Efficient Computation of Lyapunov Functions Using Deep Neural Networks for the Assessment of Stability in Controller Design

Caglar Uyulan (caglar.uyulan@ikcu.edu.tr)
İzmir Katip Çelebi University https://orcid.org/0000-0002-6423-6720

Research Article

Keywords: Deep Neural Network, Lyapunov Function, Stability Analysis, Small-Gain Condition, Curse of Dimensionality

Posted Date: December 7th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3698604/v1

Additional Declarations:
Competing interests: The authors declare no competing interests.

Efficient Computation of Lyapunov Functions Using Deep Neural Networks for the Assessment of Stability in Controller Design

Caglar Uyulan[1]

[1]İzmir Katip Çelebi University, Faculty of Engineering and Architecture, Department of Mechanical Engineering, İzmir, Turkey

ORCID and email: 0000-0002-6423-6720 / caglar.uyulan@ikcu.edu.tr

Abstract

This paper presents a deep neural network (DNN) based method to estimate approximate Lyapunov functions and their orbital derivatives, which are key to the stability of the system in control theory. Our approach addresses the challenge of the curse of dimensionality in control and optimization problems, demonstrating that the computational effort required grows only polynomically with the state dimension.

This is a significant improvement over traditional methods. We emphasize that the calculated functions are approximations of Lyapunov functions and not exact representations. This distinction is important, as validating these approximations in high-dimensional environments is challenging and opens new avenues for future research.

Our approach diverges from traditional grid-based approaches and moves away from relying on small-gain theorems and accurate subsystem knowledge. This flexibility is proven in deriving Lyapunov functions for the development of stabilizing feedback rules in nonlinear systems.

A crucial aspect of our approach is the use of ReLU activation features in neural networks, which is steady with modern deep getting-to-know traits. We also explore the feasibility of using DNNs to estimate fairly constructive Lyapunov functions, despite the demanding situations posed through uncertainty.

Our set of rules' outcomes aren't unique, highlighting the want to set up criteria for figuring out especially useful Lyapunov features. The paper culminates with the aid of emphasizing the capacity of DNNs to approximate compositional Lyapunov capabilities, especially underneath small-gain conditions, to mitigate the curse of dimensionality.

Our contributions are manifold, including scalability in dealing with systems of various dimensionality, flexibility in accommodating both low and excessive-dimensional structures, and performance in computing Lyapunov features through deep learning strategies.

However, challenges continue to be in the approximation accuracy and the verification of Lyapunov functions in better dimensions.

Keywords: Deep Neural Network, Lyapunov Function, Stability Analysis, Small-Gain Condition, Curse of Dimensionality.

1. Introduction

Lyapunov's theory, pivotal in the analysis of the stability of dynamical systems, offers a unique approach that is particularly effective in handling nonlinear systems. In contrast to other methods, which often rely on linearization or specific system models, Lyapunov's approach is more general and less restrictive. Lyapunov's stability theory, formulated by the Russian mathematician Aleksandr Lyapunov, has become a cornerstone in the analysis of dynamical systems [Lyapunov & Walker, 1994]. Unlike other stability criteria, which are predominantly model-based and require specific system knowledge, Lyapunov's theory offers a more general framework. Its primary distinction lies in its ability to handle systems without explicit reliance on the system models, as opposed to the model-dependent nature of traditional stability criteria. Lyapunov functions are crucial in the stability analysis of nonlinear systems, providing insights against perturbations. Despite their importance, explicit Lyapunov functions are often elusive, prompting significant interest in their numerical computation.

Traditional approaches, including series expansions, finite element methods, and sum-of-squares techniques, encounter the curse of dimensionality, limiting their applicability to low-dimensional systems.

First, we define the basic concept of the Lyapunov stability theory. Lyapunov's stability theory is based on the concept of energy-like functions, known as Lyapunov functions, to ascertain the stability of equilibrium points of dynamical systems. The theory is delineated into two primary approaches [Khalil, 2015; Slotine&Li, 2005]:

a-Lyapunov's Direct Method:

-It involves constructing a Lyapunov function $V(x)$; $V: \mathbb{R}^n \rightarrow \mathbb{R}^n$ for the time-invariant system $\dot{x} = f(x)$, where $x \in \mathbb{R}^n$ is the state vector.

-The function $V(x)$ is scalar, continuously differentiable, and positive definite ($V(x) > 0 \; for \; x \neq 0$ and $V(0) = 0$)

-The time derivative of $V(x)$ along the system's trajectories,$\dot{V}(x) = \nabla V(x).f(x)$, must be negative definite or negative semi-definite for the system to be stable or asymptotically stable, respectively. (If $\dot{V}(x) \leq 0$ for all x, the system is stable, if $\dot{V}(x) < 0$, it's asymptotically stable).

a-Lyapunov's Indirect Method:

-Focuses on linearizing the system at the equilibrium point and analyzing the stability of the linearized system.

-The stability of the nonlinear system is deduced from the stability of its linear approximation.

-Consider the nonlinear system $\dot{x} = f(x)$ with an equilibrium point $x = x_{eq.}$. The linearized system around this equilibrium point is $\dot{x} = Ax$, where $A = \frac{\partial f}{\partial x}\Big|_{x=x_{eq.}}$ is the Jacobian matrix evaluated at the equilibrium point.

The stability of the equilibrium point of the original nonlinear system is inferred from the stability of the linearized system. The criteria are based on the eigenvalues of the matrix A:

-If all eigenvalues of A have negative real parts, the equilibrium point is asymptotically stable.

-If any eigenvalue has a positive real part, the equilibrium point is unstable.

-If eigenvalues have non-positive real parts, but there exist eigenvalues with zero real parts, the method is inconclusive, and further analysis is required.

Underlying assumptions are that the system is locally linearizable around the equilibrium point, and the Jacobian matrix A accurately represents the dynamics near the equilibrium.

Other stability criteria, such as the Routh-Hurwitz criterion, Bode plots, and Nyquist criteria, primarily focus on linear time-invariant (LTI) systems. These methods often require explicit system models and are limited in their applicability to nonlinear systems.

Lyapunov's stability theory does not necessitate the linearization of the system or the availability of a precise mathematical model. This makes it inherently more suitable for a

broader range of systems, particularly nonlinear and time-varying systems. Furthermore, it provides a constructive approach to controller design, particularly in the field of robust and adaptive control.

Lyapunov's stability theory stands out due to its generality, applicability to nonlinear systems, and independence from specific system models. While other stability criteria offer valuable insights for LTI systems, Lyapunov's theory provides a more versatile framework, particularly useful in modern control engineering where nonlinear dynamics are prevalent. Its methodology provides a robust framework in modern control engineering where nonlinearities and uncertainties are prevalent.

For linear systems ($\dot{x} = Ax + Bu$), when the state-feedback control law that minimizes the quadratic cost function,

$J = \int_0^\infty (x^T Qx + u^T Ru)dt,\ R = R^T > 0,\ Q = Q^T \geq 0$, is given by $u = -Kx,\ K = R^{-1}B^T P,$ $P = P^T > 0$.

Quadratic functions like $V(x) = x^T Px$, where P is a positive definite matrix, are common choices for satisfying that the origin is globally asymptotically stable.

The Lyapunov's direct method involves solving the Riccati equation $A^T P + PA + Q - PBR^{-1}B^T P = 0$ for a given positive definite matrix Q>0.

For nonlinear systems; the construction is more complex and often involves creative insights. Common approaches include the use of energy functions in mechanical systems or constructing a Lyapunov function based on the system's known properties. Krasovskii's and Zubov's methods are notable techniques for nonlinear systems [Portilla et al., 2021; Szeg, 1963]

The mathematical prerequisites for understanding Lyapunov stability theory are listed as; ***differential equations*** (ability to analyze and solve linear and nonlinear ODEs, as the dynamical systems in control theory are typically represented by such equations), ***linear algebra*** (particularly in the context of eigenvalues, eigenvectors, matrix theory, and the concept of stability for linear systems), ***real analysis*** (especially those related to limits, continuity, and differentiability of functions, are foundational for Lyapunov's theory. The theory often involves examining the behavior of functions and their derivatives in the vicinity of equilibrium points), ***vector calculus*** (including gradient, divergence, and the Jacobian, is important for understanding the behavior of dynamical systems and for constructing

Lyapunov functions in multidimensional spaces), ***system dynamics*** (including the concepts of state, stability, and equilibrium).

Traditional stability analysis primarily deals with linear systems, where superposition and homogeneity principles apply. Nonlinear systems, by contrast, often exhibit behaviors (such as limit cycles, bifurcations, and chaos) that cannot be captured by linear analysis. Linear stability analysis is typically local, and applicable near equilibrium points. Nonlinear systems, however, require methods that can address global stability issues. Lyapunov's theory provides tools for both local and global stability analysis. Nonlinear dynamics can exhibit a rich array of phenomena like multiple equilibria, complex transient behavior, and sensitivity to initial conditions. Lyapunov's methods, especially in the nonlinear context, provide a systematic approach to analyzing such diverse behaviors without solving the system explicitly. One of the central challenges in applying Lyapunov's theory to nonlinear systems is the construction of an appropriate Lyapunov function. This often requires creative insight into the system's properties and is not as straightforward as in linear systems. Nonlinear systems are often subject to perturbations and uncertainties. Lyapunov's theory, particularly through the concept of Lyapunov's indirect method and robustness analysis, addresses these challenges by providing a framework to assess the stability in the presence of such uncertainties [Sastry, 1999; Doyle et al., 2009; Haddad & Chellaboina, 2008].

In feedback control, the Lyapunov stability theory is used to ensure that the closed-loop system remains stable. By selecting a suitable Lyapunov function, engineers can demonstrate that the system's state will converge to a desired equilibrium point or trajectory. This is particularly crucial in nonlinear control systems, where traditional linear stability analysis methods may not apply. For control law design, the goal is often to find a control input that makes the derivative of the Lyapunov function negatively definite. This approach ensures that the system's state converges to the desired state, implying stability. Lyapunov functions are used to prove stability in the presence of model uncertainties and external disturbances. The concept of Lyapunov's indirect method or the use of Krasovskii or LaSalle's invariance principles allows for the design of controllers that are robust against such uncertainties, external disturbances, and model inaccuracies [Helton et al., 1999; Isidori, 1995; Polyakov et al., 2015; Zhao et al., 2014].

Adaptive control deals with systems with unknown or time-varying parameters. Here, Lyapunov stability theory is instrumental in designing adaptive laws that adjust control parameters in real-time. By ensuring the negative definiteness of the derivative of the

Lyapunov function, stability can be maintained even as the system parameters change [Ioannou, 1996].

In state estimation, Lyapunov functions are used to design observers that ensure convergence of the estimated states to the true states. This approach is vital in systems where all state variables are not directly measurable. For designing estimators, particularly in the presence of noise and disturbances, Lyapunov functions provide a framework to ensure that the estimation error converges to zero or a bounded value, thus guaranteeing the reliability of the estimators [Alessandri, 2020; Mohamed et al., 2017].

In large or complex systems, numerical optimization techniques are often used to find Lyapunov functions. This involves solving optimization problems that are subject to Lyapunov inequality constraints [Johansen, 2000].

Sum-of-squares (SOS) programming is a popular technique for constructing Lyapunov functions for polynomial systems. It involves expressing the Lyapunov function as a sum of squares, which can be solved efficiently using semidefinite programming [Ahmadi & Parrilo, 2013; Parrilo & Lall, 2003].

For systems where analytical solutions are challenging, simulation-based methods like Monte Carlo simulations are used to validate Lyapunov functions and assess system stability [Ju et al., 2018].

