IMPLICIT AND DEEP LEARNING-BASED CONTROL METHODS FOR UNCERTAIN NONLINEAR SYSTEMS

By

OMKAR SUDHIR PATIL

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2023

© 2023 Omkar Sudhir Patil

To my mother Kalpalata Patil, father Sudhir Patil, and brother Sarthak Patil

ACKNOWLEDGMENTS

I would like to express my heartfelt gratitude to my advisor, Dr. Warren E. Dixon, for his unwavering guidance and support throughout my doctoral journey. His mentorship and encouragement have significantly boosted my confidence and motivated me to work ‘more faster’. I also extend my sincere thanks to my committee members Dr. Amor Menezes, Dr. Yu Wang, and Dr. Zoleikha Biron for their invaluable insights and contributions to my work. I am grateful to Dr. Shubhendu Bhasin at IIT Delhi for kindling my interest in the field of nonlinear and adaptive control. I would also like to acknowledge the support and assistance from my colleagues in the Nonlinear Control and Robotics Lab, my peers at the University of Florida, and my research collaborators, who have made this experience enriching and fulfilling. Additionally, I would like to thank the anonymous reviewers for their valuable feedback that has helped me to improve my publications. Lastly, I extend my thanks to my family and friends who have supported me through this challenging but rewarding journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

a.a.t.	Almost All Time
a.e.	Almost Everywhere
AWG	Additive White Gaussian
DNN	Deep Neural Network
EL	Euler-Lagrange
NN	Neural Network
LReLU	Leaky Rectified Linear Unit
ReLU	Rectified Linear Unit
RISE	Robust Integral of the Sign of the Error
ResNet	Deep Residual Neural Network
RMS	Root Mean Square
UUB	Uniformly Ultimately Bounded

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

IMPLICIT AND DEEP LEARNING-BASED CONTROL METHODS FOR UNCERTAIN NONLINEAR SYSTEMS

By

Omkar Sudhir Patil

May 2023

Chair: Warren E. Dixon
Major: Mechanical Engineering

Many practical engineering systems exhibit nonlinear behaviors that make it difficult to predict the output response for a given input. One approach to compensate for nonlinear effects is to feed forward a model of the nonlinearities in the control design as a means to cancel destabilizing effects: feedback linearization is a simplified example of such an approach. However, the system model typically has uncertainties such as parametric uncertainty or more general uncertainty that is often modeled as a bounded exogeneous disturbance. Adaptive and learning-based control strategies are motivated by the desire to include an approximate feedforward model in the controller or to implicitly learn the model through the feedback structure. This dissertation is focused on the development of such adaptive and learning-based controllers. Since the dissertation focuses on general nonlinear systems, a constructive Lyapunov-based design and analysis approach is used.

In Chapter 2, a deep neural network (DNN)-based adaptive controller is developed to compensate for uncertainty in a nonlinear dynamic system. Although Lyapunov-based real-time update laws are well-known for neural network (NN)-based adaptive controllers with a single-hidden-layer, developing real-time weight update laws for DNNs remains an open question. This dissertation presents the first result with Lyapunov-based real-time weight adaptation laws for each layer of a feedforward DNN-based control architecture, with stability guarantees. Additionally, the developed method allows

nonsmooth activation functions to be used in the DNN to facilitate improved transient performance. A nonsmooth Lyapunov-based stability analysis proves global asymptotic tracking error convergence. Simulation results are provided for a nonlinear system using DNNs with leaky rectified linear unit (LReLU) and hyperbolic tangent activation functions to demonstrate the efficacy and performance of the developed method. Although Chapter 2 provides weight adaptation laws for DNNs, the development is restricted for fully-connected DNNs, whereas deriving weight adaptation laws has been an open problem for deep residual neural networks (ResNets).

Chapter 3 provides the first result on Lyapunov-derived weight adaptation for a ResNet-based adaptive controller. A nonsmooth Lyapunov-based analysis is provided to guarantee global asymptotic tracking error convergence. Comparative Monte Carlo simulations are provided to demonstrate the performance of the developed ResNet-based adaptive controller. The ResNet-based adaptive controller shows a 49.52% and 54.38% improvement in the tracking and function approximation performance, respectively, in comparison to a fully-connected DNN-based adaptive controller.

Chapter 4 addresses the problem of adaptive control of systems with uncertain time-varying parameters. A continuous adaptive controller is developed for nonlinear dynamical systems with linearly parameterizable uncertainty involving time-varying uncertain parameters. Through a unique stability analysis strategy, a new adaptive feedforward term is developed, along with specialized feedback terms, to yield asymptotic tracking error convergence by compensating for the time-varying nature of the uncertain parameters. A Lyapunov-based stability analysis is shown for Euler-Lagrange systems, which ensures asymptotic tracking error convergence and boundedness of the closed-loop signals. Additionally, the time-varying uncertain function approximation error is shown to converge to zero. A simulation example of a two-link manipulator is provided to demonstrate the asymptotic tracking result.

Chapter 5 provides new stability results for a class of implicit learning controllers called Robust Integral of the Sign of the Error (RISE) controllers. RISE controllers have been published over the past two decades as a means to yield asymptotic tracking error convergence and implicit asymptotic identification of time-varying uncertainties, for classes of nonlinear systems that are subject to sufficiently smooth bounded exogenous disturbances and/or modeling uncertainties. Despite the wide application of RISE-based techniques, an open question that has eluded researchers during this time-span is whether the asymptotic tracking error convergence is also uniform or exponential. This question has remained open due to certain limitations in the traditional construction of a Lyapunov function for RISE-based error systems. In this dissertation, new insights on the construction of a Lyapunov function are used that result in an exponential stability result for RISE-based controllers. As an outcome of this breakthrough, the inherent learning capability of RISE-based controllers is shown to yield exponential identification of state-dependent disturbances/uncertainty.

CHAPTER 1
INTRODUCTION

1.1 Background

Many practical engineering systems exhibit nonlinear behaviors that make it difficult to predict the output response for a given input. One approach to compensate for nonlinear effects is to feed forward a model of the nonlinearities in the control design as a means to cancel destabilizing effects: feedback linearization is a simplified example of such an approach. However, the system model typically has uncertainties such as parametric uncertainty or more general uncertainty that is often modeled as a bounded exogeneous disturbance. Adaptive and learning-based control strategies are motivated by the desire to include an approximate feedforward model in the controller or to implicitly learn the model through the feedback structure. This dissertation is focused on the development of such adaptive and learning-based controllers. Since the dissertation focuses on general nonlinear systems, a constructive Lyapunov-based design and analysis approach is used.

One way to estimate uncertainty is using a neural network (NN)-based model. NNs are universal function approximators that are capable of modeling continuous functions over a compact domain [1]. Although NNs with a single hidden-layer are capable of approximating general nonlinear functions, deep NNs (DNNs) provide improved performance [2]. Moreover, DNNs are exponentially more expressive than shallow NNs in terms of the total number of neurons required to achieve the same accuracy in function approximation [3].

Motivated by recent advances in DNNs, researchers have explored the use of DNN-based control architectures. DNN-based techniques often employ optimization methods to train the DNN weights by minimizing a loss function over a training dataset [4]. Results in [5–7] utilize such offline DNN training techniques to approximate explicit model predictive control laws. However, such offline methods pose limitations since

training typically requires large amounts of data, and the resulting feedforward terms are implemented as an open-loop approximator based on the offline training.

In contrast to offline training and open-loop implementation, NN weight update laws derived from Lyapunov-based stability analysis methods have been developed to adjust the NN weights in real-time as an adaptive closed-loop feedforward term. Although NN-based adaptive architectures are well-established, these methods only apply to NNs with a single hidden-layer [8]. DNNs achieve improved function approximation performance because of the nested nonlinear parameterizations of the inner-layer activation functions; however, such nested nonlinear functions present a challenge that has heretofore precluded the development of real-time adaptation laws with Lyapunov-based methods.

Motivated by function approximation abilities of DNNs, emerging results in [9–12] develop real-time DNN-based adaptive architectures. In [9] and [10], real-time DNN-based adaptive architectures are developed for model reference adaptive control of linear systems. Similarly, results in [11] generalize the DNN-based adaptive architecture to general nonlinear systems. However, such results only update the output-layer weights in real-time. While the output-layer weights are updated in real-time, data is collected and used to train the inner-layer weights iteratively over discrete training periods via traditional offline techniques. In [12], insights are provided into the development of real-time adaptive weight update laws for individual layers of a feedforward DNN based on a modular design. Modular adaptive designs develop mild constraints on the adaptation laws and provide stability guarantees based on the worst-case scenario of the developed constraints. Although the modular adaptive approach provides constraints on the weight update laws, these constraints are only sufficient and lack insights on how to best design the inner-layer weight adaptation laws.

Chapter 2 and my work in [13] present the first result with Lyapunov-based real-time weight adaptation laws for each layer of a DNN for a general uncertain nonlinear

system. To address the challenges posed by nested nonlinear parameterizations of the inner-layer DNN weights, a recursive representation of the inner-layer DNN structure is developed to facilitate the analysis. Then, a Taylor's first order approximation of the uncertainty is recursively derived. Subsequently, the update laws are derived from a Lyapunov-based stability analysis, in which the first-order terms are canceled by the weight update law-based terms. The remaining terms in the Lyapunov-based analysis are eliminated using a robust control approach.

The adaptation laws developed in this dissertation depend on gradients of activation functions. The adaptation laws contain discontinuities if an activation function with a discontinuous gradient is used in the DNN architecture. Nonsmooth activation functions such as rectified linear units (ReLUs), leaky ReLUs (LReLUs) [14], maxout [15], etc. are often preferred over sigmoidal activation functions, since they empirically exhibit improved function approximation performance while also overcoming the vanishing gradient problem [4, Ch. 6]. As previously noted, the discontinuities in gradients of these activation functions pose difficulties in facilitating standard Lyapunov-based analysis methods. In this chapter, a nonsmooth analysis is performed to address the challenges of including nonsmooth activation functions. The nonsmooth Lyapunov-based analysis guarantees global asymptotic tracking error convergence. Simulation results are provided for a nonlinear system using DNNs with leaky ReLU and hyperbolic tangent activation functions to demonstrate the efficacy and performance of the developed method. A comparison of a DNN with leaky ReLU activation functions to a DNN with hyperbolic tangent activation functions shows improved tracking and function approximation performance while using the DNN with leaky ReLU activation functions.

Although Chapter 2 and my work in [13] provide Lyapunov-derived weight adaptation laws for the DNN, the development is restricted to fully-connected DNNs. There are several limitations associated with standard DNN architectures such as fully-connected and convolutional DNNs. Deeper networks typically suffer from the problem of vanishing

or exploding gradients, i.e., the rate of learning using a gradient-based update rule is highly sensitive to the magnitude of DNN weights. Challenges faced from the vanishing or exploding gradient problem are ubiquitous to both offline training [4] and real-time weight adaptation [13]. Additionally, in applications such as image recognition, the performance of a DNN is found to initially improve by increasing the depth of the DNN. However, as the depth exceeds a threshold, performance rapidly degrades [16].

To overcome the vanishing or exploding gradient problem and the degradation of performance with the increasing depth of a DNN, results in [16] introduce shortcut connections across layers, i.e., a feedforward connection between layers that are separated by more than one layer. DNNs with a shortcut connection are known as deep residual neural networks (ResNets). Offline results in [17] and [18] offer mathematical explanations for why ResNets perform better than non-residual DNNs. In [17], the parameterization of a non-residual DNN is shown to cause difficulties in training DNN layers to approximate the identity function. As explained in [17], for a DNN to achieve a good training accuracy, the DNN layers must be able to approximate the identity function well. Since a shortcut connection in ResNets is represented using an identity function, ResNets provide an improved performance when compared to non-residual DNNs. Additionally, the result in [18] provides explanations from Lyapunov stability theory on why ResNets are easier to train offline using the gradient descent algorithm as compared to non-residual DNNs. The shortcut connections in ResNets facilitate the stability of the equilibria of gradient descent dynamics for a larger set of step sizes or initial weights as compared to non-residual DNNs.

