is satisfied by design, that is, regardless of the choice of trainable parameters.

Proposition 1: Suppose the dimension of **Ker**(g) is r, and let $v_1, ..., v_r$ be the basis of **Ker**(g). Define each entry of $\Gamma_\phi(x)$ as

$$\Gamma_{\phi,ij}(x) := K_{ij}\left(\sum_{\ell=1}^{r}\beta_{\ell,ij}(x^\top v_\ell)\right), \forall\ i,j\ , \tag{7}$$

for some continuously differentiable NNs $K_{ij} : \mathbb{R} \mapsto \mathbb{R}, \beta_{\ell,ij} : \mathbb{R} \mapsto \mathbb{R}, \ell = 1,\ldots,r$. Then, taking $M_\phi(x) = \Gamma_\phi(x)^\top\Gamma_\phi(x) + \epsilon I$ satisfies (6) for all $x \in \mathcal{X}$. ■

The proof is provided in Appendix of the accompanying technical report [23]. Notably, in most real-world applications, the systems are under-actuated where the rank of the control matrix, i.e. input dimension is lower than the number of states ($m < n$). This guarantees the existence of a non-trivial subspace **Ker**(g). Interested readers are deferred to [23, Example 1] for a an example that showcases the essence of Proposition 1 by demonstrating how to parameterize NCMs for a single-input control-affine system.

NN controllers with bounded Jacobians: To compute the element-wise lower and upper bounds L_θ and U_θ appearing in Theorem 2, it is crucial to calculate the element-wise bounds of the Jacobian of the NN controller (2). To this end, we write the Jacobian as:

$$\partial_x u_\theta(x) = W_o J_N W_N J_{N-1} W_{N-1} \ldots J_1 W_1\ , \tag{8}$$

where each $J_i = \sigma'(W_i x + b_i)$ and satisfies $0 < aI \leq J_i \leq bI$. Leveraging the special structure of the proposed NN controller (2), in the following, we propose Algorithm 1 to compute element-wise lower $L_{\partial_x u}$ and upper $U_{\partial_x u}$ bounds of (8). This algorithm relies on two functions:

- **Func1:** This function, provided in Appendix of [23], calculates the element-wise lower $\tilde{L}$ and upper bound $\tilde{U}$ of a matrix product WQ. It takes the element-wise upper bound U and lower bound L of a matrix Q, along with the matrix W as inputs.
- **Func2** This function, provided in Appendix of [23], calculates the element-wise lower $\hat{L}$ and upper bound $\hat{U}$ of the matrix product $J_i P$, where J_i are the Jacobian matrices as in (8). It takes the element-wise upper $\tilde{U}$ and lower bound $\tilde{L}$ matrices of an arbitrary matrix P, and the scalars a and b (bounds on the slope of activation functions) as inputs.

Note that once the element-wise bounds on the Jacobian of the NN controller are determined, it is straightforward

Algorithm 1 Returns the upper $U_{\partial_x u}$ and lower $L_{\partial_x u}$ bound of $\partial_x u_\theta$

Set $L = I$, $U = I$
for $i = 1, \cdots, N$ **do**
  call **Func1**(W_i, L, U) and return $\tilde{L}, \tilde{U}$ (which calculates the element-wise upper and lower bound of $W_i J_{i-1} W_{i-1} \ldots J_1 W_1$)
  call **Func2**$(\tilde{L}, \tilde{U}, a, b)$ and return $\hat{L}, \hat{U}$ (which calculates the element-wise upper and lower bound of $J_i W_i J_{i-1} W_{i-1} \ldots J_1 W_1$)
  set $L = \hat{L}, U = \hat{U}$
end for
return L, U

to calculate the bounds L_θ and U_θ in Theorem 2 using the **Func3**, provided in Appendix of [23]. Moreover, one can employ bounded activation functions, e.g. $\tanh(\cdot)$ in the output layer of NNs K_{ij} in (7) to easily calculate the element-wise lower and upper bounds of the NCM $M_\phi(x)$. Likewise, the element-wise lower L_ϕ and upper U_ϕ bounds of $\dot{M}_\phi(x)$ can be computed via **Func1**.