Recently, machine learning methods, especially neural networks (NNs), have been explored for approximating Lyapunov functions in complex or high-dimensional systems [Grüne, 2021]. Traditional approaches for computing Lyapunov functions, such as linear matrix inequalities (LMIs), are grounded in convex optimization. These methods are highly efficient for linear systems or systems with mild nonlinearities. The computational efficiency here is due to the well-established nature of convex optimization algorithms, which are typically polynomial in time complexity. The use of NNs, particularly deep neural networks (DNNs), introduces a non-convex optimization problem. While they can model complex nonlinear dynamics efficiently, the training process is computationally intensive. The iterative nature of gradient descent and backpropagation, especially with large datasets and deep architectures, can be time-consuming. However, once trained, NNs can be highly efficient in evaluating Lyapunov functions. Traditional methods generally provide high accuracy for systems where the assumptions (like linearity or certain bounds on nonlinearity) hold. The Lyapunov functions obtained are analytically derived, ensuring mathematical rigor and accuracy in those scenarios. However, NNs excel in capturing complex, high-dimensional, nonlinear system

dynamics that traditional methods might not handle accurately. The accuracy of NNs heavily depends on the quality and quantity of the training data, the network architecture, and the training process. There is always a trade-off between the model's complexity and its generalization ability, which can impact accuracy.

Scalability is also a significant challenge with traditional methods, especially for high-dimensional systems. The complexity of solving LMIs grows exponentially with the system's dimensionality, making it impractical for large-scale systems. NNs offer superior scalability. Deep learning models, once trained, can handle high-dimensional inputs more effectively. The ability to learn from data enables these models to adapt to large-scale systems, although the training process itself might be resource-intensive [Huang & Zhang, 2019; Fu et al., 2016; Li et al., 2013].

In traditional methods, Lyapunov functions, $V(x)$, are often explicitly defined and structured. In contrast, when using NNs, $V(x)$ is implicitly defined by the network's architecture and weights, which are learned from data. The interpretability of $V(x)$ in the NN context is less straightforward compared to traditional methods. Traditional methods offer clear criteria for stability analysis, like checking the definiteness of P in a quadratic Lyapunov function. NNs, however, often require additional verification techniques to ensure that the learned Lyapunov function satisfies stability criteria throughout the state space [Abate et al., 2021; Forti et al., 2006; Behera et al., 2004; Lee et al., 2018].

Developing adaptive control systems that can learn and update their control strategies in real-time using NNs involves integrating online learning algorithms with control systems. For instance, using reinforcement learning where NN evolves its policy based on feedback from the system's performance. Such adaptive systems can be pivotal in robotics, autonomous vehicles, and any environment where conditions change unpredictably. Utilizing NNs for the identification and control of highly nonlinear and complex dynamical systems is also an option by training NNs to model the system dynamics from input-output data, bypassing the need for explicit mathematical modeling. This model can then be used for designing control laws. This approach is especially relevant in fields like aerospace engineering, where systems often exhibit complex nonlinear behaviors. The deep learning methods are also implemented for fault diagnosis and predictive maintenance (to detect subtle patterns or anomalies in system data that may indicate a fault or a forthcoming failure), distributed control systems (to optimize communication and control strategies among distributed agents), human-in-the-loop control systems (to model human decision-making processes and incorporating these models

into control systems for more intuitive and efficient human-machine interactions) [Handelman et al., 1989; van Luenen, 1995; Esfandiari et al., 2021].

One of the primary limitations is the difficulty in constructing suitable Lyapunov functions, especially for complex or high-dimensional nonlinear systems. There is no systematic method for constructing these functions in all cases. Lyapunov stability theory often provides local stability results. Extending these results to global stability can be challenging, especially in systems exhibiting rich nonlinear behavior. The accuracy of stability analysis using Lyapunov theory heavily relies on the accuracy of the system model. Inaccuracies or uncertainties in the model can lead to incorrect stability assessments. In large-scale or interconnected systems, the computational complexity of applying Lyapunov theory increases significantly, which can limit its practical applicability. For systems that are analyzed through linearized models, Lyapunov stability theory may fail to capture the complete dynamics, especially near points where linearization is not valid [Vamvoudakis & Jagannathan, 2016; Freeman et al., 2008]. Bifurcation theory deals with qualitative changes in system behavior due to parameter variations. It is critical in understanding phenomena such as the sudden appearance of oscillations (Hopf bifurcations) or changes in system stability. Lyapunov functions are employed to assess the stability of equilibrium points near bifurcation points. They can determine whether a bifurcation leads to a stable or unstable equilibrium. While Lyapunov theory provides local stability information, bifurcation theory extends this by illustrating how stability can change with system parameters. In saddle-node bifurcations, Lyapunov functions help identify stable and unstable manifolds associated with equilibria.

Future research should focus on developing more generalized and computationally efficient methods for constructing Lyapunov functions, especially for high-dimensional and interconnected systems. Additionally, exploring the integration of these theories with emerging areas such as machine learning and data-driven control could provide new insights and tools for control system analysis and design [Strogatz, 2024].

Related Works

The authors in the paper [Nguyen et al., 2020] addressed the complex challenge of ensuring performance and stability in control systems using deep learning-based controllers. It focuses on creating conditions for maintaining nominal stability in such systems, particularly emphasizing non-autonomous deep networks trained with data from a baseline controller, which interestingly doesn't require a mathematical definition and could be a human operator. The authors propose a novel method to determine the necessary number of hidden layers in

the network to ensure stability in the closed-loop system, incorporating this formula into the training process. This approach's effectiveness is demonstrated through a simulation of a continuously stirred tank reactor. The paper makes a significant contribution to deep learning-based control systems by using non-autonomous input-output stable networks (NAIS-Net) and Lyapunov functions to determine a minimum number of NN layers required for stability in both linear and nonlinear systems. This is particularly noteworthy for systems where the baseline controller is not mathematically defined. The study shows that this method successfully maintains closed-loop stability in a continuously stirred tank reactor, a result not guaranteed by the baseline controller. The article concludes with a vision to apply these methods to larger control systems, merging the Lyapunov-based approach with the small-gain theory. This extension is expected to significantly improve the reliability and practicality of deep learning in advanced control systems.

Learning algorithms are highly effective in simulations, enabling robots to adapt to uncertain environments and enhance performance. Yet, their practical application in safety-critical systems is limited due to the lack of safety guarantees in learned policies, which could potentially harm the robot or its surroundings. This paper [Richards et al., 2018] introduces a method for learning accurate safety certificates for nonlinear, closed-loop dynamical systems. The method involves creating an NN-based Lyapunov function and a training algorithm tailored to the largest safe region in the state space, relying only on the dynamics' inputs and outputs, not on a specific model structure. The effectiveness of this approach is showcased through a simulated inverted pendulum. Additionally, the paper explores the integration of this method in safe learning algorithms with statistical models of dynamical systems.

Stability analysis is essential in understanding the behavior of dynamical systems across theoretical and engineering disciplines. Of various stability types, the stability of equilibrium points is particularly crucial and is primarily examined through Lyapunov's stability theory. This theory necessitates the identification of a function with specific properties. However, barring a few elementary instances, there lacks a clear, constructive algorithm to derive a Lyapunov function for a general dynamical system. This paper [Mehrjou & Schölkopf, 2019] aims to address this gap by proposing an accessible yet efficient methodology for approximating the Lyapunov function using deep learning tools.

The paper introduces an innovative dynamic deep learning architecture that integrates Lyapunov control to tackle the challenges of timing latency and constraints inherent in deep learning. This dynamic aspect of the architecture allows for the adjustment of the network

depth in response to the system's complexity or nonlinearity, which is assessed using a parameterized complexity method. The study also examines the impact of parameter tuning on error reduction, leading to decreased time requirements and computational costs in network training and retraining. A control Lyapunov function is employed as an input cost function for the DNN to assess system stability. The architecture is designed to initiate a relearning process in response to disturbances or unforeseen model dynamics, thus obviating the need for an observer-based approach. This relearning mechanism broadens the algorithm's applicability to various cyber-physical systems (CPS). The paper evaluates the intelligent controller's autonomy in different scenarios, including high-frequency nonlinear references, reference changes, or the introduction of disturbances. The dynamic deep learning algorithm demonstrates its effectiveness in adapting to these changes and autonomously stabilizing the system safely [Mahmoud & Zohdy, 2022].

The paper [Dawson et al., 2021] addresses the challenge of designing controllers that ensure safety and stability in robotic systems, especially for nonlinear and uncertain models. The authors develop a model-based learning approach to synthesize robust feedback controllers that are both safe and stable. Drawing on robust convex optimization and Lyapunov theory, they introduce robust control Lyapunov barrier functions that maintain effectiveness despite model uncertainties. The paper demonstrates the approach through simulations in various scenarios such as car trajectory tracking, nonlinear control with obstacle avoidance, satellite rendezvous with safety constraints, and flight control considering a learned ground effect model. The simulations reveal that this method not only matches but may surpass the capabilities of robust Model Predictive Control (MPC), while significantly reducing computational costs.

[Chen et al.] present a learning-based approach for the Lyapunov stability analysis of piecewise affine dynamical systems interfacing with piecewise affine NN controllers. The method involves an iterative process between a learner and a verifier. In each iteration, the learner proposes a Lyapunov function candidate, derived from closed-loop system samples, as a solution to a convex program. The verifier, employing a mixed-integer program, either validates this candidate or rejects it with a counterexample indicating where the stability condition fails. These counterexamples are then used to refine the learner's sample set and consequently improve the Lyapunov function candidates. The learner and verifier are designed based on the analytic center cutting-plane method, with the verifier acting as a cutting-plane oracle. The authors demonstrate that when the set of Lyapunov functions is fully dimensional in parameter space, their procedure successfully identifies a Lyapunov

function within a finite number of iterations. The utility of this method is shown in the search for quadratic and piecewise quadratic Lyapunov functions.

The paper [Manek & Kolter, 2020] addresses the challenge of making formal claims about the properties of dynamical systems learned through deep networks. These networks are typically used for predicting the evolution of a system's state over time. The paper proposes a novel method for learning dynamical systems that are guaranteed to be stable across the entire state space. This is achieved by concurrently learning a dynamics model and a Lyapunov function, ensuring the non-expansiveness of the dynamics under the learned Lyapunov function. The authors demonstrate that this learning system can model simple dynamical systems and can be integrated with deep generative models to learn complex dynamics, such as video textures, in an end-to-end manner. This contribution was presented at the NeurIPS 2019 conference and offers a significant advancement in the field of machine learning and dynamical systems.

Dai et al. [Dai et al., 2021] tackle the challenge of the lack of theoretical performance guarantees, such as Lyapunov stability, in neural-network controllers developed through deep reinforcement learning, which is a stark contrast to traditional model-based controller designs. To bridge this gap, the authors propose a method to synthesize a Lyapunov-stable NN controller. This is accompanied by an NN Lyapunov function to certify the controller's stability. The paper presents a mixed-integer linear program (MIP) for verifying the Lyapunov condition. The MIP verifier can either certify the condition or generate counterexamples to refine the controller and Lyapunov function. The paper also introduces an optimization program to compute an inner approximation of the region of attraction for the closed-loop system. The approach is applied to various robots, including an inverted pendulum and 2D and 3D quadrotors, demonstrating superior performance over a baseline LQR controller. This work was published at the Robotics: Science and Systems conference in July 2021 and offers significant advancements in the integration of deep learning with traditional control theory in robotics.

The paper introduces a novel DNN architecture named Lyapunov-Net, designed to approximate Lyapunov functions for high-dimensional dynamical systems. A key feature of Lyapunov-Net is its inherent guarantee of positive definiteness, which facilitates easy training to meet the negative orbital derivative condition. This requirement simplifies the empirical risk function to just a single term in practice, thereby significantly reducing the number of hyper-parameters compared to existing approaches. The paper not only presents the architecture but also provides theoretical support for the approximation capabilities and

complexity bounds of Lyapunov-Net. The effectiveness of this method is demonstrated through its application to nonlinear dynamical systems with state spaces of up to 30 dimensions. The results highlight that the proposed approach markedly surpasses current state-of-the-art methods in efficiency [Gaby et al., 2022].