Although there has been significant research across various applications involving ResNets [16, 19–22], the approximation power of ResNets has not yet been explored for adaptive control problems. Developing a ResNet-based adaptive feedforward control term with real-time weight adaptation laws is an open problem. Although real-time weight adaptation laws are developed for fully-connected feedforward DNNs in [13],

the shortcut connections in ResNets pose additional mathematical challenges. Unlike fully-connected DNNs, the shortcut connection prevents a recursive application of Taylor series approximation for each layer of the ResNet. As a result, it is difficult to generate the coupling terms that are generated using the approximation strategy in [13], that can be canceled using the weight adaptation laws in the Lyapunov-based analysis.

My preliminary work in [23] and this dissertation provide the first result on Lyapunov-derived adaptation laws for the weights of each layer of a ResNet-based adaptive controller for uncertain nonlinear systems. To overcome the mathematical challenges posed by the residual network architecture, the ResNet is expressed as a composition of building blocks that involve a shortcut connection across a fully-connected DNN. Then, a constructive Lyapunov-based approach is provided to derive weight adaptation laws for the ResNet using the gradient of each DNN building block. A nonsmooth Lyapunov-based analysis is provided to guarantee global asymptotic tracking error convergence. Unlike [23], which involved a ResNet with only one shortcut connection, this dissertation provides weight adaptation laws for a general ResNet that has an arbitrary number of shortcut connections. The development of adaptation laws for ResNets with an arbitrary number of shortcut connections is challenging due to the complexity of the architecture. This challenge is addressed by constructing a recursive representation of the ResNet which involves a composition of an arbitrary number of building blocks. Then, based on the recursive representation of the ResNet architecture, a first-order Taylor series approximation is applied, which is then utilized to yield the Lyapunov-based adaptation laws. Additionally, unlike [23] which did not provide simulations, this dissertation provides comparative Monte Carlo simulations to demonstrate the performance of the developed ResNet-based adaptive controller, and the results are compared with an equivalent fully-connected DNN-based adaptive controller [13]. Since the performance of ResNet and DNN-based adaptive controllers is sensitive to weight initialization, the Monte Carlo approach is used to provide a fair comparison between the

two architectures. In the Monte Carlo comparison, 10,000 simulations are performed, where the initial weights in each simulation are selected from a uniform random distribution, and a cost function is evaluated for each simulation. Then, the simulation results yielding the least cost for both architectures are compared. The ResNet-based adaptive controller shows a 49.52% and 54.38% improvement in the tracking and function approximation performance, respectively, in comparison to a fully-connected DNN-based adaptive controller.

Although DNNs can compensate for the uncertainty as discussed above, the uncertainty is assumed to be time-invariant. Since DNNs can approximate functions only over a compact domain, it is challenging to approximate time-varying uncertainties over an unbounded time-domain. Perhaps one way to address this problem could be to model the time-varying uncertainty using an architecture with unknown time-varying parameters. However, it is not known how to develop adaptive controllers for systems with time-varying parameters.

Adaptive control of nonlinear dynamical systems with time-varying uncertain parameters is a problem that has received recent attention because of its practical importance for some applications. It has been well established that traditional gradient-based update laws can compensate for constant unknown parameters yielding asymptotic convergence. Moreover, the development of robust modifications of such adaptive update laws result in uniformly ultimately bounded (UUB) results for slowly varying parametric uncertainty using a Lyapunov-based analysis, under the assumption of bounded parameters and their time-derivatives [24–26].

More recent results focus on tracking and parameter estimation improvement using various adaptive control approaches for systems with unknown time-varying parameters. One such approach involves a fast adaptation law [27], where a matrix of time-varying learning rates is utilized to improve the tracking and estimation performance under a finite excitation condition. Another approach uses a set-theoretic control architecture

[28] to reject the effects of parameter variation, while restricting the system error within a prescribed performance bound. While the aforementioned approaches can potentially yield improved transient response, they yield UUB error systems.

Motivation exists to obtain asymptotic convergence of the tracking error to zero, despite the time-varying nature of the uncertain parameters. Results such as [29] and [30] yield asymptotic tracking for linear systems with asymptotically vanishing time-varying parameter variations. For nonlinear systems involving periodic time-varying uncertain parameters with known periodicity, repetitive/iterative learning based approaches such as [31] yield asymptotic tracking. However, it is challenging to extend these results to nonlinear systems where the uncertain parameter variation is non-vanishing and aperiodic.

Robust adaptive control approaches such as [32, Section IV] yield asymptotic adaptive tracking for systems with time-varying uncertain parameters using an adaptive sliding mode-like design, and [32, Section VII], [33] use a continuous robust design; however, such approaches exploit high-gain or high-frequency feedback, without any additional adaptive feedforward term that is specifically designed to target the uncertainty through adaptation. Recent results in [34] yield asymptotic tracking using a method called congelation of variables, where each unknown time-varying parameter is treated as a nominal constant unknown parameter with a time-varying perturbation, and the control input consists of an adaptive feedforward term to compensate for the nominal constant parameters, while a robust high-gain term is designed to compensate the time-varying perturbation. While the congelation of variables based approach can compensate for fast-varying parameters, it requires the regression matrix to vanish with the state, which might be restrictive for a wide variety of applications.

Results such as [35–37] investigate the identification of systems with time-varying parameters. A more recent result in [38] utilizes the dynamic regressor extension and mixing technique to yield finite-time parameter convergence for systems with unknown

piecewise linearly time-varying parameters. Note that these results concern only adaptive parameter estimation, without developing an adaptive feedforward control term for closed-loop implementation.

In the field of fault-tolerant control design, system faults are typically modeled as unknown piecewise constant time-varying parameters such as in [39], for which, classical adaptive control techniques are used. In this dissertation, the more challenging problem of continuously time-varying parameters is considered, which necessitates an alternative adaptive update law.

To illustrate the technical challenges associated with developing an adaptive feedforward term for systems with time-varying parametric uncertainty, consider the scalar dynamical system

$$\dot{x}(t) = a(t)x(t) + b(t)\cos(x(t)) + u(t), \tag{1–1}$$

with the controller $u(t) = -kx(t) - \hat{a}(t)x(t) - \hat{b}(t)\cos(x(t))$, where k is a positive constant gain, $a(t)$ and $b(t)$ are unknown time-varying parameters, $\hat{a}(t)$ and $\hat{b}(t)$ are the parameter estimates of $a(t)$ and $b(t)$, respectively, and the parameter estimation errors $\tilde{a}(t)$ and $\tilde{b}(t)$ are defined as $\tilde{a}(t) \triangleq a(t) - \hat{a}(t)$ and $\tilde{b}(t) \triangleq b(t) - \hat{b}(t)$, respectively. The traditional stability analysis approach for such problems is to consider the candidate Lyapunov function $V(x(t), \tilde{a}(t), \tilde{b}(t)) = \frac{1}{2}x^2(t) + \frac{1}{2\gamma_a}\tilde{a}^2(t) + \frac{1}{2\gamma_b}\tilde{b}^2(t)$, where γ_a and γ_b are positive constant gains. The given definitions and controller yield the following time-derivative of the candidate Lyapunov function: $\dot{V}(t) = -kx^2(t) + \tilde{a}(t)x^2(t) + \tilde{b}(t)x(t)\cos(x(t)) + \frac{\tilde{a}(t)}{\gamma_a}(\dot{a}(t) - \dot{\hat{a}}(t)) + \frac{\tilde{b}(t)}{\gamma_b}(\dot{b}(t) - \dot{\hat{b}}(t))$. For the constant parameter case, i.e., $\dot{a}(t) = \dot{b}(t) = 0$, the well-known adaptive update laws $\dot{\hat{a}}(t) = \gamma_a x^2(t)$ and $\dot{\hat{b}}(t) = \gamma_b x(t)\cos(x(t))$, respectively, will cancel $\tilde{a}(t)x^2(t)$ and $\tilde{b}(t)x(t)\cos(x(t))$ in $\dot{V}(t)$, leading to Lyapunov stability and asymptotic tracking. However, when the parameters are time-varying, it is unclear how to address $\dot{a}(t)$ and $\dot{b}(t)$ via a feedforward adaptive update law such that $\dot{V}(t)$ becomes at least negative semi-definite. Alternatively to

obtaining a negative semi-definite derivative of the Lyapunov-like function (which is a contribution of this dissertation), the typical approach to design adaptive controllers for the time-varying parameter case is to consider a robust modification of the update laws and assume some constant upper bounds on $|a(t)|, |b(t)|$, $|\dot{a}(t)|$, and $\left|\dot{b}(t)\right|$ to obtain a UUB result. For instance, consider a standard gradient update law with sigma-modification [26], $\dot{\hat{a}}(t) = \gamma_a x^2(t) - \gamma_a \sigma \hat{a}(t)$, $\dot{\hat{b}}(t) = \gamma_b x(t)\cos(x(t)) - \gamma_b \sigma \hat{b}(t)$, which yields $\dot{V}(t) = -kx^2(t) - \sigma\tilde{a}^2(t) - \sigma\tilde{b}^2(t) + \tilde{a}(t)(\frac{\dot{a}(t)}{\gamma_a} + \sigma a(t)) + \tilde{b}(t)(\frac{\dot{b}(t)}{\gamma_b} + \sigma b(t))$, implying a UUB result when the parameters $a(t)$ and $b(t)$, and their time-derivatives $\dot{a}(t)$ and $\dot{b}(t)$ are bounded. More modern approaches (cf., [40]) provide additional modifications to yield UUB results with improved transient performance.

It would be desirable to have a sliding mode-like term based on $\tilde{a}(t)$ and $\tilde{b}(t)$ (i.e., $\text{sgn}(\tilde{a})$ and $\text{sgn}(\tilde{b})$ in the adaptation law), if only $\tilde{a}(t)$ and $\tilde{b}(t)$ were known. Another approach could be to use a pure robust controller, e.g., $u(t) = -kx(t) - \bar{a}x(t) - \bar{b}\text{sgn}(x(t))$, where $\bar{a}$ and $\bar{b}$ are known constant upper bounds on the norms of parameters $|a(t)|$ and $|b(t)|$, respectively. If the bounds $\bar{a}$ and $\bar{b}$ are unknown, an adaptation law could be designed to yield their adaptive estimates, i.e., $\hat{\bar{a}}$ and $\hat{\bar{b}}$. Either of these approaches would yield an asymptotic tracking result (cf., [32]), but as stated earlier, these approaches require a discontinuous sliding-mode term in the control input, and do not include an adaptive feedforward term to compensate for the uncertainty. The congelation of variables based approach in [34] may help avoid some of the aforementioned challenges; however, it is not applicable for uncertain terms like $b(t)\cos(x(t))$, which do not vanish with the state.

The major challenge in achieving asymptotic tracking is that the time-derivative of the parameter acts like an unknown exogenous disturbance in the parameter estimation dynamics, which is difficult to cancel with an adaptive update law in a Lyapunov-based stability analysis. This technical challenge is addressed in Chapter 4 and in my previous paper [41], through new insights into the closed-loop error system development and

stability analysis, coupled with a new adaptive update law design. Specifically, because of challenges associated with including the uncertain parameter estimation error in the Lyapunov function, such terms are omitted, and include a P-function based on [42], while also formulating the closed-loop error system so that they appear in the Lyapunov-based derivative in a manner that facilitates an adaptive update law. The unique challenge associated with incorporating the time-varying parameter estimation error is addressed in the analysis by formulating the update law so that it contains a signum function of the tracking error term multiplied by a desired regressor. The update law also involves a projection algorithm to ensure that the parameter estimates stay within a known bounded set. However, the projection algorithm introduces a potentially destabilizing term in the time-derivative of the candidate Lyapunov function, leading to an additional technical obstacle to obtain asymptotic tracking. This challenge is resolved by using an additional term in the control input, which compensates for terms that result from using a projection operator. The developed Lyapunov-based stability analysis yields semi-global asymptotic tracking, and boundedness of the closed-loop signals. Additionally, the time-varying uncertain function approximation error is shown to converge to zero. The dynamics of a two-link manipulator are used in a simulation to demonstrate the asymptotic tracking and function approximation error convergence result, and the tracking performance is compared with a robust e-modification update law [43] based controller.