Remark 1 (Computational complexity of Algorithm 1): The complexity of **Func1**(W, L, U), where $W \in \mathbb{R}^{m\times n}$, and $L, U \in \mathbb{R}^{n\times p}$ is $\mathcal{O}(mnp)$ and for **Func2**(L, U, a, b) where $L, U \in \mathbb{R}^{n\times p}$ is $\mathcal{O}(np)$. Finally, for Algorithm 1, we have $\mathcal{O}(Nmnp)$, where N is the total number of layers in the NN controller (2). In the worst case scenario, i.e., $m = n = p$, our verification method has the computational complexity of $\mathcal{O}(Nn^3)$. On the other hand, both MILP and SMT-based verification methods have significant computational overhead and their complexity grows exponentially with the number of neurons [2].

Non-uniform bounded Jacobian of $f(x)$: After determining the element-wise lower L and upper U bounds inequality (5), our goal becomes computing the positive scalar $\eta = -(c_1 + c_2)$ to leverage Theorem 2 for designing contractivity-enforcing regularizers. Finding c_2 is crucial for this step, and it necessitates a uniformly bounded Jacobian $\partial_x f$ for all $x \in \mathcal{X}$. In many real-world applications (e.g., globally Lipschitz-bounded nonlinear systems), the Jacobian of the underlying dynamics exhibits uniform global boundedness (see Section IV). However, for systems that lack this property, careful gridding in the neighborhood of the desired equilibrium point x^* can be employed. Nevertheless, to provide deterministic guarantees that this uniform bound holds across all $x \in \mathcal{X}$, it is necessary to define an upper bound on the distance between two adjacent gridding samples [2].

Given a Lipschitz continuous function $h : \mathcal{X} \to \mathbb{R}$ endowed with a Lipschitz constant L_h, assume one discretizes the domain $\mathcal{X}$ such that the distance between any sampled point x_i and its nearest neighbor x_j is less than $\|x_j - x_i\| < \tau$. If $h(x_i) < -L_h\tau$ holds for all sampled points $x_i \in \mathcal{X}$, then $h(x) < 0$ holds for all $x \in \mathcal{X}$. The following proposition provides deterministic guarantees of the existence of c_2 and provides an upper bound on the resolution of the gridding

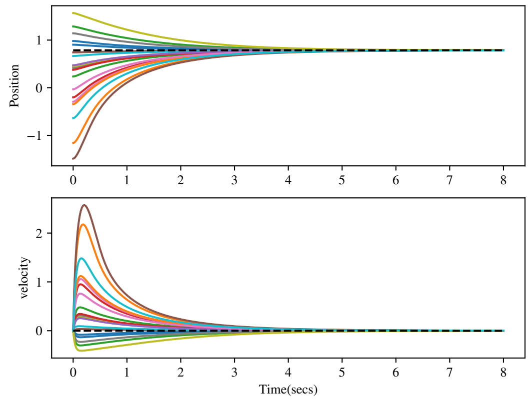

Fig. 1. Closed-loop response for 20 different initial conditions demonstrating stabilization at $(\pi/4, 0)$.

samples.

Proposition 2: Let L_{δ_x} and L_M be the Lipschitz constants of $\partial_x f$, and $M_\phi(x)$, respectively. Then, $\bar{\lambda}(\text{Sym}[M_\phi(x)\partial_x f(x)])$ has the Lipschitz constant of $2(S_M L_{\delta_f} + S_{\delta_f} L_M)$, where S_M and S_{δ_f} satisfy $\|M_\phi\|_2 \leq S_M$, and $\|\partial_x f(x)\|_2 \leq S_{\delta_f}$, respectively. ■

The detailed proof is given in Appendix of [23]. Likewise, Proposition 2 can be utilized to compute the constant c_1 in Theorem 2. Moreover, the readers are deferred to [2, Appendix B.2] for more information on the calculations of these Lipschitz constants.