In "Neural Lyapunov Control," Chang, Roohi, and Gao [Chang et al., 2020] propose innovative methods for learning control policies and NN-based Lyapunov functions for nonlinear control problems, offering a provable guarantee of stability. The methodology comprises a learner tasked with identifying control and Lyapunov functions, and a falsifier that generates counterexamples to efficiently direct the learner towards viable solutions. The process concludes successfully when the falsifier can no longer find counterexamples, thereby establishing the stability of the controlled nonlinear system. This approach simplifies Lyapunov control design and ensures end-to-end correctness. It is also capable of attaining much larger regions of attraction compared to traditional methods like LQR and SOS/SDP. The paper includes experiments demonstrating how the new methods achieve high-quality solutions for complex control challenges.

The paper [Zaki et al., 2020] introduces a novel feed-forward NN deep learning controller specifically designed for nonlinear systems. This controller merges the attributes of a multilayer feed-forward NN with those of a restricted Boltzmann machine (RBM). The RBM plays a critical role in this deep learning controller by initializing a multilayer feed-forward NN through unsupervised pretraining, where all the weights start from zero. The weight laws for the proposed network are formulated using the Lyapunov stability method.

A comparison is drawn between the proposed feed-forward neural network deep learning controller (FFNNDLC), a standard FFNN controller (FFNNC), and other controller types, noting that all these controllers initially set their weights to zero. The FFNNDLC demonstrates superior performance in responding to system uncertainties and external disturbances, as evidenced in the simulation results. To validate the controller's practical applicability, it is implemented on an ARDUINO DUE kit microcontroller to control an electromechanical system. The proposed FFNNDLC not only outpaces other FFNNCs, where parameters are learned using the backpropagation method but also adeptly handles variations in both the disturbance and the system parameters.

In "Neural Lyapunov Redesign," Mehrjou, Ghavamzadeh, and Schölkopf [Mehrjou et al., 2020] explore the integration of Lyapunov functions in learning controllers that not only focus on performance metrics but also ensure safety, a critical aspect often required in practical

applications. The paper discusses the importance of stability as a safety measure and the effectiveness of Lyapunov functions in assessing stability in nonlinear dynamical systems. The authors present a novel two-player collaborative algorithm that alternates between estimating a Lyapunov function and developing a controller. This iterative process aims to progressively expand the stability region of the closed-loop system, thus achieving control policies with larger safe regions. Theoretical insights into the class of systems suitable for this algorithm are provided, and the paper also includes empirical evaluations of the method's effectiveness using a representative dynamical system.

The paper [Chang & Gao, 2021] addresses a crucial limitation in the practical application of learning-based methods in robotics control problems, particularly the absence of a stability guarantee. To overcome this challenge, the authors develop innovative methods for learning neural control policies and neural Lyapunov critic functions within the model-free reinforcement learning (RL) framework. The approach utilizes sample-based methods and the Almost Lyapunov function conditions to estimate the region of attraction and invariance properties via the learned Lyapunov critic functions. These methods are designed to enhance the stability of neural controllers across different nonlinear systems, including automobile and quadrotor control, thereby bridging a significant gap in the field of robotics and control systems.

The structure of this paper is organized as follows: Section 2 introduces the theoretical background for the Lyapunov Stability Theory. Section 3 is comprehensive, covering multiple aspects: it starts by exploring the concept of compositional Lyapunov functions and their relation to small-gain theory. It then provides a concise overview of NNs, primarily focusing on clarifying the terminology used. Furthermore, in this section, we delve into the DNN architecture and its ability to approximate compositional Lyapunov functions, thus overcoming the curse of dimensionality. Additionally, Section 3 outlines the loss functions for a training algorithm aimed at computing Lyapunov functions using the NNs discussed. Section 4 is dedicated to showcasing numerical examples that demonstrate the effectiveness of our method. Finally, Section 5 discusses various aspects and potential extensions of our methodology, culminating in the conclusion of the paper.

2. Theoretical Background

Control systems, particularly linear time-invariant (LTI) systems, are central to engineering and physics. The stability of such systems is a pivotal concern in control theory. The core of this stability analysis often lies in the spectral properties of the system matrix, specifically its eigenvalues and eigenvectors.

An LTI system can be represented in state-space form as:

$$\dot{x} = Ax(t) + Bu(t) \tag{1}$$

where $x(t)$ is the state vector, A is the system matrix, B is the input matrix, and $u(t)$ represents the input vector. The eigenvalues and eigenvectors of the matrix A play a crucial role in determining the system's behavior over time.

The eigenvalues of the matrix A, denoted as λ_i (where i ranges over the number of states), directly influence the system's stability.

The key principles are:

1- A system is stable if and only if all the eigenvalues of its system matrix have negative real parts. Mathematically, for all i, $Re(\lambda_i) < 0$.
2- If any eigenvalue has a real part equal to zero (and none with positive real parts), the system is marginally stable.
3- If any eigenvalue has a positive real part, the system is unstable.

Eigenvectors associated with the eigenvalues of A define the modes of the system. For a given eigenvalue λ_i, its corresponding eigenvector v_i dictates the direction in the state space along which the system state evolves. The overall system response is a superposition of these modes. Understanding the eigenstructure of the system matrix allows engineers to predict and modify the system's behavior. For example, in controller design, the placement of poles (eigenvalues) is a common technique to achieve desired stability and performance characteristics [Kailath, 2016; Nise, 2011; Ogata, 2016].

The focus is on nonlinear ordinary differential equations (ODEs) and the computation of Lyapunov functions, specifically targeting the challenge of the curse of dimensionality in higher-dimensional nonlinear systems.

A system of nonlinear ordinary differential equations is defined as follows:

$$\dot{x}(t) = f(x(t)) \tag{2}$$

where $f: \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitz continuous, and $x = 0$ is an asymptotically stable equilibrium. The domain of attraction is defined within a compact set $K_n \subset \mathbb{R}^n$.

A continuously differentiable function V: $O \to \mathbb{R}$ (with O being an open set containing the origin) is a Lyapunov function if it satisfies:

$V(0) = 0$

$V(x) > 0$ for all $x \neq 0$

The orbital derivative $\nabla V(x).f(x) \leq -h(x)$ for a function h: $\mathbb{R}^n \to \mathbb{R}$ with $h(x) > 0$ for all $x \in O \backslash \{0\}$ (3)

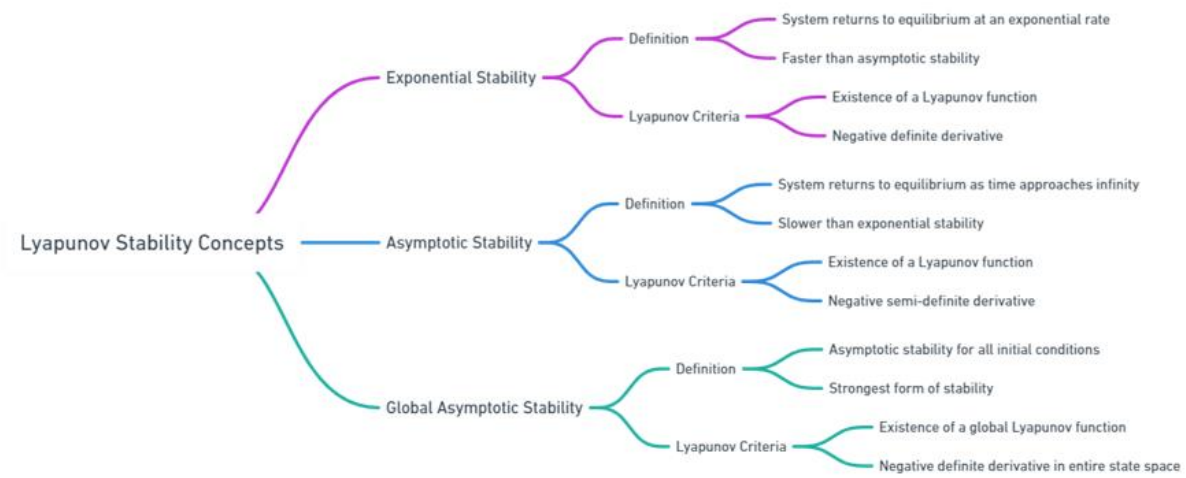
If $O = \mathbb{R}^n$ and $V(x) \to \infty$ as $\|x\| \to \infty$, then V is a global Lyapunov function.

It's applicable even without a complete model of the system, making it versatile for systems where the exact dynamics are unknown or complex. The method is inherently robust to uncertainties in the system model. Depending on the choice of Lyapunov function, it can provide insights into global or local stability. Finding an appropriate Lyapunov function can be challenging and often requires insight into the system's behavior. Lyapunov's indirect method cannot capture the richness of nonlinear dynamics and behaviors (e.g., limit cycles, chaos) that might be present in the system. The direct method is more versatile in its applicability, especially in situations where the system model is unknown or highly nonlinear. In contrast, the indirect method is limited to systems that can be accurately linearized. The direct method can potentially provide global stability information, whereas the indirect method is inherently local. The indirect method is typically easier to apply due to its reliance on linear system theory, which is well-understood and has a wide array of analytical tools. The direct method, while more versatile, often presents significant challenges in identifying an appropriate Lyapunov function. The direct method is inherently more robust to model uncertainties, making it preferable in real-world applications where such uncertainties are inevitable. The direct method is capable of providing insights into nonlinear dynamics, which are often overlooked by the indirect method [Zhang et al., 2019; Zhou et al., 2014; Nakakuki et al., 2008; Yang & Minimis, 1991].

A system is asymptotically stable if it is Lyapunov stable and if, additionally, $lim_{t\to\infty}x(t) = 0$ for every initial condition in a neighborhood of the equilibrium. In Lyapunov terms, this implies that $\dot{V}(x)$ must be strictly negative outside the equilibrium.

In a stronger form of stability, a system is exponentially stable if it returns to equilibrium at a rate proportional to the initial displacement. Mathematically, this requires $V(x)$ to satisfy certain bounds that ensure a convergence rate, like $V(x) \leq Ke^{-\lambda t}$ for constants K and λ. Global stability refers to stability for all initial conditions, not just those in a neighborhood of the equilibrium. A Lyapunov function that is valid globally (i.e., for all $x \epsilon \mathbb{R}^n$) can establish global stability [Zhang et al., 1999; Yan & Baotong, 2006; Ghrissi et al., 2017; Liu, 2006; Carnevale et al., 2007; Vernigora, 2006]. The above-mentioned terms are visualized in Fig.1a and Fig.1b, respectively.

a)



b)

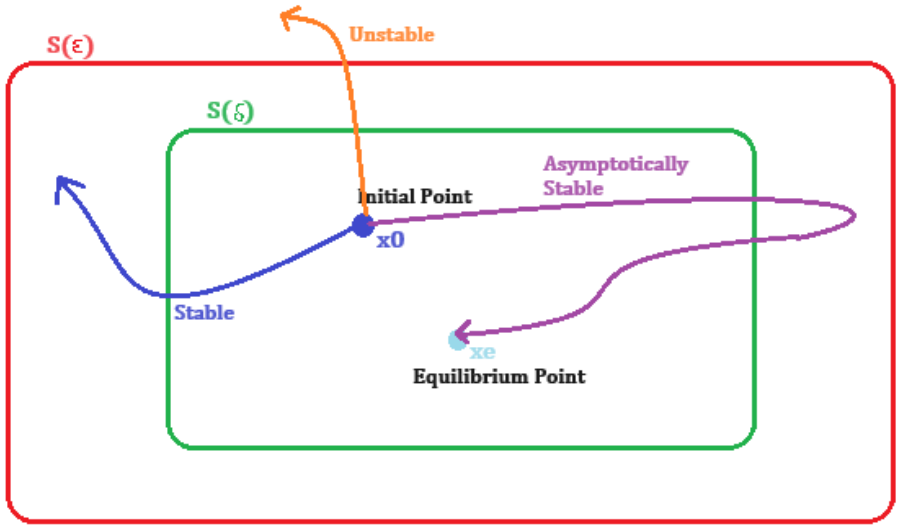


Figure 1: a) A diagram that gives the Lyapunov stability concepts. b) Geometric representation of stability concepts from the perspective of Lyapunov theorem.

The figure demonstrates: **a) Stability:** The system's state returns to the equilibrium point x_e, after a small disturbance $S(\delta)$, but it doesn't necessarily get closer to the equilibrium as time progresses; **b) Asymptotic stability:** Not only does the system's state return to the equilibrium after a small disturbance $S(\delta)$, but it also gets closer to x_e as time progresses until it converges to the equilibrium; **c) Instability:** The system's state diverges from the equilibrium point x_e, after a small disturbance.