Another way to address the challenges posed by the uncertainty in the system is using a class of continuous robust controllers termed Robust Integral of the Sign of the Error (RISE) controllers. RISE controllers have been published over the past two decades [32, 42, 44–55] as a means to yield asymptotic tracking error convergence and asymptotic identification of time-varying uncertainties, for classes of nonlinear systems that are subject to sufficiently smooth bounded exogenous disturbances and/or modeling uncertainties. RISE-based methods have been used for a wide variety of

applications involving control [45–61], estimation [44, 62, 63], and optimization [64]. Despite the wide application of RISE-based techniques, an open question that has eluded researchers is whether the asymptotic tracking error convergence is also uniform or exponential. This question has remained open due to certain limitations in the traditional construction of a Lyapunov function for RISE-based error systems.

Traditional analysis methods for a RISE-based error system involve a Lyapunov-based approach, where the candidate Lyapunov function (denoted by V_L) includes a P-function (denoted by P) in addition to a typical sum of norm squared error terms. The P-function is designed by selecting $\dot{P}$ to cancel disturbance-based terms in $\dot{V}_L$ and is the essential analysis and design tool to enable asymptotic convergence (instead of uniformly ultimately bounded tracking) despite the presence of a disturbance term that is only upper bounded by a constant. Previous results, including the result in Chapter 5 and my previous paper [65], determine P as a function of the initial conditions of the system that is proven to be non-negative under certain gain conditions. Evaluating $\dot{V}_L$ along the closed-loop error trajectories yields a negative semi-definite $\dot{V}_L$. Then the extension of the LaSalle-Yoshizawa theorem for nonsmooth systems in [66] is invoked to prove asymptotic tracking error convergence. Since the LaSalle-Yoshizawa theorem is based on Barbalat's lemma, the traditional analysis methodology does not guarantee uniform tracking error convergence, and the non-strictness of V_L precluded exponential stability of the closed-loop error system's origin.

To prove exponential stability, it would be sufficient to select a positive-definite V_L such that $\dot{V}_L \leq -\lambda_L V_L$ for almost all time, with some positive constant λ_L. Then exponential stability can be established using the comparison principle. Such a Lyapunov function is developed in [55], which is the only known RISE-based exponential tracking result. The result in [55] was developed for a specific application under the assumption that the first and second derivatives of the uncertainty are bounded by known constants. However, RISE-based controllers have been applied to a broader set of applications

where this assumed bound would not hold. For example, in results like [49, 51] and [52] that involve dynamic compensator-based auxiliary control terms, the first or second derivative of the uncertainty have bounds that are state-dependent. It is not clear how the analysis approach in [55] can be extended for such cases.

In Chapter 5 and my previous paper [65], a novel P-function design is developed that results in a strict Lyapunov function. The new analysis results in exponential stability of the closed-loop error system's origin using a comparison theorem-based argument. The novel P-function is shown to be non-negative under certain gain conditions by examining the analytically derived solution to the dynamics in $\dot{P}$. Unlike the analysis approach in [55], the developed P-function can be easily modified for various bounds on the first and second derivatives of uncertainty. To rule out the existence of extra solutions for P that could be potentially negative over some time interval, the derived solution for P is shown to be unique corresponding to a given closed-loop error trajectory. Additionally, solution-dependent arguments are employed to show the sign of the error term is integrable, and $\dot{V}_L \leq -\lambda_L V_L$ for almost all time, which involves showing that the set of time-instants where $\dot{V}_L \leq -\lambda_L V_L$ may not be true have Lebesgue measure zero. Furthermore, the disturbance/uncertainty is shown to be estimated with exponential convergence of the disturbance identification error, while prior results only indicated asymptotic convergence.

1.2 Outline of the dissertation

In Chapter 2, a DNN-based adaptive controller is developed with Lyapunov-based real-time weight adaptation laws for each layer of the feedforward DNN, with stability guarantees. Additionally, the developed method allows nonsmooth activation functions to be used in the DNN to facilitate improved transient performance. A nonsmooth Lyapunov-based stability analysis proves global asymptotic tracking error convergence. Simulation results are provided for a nonlinear system using DNNs with LReLU and

hyperbolic tangent activation functions to demonstrate the efficacy of the developed method.

Chapter 3 provides the first result on Lyapunov-derived weight adaptation for a ResNet-based adaptive controller. A nonsmooth Lyapunov-based analysis is provided to guarantee global asymptotic tracking error convergence. Comparative Monte Carlo simulations are provided to demonstrate the performance of the developed ResNet-based adaptive controller. The ResNet-based adaptive controller shows a 49.52% and 54.38% improvement in the tracking and function approximation performance, respectively, in comparison to a fully-connected DNN-based adaptive controller.

In Chapter 4, a continuous adaptive controller is developed for nonlinear dynamical systems with linearly parameterizable uncertainty involving time-varying uncertain parameters. Through a unique stability analysis strategy, a new adaptive feedforward term is developed along with specialized feedback terms, to yield an asymptotic tracking error convergence result by compensating for the time-varying nature of the uncertain parameters. A Lyapunov-based stability analysis is shown for Euler-Lagrange systems, which ensures asymptotic tracking error convergence and boundedness of the closed-loop signals. Additionally, the time-varying uncertain function approximation error is shown to converge to zero. A simulation example of a two-link manipulator is provided to demonstrate the asymptotic tracking result.

Chapter 5 provides an exponential stability result for RISE controllers. New insights on the construction of a Lyapunov function are used that result in an exponential stability result for RISE-based controllers. As an outcome of this breakthrough, the inherent learning capability of RISE-based controllers is shown to yield exponential identification of state-dependent disturbances/uncertainty.

In Chapter 6, the conclusions of this dissertation are provided.

1.3 Preliminaries

To facilitate the readability, this section provides some preliminary information that defines much of the mathematical notation and definitions used in the dissertation.

The space of essentially bounded Lebesgue measurable functions is denoted by $\mathcal{L}_\infty$. The right-to-left matrix product operator is represented by $\overleftarrow{\prod}$, i.e., $\overleftarrow{\prod_{p=1}^{m}} A_p = A_m \dots A_2 A_1$ and $\overleftarrow{\prod_{p=a}^{m}} A_p = 1$ if $a > m$. The vectorization operator is denoted by $\text{vec}(\cdot)$, i.e., given $A \triangleq [a_{i,j}] \in \mathbb{R}^{n \times m}$, $\text{vec}(A) \triangleq [a_{1,1}, \dots, a_{1,m}, \dots, a_{n,1}, \dots, a_{n,m}]^T$. The p-norm is denoted by $\|\cdot\|_p$, where the subscript is suppressed when $p = 2$. The Frobenius norm is denoted by $\|\cdot\|_F \triangleq \|\text{vec}(\cdot)\|$. The Kronecker product is denoted by $\otimes$. Function compositions are denoted using the symbol $\circ$, e.g., $(g \circ h)(x) = g(h(x))$, given suitable functions f and g. Given some functions f and g, the notation $f(y) = \mathcal{O}^m(g(y))$ means that there exists some constants $M \in \mathbb{R}_{>0}$ and $y_0 \in \mathbb{R}$ such that $\|f(y)\| \leq M \|g(y)\|^m$ for all $y \geq y_0$. The space of continuous functions with continuous first m derivatives is denoted by $\mathcal{C}^m$. The Filippov set-valued map defined in [67, Equation 2b] is denoted by $K[\cdot]$. Consider a Lebesgue measurable and locally essentially bounded function $h : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. Then, the function $y : \mathcal{I} \to \mathbb{R}^n$ is called a *Filippov solution* of $\dot{y} = h(y,t)$ on the interval $\mathcal{I} \subseteq \mathbb{R}_{\geq 0}$ if y is absolutely continuous on $\mathcal{I}$ and $\dot{y} \in K[h](y,t)$ for almost all $t \in \mathcal{I}$. The notation $F : A \rightrightarrows B$ denotes a set-valued map from set A to set B. A solution is called *complete* if $\mathcal{I}_y$ is unbounded. A solution $y_2 : [t_0, t_2) \to \mathbb{R}^n$ to $\dot{y} = h(y,t)$ is called a *proper right extension* of a solution $y_1 : [t_0, t_1) \to \mathbb{R}^n$ to $\dot{y} = h(y,t)$ if $t_2 > t_1$ and $y_2(t) = y_1(t)$, $\forall t \in [t_0, t_1)$. A solution to $\dot{y} = h(y,t)$ is called *maximal* if it does not have a proper right extension which is also a solution to $\dot{y} = h(y,t)$. If a solution is maximal and if the closure of its range, $\overline{\{y(t) \in \mathbb{R}^n | t \in \mathcal{I}_y\}}$, is compact, then the solution is called *precompact*.

The following fact plays a key role in facilitating the analysis in Chapters 2 and 3.

Fact 1.1. *[68, Proposition 7.1.9] Given any $A \in \mathbb{R}^{p\times a}$, $B \in \mathbb{R}^{a\times r}$, and $C \in \mathbb{R}^{r\times s}$,*

$$\mathrm{vec}(ABC) \;=\; (C^T \otimes A)\mathrm{vec}\,(B)\,. \tag{1–2}$$

Differentiating (1–2) on both sides with respect to $\mathrm{vec}\,(B)$ yields the property

$$\frac{\partial}{\partial \mathrm{vec}\,(B)}\mathrm{vec}(ABC) \;=\; (C^T \otimes A). \tag{1–3}$$

CHAPTER 2
LYAPUNOV-DERIVED CONTROL AND ADAPTIVE UPDATE LAWS FOR INNER AND OUTER LAYER WEIGHTS OF A DEEP NEURAL NETWORK

Lyapunov-based real-time update laws are well-known for single hidden layer NN-based adaptive controllers. However, developing stability-driven real-time weight update laws for DNNs remains an open question. This dissertation and my work in [13] are the first results with Lyapunov-based real-time weight adaptation laws for each layer of a feedforward DNN-based control architecture, with stability guarantees. Additionally, the developed method allows nonsmooth activation functions to be used in the DNN to facilitate improved transient performance. A nonsmooth Lyapunov-based stability analysis proves global asymptotic tracking error convergence. Simulation results are provided for a nonlinear system using DNNs with leaky rectified linear unit (LReLU) and hyperbolic tangent activation functions to demonstrate the efficacy of the developed method.

2.1 Unknown System Dynamics and Control Design

Consider a control-affine nonlinear dynamic system modeled as

$$\dot{x} = f(x) + u, \tag{2–1}$$

where $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ denotes a Filippov solution[1] to (2–1), $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes an unknown differentiable function, and $u : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ denotes a control input. The control objective is to track a user-defined reference trajectory $x_d : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$. The reference trajectory is designed to be continuously differentiable, such that $x_d(t) \in \Omega$, $\forall t \in \mathbb{R}_{\geq 0}$, and $\dot{x}_d \in \mathcal{L}_\infty$, where $\Omega \subset \mathbb{R}^n$ denotes a known compact set. To quantify the tracking

[1] Generalized solutions such as Filippov or Krasovskii solutions are considered instead of classical solutions to facilitate a nonsmooth control design. These solutions are guaranteed to exist for nonsmooth systems with Lebesgue measurable and locally essentially bounded right-hand-sides [69, Proposition 3], whereas classical solutions might not exist.