Control and NCM design: Based on Theorem 2, and Algorithm 1, we define the following optimization problem to simultaneously train the NN controller (2) and NCM $M_\phi(x)$:

$$\min_{\theta,\phi} \ell_1(x^*) + \nu \ell_2(\eta, L, U) \;, \tag{9}$$

with

$$\begin{aligned}\ell_1(x^*) &= ||f(x^*) + g u_\theta(x^*)|| \\ \ell_2(\eta, L, U) &= [2U_{ii} + \eta \\ &\quad + \sum_{j\neq i} \max(|L_{ij} + L_{ji}|, |U_{ij} + U_{ji}|)]_+ \;,\end{aligned}$$

where $[x]_+ = \max\{0, x\}$ operates element-wise. The regularizer ℓ_1 assigns the equilibrium condition, the regularizer ℓ_2 enforces the contractivity, rendering the equilibrium point exponentially stable, and ν trades off both regularizers. If both losses converge to zero, we have a valid certificate NCM for the corresponding state-feedback controller policy $u_\theta(x)$.

IV. Experiments

In this section, we discuss two numerical examples to validate the efficacy of our method. In Section IV-A, the objective is to stabilize a standard pendulum. In Section IV-B, we consider a pre-stabilized non-linear system with an LQR controller and then, we leverage our method to design an NN controller to expand the region of attraction.

A. Exponential set-point stabilization

Standard pendulum: Consider the following dynamics

$$\dot{x} = \begin{bmatrix} x_2 \\ \frac{-mgl\sin(x_1) - 0.1x_2 + u}{ml^2} \end{bmatrix} \;, \tag{10}$$

where x_1 and x_2 are the positions and the angular velocity, respectively. Moreover, $g = 9.81$, $m = 0.15$, and $l = 0.5$. In this experiment, our goal is to design a NN controller (2) such that the closed-loop dynamics are contractive and have the equilibrium point $(x_1^\star, x_2^\star) = (\pi/4, 0)$. We choose CCM $M = I$, and for the NN controller, we choose a three-layered network with 32 neurons in the hidden layer, and the activation function (3) with $a = 0.3$, and $b = 1.0$. In this case, the Jacobian of the dynamics is

$$\partial_x f(x) = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l}\cos(x_1) & -\frac{0.1}{ml^2} \end{bmatrix},$$

and we can uniformly bound its eigenvalues as $c_2 = \sup_{x\in\mathbb{R}^n} \bar{\lambda}(\text{Sym}[\partial_x f]) = 20.45$. We trained the NN controller using the loss $\ell = \ell_1 + \ell_2$ and in this case, the optimal value of ℓ was zero. This is also evident from the evolution of the closed-loop trajectories starting from 20 arbitrary initial conditions depicted in Fig. 1, where all the trajectories converge to the desired equilibrium point.[1] Curious readers are deferred to [23, Section IV] for a similar experiment with an inverted pendulum.

B. Enhancing the region of attraction

As discussed in a prior work [13] and references therein, researchers have investigated the challenge of unifying locally optimal and globally stabilizing controllers. This task is particularly difficult within the Lyapunov framework because the set of control Lyapunov functions for a system is non-convex. However, the CCM framework offers a straightforward approach to achieving this goal. Let us illustrate with an example taken from [13], where state $x = [x_1, x_2, x_3]^\top$ and the system follows the dynamics (1) with

$$f(x) = \begin{bmatrix} -x_1 + x_3 \\ x_1^2 - x_2 - 2x_1x_3 + x_3 \\ -x_2 \end{bmatrix}, g = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{11}$$

We first solve the linear quadratic regulator (LQR) problem for the system linearized at the origin with cost function $\int_0^\infty (x^\top x + ru^2)dt$, where $r = 1$, obtaining a solution $P = P^\top > 0$ of the algebraic Riccati equation and the locally optimal controller $u = -r^{-1}B^\top Px$.