Understanding the criteria for selecting and constructing Lyapunov functions is crucial for effectively applying Lyapunov stability theory. Constructing a Lyapunov function often involves insight into the system's energy-like properties. Common approaches include:

-For mechanical systems, potential and kinetic energy can guide the construction.

-In linear systems, quadratic forms are common.

-Sum of squares (SOS) techniques are used in polynomial nonlinear systems, where $V(x)$ is expressed as a sum of squared terms [Pota & Moylan, 1992; Fu & Abed, 1990; Haddad & Bernstein, 1994].

In nonlinear systems, the construction of $V(x)$ is more intricate and system-specific. It involves capturing the nonlinearities of the system and ensuring the negative definiteness of $\dot{V}(x)$.

The main challenge lies in finding a suitable $V(x)$ that accurately reflects the system's dynamics and requires the positive definite $V(x)$ and negative definite $\dot{V}(x)$ over the entire state space [Prajna et al., 2004; Sontag, 1989; Briat, 2012; Grimm et al., 2005; Ito, 1994; Teel et al., 2013].

3. Problem Statement and Methodology

This paper addresses the limitations of conventional methods in computing Lyapunov functions for high-dimensional systems. Our focus is on devising a DNN-based methodology that overcomes the exponential growth of computational requirements with increased state space dimensions [Hafstein, 2007].

The primary goal is to design a DNN capable of efficiently computing an approximation of a Lyapunov function on K_n. Efficiently here refers to moderate growth in computational and storage efforts with increasing space dimension. The paper proposes to achieve this for Lyapunov functions that satisfy a specific structure termed "compositional Lyapunov functions".

This structure is motivated by recent results on the approximation properties of NNs and corresponds to a specific kind of small-gain condition in systems theory. The system is divided into subsystems, and the state vector and vector field are split accordingly. This division allows for a more manageable representation and computation of the dynamics.

The method of using DNNs to approximate Lyapunov functions is innovative, especially in addressing the curse of dimensionality. This approach has significant implications for control theory and the stability analysis of high-dimensional nonlinear systems. A notable aspect is the focus on complexity analysis, a critical factor when dealing with high-dimensional systems. The paper's attempt to balance computational efficiency with accuracy is commendable. While the approach is promising, its applicability to a wide range of systems remains to be thoroughly explored. The assumption of the existence of a Lyapunov function with a specific structure may not hold for all systems, potentially limiting the method's generalizability.

3.1. Compositional Lyapunov Functions and Small-Gain Theory

First, we focus on the concept of compositional Lyapunov functions and their relation to small-gain theory in the context of control systems [Zhong & Marcels, 1997]. This section delineates the conditions under which such functions exist and their relevance in stabilizing large-scale nonlinear systems.

The overall Lyapunov function $V(x)$ for a system is termed compositional if it can be expressed as a sum of C^1 −functions $V_i(z_i)$ of the form:

$$V(x) = \sum_{i=1}^{s} \alpha_i V_i(z_i) \tag{4}$$

where z_i are state vectors of subsystems, $\alpha_i > 0$ are weighting factors.

The system is divided into s subsystems, each with its state and dynamics, and the state vector x and vector field f are accordingly split. The overall stability is deduced from the interconnection of these subsystems and their respective Lyapunov functions [Lin et al., 1996; Balestrino et al., 2012; Breiten & Höveler, 2023].

Stability is inferred through conditions like:

$$\sum_{i=1}^{s} \nabla V_i(z_i) \,.\, f_i(z_i, u_i) < 0 \tag{5}$$

The concept of subsystem decomposition in the formulation of compositional Lyapunov functions is a pivotal aspect in the analysis and control of large-scale or interconnected systems. This approach is grounded in the principle that a complex system can be decomposed into smaller, more manageable subsystems. The main challenge lies in ensuring the stability of the overall system based on the stability of its components. This decomposition is guided by the system's inherent structure, such as physical, functional, or control interconnections. Analyzing the effect of interconnections between subsystems is critical. This involves ensuring that the interconnection terms do not violate the overall stability condition. The coupling effects must be accounted for in such a way that the derivative of the global Lyapunov function, remains negative definite. In large-scale systems, the scalability of the Lyapunov approach is crucial. This often involves ensuring that the compositional method remains valid even as the system scales. Additionally, robustness against uncertainties and perturbations in subsystems and interconnections must be considered [Yao, 2007; Jönsson et al., 2007; Pai & Vittal, 1983].

Small-gain theory is used to analyze the stability of interconnected or large-scale systems [Nwokah, 1986]. It is particularly relevant when subsystems are coupled but the coupling

does not destabilize the overall system. The concept of input-to-state stability (ISS) is crucial, where the stability of each subsystem is influenced by the states of other subsystems. This is quantified using ISS-Lyapunov functions and ISS gains. The small-gain condition is formulated using ISS-Lyapunov functions and involves certain bounded positive definite functions and gain mappings. It ensures that the gains are sufficiently small to maintain system stability. Under the small-gain condition, a Lyapunov function V can be constructed from ISS-Lyapunov functions of the subsystems, providing a tool for stability analysis. This approach leverages the compositional structure of Lyapunov functions and the capabilities of DNNs in function approximation [Hornik et al., 1989].

The small-gain approach provides a framework for assessing the stability of a system based on the interplay between the gains of its components. When combined with ISS and ISS-Lyapunov functions, this theory offers a robust methodology for handling nonlinearities and external disturbances. The small-gain theorem asserts that a feedback interconnection of two stable systems remains stable if the product of their gains is less than one. Mathematically, if we have two systems with transfer functions $G_1(s)$ and $G_2(s)$, and if $\|G_1 G_2\| < 1$ in a certain norm (typically H_∞), then the feedback interconnection is stable. The overall stability is assessed by examining the gains and their interconnections, ensuring that the composite gain around any feedback loop in the network satisfies the small-gain condition. Input-to-state stability (ISS) extends the concept of stability to systems with external inputs. A system is ISS if every bounded input leads to a bounded state. ISS provides a framework to analyze how external disturbances affect the system's state. In applying small-gain theory to ISS systems, the gains are considered in the context of the input-output behavior influenced by disturbances. ISS-Lyapunov functions are used to establish the ISS property of each subsystem. The small-gain theorem is then applied to ensure that the interconnections do not lead to instability, taking into account the ISS properties of individual subsystems [Laila & Nesic, 2022; Ito, 2006; Dashkovskiy et al., 2012; Mironchenko & Ito, 2014; Jiang et al., 1996; Stein et al., 2015; Rueffer, 2010]. One of the key advantages of the small-gain approach is its contribution to system robustness. By ensuring a margin within the gain product condition, the system can tolerate certain levels of parameter variations and uncertainties without becoming unstable [Daskovskiy & Kosymkov, 2013; Ito & Zhong-Ping Jiang, 2009].

3.2. Deep Neural Network-Based Approach

We provide an overview of DNNs, particularly their capacity to approximate $C1$ −funtions. Our analysis reveals that under specific structural assumptions, an NN can efficiently approximate compositional Lyapunov functions, thereby evading the curse of dimensionality.

After decomposing the complex system into interconnected subsystems, each described by its state dynamics, a NN is designed for each subsystem to approximate its local Lyapunov function. These networks take the subsystem's state as input and output a scalar value representing the Lyapunov function's value at that state. Training data is generated by simulating subsystems. This data should include various state trajectories, emphasizing states near the stability boundaries. Each NN is trained using the generated data. The objective is to minimize a loss function that ensures the NN output meets the criteria for a Lyapunov function (positive definiteness and negative definiteness of its time derivative). Then, a global Lyapunov function for the entire system is constructed by approximately combining the outputs of the NNs corresponding to each subsystem. This could involve weighted sums or other mathematical operations that ensure compliance with Lyapunov's stability conditions. The composed global Lyapunov function is used to analyze the stability of the overall system. The NNs' outputs should collectively indicate system stability under various operating conditions. This method allows leveraging empirical data, which can be especially beneficial in systems where theoretical modeling is challenging [Gaby et al., 2022; Li et al., 2017; Chen et al., 2014].

The drawbacks are listed as:

Data Dependency: The accuracy of the approximation heavily depends on the quality and representativeness of the training data.

Overfitting: There's a risk of NNs overfitting the training data, leading to poor generalization to unseen states.

Verification Challenges: Verifying that the NN truly approximates a valid Lyapunov function across the entire state space can be challenging.

Computational Demands: Training NNs for complex systems can be computationally intensive.

The integration of bridging theory and data-driven method signifies a move towards combining theoretical control principles with data-driven computational techniques, opening

new avenues in control theory research. It could lead to a deeper understanding of stability in complex, high-dimensional systems through empirical data analysis. In practical terms, this approach can facilitate the design of controllers for systems that are traditionally difficult to model and analyze using conventional control theory methods. NNs can adapt to changes in system dynamics, potentially leading to more robust and adaptable control systems [Fu et al., 2013; Grüne, 2021].

Here, we propose a DNN architecture tailored to approximate compositional Lyapunov functions. We demonstrate that this architecture requires a polynomically growing number of neurons relative to the system dimension, marking a significant advancement over traditional methods.

This network is a feedforward network with multiple hidden layers. The input vector $x = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ is processed through these layers to compute the scalar output $W(x;\theta) \in \mathbb{R}$, where $\theta \in \mathbb{R}^p$ represents the tunable parameters of the network.

The goal is to approximate a Lyapunov function V(x) using the NN output $W(x;\theta)$. The Lyapunov function's properties are expressed as a partial differential equation (PDE), specifically a Zubov-type equation.

$$\nabla W(x;\theta) f(x) = -\|x\|^2 \tag{6}$$

where $\nabla W(x;\theta)$ is the derivative of W w.r.t x, and $f(x)$ is the system's dynamics.

The boundary conditions for the PDE are reformulated for numerical feasibility.

$$\alpha_1\|x\| \leq \nabla W(x;\theta) \leq \alpha_2\|x\| \qquad \forall x \in K_n \tag{7}$$

where $\alpha_1, \alpha_2 \in K$ are appropriately chosen functions. This reformulation helps in finding a solution to the PDE that satisfies these conditions.

The training of the network involves minimizing a loss function for training NNs, based on partial differential inequalities L defined as

$$L(w,p,x) = ([pf(x) + \|x\|^2]^+)^2 + \delta([w - \alpha_1(\|x\|)]^-)^2 + \delta([w - \alpha_2(\|x\|)]^+)^2 \tag{8}$$

where $[a]^- = min\{a, 0\}$, $[a]^+ = max\{a, 0\}$, and $\delta > 0$ is a weighting parameter. The minimization of this loss function w.r.t θ is expected to yield a Lyapunov function.

This methodology elaborates on the algorithmic aspect of our approach, detailing the process of computing Lyapunov functions using the proposed network. The number of neurons

needed for a fixed approximation accuracy grows exponentially with the dimension of the system.

4. Numerical Results

The elaboration of our introduced approach is demonstrated through two examples. The first is a lower-dimensional case, highlighting the effectiveness of the loss function mentioned in Equation 8. The second example, of a higher dimension, illustrates the method's capability to identify Lyapunov functions in more complex scenarios. The computational work was carried out using the following setup: Python 3.7 backend, Colab Pro with Tensorflow 2.1.0 environment, V100 GPU High-Ram, equipped with 51 GB of memory.

For those interested, the Python and Tensorflow-Keras scripts, along with the trained DNNs, are accessible at the following URL: DeepDynaSim Repository

(https://github.com/DeepDynaSim/DeepLyapunovFunction).

To establish the effectiveness of the proposed *"Deep Lyapunov"* technique, we conducted a series of numerical experiments. These evaluations are performed in spaces of up to four dimensions, thereby demonstrating the practicality and robustness of our method. Our process begins by specifying a system and a candidate for the Lyapunov function, ensuring compliance with the Lyapunov stability criteria. Following this, we proceed to configure a DNN.