objective, the tracking error $e : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is defined as

$$e \triangleq x - x_d. \tag{2–2}$$

2.1.1 Deep Neural Network Architecture

A variety of DNN architectures are known to approximate any given continuous function on a compact set, based on universal approximation theorems that can be invoked case-by-case for DNN architectures [70]. Let $\Phi : \mathbb{R}^n \times \mathbb{R}^{L_0 \times L_1} \times \ldots \times \mathbb{R}^{L_k \times L_{k+1}} \to \mathbb{R}^n$ denote the feedforward DNN architecture defined as

$$\Phi(x_d, V_0, V_1, \ldots, V_k) \triangleq \left(V_k^T \phi_k \circ ... \circ V_1^T \phi_1\right)\left(V_0^T x_{da}\right), \tag{2–3}$$

where $x_{da} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n+1}$ denotes the augmented desired state $x_{da} \triangleq \begin{bmatrix} x_d^T & 1 \end{bmatrix}^T$, and $k \in \mathbb{N}$ denotes the total number of hidden-layers. The matrix of weights and biases at the j^{th} layer is denoted by $V_j \in \mathbb{R}^{L_j \times L_{j+1}}$, where $L_j \in \mathbb{N}$ denotes the number of nodes in the j^{th} inner-layer for all $j \in \{0, \ldots, k\}$, with $L_0 \triangleq n+1$ and $L_{k+1} \triangleq n$. The vector of smooth activation functions at the j^{th} layer is denoted by $\phi_j : \mathbb{R}^{L_j} \to \mathbb{R}^{L_j}$. Although ϕ_j is defined as a smooth function, the subsequent analysis allows the inclusion of nonsmooth activation functions by modeling them via a switching mechanism involving smooth functions. If the DNN involves multiple types of activation functions at each layer, then ϕ_j may be represented as $\phi_j \triangleq \begin{bmatrix} \varsigma_{j,1} & \ldots & \varsigma_{j,L_j-1} & 1 \end{bmatrix}^T$, where $\varsigma_{j,i} : \mathbb{R} \to \mathbb{R}$ denotes the activation function at the i^{th} node of j^{th} layer. Note that x_{da} and ϕ_j are augmented with 1 to facilitate the inclusion of a bias term. The DNN architecture in (2–3) can also be represented recursively as

$$\Phi_j \triangleq \begin{cases} V_j^T \phi_j\left(\Phi_{j-1}\right), & j \in \{1, \ldots, k\}, \\ V_0^T x_{da}, & j = 0, \end{cases} \tag{2–4}$$

and $\Phi(x_d, V_0, \ldots, V_k) = \Phi_k$, where $\Phi_j : \mathbb{R}^n \times \mathbb{R}^{L_0 \times L_1} \times \ldots \times \mathbb{R}^{L_j \times L_{j+1}} \to \mathbb{R}^{L_{j+1}}$ denotes $(x_d, V_0, \ldots, V_j) \mapsto \Phi_j(x_d, V_0, \ldots, V_j)$. The universal function approximation property states that the function space of DNNs given by (2–3) is dense in $\mathcal{C}(\Omega)$ [70, Thm. 3.2], where $\mathcal{C}(\Omega)$ denotes the space of functions continuous over Ω. For any given $f \in \mathcal{C}(\Omega)$ and prescribed $\overline{\varepsilon} \in \mathbb{R}_{>0}$, there exist some $k, L_j \in \mathbb{N}$, and corresponding ideal weights and biases, $V_j^* \in \mathbb{R}^{L_j \times L_{j+1}}$, $\forall j \in \{0, \ldots, k\}$, such that $\sup_{x_d \in \Omega} \|f(x_d) - \Phi(x_d, V_0^*, V_1^*, \ldots, V_k^*)\| \leq \overline{\varepsilon}$. Then the unknown function in (2–1) can be modeled as

$$f(x_d) = \Phi(x_d, V_0^*, V_1^*, \ldots, V_k^*) + \varepsilon(x_d), \tag{2–5}$$

where $\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ denotes the unknown function approximation error such that $\sup_{x_d \in \Omega} \|\varepsilon(x_d)\| \leq \overline{\varepsilon}$. It is assumed there exists a known constant $\overline{V} \in \mathbb{R}_{>0}$ such that $\sup_{x_d \in \Omega, \forall j} \left\|V_j^*\right\|_F \leq \overline{V}$ (cf., [71, Assumption 1]).

2.1.2 Control Law Development

The universal approximation property makes DNN-based adaptive control architectures well-suited for unknown dynamics, as in (2–1) where $f(\cdot)$ is unknown [70, Thm. 3.2]. The adaptive feedforward DNN term is designed as $\hat{\Phi} \triangleq \Phi(x_d, \hat{V}_0, \ldots, \hat{V}_k)$, where $\hat{V}_j : \mathbb{R}_{\geq 0} \to \mathbb{R}^{L_j \times L_{j+1}}$ for all $j \in \{0, \ldots, k\}$ denotes the estimated weight matrix for the j^{th} layer. The weight estimation error of the ideal inner-layer weights $\widetilde{V}_j : \mathbb{R}_{\geq 0} \to \mathbb{R}^{L_j \times L_{j+1}}$ for all $j \in \{0, \ldots, k\}$ is defined as $\widetilde{V}_j \triangleq V_j^* - \hat{V}_j$. The gradient of the activation function vector at the j^{th} layer is denoted as $\phi_j' : \mathbb{R}^{L_j} \to \mathbb{R}^{L_j \times L_j}$, and $\phi_j'(y) \triangleq \frac{\partial}{\partial z} \phi_j(z)\big|_{z=y}$, $\forall y \in \mathbb{R}^{L_j}$. To facilitate the subsequent stability analysis, let the function $f_e : \mathbb{R}^n \times \Omega \to \mathbb{R}^n$ be defined as $f_e \triangleq f(x) - f(x_d)$. By [72, Lemma 5], the function $(x, x_d) \mapsto f_e$ is bounded as $\|f_e\| \leq \rho(\|e\|)\|e\|$ for all $x \in \mathbb{R}^n$ and $x_d \in \Omega$, where $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes a known strictly increasing function. Based on the subsequent stability analysis, the control input is designed as

$$u \triangleq \dot{x}_d - \rho(\|e\|)e - k_1 e - k_s \text{sgn}(e) - \hat{\Phi}, \tag{2–6}$$

where $k_1, k_s \in \mathbb{R}_{>0}$ are user-defined control gains, and $\text{sgn}(\cdot)$ denotes the vector signum function. The following short-hand notations are introduced for brevity in the subsequent analysis: $\Phi_j^* \triangleq \Phi_j(x_d, V_0^*, \ldots, V_j^*)$, $\hat{\Phi}_j \triangleq \Phi_j(x_d, \hat{V}_0, \ldots, \hat{V}_j)$, $\widetilde{\Phi}_j \triangleq \Phi_j^* - \hat{\Phi}_j$, $\Phi^* \triangleq \Phi_k^*$, $\widetilde{\Phi} \triangleq \widetilde{\Phi}_k = \Phi^* - \hat{\Phi}$, $\phi_j^* \triangleq \phi_j\left(\Phi_{j-1}^*\right)$, $\hat{\phi}_j \triangleq \phi_j\left(\hat{\Phi}_{j-1}\right)$, and $\hat{\phi}_j' \triangleq \phi_j'\left(\hat{\Phi}_{j-1}\right)$.

Based on the subsequent analysis, the input layer weight adaptation law is designed as

$$\text{vec}(\dot{\hat{V}}_0) \triangleq \text{proj}(\Gamma_0((\prod_{l=1}^{\overset{\curvearrowleft}{k}} \hat{V}_l^T \hat{\phi}_l')(I_{L_1} \otimes x_{da}^T))^T e), \tag{2–7}$$

and the j^{th} layer weight adaptation law is designed as

$$\text{vec}(\dot{\hat{V}}_j) \triangleq \text{proj}(\Gamma_j((\prod_{l=j+1}^{\overset{\curvearrowleft}{k}} \hat{V}_l^T \hat{\phi}_l')(I_{L_{j+1}} \otimes \hat{\phi}_j^T))^T e), \tag{2–8}$$

$\forall j \in \{1, ..., k\}$, where $\Gamma_j \in \mathbb{R}^{L_j L_{j+1} \times L_j L_{j+1}}$ is a positive-definite adaptation gain matrix for all $j \in \{0, ..., k\}$. The operator $\text{proj}(\cdot)$ denotes the projection operator defined in [73, Appendix E, Eq. E.4], which is used to ensure $\hat{V}_j(t) \in \mathcal{B}_j \triangleq \{\theta \in \mathbb{R}^{L_j L_{j+1}} : \|\theta\|_F \leq \overline{V}\}, \forall (t, j) \in \mathbb{R}_{\geq 0} \times \{0, 1, \ldots, k\}$.

2.2 Stability Analysis

2.2.1 Closed-Loop Error System Development

Subtracting $\hat{\Phi}_j$ from Φ_j^*, using (2–4), adding and subtracting $V_j^{*T} \hat{\phi}_j$, and rearranging terms yields

$$\widetilde{\Phi}_j = \widetilde{V}_j^T \hat{\phi}_j + V_j^{*T}(\phi_j^* - \hat{\phi}_j), \tag{2–9}$$

$\forall j \in \{1, \ldots, k\}$, and $\widetilde{\Phi}_0 = \widetilde{V}_0^T x_{da}$. Using (2–1), (2–2), and (2–6) yields the closed-loop error system

$$\dot{e} = f_e + \widetilde{\Phi} + \varepsilon(x_d) - \rho(\|e\|)e - k_1 e - k_s \text{sgn}(e). \tag{2–10}$$

The term $\widetilde{\Phi}$ in (2–10) has a nested nonlinear parameterization in V_j^* and $\hat{V}_j$, which precludes the application of traditional analysis techniques that are used for linearly

parameterized adaptive systems. A first-order Taylor series approximation is developed in [71] to overcome the challenges presented by the nonlinear parameterization for three-layer neural networks. To overcome the nested structure of nonlinear parameterization in DNNs, a recursive approach is used to develop a first-order Taylor series approximation for ϕ_j^*, $\widetilde{\Phi}_j$, and $\widetilde{\Phi}$. Using the first-order Taylor series approximation in [71, Eq. 22] yields

$$\phi_j^* = \hat{\phi}_j + \hat{\phi}_j' \widetilde{\Phi}_{j-1} + \mathcal{O}^2\left(\widetilde{\Phi}_{j-1}\right), \tag{2–11}$$

$\forall j \in \{1, \ldots, k\}$. Substituting (2–11) into (2–9), adding and subtracting $\hat{V}_j^T \hat{\phi}_j' \widetilde{\Phi}_{j-1}$, and rearranging terms yields

$$\widetilde{\Phi}_j = \widetilde{V}_j^T \hat{\phi}_j + \hat{V}_j^T \hat{\phi}_j' \widetilde{\Phi}_{j-1} + \Delta_j, \tag{2–12}$$

where $\Delta_j : \mathbb{R}_{\geq 0} \to \mathbb{R}^{L_{j+1}}$is defined as

$$\Delta_j \triangleq \widetilde{V}_j^T \hat{\phi}_j' \widetilde{\Phi}_{j-1} + V_j^{*T} \mathcal{O}^2\left(\widetilde{\Phi}_{j-1}\right), \tag{2–13}$$

$\forall j \in \{1, \ldots, k\}$. Since the term $\widetilde{V}_j^T \hat{\phi}_j$ is a vector, $\widetilde{V}_j^T \hat{\phi}_j = \operatorname{vec}(\widetilde{V}_j^T \hat{\phi}_j) = \operatorname{vec}(\hat{\phi}_j^T \widetilde{V}_j) = \operatorname{vec}(\hat{\phi}_j^T \widetilde{V}_j I_{L_{j+1}})$. Applying Fact 1.1 on $\operatorname{vec}(\hat{\phi}_j^T \widetilde{V}_j I_{L_{j+1}})$ yields

$$\widetilde{V}_j^T \hat{\phi}_j = \left(I_{L_{j+1}} \otimes \hat{\phi}_j^T\right) \operatorname{vec}\left(\widetilde{V}_j\right). \tag{2–14}$$

Substituting (2–14) into (2–12) yields the recursive representation

$$\widetilde{\Phi}_j = \left(I_{L_{j+1}} \otimes \hat{\phi}_j^T\right) \operatorname{vec}\left(\widetilde{V}_j\right) + \hat{V}_j^T \hat{\phi}_j' \widetilde{\Phi}_{j-1} + \Delta_j, \tag{2–15}$$

$\forall j \in \{1, \ldots, k\}$. To facilitate the subsequent analysis, the following lemma yields a generalized expression for $\widetilde{\Phi}_j$.