Next, we trained an NN controller for 2000 epochs to achieve a desired equilibrium point of $(x_1^\star, x_2^\star, x_3^\star) = (0, 0, 0)$ for the pre-stabilized closed-loop system. The NN controller architecture consisted of a single hidden layer with 64 neurons and utilized the activation function (3) with $a = 0.3$ and $b = 1.0$. For the sake of simplicity, we choose NCM $M_\phi(x) = I$.

Figure 2 demonstrates that for small initial conditions, the performance of the NN controller is nearly identical to that of

[1] Our code is available at https://github.com/DecodEPFL/Neural-Exponential-Stabilization

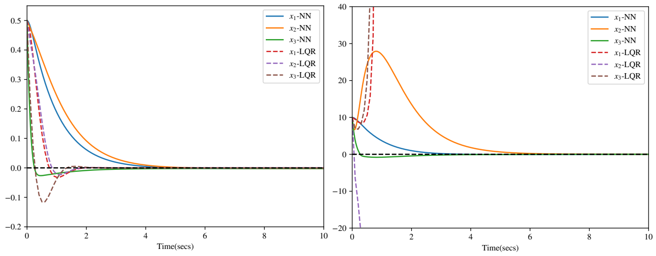

Fig. 2. Response of nonlinear system (11) with NCM and LQR control to initial state $x(0) = [0.5, 0.5, 0.5]^\top$ (left) and $x(0) = [10, 10, 10]^\top$ (right). This exhibits the locally optimal and increased region of attraction of the NCM controller.

the LQR controller. Additionally, with a contraction rate of ρ and $M_\phi(x)$, the NN control law (refer to (2)) approximates a basic linear feedback on the error $x - x^\star$. In contrast, the LQR controller fails to achieve stability for larger initial conditions, while the NN controller successfully stabilizes the system. The simulations under LQR control shown in the right panel of Figure 2 diverge rapidly after approximately 1 second. Conversely, the NN control law demonstrates its efficacy by achieving stability even for a large initial state of $x_0 = [10.0, 10.0, 10.0]^\top$.

V. Concluding Remarks

For decades, automatic certificate synthesis in control theory has been elusive. Traditional methods like LP-based synthesis and SOS programming provide partial solutions but lack scalability. This paper details a novel co-synthesis approach for a NN controller and its NCM certificate for exponential stabilization of a class of nonlinear systems. Specifically, we transformed an intractable NCM computation (an infinite-dimensional SDP) into a tractable condition based on some element-wise upper and lower bounds of some matrix-valued functions. Our proposed algorithm efficiently computes these bounds. We leveraged this to design contraction-enforcing regularizers for equilibrium-assignment problems and enhancing region of attraction. Furthermore, under zero training loss, this training scheme provides us with a formal verification of the NN controller. Finally, we introduce a novel NCM parametrization ensuring positive definiteness and a PDE constraint satisfaction regardless of trainable parameters. The future efforts will be devoted to generalizing our approach to other classes of nonlinear systems.

References

[1] W. Xiao, T.-H. Wang, R. Hasani, M. Chahine, A. Amini, X. Li, and D. Rus, "Barriernet: Differentiable control barrier functions for learning of safe robot control," *IEEE Transactions on Robotics*, 2023.

[2] D. Sun, S. Jha, and C. Fan, "Learning certified control using contraction metric," in *Conference on Robot Learning*. PMLR, 2021, pp. 1519–1539.

[3] A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames, "Episodic learning with control Lyapunov functions for uncertain robotic systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6878–6884.