In the first example; Let's consider a simple nonlinear system:

$$\dot{x}_1 = -x_1^3$$

$$\dot{x}_2 = -x_2 \tag{9}$$

A suitable Lyapunov function candidate for this system could be:

$$V(x) = \frac{1}{2}x_1^4 + \frac{1}{2}x_2^2 \tag{10}$$

The derivative of V along the trajectories of the system is:

$$\dot{V}(x) = \frac{\partial V}{\partial x_1}\dot{x}_1 + \frac{\partial V}{\partial x_2}\dot{x}_2 = 2x_1^3(-x_1^3) + x_2(-x_2) = -2x_1^6 - x_2^2 \leq 0 \tag{11}$$

Since $\dot{V}(x)$ is negative definite, $V(x)$ is a valid Lyapunov function, confirming the system's stability.

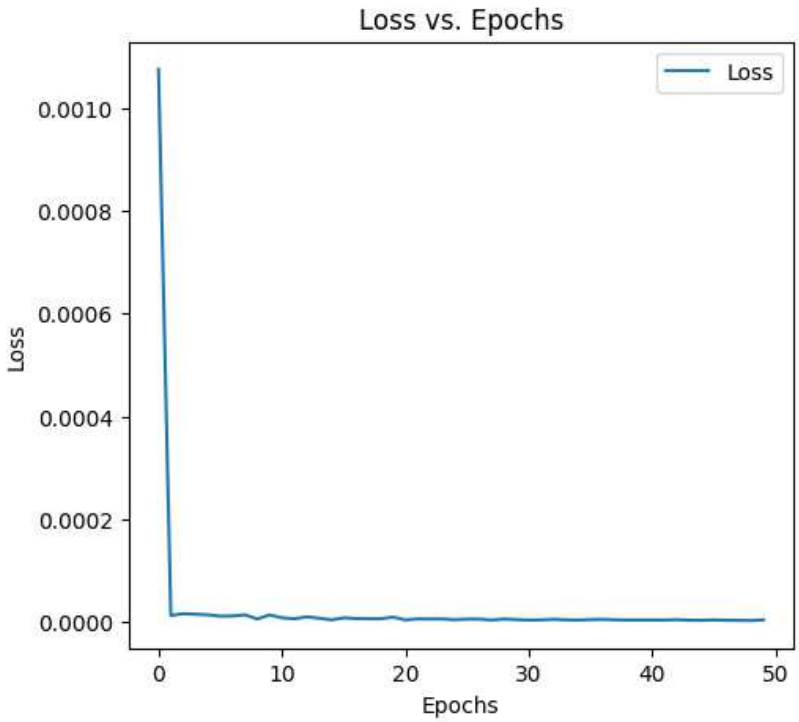
In our approach, we construct a DNN configured with various parameters. Specifically, we deploy a network comprising three layers, each consisting of 64 neurons, and employ the ReLU (Rectified Linear Unit) as the activation function. We also fine-tuned several hyperparameters, including setting the learning rate to 0.001 and opting for a batch size of 64. The selection of the network's architecture and the activation functions is tailored to align with the specific demands of our problem. For training, we generate synthetic data points. Ideally, in practical applications, this data ought to comprehensively represent the state space of the system in question. Our training process utilizes the Adam optimizer, a popular choice for its efficiency in handling large datasets and complex NN architectures. The model is trained to minimize the "mean squared error", a common loss function in regression problems, reflecting the difference between the predicted and actual values.

This training strategy is applied within the context of the mentioned Python code, which includes TensorFlow and NumPy libraries. The code outlines the system dynamics and a candidate Lyapunov function. It also details the generation of training data, construction of the NN model, compilation of the model with specified optimizer and loss function, and the training procedure. Optionally, the model can be evaluated on a separate test dataset to assess its performance.

The DNN is trained with 100000 test points and converged after 6 epochs, and the computed Lyapunov function along with its derivative is graphically represented, confirming the method's success.

Fig.2a depicts the progression of loss functions concerning the number of epochs, while Fig. 2b provides a graphical representation of the error metrics corresponding to each epoch. These visualizations are instrumental in evaluating the performance and convergence behavior of the proposed model for Example 1.

a)



b)

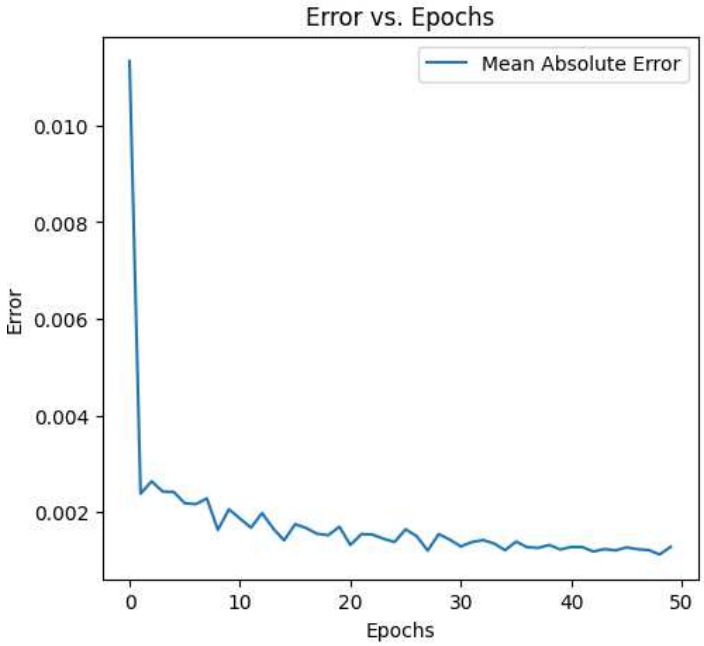
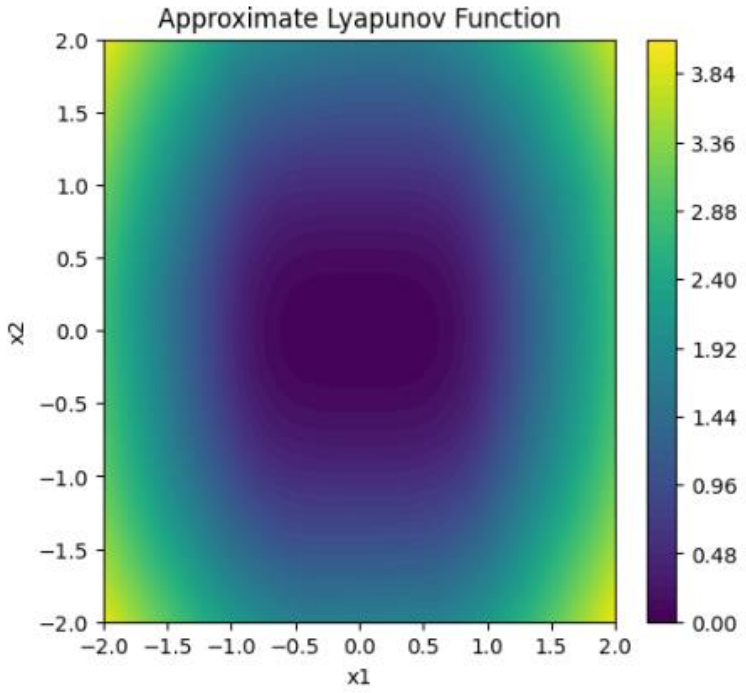


Figure 2: Convergence and performance behavior of the proposed model for Example 1 a) Loss vs. Epochs. b) Error vs. Epochs.

A practical implementation for approximating and visualizing Lyapunov functions using DNN is also done. The primary objective is to demonstrate how a trained NN can be utilized to approximate a Lyapunov function for a given dynamical system and to compute its orbital derivative, which is crucial for analyzing the system's stability. This representation is essential for understanding how the state of the system evolves. The NN model, assumed to be pre-built and trained, represents the approximate Lyapunov function. This model is a crucial component, as it embodies the deep learning approach to approximating complex functions, which might be analytically intractable. The architecture of the NN, including the number of layers and neurons, activation functions, and training methodology, is designed to capture the underlying characteristics of the Lyapunov function for the given system. The orbital_derivative function computes the derivative of the Lyapunov function along the system's trajectories. This is achieved using TensorFlow's automatic differentiation capabilities, which are essential for evaluating the change of the Lyapunov function along the system's dynamics. The orbital derivative is a key factor in assessing the stability of the system, as it indicates whether the Lyapunov function is decreasing along the system's trajectories.

Fig.3 is responsible for visualizing both the approximate Lyapunov function and its orbital derivative. This is done by creating a 2D grid of the state space and evaluating the NN model and the orbital derivative at each point in this grid. The use of contour plots provides an intuitive visual representation of how the Lyapunov function and its derivative behave across different states of the system. This visualization is instrumental in conveying complex mathematical concepts in a more accessible manner.

a)



b)

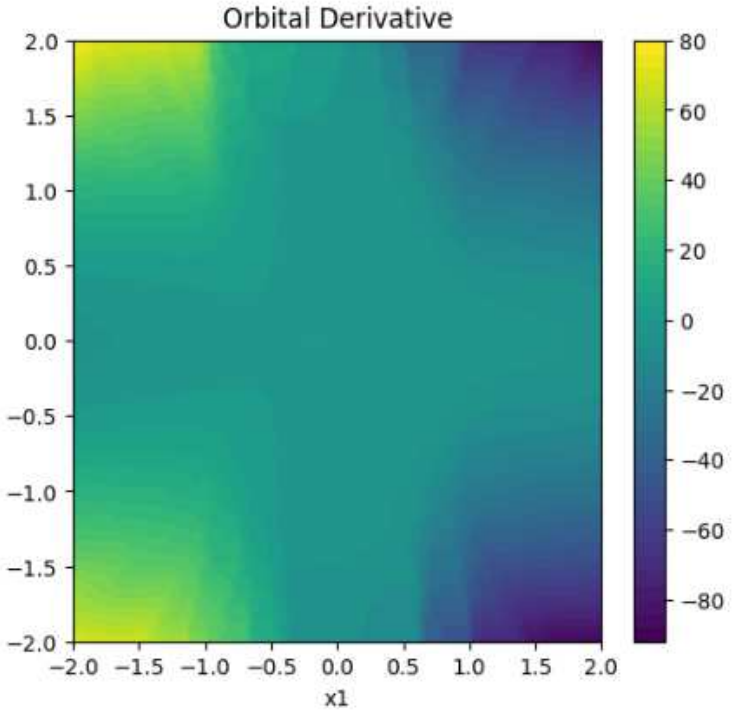


Figure 3: a) Approximate Lyapunov function $W(x, \theta^*)$ b) Its orbital derivative $\nabla W(x, \theta^*) f(mesh)$ for Example 1.

In the second example; a more complex system involving a 4-dimensional vector field $f(x)$ with linear and nonlinear components is proposed.

$$\dot{x}_1 = -x_1^2$$

$$\dot{x}_2 = -x_2$$

$$\dot{x}_3 = -x_3^2$$

$$\dot{x}_4 = -x_4 \quad (12)$$

A suitable Lyapunov function candidate for this system could be:

$$V(x) = x_1^4 + x_2^2 + x_3^6 + x_4^2 \quad (13)$$

The derivative of V along the trajectories of the system is:

$$\dot{V}(x) = \frac{\partial V}{\partial x_1}\dot{x}_1 + \frac{\partial V}{\partial x_2}\dot{x}_2 + \frac{\partial V}{\partial x_3}\dot{x}_3 + \frac{\partial V}{\partial x_4}\dot{x}_4 = 4x_1^3(-x_1^2) + 2x_2(-x_2) + 6x_3^5(-x_3^2) + 2x_4(-x_4) \leq 0 \quad (14)$$

Since $\dot{V}(x)$ is negative definite, $V(x)$ is a valid Lyapunov function, confirming the system's stability.