Lemma 2.1. *For all $j \in \{0, \ldots, k\}$, the term $\widetilde{\Phi}_j$ can be expressed as*

$$\widetilde{\Phi}_j = \sum_{i=1}^{j} \left(\prod_{l=i+1}^{\overset{\curvearrowleft}{j}} \hat{V}_l^T \hat{\phi}_l' \right) \left(I_{L_{i+1}} \otimes \hat{\phi}_i^T\right) \operatorname{vec}\left(\widetilde{V}_i\right) + \left(\prod_{l=1}^{\overset{\curvearrowleft}{j}} \hat{V}_l^T \hat{\phi}_l' \right) \left(I_{L_1} \otimes x_{da}^T\right) \operatorname{vec}\left(\widetilde{V}_0\right)$$

$$+\sum_{i=1}^{j}\left(\prod_{l=i+1}^{\overset{\curvearrowleft}{j}}\hat{V}_l^T\hat{\phi}_l'\right)\Delta_i. \tag{2–16}$$

Proof. This lemma can be proved using mathematical induction. Using (2–4) and Fact 1.1 yields $\widetilde{\Phi}_0 = \widetilde{V}_0^T x_{da} = (I_{L_1}\otimes x_{da}^T)\mathrm{vec}(\widetilde{V}_0)$. Since $\prod_{l=1}^{\overset{\curvearrowleft}{0}}\hat{V}_l^T\hat{\phi}_l' = 1$, it can be verified that (2–16) also yields $\widetilde{\Phi}_0 = (I_{L_1}\otimes x_{da}^T)\mathrm{vec}(\widetilde{V}_0)$. Thus, Lemma 2.1 holds for $j = 0$. To use induction, assume (2–16) applies for $j = h-1$, given any arbitrary $h \in \{1,\ldots,k\}$, and evaluate $\widetilde{\Phi}_{h-1}$. Then, using (2–15) with $j = h$ yields

$$\widetilde{\Phi}_h = (I_{L_{h+1}}\otimes\hat{\phi}_h^T)\mathrm{vec}\left(\widetilde{V}_h\right) + \hat{V}_h^T\hat{\phi}_h'\widetilde{\Phi}_{h-1} + \Delta_h \tag{2–17}$$

Substituting $\widetilde{\Phi}_{h-1}$ into (2–17) and rearranging terms, it can be verified that the obtained expression is the same as that obtained using (2–16). Thus, Lemma 2.1 also applies for $j = h$. □

2.2.2 Nonsmooth Analysis

Let $\Xi_0 \triangleq \prod_{l=1}^{\overset{\curvearrowleft}{k}}\hat{V}_l^T\hat{\phi}_l'$, $\Xi_j \triangleq \prod_{l=j+1}^{\overset{\curvearrowleft}{k}}\hat{V}_l^T\hat{\phi}_l'$, $\Lambda_0 \triangleq \Xi_0(I_{L_1}\otimes x_{da}^T)$, and $\Lambda_j \triangleq \Xi_j(I_{L_{j+1}}\otimes\hat{\phi}_j^T)$, $\forall j \in \{1,\ldots,k\}$, for notational brevity, where $\Xi_j : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n\times L_{j+1}}$ and $\Lambda_j : \mathbb{R}_{\geq 0} \to \mathbb{R}^{L_{j+1}\times L_j L_{j+1}}$, respectively, $\forall j \in \{0,\ldots,k\}$. The subsequent analysis is structured to account for nonsmooth systems. Thus, state-dependent switching between smooth activation functions can also be considered in the analysis. Specifically, a nonsmooth activation function with a finite number of discontinuities in its gradient can be modeled by a switched function involving a collection of smooth activation functions. Let $\sigma \in \mathcal{N}$ denote the switching index considering the total number of switching between activation functions in the entire DNN, where $\mathcal{N} \subset \mathbb{N}$ denotes the set of all possible switching indices. Then, the function approximation in (2–5) can be represented as $f(x_d) = \Phi_{k,\sigma}(x_d, V_0^*, V_1^*, \ldots, V_k^*) + \varepsilon_\sigma(x_d)$, such that $(x_d, V_0, \ldots, V_j) \mapsto \Phi_{k,\sigma}(x_d, V_0, \ldots, V_j)$ is smooth for each σ with the corresponding approximation error $\varepsilon_\sigma(x_d)$. Thus, ϕ_j^*, $\hat{\phi}_j$, $\hat{\phi}_j'$, $\widetilde{\Phi}_j$ Ξ_j, Λ_j, and Δ_j can also be represented as the switched functions $\phi_{j,\sigma}^*$, $\hat{\phi}_{j,\sigma}$, $\hat{\phi}_{j,\sigma}'$, $\widetilde{\Phi}_{j,\sigma}$, $\Xi_{j,\sigma}$,

$\Lambda_{j,\sigma}$, and $\Delta_{j,\sigma}$, respectively, such that they are continuous for each σ. It is assumed that the bound $\sup_{x_d \in \Omega} \|\varepsilon_\sigma(x_d)\| \leq \overline{\varepsilon}$ holds for all $\sigma \in \mathcal{N}$. Using Lemma 2.1 yields $\widetilde{\Phi} = \widetilde{\Phi}_{k,\sigma} = \sum_{j=0}^{k} \Lambda_{j,\sigma} \mathrm{vec}(\widetilde{V}_j) + \sum_{j=1}^{k} \Xi_{j,\sigma} \Delta_{j,\sigma}$. Substituting $\widetilde{\Phi}$ into (2–10) yields

$$\dot{e} = f_e + \sum_{j=0}^{k} \Lambda_{j,\sigma} \mathrm{vec}(\widetilde{V}_j) + \sum_{j=1}^{k} \Xi_{j,\sigma} \Delta_{j,\sigma} + \varepsilon_\sigma(x_d) - \rho(\|e\|)e - k_1 e - k_s \mathrm{sgn}(e). \tag{2–18}$$

Additionally, the adaptation laws in (2–7) and (2–8) can be represented using $\mathrm{vec}(\dot{\hat{V}}_j) \triangleq \mathrm{proj}(\Gamma_j \Lambda_{j,\sigma}^T e), \forall j \in \{0, \ldots, k\}$. Consequently, $\mathrm{vec}(\dot{\widetilde{V}}_j) = -\mathrm{proj}(\Gamma_j \Lambda_{j,\sigma}^T e), \forall j \in \{0, \ldots, k\}$. Since $\left\|V_j^*\right\|_F \leq \overline{V}$ and $\left\|\hat{V}_j\right\|_F \leq \overline{V}, \forall j \in \{0, \ldots, k\}$, it follows that $\left\|\widetilde{V}_j\right\|_F = \left\|V_j^* - \hat{V}_j\right\|_F \leq 2\overline{V}$. Moreover, since x_{da} is bounded, and $\phi_{j,\sigma}$ and $\phi'_{j,\sigma}$ are continuous for each $\sigma \in \mathcal{N}$, it follows from (2–9) that $\phi^*_{j,\sigma}$, $\hat{\phi}_{j,\sigma}$, $\hat{\phi}'_{j,\sigma}$, $\widetilde{\Phi}_{j,\sigma}$, and $\Xi_{j,\sigma}$ can be bounded by known constants for all $(j, \sigma) \in \{0, \ldots, k\} \times \mathcal{N}$. Therefore, based on (2–13), Δ_j can be bounded by known constants for all $j \in \{1, \ldots, k\}$, and it follows that there exists a known constant $c \in \mathbb{R}_{>0}$ such that

$$\left\| \sum_{i=1}^{k} \Xi_{i,\sigma} \Delta_{i,\sigma} \right\| \leq c. \tag{2–19}$$

Let $z : \mathbb{R}_{\geq 0} \to \mathbb{R}^\Psi$ denote the concatenated function, $z \triangleq \left[e^T, \ \mathrm{vec}\left(\tilde{V}_0\right)^T, \ \ldots, \ \mathrm{vec}\left(\tilde{V}_k\right)^T \right]^T$, where $\Psi \triangleq n + \sum_{j=0}^{k} L_j L_{j+1}$ is defined for notational brevity. Let $w_\sigma : \mathbb{R}^\Psi \times \mathbb{R}_{\geq 0} \to \mathbb{R}^\Psi$ denote the concatenated right hand sides of (2–18) and $\mathrm{vec}(\dot{\widetilde{V}}_j) = -\mathrm{proj}(\Gamma_j \Lambda_{j,\sigma}^T e)$. Then (2–18) and $\mathrm{vec}(\dot{\widetilde{V}}_j)$ can be represented by the collection of subsystems $\dot{z} = w_\sigma(z, t)$, and the corresponding switched system is represented by

$$\dot{z} = w_{\varrho(z,t)}(z, t), \tag{2–20}$$

where $\varrho : \mathbb{R}^{\Psi} \times \mathbb{R}_{\geq 0} \to \mathcal{N}$ denotes a state-dependent switching signal that satisfies [74, Assumption 1].[2] Based on the result in [74], the invariance properties of (2–20) are established by establishing the invariance properties of $\dot{z} = w_{\sigma}(z,t)$ for each $\sigma \in \mathcal{N}$. Let $F_{\sigma} : \mathbb{R}^{\Psi} \times \mathbb{R}_{\geq 0} \rightrightarrows \mathbb{R}^{\Psi}$ denote $K[w_{\sigma}](z,t)$. Then $F_{\sigma}(z,t) \subseteq F'_{\sigma}(z,t)$, where $F'_{\sigma} : \mathbb{R}^{\Psi} \times \mathbb{R}_{\geq 0} \rightrightarrows \mathbb{R}^{\Psi}$ is defined as $F'_{\sigma}(z,t) \triangleq [\{\sum_{j=0}^{k} \Lambda_{j,\sigma} \mathrm{vec}(\widetilde{V}_j) + \sum_{j=1}^{k} \Xi_{j,\sigma}\Delta_{j,\sigma} + f_e + \varepsilon_{\sigma}(x_d) - \rho(\|e\|)e - k_1 e\} - k_s K[\mathsf{sgn}](e); -K[\mathrm{proj}](\Gamma_0 \Lambda_{0,\sigma}^T e); \ldots; -K[\mathrm{proj}](\Gamma_k \Lambda_{k,\sigma}^T e)]$.

Theorem 2.1. *For the dynamical system in* (2–1)*, the controller in* (2–6) *and the adaptation laws in* (2–7) *and* (2–8) *ensure global asymptotic tracking error convergence in the sense that* $\lim_{t\to\infty} \|e(t)\| = 0$*,* $\forall (e(0), \hat{V}_0, \ldots, \hat{V}_k) \in \mathbb{R}^n \times \mathcal{B}_0 \times \ldots \times \mathcal{B}_k$*, provided the gain condition* $k_s > \overline{\varepsilon} + c$ *is satisfied.*

Proof. Consider the candidate common Lyapunov function $\mathcal{V}_L : \mathbb{R}^{\Psi} \to \mathbb{R}_{\geq 0}$ defined as

$$\mathcal{V}_L(z) \triangleq \frac{1}{2} e^T e + \frac{1}{2} \sum_{j=0}^{k} \mathrm{vec}(\widetilde{V}_j)^T \Gamma_j^{-1} \mathrm{vec}(\widetilde{V}_j), \tag{2–21}$$

which satisfies the inequality $\underline{\alpha} \|z\|^2 \leq \mathcal{V}_L(z) \leq \overline{\alpha} \|z\|^2$, where $\underline{\alpha}, \overline{\alpha} \in \mathbb{R}_{\geq 0}$ are known constants. Using [74, Def. 3], the generalized time-derivative of $\mathcal{V}_L$ can be computed as $\dot{\overline{\mathcal{V}}}_{\sigma}(z,t) \triangleq \max_{p \in \partial \mathcal{V}_L(z)} \max_{q \in F_{\sigma}(z,t)} p^T q$, where $\partial \mathcal{V}_L$ denotes the Clarke gradient of $\mathcal{V}_L$ defined in [75, p. 39]. Since $z \mapsto \mathcal{V}_L(z)$ is continuously differentiable, $\partial \mathcal{V}_L(z) = \{\nabla \mathcal{V}_L(z)\}$, where ∇ denotes the standard gradient operator. Thus,

$$\begin{aligned} \dot{\overline{\mathcal{V}}}_{\sigma}(z,t) &= \max_{q \in F_{\sigma}(z,t)} \left(\nabla \mathcal{V}_L(z)\right)^T q \\ &\overset{a.e.}{\leq} \max_{q \in F'_{\sigma}(z,t)} \left(\nabla \mathcal{V}_L(z)\right)^T q, \end{aligned}$$

[2] The assumption [74, Assumption 1] is equivalent to the assumption that ϱ is locally bounded. Since the switched system in (2–20) involves a finite number of subsystems, the assumption is always satisfied in this dissertation.