[4] A. Peruffo, D. Ahmed, and A. Abate, "Automated and formal synthesis of neural barrier certificates for dynamical models," in *International conference on tools and algorithms for the construction and analysis of systems*. Springer, 2021, pp. 370–388.

[5] H. Tsukamoto, S.-J. Chung, and J.-J. E. Slotine, "Contraction theory for nonlinear stability analysis and learning-based control: A tutorial overview," *Annual Reviews in Control*, vol. 52, pp. 135–169, 2021.

[6] Y.-C. Chang, N. Roohi, and S. Gao, "Neural Lyapunov control," *Advances in neural information processing systems*, vol. 32, 2019.

[7] S. M. Richards, F. Berkenkamp, and A. Krause, "The Lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems," in *Conference on Robot Learning*. PMLR, 2018, pp. 466–476.

[8] J. Choi, F. Castaneda, C. J. Tomlin, and K. Sreenath, "Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions," *arXiv preprint arXiv:2004.07584*, 2020.

[9] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multi-agent control with decentralized neural barrier certificates," *arXiv preprint arXiv:2101.05436*, 2021.

[10] L. B. Armenio, E. Terzi, M. Farina, and R. Scattolini, "Model predictive control design for dynamical systems learned by echo state networks," *IEEE Control Systems Letters*, vol. 3, no. 4, pp. 1044–1049, 2019.

[11] F. Bonassi, M. Farina, J. Xie, and R. Scattolini, "On recurrent neural networks for learning-based control: recent results and ideas for future developments," *Journal of Process Control*, vol. 114, pp. 92–104, 2022.

[12] E. Terzi, F. Bonassi, M. Farina, and R. Scattolini, "Learning model predictive control with long short-term memory networks," *International Journal of Robust and Nonlinear Control*, vol. 31, no. 18, pp. 8877–8896, 2021.

[13] I. R. Manchester and J.-J. E. Slotine, "Control contraction metrics: Convex and intrinsic criteria for nonlinear feedback design," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 3046–3053, 2017.

[14] W. Tan and A. Packard, "Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming," *IEEE Transactions on Automatic Control*, vol. 53, no. 2, pp. 565–571, 2008.

[15] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural Lyapunov, barrier, and contraction methods for robotics and control," *IEEE Transactions on Robotics*, 2023.

[16] H. Dai, B. Landry, M. Pavone, and R. Tedrake, "Counter-example guided synthesis of neural network Lyapunov functions for piecewise linear systems," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 1274–1281.

[17] R. Schwan, C. N. Jones, and D. Kuhn, "Stability verification of neural network controllers using mixed-integer programming," *IEEE Transactions on Automatic Control*, 2023.

[18] A. Abate, D. Ahmed, M. Giacobbe, and A. Peruffo, "Formal synthesis of Lyapunov neural networks," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 773–778, 2020.

[19] S. Chen, M. Fazlyab, M. Morari, G. J. Pappas, and V. M. Preciado, "Learning Lyapunov functions for hybrid systems," in *Proceedings of the 24th International Conference on Hybrid Systems: Computation and Control*, 2021, pp. 1–11.

[20] H. Tsukamoto and S.-J. Chung, "Neural contraction metrics for robust estimation and control: A convex optimization approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 211–216, 2020.

[21] W. Lohmiller and J.-J. E. Slotine, "On contraction analysis for non-linear systems," *Automatica*, vol. 34, no. 6, pp. 683–696, 1998.

[22] L. Li, Y. Duan, G. Ji, and Y. Cai, "Minimum width of leaky-ReLU neural networks for uniform universal approximation," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 19 460–19 470.

[23] M. Zakwan, L. Xu, and G. Ferrari-Trecate, "Neural exponential stabilization of control-affine nonlinear systems," *arXiv preprint arXiv:2403.17793*, 2024.

[24] L. Xu, M. Zakwan, and G. Ferrari-Trecate, "Neural energy casimir control for Port-Hamiltonian systems," in *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 4053–4058.