In this scenario, we are designing a DNN tailored for a specific control engineering application. First, we define the system dynamics using a Python function. This function models the behavior of a four-dimensional system, with each dimension's dynamics specified as a function of the system's current state. Next, we define a candidate Lyapunov function. This function is crucial in control theory, particularly for stability analysis. The chosen Lyapunov function here is a polynomial function of the system's state, intended to help in verifying the stability of the system's equilibrium point. For training the NN, we generate a large dataset (200,000 samples) of random points within the state space. This dataset is created to capture a wide range of possible system states. Each data point consists of a state vector (with four elements) and a corresponding value of the Lyapunov function.

The NN model we construct is more complex than usual, consisting of four layers with 100 neurons each. This complexity is likely due to the higher dimensionality and complexity of the system being modeled. The sigmoid activation function is used in each layer, with a final

layer employing a linear activation function to output a scalar value, which is the predicted value of the Lyapunov function.

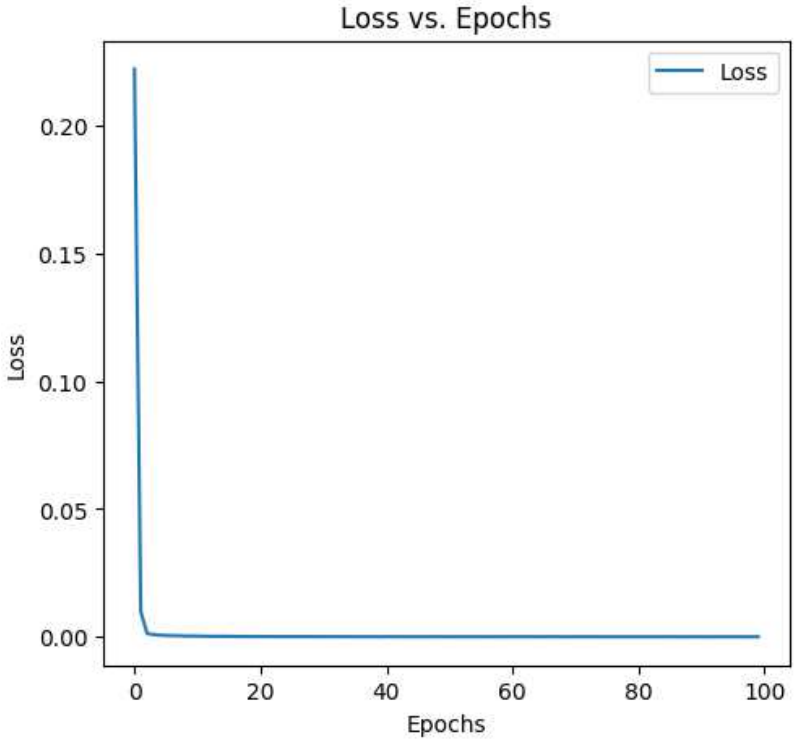
The model is compiled using the Adam optimizer, a popular choice for training DNNs due to its efficiency in converging to a solution. The learning rate is set to 0.01, and the loss function is the mean squared error, which measures the difference between the predicted and actual values of the Lyapunov function. Training is conducted over 100 epochs with a batch size of 50. This longer training period and smaller batch size could be indicative of the model's need to learn complex patterns in the data.

Overall, this method represents a sophisticated approach to modeling and analyzing a complex dynamical system using NNs, with an emphasis on stability analysis through Lyapunov functions.

The NN is trained with 200000 test points and converged after epochs, and the computed Lyapunov function along with its derivative is graphically represented, confirming the method's success.

Fig.4a depicts the progression of loss functions concerning the number of epochs, while Fig. 4b provides a graphical representation of the error metrics corresponding to each epoch. These visualizations are instrumental in evaluating the performance and convergence behavior of the proposed model for Example 2.

a)



b)

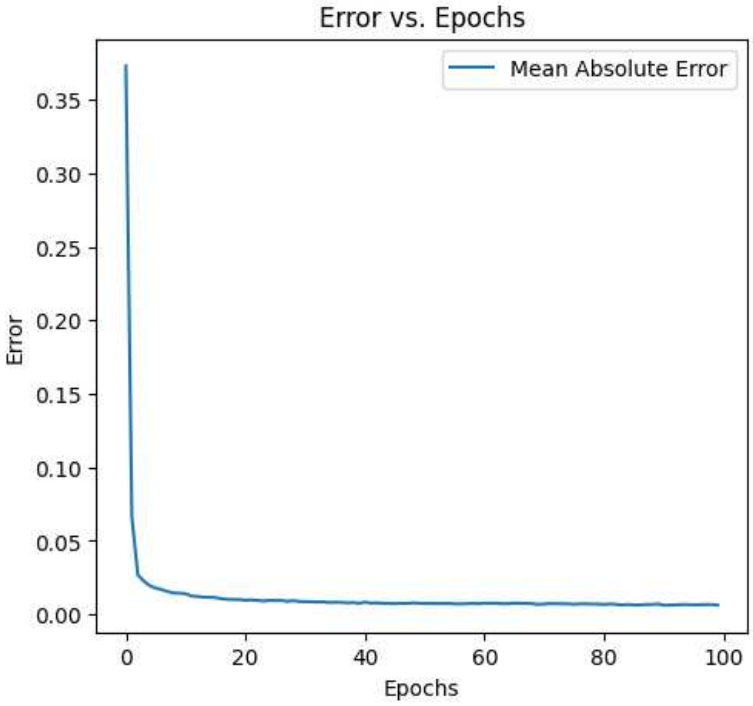
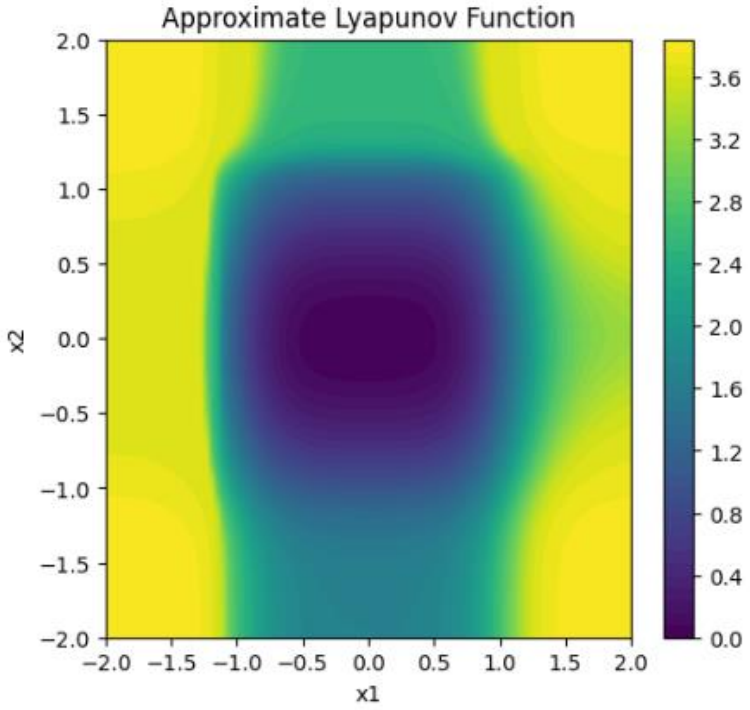


Figure 4: Convergence and performance behavior of the proposed model for Example 2 a) Loss vs. Epochs. b) Error vs. Epochs.

We present a computational approach to approximate and visualize Lyapunov functions for high-dimensional dynamical systems using DNN. This method is particularly useful for systems where traditional analytical methods for Lyapunov function computation are infeasible due to complexity or nonlinearity. We illustrate the implementation of this approach for a hypothetical 4-dimensional system, as described in Example 2 in Fig.5.

a)



b)

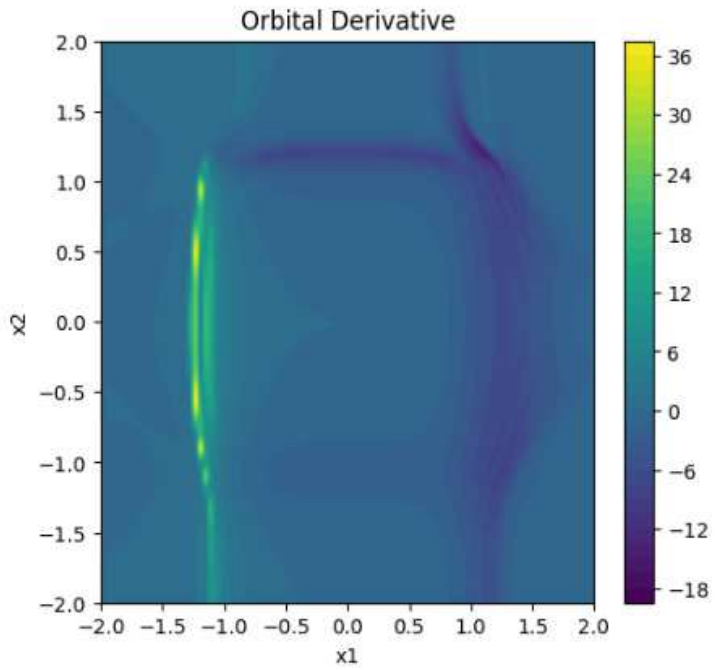


Figure 5: a) Approximate Lyapunov function $W(x, \theta^*)$ b) Its orbital derivative $\nabla W(x, \theta^*) f(mesh)$ for Example 2.

We use TensorFlow's automatic differentiation to compute the gradient of the Lyapunov function concerning the state variables and then calculate the orbital derivative. We create a 2D grid for two state variables (x1 and x2) and compute the approximate Lyapunov function and its orbital derivative at each point in this grid. We then use contour plots to visualize these functions. For visualization, we fix the values of x3 and x4. These values can be changed or additional plots for different fixed values can be added to explore the behavior in different regions of the state space. This method provides a visual representation of the approximate Lyapunov function and its orbital derivative, offering insights into the stability properties of the system as learned by the NN. The training successfully computed a Lyapunov function, as evidenced by the following graphical representations.

A comparative table that outlines the key differences between the two DNNs as described in Example 1 and Example 2, is given in Table 1.

Table 1: A comparative table for the proposed DNN models.

Feature	Example 1	Example 2
Number of Layers	3	5
Neurons per Layer	64 in each layer	100 in each layer
Activation Functions	ReLU for the first 2 layers, Linear for output	Sigmoid for first 4 layers, Linear for output
Input Dimensions	2	4
Output Layer Activation	Linear	Linear
Learning Rate	0.001	0.01
Batch Size	64	50
Epochs	50 (converged in 6 epochs)	100 (converged in epochs)
Loss Function	Mean Squared Error	Mean Squared Error
Optimizer	Adam	Adam
Training Samples	100000	200000
System Dynamics Function	**system_dynamics** (2 variables)	**system_dynamics** (4 variables)
Lyapunov Function	**lyapunov_function** (2 variables)	**lyapunov_function** (4 variables)

This table provides a clear comparison of the two DNN models, highlighting their structural differences and training configurations.

In the APPENDIX section, the implementation of the approximated Lyapunov function and its orbital derivative for use in Simulink for control system applications is explained step-by-step.

5. Discussion and Conclusive Summary

In this paper, we have presented a thorough methodology for computing Lyapunov functions using DNNs, effectively addressing the long-standing issue of the curse of dimensionality in control and optimization problems. Our approach demonstrates that, unlike traditional methods, the computational effort here grows only polynomially with the state dimension, marking a substantial advancement in the field.

Crucially, we underscore the fact that the functions computed through our method are approximations of Lyapunov functions, rather than exact representations. This distinction is pivotal in control theory, where Lyapunov functions are the cornerstone for establishing system stability. While these approximations bear resemblance to true Lyapunov functions, their validation in higher-dimensional spaces poses a significant challenge and opens avenues for future research.

Our method diverges from conventional grid-based techniques by eschewing the necessity of small-gain theorems and precise subsystem knowledge, showcasing remarkable flexibility and adaptability. We also propose the potential expansion of this methodology to control Lyapunov functions, which are fundamental in developing stabilizing feedback laws for nonlinear systems. This extension, while promising, is acknowledged as a complex and intricate future research direction.

Another key aspect of our approach is the use of ReLU activation functions in NNs, aligning with current trends in deep learning applications. This choice, however, introduces the need for nonsmooth analysis or integral representations, highlighting a critical area for further investigation to balance the advantages of ReLU functions with the intricacies of nonsmooth analysis.