where the notation $\overset{a.e.}{(\cdot)}$ denotes that the relation $(\cdot)$ holds for almost all $t \in \mathbb{R}_{\geq 0}$. Additionally, using [73, Lemma E.1.IV][3] , the update law-based terms that appear after evaluating $\max_{q \in F'_\sigma(z,t)} (\nabla \mathcal{V}_L(z))^T q$ can be upper-bounded as

$$-\text{vec}(\tilde{V}_j)^T \Gamma_j^{-1} K\,[\text{proj}]\,(\Gamma_j \Lambda_{j,\sigma}^T e) \leq -\text{vec}(\widetilde{V}_j)^T \Lambda_{j,\sigma}^T e, \tag{2–22}$$

$\forall j \in \{0, \ldots, k\}$. Thus, evaluating $\max_{q \in F'_\sigma(z,t)} (\nabla \mathcal{V}_L(z))^T q$, using (2–22) and the fact that $e^T K\,[\text{sgn}]\,(e) = \{\|e\|_1\}$ yields

$$\dot{\tilde{\mathcal{V}}}_\sigma(z,t) \overset{a.e.}{\leq} e^T (f_e + \sum_{j=1}^{k} \Xi_{j,\sigma}\Delta_{j,\sigma} + \varepsilon_\sigma(x_d) - \rho(\|e\|)e - k_1 e) + \max \sum_{j=0}^{k} \{e^T \Lambda_{j,\sigma} \text{vec}(\widetilde{V}_j) - \text{vec}(\widetilde{V}_j)^T \Lambda_{j,\sigma}^T e\} - k_s \|e\|_1 . \tag{2–23}$$

Noting that $e^T \Lambda_{j,\sigma} \text{vec}(\widetilde{V}_j) = (e^T \Lambda_{j,\sigma} \text{vec}(\widetilde{V}_j))^T = \text{vec}(\widetilde{V}_j)^T \Lambda_{j,\sigma}^T e$ since they are scalar, the term $\max \sum_{j=0}^{k} \{e^T \Lambda_{j,\sigma} \text{vec}(\widetilde{V}_j) - \text{vec}(\widetilde{V}_j)^T \Lambda_{j,\sigma}^T e\} = 0$, $\forall \sigma \in \mathcal{N}$. Thus, substituting $\left\| \sum_{j=1}^{k} \Xi_j \Delta_j \right\| \leq c$, $\|\varepsilon_\sigma(x_d)\| \leq \bar{\varepsilon}$, and $e^T f_e \leq \rho(\|e\|)\|e\|^2$, (2–23) can be upper bounded as

$$\dot{\tilde{\mathcal{V}}}_\sigma(z,t) \overset{a.e.}{\leq} e^T (c + \bar{\varepsilon} - k_1 e) - k_s \|e\|_1 .$$

Using $-k_s \|e\|_1 \leq -k_s \|e\|$, and selecting k_s according to the theorem statement yields

$$\dot{\tilde{\mathcal{V}}}_\sigma(z,t) \overset{a.e.}{\leq} -k_1 \|e\|^2 . \tag{2–24}$$

By invoking [74, Theorem 2], $z \in \mathcal{L}_\infty$ and $\|e(t)\| \to 0$ as $t \to \infty$. Additionally, $z \in \mathcal{L}_\infty$ implies $\widetilde{V}_j, \hat{V}_j \in \mathcal{L}_\infty$, $\forall j \in \{0, \ldots, k\}$. Since Φ is a locally essentially bounded function, it follows that $\hat{\Phi}$ is bounded. Therefore, since all terms on the right hand side of (2–6) are bounded, it follows that $u \in \mathcal{L}_\infty$. Moreover, since ϕ_j and ϕ'_j are locally essentially

[3] The lemma says $-\tilde{\theta}^T \Gamma^{-1} \text{proj}(\mu) \leq -\tilde{\theta}^T \Gamma^{-1} \mu$. This property also holds after replacing $\text{proj}(\mu)$ with $K\,[\text{proj}]\,(\mu)$, since $K\,[\text{proj}]\,(\mu)$ evaluates as the set of convex combinations of $\text{proj}(\mu)$ and μ, whenever $\text{proj}(\mu)$ is discontinuous.

bounded functions for all $j \in \{0, \ldots, k\}$, it follows from (2–7) and (2–8) that $\dot{\hat{V}}_j \in \mathcal{L}_\infty$, $\forall j \in \{0, \ldots, k\}$. □

2.3 Simulations

Four simulation examples are provided to demonstrate the efficacy of the developed method, and the results are quantitatively compared with known baseline methods such as offline pre-training and output-layer adaptation [11]. The nonlinear system in (2–1) is considered with $f(x) = [x_1 x_2 \tanh(x_2) + \operatorname{sech}^2(x_1), \operatorname{sech}^2(x_1 + x_2) - \operatorname{sech}^2(x_2)]^T$, where $x = [x_1, x_2]^T$. The desired trajectory is $x_d(t) = [\sin(2t), -\cos(t)]^T$, the initial condition is $x(0) = [1, 2]^T$, the control gains are selected as $k_1 = 20$ and $k_s = 1$, and the bound for projection operator is selected as $\overline{V} = 5000$. The DNNs in the first and second examples, i.e., DNN1 and DNN2, consist of 6 layers, with 7 neurons in each layer; hence, there is a total of 231 individual weights in the first two examples. The DNNs in the third and fourth examples, i.e., DNN3 and DNN4, consist of 10 layers, with 30 neurons in each layer; hence, there is a total of 90150 individual weights in the third and fourth examples. Each simulation is performed for 10 seconds. To prevent the DNN term from having a large initial value, the inner and output layer weights are initialized as random values from the uniform distributions $U(0, 0.5)$ and $U(0, 0.01)$, respectively.

DNN1 and DNN3 contain LReLU activation functions given by $\varsigma(y) = y$ for $y \geq 0$, and $\varsigma(y) = 0.01y$, otherwise. The adaptation gain for DNN1 and DNN3 is selected using the switched rule: $\Gamma_j = 10 I_{L_j L_{j+1}}$, if $\left\| \left[\begin{array}{ccc} \text{vec}^T\left(\tilde{V}_0\right) & \ldots & \text{vec}^T\left(\tilde{V}_k\right) \end{array} \right]^T \right\| \leq 5$, and $\Gamma_j = I_{L_j L_{j+1}}$, otherwise, for all $j \in \{0, ..., k\}$. DNN2 and DNN4 contain hyperbolic tangent activation functions given by $\varsigma(y) = \tanh(y)$. Unlike LReLUs, saturating activation functions like hyperbolic tangents suffer from the vanishing gradient problem [4], i.e., the gradient terms in the update law vanish as the activation function saturates, which slows down the weight updates. To compensate for vanishing gradients and for a fair comparison with the LReLU-based DNNs, the adaptation gain for the hyperbolic

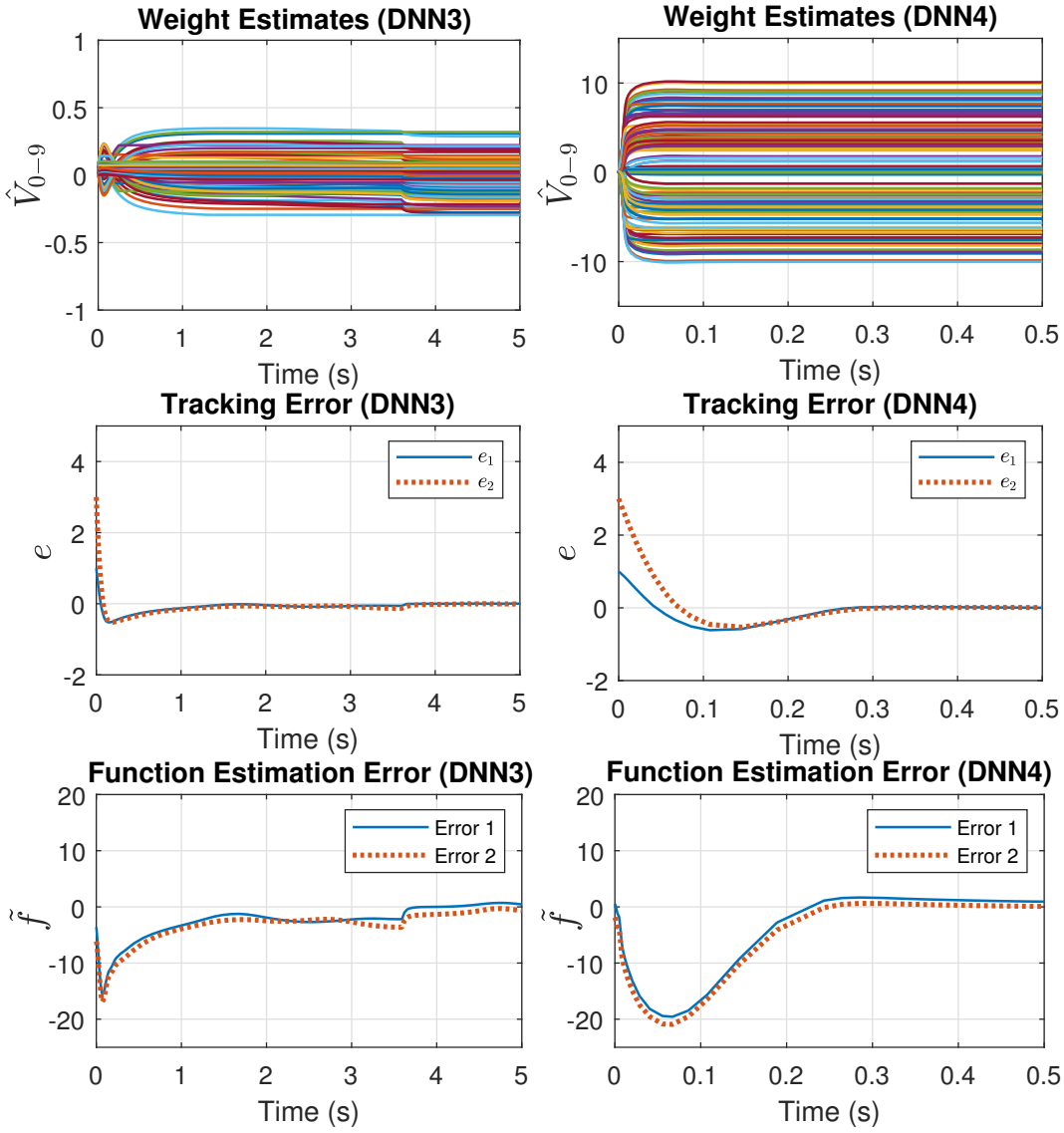

Figure 2-1. Plots of DNN weight estimates, tracking error, and function approximation error for DNN3 and DNN4. The simulation is performed for 10 seconds. For a better visualization of the transient performance, the plots for LReLU and tanh are shown for 5 and 0.5 seconds, respectively. Additionally, 150 arbitrarily selected weight estimates are shown out of the total 90150 weights for a tractable visualization.

Table 2-1. DNN Performance Comparison

Method	$\|e_{\text{RMS}}\|$	$\|e_{\text{RMS,SS}}\|$	$\left\|\tilde{f}_{\text{RMS,SS}}\right\|$
Developed (DNN1)	0.7014	0.0059	0.1204
Developed (DNN2)	1.2687	0.0108	0.2178
Developed (DNN3)	0.4326	0.0056	0.6824
Developed (DNN4)	0.3844	0.0063	0.7823
Offline (DNN1)	0.2952	0.0216	1.2165
[11] (DNN1)	0.6831	0.0105	0.2143

tangent activation function-based DNNs is selected with a relatively larger value of $\Gamma_j = 500 I_{L_j L_{j+1}}$.

Figure 2-1 shows the plots of DNN weight estimates, tracking error, and function estimation error for DNN3 and DNN4, where $\tilde{f} \triangleq f(x) - \hat{\Phi}$ denotes the function estimation error. The plots demonstrate that asymptotic convergence of the tracking error e is achieved in 0.5 s for both the examples. Table 2-1 provides a quantitative comparison of the developed method with offline training and output-layer adaptation [11], where e_{RMS} denotes the root mean square (RMS) of e over the time interval $[0, 10]$, and $e_{\text{RMS},SS}$ and $\tilde{f}_{\text{RMS,SS}}$ denote the RMS of e and $\tilde{f}$, respectively, over the time interval $[5, 10]$ (i.e., in steady state). For the simulations in Table 2-1, the robustifying term $k_s \text{sgn}(e)$ is removed to better quantitatively compare the effects of the DNN term. The offline pre-trained DNN is trained using data collected from 600 seconds of an *a priori* simulation. Using LReLUs yields improvement in the steady state tracking and function estimation performance as compared to hyperbolic tangent units as evident from the $\|e_{\text{RMS,SS}}\|$ and $\left\|\tilde{f}_{\text{RMS,SS}}\right\|$ values for DNN1 vs. DNN2 and DNN3 vs. DNN4. The developed method provides a decreased $\|e_{\text{RMS,SS}}\|$ but an increased $\|e_{\text{RMS}}\|$ as compared to offline pre-training or using adaptation for only the output-layer. This discrepancy is due to the initial overshoot in tracking error due to weight adaptation. The developed method provides a tenfold and twofold improvement in steady-state function estimation as compared to offline pre-training and output-layer adaptation, respectively.

2.4 Conclusion

A contribution in this chapter is the development of Lyapunov-based real-time weight update laws for each layer of a feedforward DNN. Additionally, the developed method also allows nonsmooth activation functions to be used in the DNN architecture. A nonsmooth Lyapunov-based stability analysis is provided to guarantee global asymptotic tracking error convergence. Simulation results are provided for a nonlinear system using DNNs involving leaky ReLU and hyperbolic tangent activation functions to demonstrate the efficacy of the developed method. Using LReLUs yields improvement in the steady-state tracking and function estimation performance when compared to hyperbolic tangent activation functions. Although adapting for more layers might cause initial overshoot in the tracking error, the developed method provides tenfold and twofold improvement in steady-state function estimation as compared to offline pre-training and output-layer adaptation, respectively.

CHAPTER 3
DEEP RESIDUAL NEURAL NETWORK (RESNET)-BASED ADAPTIVE CONTROL: A LYAPUNOV-BASED APPROACH

Deep Neural Network (DNN)-based controllers have emerged as a tool to compensate for unstructured uncertainties in nonlinear dynamical systems. Chapter 2 provided a Lyapunov-based approach to derive weight adaptation laws for each layer of a fully-connected feedforward DNN-based adaptive controller. However, deriving weight adaptation laws from a Lyapunov-based analysis remains an open problem for deep residual neural networks (ResNets). This chapter and my preliminary work [23] provide the first result on Lyapunov-derived weight adaptation for a ResNet-based adaptive controller. A nonsmooth Lyapunov-based analysis is provided to guarantee global asymptotic tracking error convergence. Comparative Monte Carlo simulations are provided to demonstrate the performance of the developed ResNet-based adaptive controller. The ResNet-based adaptive controller shows a 49.52% and 54.38% improvement in the tracking and function approximation performance, respectively, in comparison to a fully-connected DNN-based adaptive controller.

3.1 Control Design

We consider the same system and tracking control objective as in Chapter 2. The control objective is to design a ResNet-based adaptive controller that achieves global asymptotic tracking error convergence for the system in (2–1).

3.1.1 ResNet Architecture

The unknown drift vector field f can be approximated using a ResNet. A ResNet is modeled using building blocks that involve a shortcut connection across a fully-connected DNN [16]. Let $\Phi_p : \mathbb{R}^{L_{p,0}} \times \mathbb{R}^{L_{p,0}\times L_{p,1}} \times \ldots \times \mathbb{R}^{L_{p,k_p}\times L_{p,k_p+1}} \to \mathbb{R}^{L_{p,k_p+1}}$ denote the p^{th} fully-connected DNN block defined as $\Phi_p(\eta_p, V_{p,0}, \ldots, V_{p,k_p}) \triangleq \left(V_{p,k_p}^T \phi_{p,k_p} \circ \ldots \circ V_{p,1}^T \phi_{p,1}\right)\left(V_{p,0}^T \eta_p\right)$ for all $p \in \{1, \ldots, m\}$, where $\eta_p \in \mathbb{R}^{L_{p,0}}$ denotes the input of Φ_p, $k_p \in \mathbb{Z}_{>0}$ denotes the number of hidden layers in Φ_p, and $m \in \mathbb{Z}_{>0}$ denotes the number of building blocks. Additionally, $L_{p,j} \in \mathbb{Z}_{>0}$ denotes

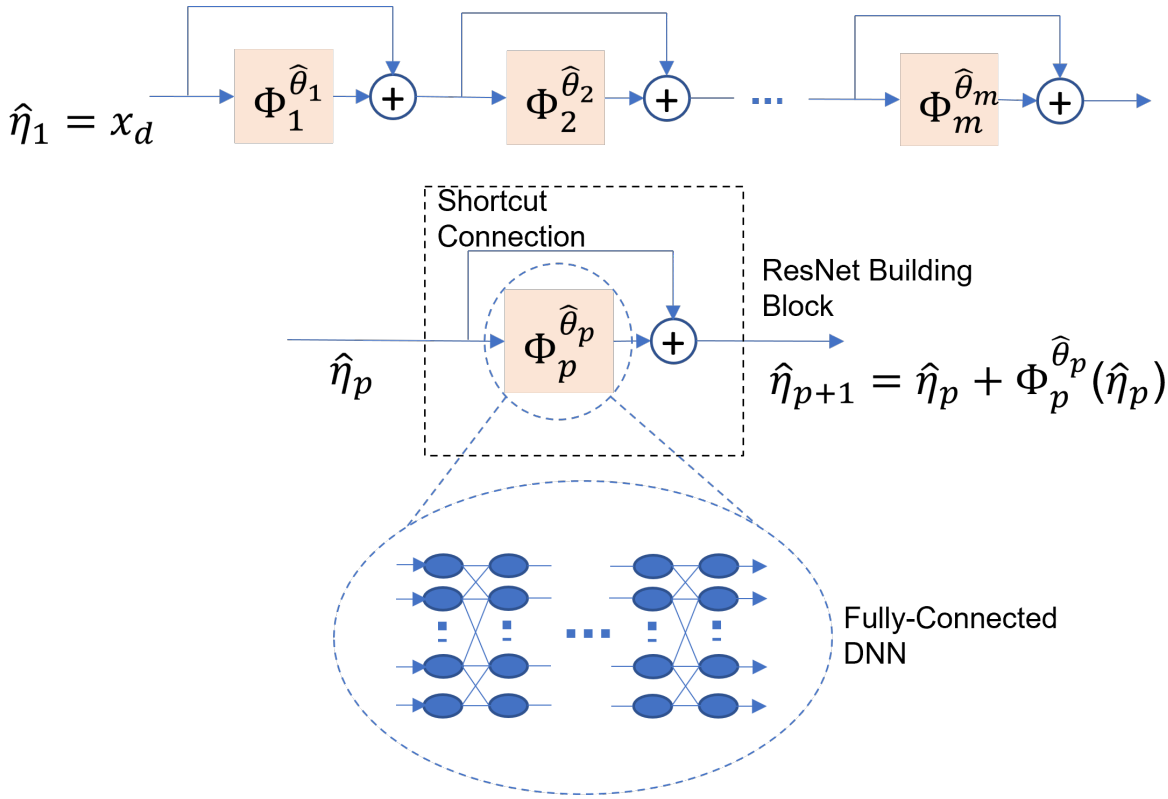

Figure 3-1. Illustration of the ResNet architecture in (3–2). The ResNet is shown at the top of the figure and is composed of building blocks that involve a shortcut connection across a fully-connected DNN component. The fully-connected DNN component for the p^{th} building block (bottom) is denoted by $\Phi_p^{\theta_p}$ for all $p \in \{1, \ldots, m\}$, where the input and the vector of weights of Φ_p are denoted by η_p and θ_p, respectively. Then the output of the p^{th} building block after considering the shortcut connection is represented by $\eta_{p+1} = \eta_p + \Phi_p^{\theta_p}(\eta_p)$ for all $p \in \{1, \ldots, m-1\}$, and the output of the ResNet is $\eta_m + \Phi_m^{\theta_m}(\eta_m)$.

the number of nodes, and $V_{p,j} \in \mathbb{R}^{L_{p,j} \times L_{p,j+1}}$ denotes the weight matrix in the j^{th} layer of Φ_p for all $(p,j) \in \{1,\ldots,m\} \times \{0,\ldots,k_p\}$. Similarly, $\phi_{p,j} : \mathbb{R}^{L_{p,j}} \to \mathbb{R}^{L_{p,j}}$ denotes a vector of smooth activation functions.[1] If the ResNet involves multiple types of activation functions at each layer, then $\phi_{p,j}$ may be represented as $\phi_{p,j} \triangleq \begin{bmatrix} \varsigma_{p,j,1} & \ldots & \varsigma_{p,j,L_{p,j}} \end{bmatrix}^T$, where $\varsigma_{p,j,i} : \mathbb{R} \to \mathbb{R}$ denotes the activation function at the i^{th} node of the j^{th} layer of Φ_p.[2] All the weights of Φ_p can be represented by the vector $\theta_p \triangleq \begin{bmatrix} \text{vec}(V_{p,0})^T & \ldots & \text{vec}(V_{p,k_p})^T \end{bmatrix}^T \in \mathbb{R}^{\Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}}$. The fully-connected block Φ_p can be expressed as $\Phi_p = V_{p,k_p}^T \varphi_{p,k_p}$, where $\varphi_{p,0} : \mathbb{R}^{L_{p,0}} \to \mathbb{R}^{L_{p,0}}$ and $\varphi_{p,j} : \mathbb{R}^{L_{p,0}} \times \mathbb{R}^{L_{p,0} \times L_{p,1}} \times \ldots \times \mathbb{R}^{L_{p,j-1} \times L_{p,j}} \to \mathbb{R}^{L_{p,j}} \; \forall j \in \{1,\ldots,k_p\}$ denote the recursive relation defined as

$$\varphi_{p,j} \triangleq \begin{cases} \phi_{p,j}\left(V_{p,j-1}^T \varphi_{p,j-1}\right), & j \in \{1,\ldots,k_p\}, \\ \eta_p, & j = 0. \end{cases} \tag{3–1}$$

The arguments of $\varphi_{p,j}$ are suppressed for notational brevity. Let $\phi'_{p,j} : \mathbb{R}^{L_{p,j}} \to \mathbb{R}^{L_{p,j} \times L_{p,j}}$ be defined as $\phi'_{p,j}(y) \triangleq \frac{\partial}{\partial y} \phi_{p,j}(y) \; \forall y \in \mathbb{R}^{L_{p,j}}$. The short-hand notation $\Phi_p^{\theta_p}(\eta_p) \triangleq \Phi_p(\eta_p, V_{p,0}, V_{p,1}, \ldots, V_{p,k_p})$ is defined for notational brevity in the subsequent development. Then the output of the p^{th} building block is given by $\eta_p + \Phi_p^{\theta_p}(\eta_p)$, where the addition of the input term η_p represents the shortcut connection across Φ_p. As shown in Figure 3-1, the ResNet $\Phi : \mathbb{R}^n \times \mathbb{R}^{\Sigma_{p=1}^{m} \Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}} \to \mathbb{R}^n$, which defines the mapping $(\eta_1, \theta) \mapsto \Phi^{\theta}(\eta_1)$, is modeled as [16]

$$\Phi^{\theta}(\eta_1) \triangleq \eta_m + \Phi_m^{\theta_m}(\eta_m), \tag{3–2}$$

[1] For the case of DNNs with nonsmooth activation functions (e.g., rectified linear unit (ReLU), leaky ReLU, maxout etc.), the reader is referred to [13] where a switched analysis is provided to account for the nonsmooth nature of activation functions. To better focus on our main contribution without loss of generality, we restrict our attention to smooth activation functions.

[2] Bias terms are omitted for simplicity of the notation.

where $\theta \triangleq \begin{bmatrix} \theta_1^T & \dots & \theta_m^T \end{bmatrix}^T \in \mathbb{R}^{\Sigma_{p=1}^{m}\Sigma_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$ denotes the vector of weights for the entire ResNet, and η_m is evaluated using the recursive relation

$$\eta_p = \begin{cases} \eta_{p-1} + \Phi_{p-1}^{\theta_{p-1}}(\eta_{p-1}), & p \in \{2, \dots, m\}, \\ x_d, & p = 1. \end{cases} \tag{3–3}$$

The recursive relation in (3–3) has valid dimensions under the constraint $L_{1,0} = L_{1,k_1+1} = L_{2,0} = L_{2,k_2+1} = \dots = L_{m,0} = L_{m,k_m+1} = n$. To facilitate the subsequent development, the following assumption is made.