We recognize that the outcomes of our algorithm are not unique, pointing to the necessity of establishing criteria to identify particularly beneficial Lyapunov functions, an idea resonating with other approximation methodologies.

The paper culminates by emphasizing the potential of deep neural networks for approximating Lyapunov functions with a compositional structure, especially under small-gain conditions.

To summarize, our contributions are manifold:

Scalability: Our method adeptly handles the curse of dimensionality, making it scalable for systems of varying dimensionality.

Flexibility: The NN-based approach demonstrates a remarkable capacity to accommodate both low and high-dimensional systems.

Efficiency: The application of deep learning techniques enables efficient computation of Lyapunov functions, a critical advantage for complex systems.

However, we also face challenges:

Approximation Accuracy: The computed functions are approximations, and their fidelity as true Lyapunov functions across all scenarios is not absolute.

Verification in High Dimensions: The task of confirming Lyapunov properties in higher dimensions remains complex and challenging.

For future research, we propose:

Extension to Control Systems: Adapting this methodology to the applications of the feedback control theory.

Nonsmooth Activation Functions: Investigating the implications of using other nonsmooth activation functions is important, though it introduces additional complexity.

In conclusion, this paper contributes significantly to control theory by offering a scalable and efficient method for the stability analysis of complex systems, as demonstrated through the numerical examples in Section 4.

However, it is imperative to continue research efforts to refine these approximations and extend the methodology's applicability in control engineering.

APPENDIX

Simulink Implementation

To extract the approximated Lyapunov function and its orbital derivative for use in Simulink for control system applications, you need to follow a multi-step process. This involves exporting the trained NN model from TensorFlow, integrating it into Simulink, and then using it within your control system design.

Here's a step-by-step guide:

1. Export the Trained Neural Network Model

First, you need to export the trained TensorFlow model to a format that can be used in Simulink. One common approach is to convert the model to the ONNX (Open Neural Network Exchange) format, which is a widely supported format for NN models.

```
“import tf2onnx

import tensorflow as tf

# Assuming 'model' is your trained TensorFlow model

model_proto, _ = tf2onnx.convert.from_keras(model)

with open("lyapunov_model.onnx", "wb") as f:

f.write(model_proto.SerializeToString())”
```

2. Import the Model into Simulink

Simulink supports the integration of ONNX models through the Deep Learning Toolbox. You can use the Import ONNX Network feature in Simulink to import the model. Once imported, you can use the model within the Simulink environment.

3. Create a Custom Block for the Orbital Derivative

Since the orbital derivative involves both the NN model and the system dynamics, you might need to create a custom Simulink block. This block would take the state of the system as input, compute the orbital derivative using both the NN model and the system dynamics, and output the result. You can create a MATLAB function block in Simulink that implements the orbital derivative calculation. This block would be called the imported NN model and include the system dynamics equations.

4. Integrate the Model and Custom Block into Your Control System

With the Lyapunov function model and the custom orbital derivative block in Simulink, you can now integrate them into your control system design. These components can be used for stability analysis or as part of a control strategy that leverages Lyapunov-based control methods.

5. Testing and Validation

Before deploying the control system, it's crucial to test and validate the integrated model within Simulink. Ensure that the NN model and the orbital derivative calculation are functioning as expected in the context of your control system.

6. Deployment

Once validated, the Simulink model, including the NN and the orbital derivative block, can be used for real-time control applications or further simulations.

Considerations

Compatibility: Ensure that the versions of TensorFlow, tf2onnx, and Simulink are compatible.

Real-Time Performance: If the model is to be used in real-time applications, consider the computational complexity and execution time of the NN model.

Validation: Rigorous testing is essential to ensure that the NN model behaves as expected within the Simulink environment and in the context of the control system.

By following these steps, you can effectively utilize the power of machine learning within traditional control system environments like Simulink, opening up new possibilities for advanced control strategies.

References

Abate, A., Ahmed, D., Giacobbe, M., & Peruffo, A. (2021). Formal synthesis of Lyapunov neural networks. *IEEE Control Systems Letters*, *5*(3), 773–778. https://doi.org/10.1109/lcsys.2020.3005328

Ahmadi, A. A., & Parrilo, P. A. (2013). Stability of polynomial differential equations: Complexity and converse lyapunov questions. arXiv preprint arXiv:1308.6833.

Alessandri, A. (2020). Lyapunov functions for state observers of dynamic systems using Hamilton–Jacobi inequalities. *Mathematics*, *8*(2), 202. https://doi.org/10.3390/math8020202

Balestrino, A., Caiti, A., & Grammatico, S. (2012). A new class of Lyapunov functions for the constrained stabilization of linear systems. *Automatica*, *48*(11), 2951–2955. https://doi.org/10.1016/j.automatica.2012.06.099

Behera, L., Kumar, S., & Patnaik, A. (2004). A novel learning algorithm for Feedforward Networks using Lyapunov function approach. *International Conference on Intelligent Sensing and Information Processing, 2004. Proceedings Of.* https://doi.org/10.1109/icisip.2004.1287667

Breiten, T., & Höveler, B. (2023). On the approximability of Koopman-based operator Lyapunov equations. *SIAM Journal on Control and Optimization*, *61*(5), 3131–3155. https://doi.org/10.1137/23m1552693

Briat, C. (2012). Robust stability and stabilization of uncertain linear positive systems via integral linear constraints: $l1$-gain and $l\infty$-gain characterization. *International Journal of Robust and Nonlinear Control*, *23*(17), 1932–1954. https://doi.org/10.1002/rnc.2859

Carnevale, D., Teel, A. R., & Nesic, D. (2007). A Lyapunov proof of an improved maximum allowable transfer interval for networked control systems. *IEEE Transactions on Automatic Control*, *52*(5), 892–897. https://doi.org/10.1109/tac.2007.895913

Chang, Y.-C., & Gao, S. (2021). Stabilizing neural control using self-learned almost Lyapunov critics. *2021 IEEE International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1109/icra48506.2021.9560886

Chang, Y.-C., Roohi, N., & Gao, S. (2020). Neural Lyapunov Control. NeurIPS 2019. arXiv:2005.00611v4 [cs.LG]. https://doi.org/10.48550/arXiv.2005.00611

Chen, C. L., Wen, G.-X., Liu, Y.-J., & Wang, F.-Y. (2014). Adaptive Consensus Control for a class of nonlinear multiagent time-delay systems using neural networks. IEEE Transactions on Neural Networks and Learning Systems, 25(6), 1217–1226. https://doi.org/10.1109/tnnls.2014.2302477

Chen, S., Fazlyab, M., Morari, M., Pappas, G. J., & Preciado, V. M. (2020). Learning Lyapunov Functions for Piecewise Affine Systems with Neural Network Controllers. arXiv:2008.06546v2 [math.OC]. https://doi.org/10.48550/arXiv.2008.06546

Dai, H., Landry, B., Yang, L., Pavone, M., & Tedrake, R. (2021). Lyapunov-stable neural-network control. Robotics: Science and Systems (RSS), July 2021. arXiv:2109.14152v1 [cs.RO]. https://doi.org/10.48550/arXiv.2109.14152

Dashkovskiy, S. N., Ruffer, B. S., & Wirth, F. R. (2012). Small gain theorems for large scale systems and construction of ISS Lyapunov functions. *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. https://doi.org/10.1109/cdc.2012.6426856

Dashkovskiy, S.; Kosmykov, M. (2013). Reduction of the small gain condition for large-scale interconnections. International Journal of Robust and Nonlinear Control, 25(6), 842–864. https://doi.org/10.1002/rnc.3111

Dawson, C., Qin, Z., Gao, S., & Fan, C. (2021). Safe Nonlinear Control Using Robust Neural Lyapunov-Barrier Functions. arXiv:2109.06697v2 [eess.SY]. Accepted to the 5th Annual Conference on Robot Learning (CoRL 21). https://doi.org/10.48550/arXiv.2109.06697

Doyle, J. C., Francis, B. A., & Tannenbaum, A. R. (2009). *Feedback control theory*. Dover.

Esfandiari, K., Abdollahi, F., & Talebi, H. A. (2021). NN-based adaptive control of affine nonlinear systems. Neural Network-Based Adaptive Control of Uncertain Nonlinear Systems, 17–58. https://doi.org/10.1007/978-3-030-73136-6_3

Forti, M., Grazzini, M., Nistri, P., & Pancioni, L. (2006). Generalized Lyapunov approach for convergence of neural networks with discontinuous or non-Lipschitz Activations. *Physica D: Nonlinear Phenomena*, *214*(1), 88–99. https://doi.org/10.1016/j.physd.2005.12.006

Freeman, Randy A., & Kokotović, P. (2008). Robust control Lyapunov functions. Robust Nonlinear Control Design, *33–63*. https://doi.org/10.1007/978-0-8176-4759-9_3

Fu, J.-H., & Abed, E. H. (1990). Families of Lyapunov functions for nonlinear systems in critical cases. *29th IEEE Conference on Decision and Control*. https://doi.org/10.1109/cdc.1990.203818

Fu, Z.-J., Xie, W.-F., & Luo, W.-D. (2013). Robust on-line nonlinear systems identification using multilayer dynamic neural networks with two-time scales. *Neurocomputing*, *113*, 16–26. https://doi.org/10.1016/j.neucom.2012.11.041

Fu, Z.-J., Xie, W.-F., & Na, J. (2016). Robust adaptive nonlinear observer design via multi-time Scales Neural Network. *Neurocomputing*, *190*, 217–225. https://doi.org/10.1016/j.neucom.2016.01.015

Gaby, N., Zhang, F., & Ye, X. (2022). Lyapunov-Net: A deep neural network architecture for Lyapunov function approximation. *2022 IEEE 61st Conference on Decision and Control (CDC)*. https://doi.org/10.1109/cdc51059.2022.9993006

Ghrissi, T., Hammami, M. A., Hammi, M., & Mabrouk, M. (2017). New criterion for asymptotic stability of time-varying dynamical systems. *Kybernetika*, *331–353*. https://doi.org/10.14736/kyb-2017-2-0331

Grimm, G., Messina, M. J., Tuna, S. E., & Teel, A. R. (2005). Model predictive control: For want of a local control Lyapunov function, all is not lost. *IEEE Transactions on Automatic Control*, *50*(5), 546–558. https://doi.org/10.1109/tac.2005.847055

Grüne, L. (2021). Computing Lyapunov functions using Deep Neural Networks. Journal of Computational Dynamics, 8(2), 131. https://doi.org/10.3934/jcd.2021006

Grüne, L. (2021). Overcoming the curse of dimensionality for approximating Lyapunov functions with deep neural networks under a small-gain condition. *IFAC-PapersOnLine*, *54*(9), 317–322. https://doi.org/10.1016/j.ifacol.2021.06.152

Haddad, W. M., & Bernstein, D. S. (1994). Explicit construction of quadratic Lyapunov functions for the small gain, positivity, Circle, and Popov theorems and their application to robust stability. part II: Discrete-time theory. *International Journal of Robust and Nonlinear Control*, *4*(2), 249–265. https://doi.org/10.1002/rnc.4590040203

Haddad, W. M., & Chellaboina, V. (2008). *Nonlinear Dynamical Systems and control: A Lyapunov-based approach.* Princeton University Press.

Hafstein, S. F. (2007). Algorithm for constructing Lyapunov functions. *Electronic Journal of Differential Equations*, *1*(Mon. 01-09), 08. https://doi.org/10.58997/ejde.mon.08

Handelman, D. A., Lane, S. H., & Gelfand, J. J. (1989). Integrating Neural Networks and knowledge-based systems for robotic control. Proceedings, 1989 International Conference on Robotics and Automation. https://doi.org/10.1109/robot.1989.100184

Helton, J. W., & James, M. R. (1999). *Extending H∞ Control to Nonlinear Systems: Control of Nonlinear Systems to Achieve Performance Objectives.* https://doi.org/10.1137/1.9780898719840

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Huang, C., & Zhang, H. (2019). Periodicity of non-autonomous inertial neural networks involving proportional delays and non-reduced order method. *International Journal of Biomathematics*, *12*(02), 1950016. https://doi.org/10.1142/s1793524519500165

Ioannou, P. A. (1996). *Robust Adaptive Control.* Prentice-Hall, 1996.

Isidori, A. (1995). *Nonlinear Control Systems.* Springer.

Ito, H. (1994). Geometry and analysis in Dynamical Systems. Geometry and Analysis in Dynamical Systems. https://doi.org/10.1142/9789814534130

Ito, H. (2006). State-dependent scaling problems and stability of interconnected IISS and ISS Systems. *IEEE Transactions on Automatic Control*, *51*(10), 1626–1643. https://doi.org/10.1109/tac.2006.882930

Ito, H., & Zhong-Ping Jiang. (2009). Necessary and sufficient small gain conditions for integral input-to-state stable systems: A Lyapunov perspective. IEEE Transactions on Automatic Control, *54*(10), 2389–2404. https://doi.org/10.1109/tac.2009.2028980

Jiang, Z.-P., Mareels, I. M. Y., & Wang, Y. (1996). A Lyapunov formulation of the nonlinear small-gain theorem for interconnected ISS Systems. *Automatica*, *32*(8), 1211–1215. https://doi.org/10.1016/0005-1098(96)00051-9

Johansen, T. A. (2000). Computation of Lyapunov functions for smooth nonlinear systems using convex optimization. Automatica, *36*(11), 1617–1626. https://doi.org/10.1016/s0005-1098(00)00088-1

Jönsson, U., Kao, C.-Y., & Fujioka, H. (2007). Low dimensional stability criteria for large-scale interconnected systems. *2007 European Control Conference (ECC)*. https://doi.org/10.23919/ecc.2007.7068616

Ju, P., Li, H., Gan, C., Liu, Y., Yu, Y., & Liu, Y. (2018). Analytical assessment for transient stability under stochastic continuous disturbances. IEEE Transactions on Power Systems, *33*(2), 2004–2014. https://doi.org/10.1109/tpwrs.2017.2720687

Kailath, T. (2016). Linear Systems. Pearson India Education Services.

Khalil, H. K. (2015). *Nonlinear control*. Pearson, 2015.

Laila, D. S., & Nesic, D. (2002). Lyapunov based small-gain theorem for parameterized discrete-time interconnected ISS Systems. *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* https://doi.org/10.1109/cdc.2002.1184874

Lee, T. H., Trinh, H. M., & Park, J. H. (2018). Stability Analysis of neural networks with time-varying delay by constructing novel Lyapunov functionals. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(9), 4238–4247. https://doi.org/10.1109/tnnls.2017.2760979

Li, H., Bai, L., Wang, L., Zhou, Q., & Wang, H. (2017). Adaptive neural control of uncertain nonstrict-feedback stochastic nonlinear systems with output constraint and unknown dead zone. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(8), 2048–2059. https://doi.org/10.1109/tsmc.2016.2605706

Li, T., Wang, T., Song, A., & Fei, S. (2013). Combined convex technique on delay-dependent stability for delayed neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(9), 1459–1466. https://doi.org/10.1109/tnnls.2013.2256796

Lin, Y., Sontag, E. D., & Wang, Y. (1996). A smooth converse Lyapunov theorem for robust stability. *SIAM Journal on Control and Optimization*, *34*(1), 124–160. https://doi.org/10.1137/s0363012993259981

Liu, M. (2006). Stability Analysis of neutral-type nonlinear delayed systems: An LMI approach. *Journal of Zhejiang University-SCIENCE A*, *7*(S2), 237–244. https://doi.org/10.1631/jzus.2006.as0237

Lyapunov, A. M., & Walker, J. A. (1994). The general problem of the stability of motion. *Journal of Applied Mechanics*, *61*(1), 226–227. https://doi.org/10.1115/1.2901415

Mahmoud, A., & Zohdy, M. (2022). Dynamic Lyapunov machine learning control of nonlinear magnetic levitation system. *Energies*, *15*(5), 1866. https://doi.org/10.3390/en15051866

Manek, G., & Kolter, J. Z. (2020). Learning Stable Deep Dynamics Models. NeurIPS 2019. arXiv:2001.06116v1 [cs.LG]. https://doi.org/10.48550/arXiv.2001.06116

Mehrjou, A., & Schölkopf, B. (2019). Deep Lyapunov Function: Automatic Stability Analysis for Dynamical Systems. arXiv:1901.08403 [math.DS]. https://doi.org/10.48550/arXiv.1901.08403

Mehrjou, A., Ghavamzadeh, M., & Schölkopf, B. (2020). Neural Lyapunov Redesign. arXiv:2006.03947v2 [eess.SY]. https://doi.org/10.48550/arXiv.2006.03947

Mironchenko, A., & Ito, H. (2014). Integral input-to-state stability of bilinear infinite-dimensional systems. *53rd IEEE Conference on Decision and Control*. https://doi.org/10.1109/cdc.2014.7039876

Mohamed, M., Yan, X.-G., Spurgeon, S. K., & Mao, Z. (2017). Variable structure observers for Nonlinear Interconnected Systems. *Advances in Variable Structure Systems and Sliding Mode Control—Theory and Applications*, 195–221. https://doi.org/10.1007/978-3-319-62896-7_8

Nakakuki, T., Shen, T., & Tamura, K. (2008). Adaptive control design for a class of nonsmooth nonlinear systems with matched and linearly parameterized uncertainty. *International Journal of Robust and Nonlinear Control*, *19*(2), 243–255. https://doi.org/10.1002/rnc.1317

Nguyen, H. H., Matschek, J., Zieger, T., Savchenko, A., Noroozi, N., Findeisen, R. (2020). Towards nominal stability certification of Deep Learning-based controllers. 2020 American Control Conference (ACC). https://doi.org/10.23919/acc45564.2020.9147564

Nise, N. S. (2011). Control Systems Engineering. Wiley.

Nwokah, O. D. (1986). On decentralized control of complex feedback systems. *1986 American Control Conference*. https://doi.org/10.23919/acc.1986.4788958

Ogata, K. (2016). Modern Control Engineering. Pearson

Pai, M. A., & Vittal, V. (1983). Multimachine stability analysis using vector Lyapunov functions with inertial-centre decomposition. *International Journal of Electrical Power & Energy Systems*, *5*(3), 139–144. https://doi.org/10.1016/0142-0615(83)90001-7

Parrilo, P. A., & Lall, S. (2003). Semidefinite programming relaxations and algebraic optimization in control. European Journal of Control, 9(2–3), 307–321. https://doi.org/10.3166/ejc.9.307-32

Polyakov, A., Efimov, D., Perruquetti, W., & Richard, J.-P. (2015). Implicit Lyapunov-Krasovski functionals for Stability Analysis and control design of time-delay systems. *IEEE Transactions on Automatic Control*, *60*(12), 3344–3349. https://doi.org/10.1109/tac.2015.2422451

Pota, H. R., & Moylan, P. J. (1992). A new Lyapunov function for interconnected Power Systems. *IEEE Transactions on Automatic Control*, *37*(8), 1192–1196. https://doi.org/10.1109/9.151102

Prajna, S., Parrilo, P. A., & Rantzer, A. (2004). Nonlinear control synthesis by convex optimization. *IEEE Transactions on Automatic Control*, *49*(2), 310–314. https://doi.org/10.1109/tac.2003.823000

Richards, S. M., Berkenkamp, F., & 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 (CoRL 2018). arXiv:1808.00924v2 [cs.SY]. https://doi.org/10.48550/arXiv.1808.00924

Rueffer, B. S. (2010). Small-gain conditions and the comparison principle. IEEE Transactions on Automatic Control, *55*(7), 1732–1736. https://doi.org/10.1109/tac.2010.2048053

Sastry, S. (1999). *Nonlinear Systems Analysis, stability, and Control*. Springer.

Slotine, J.-J. E., & Li, W. (2005). *Applied Nonlinear Control*. Prentice Education Taiwan Ltd.

Portilla, G., Alexandrova, I. V., & Mondie, S. (2021). Estimates for weighted homogeneous delay systems: A lyapunov-krasovskii-razumikhin approach. *2021 American Control Conference (ACC)*. https://doi.org/10.23919/acc50511.2021.9483011

Sontag, E. D. (1989). Smooth stabilization implies coprime factorization. *IEEE Transactions on Automatic Control*, *34*(4), 435–443. https://doi.org/10.1109/9.28018

Stein Shiromoto, H., Andrieu, V., & Prieur, C. (2015). A region-dependent gain condition for asymptotic stability. Automatica, *52*, 309–316. https://doi.org/10.1016/j.automatica.2014.12.017

Strogatz, S. H. (2024). *Nonlinear Dynamics and Chaos: With applications to physics, biology, chemistry, and engineering*. CHAPMAN & HALL CRC.

Szeg, P. (1963). New methods for constructing Liapunov functions for time-invariant control systems. *IFAC Proceedings Volumes*, *1*(2), 584–589. https://doi.org/10.1016/s1474-6670(17)69693-9

Teel, A. R., Forni, F., & Zaccarian, L. (2013). Lyapunov-based sufficient conditions for exponential stability in hybrid systems. IEEE Transactions on Automatic Control, *58*(6), 1591–1596. https://doi.org/10.1109/tac.2012.2228039

Vamvoudakis, K. G., & Jagannathan, S. (2016). Control of complex systems: Theory and applications. Elsevier.

van Luenen, W. T. (1995). Learning controllers using neural networks. Lecture Notes in Computer Science, 205–233. https://doi.org/10.1007/bfb0027031

Vernigora, I. V. (2006). Stability of solutions of dynamic systems with randomly structured aftereffect. *Cybernetics and Systems Analysis*, *42*(2), 188–194. https://doi.org/10.1007/s10559-006-0053-z

Yan, J., & Baotong, C. (2006). Novel stability analysis of high-order Cohen-Grossberg neural networks with time-varying delays. *2007 Chinese Control Conference*. https://doi.org/10.1109/chicc.2006.4347370

Yang, X., & Miminis, G. (1991). Stability Analysis for Discrete Stochastic Nonlinear Systems. *1991 American Control Conference*. https://doi.org/10.23919/acc.1991.4791886

Yao, J. (2007). Decentralized stabilization of interconnected neutral delay large-scale systems. *International Mathematical Forum*, *2*, 1989–1995. https://doi.org/10.12988/imf.2007.07180

Zaki, A. M., El-Nagar, A. M., El-Bardini, M., & Soliman, F. A. (2020). Deep Learning Controller for nonlinear system based on Lyapunov Stability Criterion. *Neural Computing and Applications*, *33*(5), 1515–1531. https://doi.org/10.1007/s00521-020-05077-1

Zhang, S., He, X., Chen, Q., & Zhu, Z. (2019). Partially saturated coupling-based control for underactuated overhead cranes with experimental verification. *Mechatronics*, *63*, 102284. https://doi.org/10.1016/j.mechatronics.2019.102284

Zhang, T., Ge, S. S., & Hang, C. C. (1999). Stable Adaptive Control for a class of nonlinear systems using a modified Lyapunov function. *IFAC Proceedings Volumes*, *32*(2), 5537–5542. https://doi.org/10.1016/s1474-6670(17)56943-8

Zhao, X., Deng, F., & Zhong, X. (2014). LaSalle-type theorems for general nonlinear stochastic functional differential equations by multiple Lyapunov functions. *Abstract and Applied Analysis*, *2014*, 1–11. https://doi.org/10.1155/2014/486191

Zhong-Ping Jiang, & Marcels, I. M. Y. (1997). A small-gain control method for nonlinear cascaded systems with dynamic uncertainties. *IEEE Transactions on Automatic Control*, *42*(3), 292–308. https://doi.org/10.1109/9.557574

Zhou, J., Zhang, Q., Li, J., Men, B., & Ren, J. (2014). Dissipative control for a class of Nonlinear Descriptor Systems. *International Journal of Systems Science*, *47*(5), 1110–1120. https://doi.org/10.1080/00207721.2014.913082