Assumption 3.1. The function space of ResNets given by (3–2) is dense in $\mathcal{C}(\Omega)$, where $\mathcal{C}(\Omega)$ denotes the space of functions continuous over Ω.

Remark 3.1. Assumption 3.1 implies that ResNets satisfy the universal function approximation property that is well-known for various DNN architectures [70]. The universal function approximation property of ResNets is a common assumption that is widely used in the deep learning literature, and has been rigorously established for ResNets with specific activation functions in [76] and [77].

Consider any vector field $f \in \mathcal{C}(\Omega)$ and a prescribed accuracy $\bar{\varepsilon} \in \mathbb{R}_{>0}$. Then by Assumption 3.1, there exists a ResNet Φ with a corresponding vector of ideal weights $\theta^* \in \mathbb{R}^{\Sigma_{p=1}^{m}\Sigma_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$ such that $\sup_{x_d \in \Omega} \left\| f(x_d) - \Phi^{\theta^*}(x_d) \right\| \leq \bar{\varepsilon}$. Therefore, the drift vector field $x_d \mapsto f(x_d)$ can be modeled as

$$f(x_d) = \Phi^{\theta^*}(x_d) + \varepsilon(x_d), \tag{3–4}$$

where $\varepsilon : \mathbb{R}^n \to \mathbb{R}^n$ denotes an unknown function reconstruction error that can be bounded as $\sup_{x_d \in \Omega} \|\varepsilon(x_d)\| \leq \bar{\varepsilon}$.

Remark 3.2. In (3–4), the ResNet is used to approximate the drift vector field along the reference trajectory x_d instead of the actual trajectory x. As a result, the universal function approximation property holds since x_d lies within the compact set Ω by design. The

main benefit of drift compensation along the reference trajectory in this dissertation is that it facilitates a global asymptotic tracking result in the subsequent stability analysis.

To facilitate the subsequent analysis, the following assumption is made (cf., [71, Assumption 1]).

Assumption 3.2. There exists a known constant $\overline{\theta} \in \mathbb{R}_{>0}$ such that the unknown ideal ResNet weights can be bounded as $\|\theta^*\| \leq \overline{\theta}$.

3.1.2 Adaptation Laws

The ResNet-based model in (3–4) can be leveraged to approximate the unknown drift vector field f. However, since the ideal weights are unknown, adaptive weight estimates are developed. The adaptive weight estimate for the j^{th} layer of Φ_p is denoted by $\hat{V}_{p,j} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{L_{p,j} \times L_{p,j+1}}$ $\forall (p,j) \in \{1,\ldots,m\} \times \{0,\ldots,k_p\}$. The weight estimate for the p^{th} building block $\hat{\theta}_p : \mathbb{R}_{\geq 0} \to \mathbb{R}^{\Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}}$ is defined as $\hat{\theta}_p = \left[\text{vec}(\hat{V}_{p,0})^T, \ldots, \text{vec}(\hat{V}_{p,k_p})^T\right]^T$ for all $p \in \{1,\ldots,m\}$, the weight estimate for the ResNet $\hat{\theta} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{\Sigma_{p=1}^{m} \Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}}$ is defined as $\hat{\theta} \triangleq \begin{bmatrix} \hat{\theta}_1^T & \ldots & \hat{\theta}_m^T \end{bmatrix}^T$, and the ResNet-based adaptive estimate of $f(x_d)$ $\forall x_d \in \Omega$ is denoted by $\Phi^{\hat{\theta}}(x_d)$. The weight estimation error $\tilde{\theta} : \mathbb{R}_{\geq 0} \to \mathbb{R}^{\Sigma_{p=1}^{m} \Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}}$ is defined as $\tilde{\theta} \triangleq \theta^* - \hat{\theta}$. Based on the subsequent stability analysis, the adaptation law for the weight estimates of the ResNet in (3–2) is designed as

$$\dot{\hat{\theta}} \triangleq \text{proj}\left(\Gamma \Phi'^T e\right), \tag{3–5}$$

where $\Gamma \in \mathbb{R}^{\Sigma_{p=1}^{m} \Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1} \times \Sigma_{p=1}^{m} \Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}}$ denotes a positive-definite adaptation gain matrix, and $\Phi' \in \mathbb{R}^{n \times \Sigma_{p=1}^{m} \Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}}$ is a short-hand notation denoting the $\Phi' \triangleq \frac{\partial \Phi^{\hat{\theta}}(x_d)}{\partial \hat{\theta}}$. In (3–5), $\text{proj}(\cdot)$ denotes the projection operator defined in [73, Appendix E, Eq. E.4], which is used to ensure $\hat{\theta}(t) \in \mathcal{B} \triangleq \{\theta \in \mathbb{R}^{\Sigma_{p=1}^{m} \Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}} : \|\theta\| \leq \overline{\theta}\}$ $\forall t \in \mathbb{R}_{\geq 0}$. The term Φ' can be evaluated as follows. Let $\hat{\eta}_p \in \mathbb{R}^{L_{p,0}}$ be defined as

$$\hat{\eta}_p = \begin{cases} x_d, & p = 1, \\ \hat{\eta}_{p-1} + \Phi_{p-1}^{\hat{\theta}_{p-1}}\left(\hat{\eta}_{p-1}\right), & p \in \{2,\ldots,m\}. \end{cases} \tag{3–6}$$

Then, it follows that $\Phi^{\hat{\theta}}(x_d) = \hat{\eta}_m + \Phi_m^{\hat{\theta}_m}(\hat{\eta}_m)$. To facilitate the subsequent development, the short-hand notations $\Phi'_p \triangleq \left(\frac{\partial \Phi^{\hat{\theta}}(x_d)}{\partial \hat{\theta}_p}\right)$, $\Lambda_p \triangleq \frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \hat{\theta}_p}$, $\Lambda_{p,j} \triangleq \frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \mathrm{vec}(\hat{V}_{p,j})}$, and $\Xi_p \triangleq \frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \hat{\eta}_p}$ are introduced. Then $\Phi' = \left[\left(\frac{\partial \Phi^{\hat{\theta}}(x_d)}{\partial \hat{\theta}_1}\right), \ldots, \left(\frac{\partial \Phi^{\hat{\theta}}(x_d)}{\partial \hat{\theta}_m}\right)\right]$ can be expressed as

$$\Phi' \triangleq \begin{bmatrix} \Phi'_1, & \ldots, & \Phi'_m \end{bmatrix}. \tag{3–7}$$

Using the chain rule, the term Φ'_p can be computed as

$$\Phi'_p = \left(\prod_{l=p+1}^{\overset{\curvearrowleft}{m}} (I_n + \Xi_l)\right) \Lambda_p, \ \forall p \in \{1, \ldots, m\}. \tag{3–8}$$

In (3–8), the terms Λ_p and Ξ_p, for all $p \in \{1, \ldots, m\}$, can be computed as follows. Since $\hat{\theta}_p = \left[\mathrm{vec}(\hat{V}_{p,0})^T, \ldots, \mathrm{vec}(\hat{V}_{p,k_p})^T\right]^T$, it follows that $\frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \hat{\theta}_p} = \left[\left(\frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \mathrm{vec}(\hat{V}_{p,0})}\right), \ldots, \left(\frac{\partial \Phi_p^{\hat{\theta}_p}(\hat{\eta}_p)}{\partial \mathrm{vec}(\hat{V}_{p,k_p})}\right)\right]$. Therefore, using the definitions of Λ_p and $\Lambda_{p,j}$ yields

$$\Lambda_p = \begin{bmatrix} \Lambda_{p,0} & \Lambda_{p,1} & \ldots & \Lambda_{p,k_p} \end{bmatrix}, \forall p \in \{1, \ldots, m\}. \tag{3–9}$$

For brevity in the subsequent development, the short-hand notations $\hat{\varphi}_{p,j} \triangleq \varphi_{p,j}(\hat{\eta}_p, \hat{V}_{p,0}, \ldots, \hat{V}_{p,j})$ and $\hat{\varphi}'_{p,j} \triangleq \varphi'_{p,j}(\hat{\eta}_p, \hat{V}_{p,0}, \ldots, \hat{V}_{p,j})$ are introduced. Using (3–1), the chain rule, and the property of vectorization operators in (1–3), the terms $\Lambda_{p,0}$ and $\Lambda_{p,j}$ in (3–9) can be computed as

$$\Lambda_{p,0} = \left(\prod_{l=1}^{\overset{\curvearrowleft}{k_p}} \hat{V}_{p,l}^T \hat{\varphi}'_{p,l}\right) (I_{L_{p,1}} \otimes \hat{\eta}_p^T), \tag{3–10}$$

and

$$\Lambda_{p,j} = \left(\prod_{l=j+1}^{\overset{\curvearrowleft}{k_p}} \hat{V}_{p,l}^T \hat{\varphi}'_{p,l}\right) (I_{L_{p,j+1}} \otimes \hat{\varphi}_{p,j}^T), \tag{3–11}$$

for all $(p, j) \in \{1, \ldots, m\} \times \{1, \ldots, k_p\}$, respectively. Similarly, the term Ξ_p can be computed as

$$\Xi_p = \left(\prod_{l=1}^{\overset{\curvearrowleft}{k_p}} \hat{V}_{p,l}^T \hat{\varphi}'_{p,l} \right) \hat{V}_{p,0}^T, \forall p \in \{1, \ldots, m\}. \tag{3–12}$$

Remark 3.3. If Φ_p suffers from the vanishing gradient problem, i.e., $\|\Xi_l\|_F \approx 0$ for all $l \in \{p+1, \ldots, m\}$, then $\Phi'_p = \left(\prod_{l=p+1}^{\overset{\curvearrowleft}{m}} (I_n + \Xi_l) \right) \Lambda_p \approx \Lambda_p$. For an equivalent fully-connected DNN, i.e., in absence of shortcut connections, $\left\|\Phi'_p\right\|_F = \left\| \left(\prod_{l=p+1}^{\overset{\curvearrowleft}{m}} \Xi_l \right) \Lambda_p \right\| \approx 0$. Thus, the shortcut connection circumvents the vanishing gradient problem in the ResNet when Φ_p has a vanishing gradient.

3.1.3 Control Law Development

Let the function $f_e : \mathbb{R}^n \times \Omega \to \mathbb{R}^n$ be defined as $f_e \triangleq f(x) - f(x_d)$. Using [78, Appendix A], the function $(x, x_d) \mapsto f_e$ can be bounded as $\|f_e\| \leq \rho(\|e\|)\|e\|$, for all $x \in \mathbb{R}^n$ and $x_d \in \Omega$, where $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes a known strictly increasing function. Based on the subsequent stability analysis, the control input is designed as

$$u \triangleq \dot{x}_d - \Phi^{\hat{\theta}}(x_d) - \rho(\|e\|)e - \sigma_e e - \sigma_s \text{sgn}(e), \tag{3–13}$$

where $\sigma_e, \sigma_s \in \mathbb{R}_{>0}$ are constant control gains, and $\text{sgn}(\cdot)$ denotes the vector signum function.

Taking the time-derivative of (2–2), substituting in (2–1) and (3–13), adding and subtracting $f(x_d)$, and substituting in (3–4) yields the closed-loop error system

$$\dot{e} = f_e + \Phi^{\theta^*}(x_d) - \Phi^{\hat{\theta}}(x_d) + \varepsilon(x_d) - \rho(\|e\|)e - \sigma_e e - \sigma_s \text{sgn}(e). \tag{3–14}$$

The ResNet in (3–2) is nonlinear in terms of the weights. Adaptive control design for nonlinearly parameterized systems is known to be a difficult problem [79]. A number of adaptive control methods have been developed to address the challenges posed

by a nonlinear parameterization [13, 79–85]. In particular, first-order Taylor series approximation-based techniques have shown promising results for neural network-based adaptive controllers [13, 85, 86]. Specifically, the result in [13] uses a first-order Taylor series approximation to derive weight adaptation laws for a fully-connected DNN-based adaptive controller. Thus, motivation exists to explore a Taylor series approximation-based design to derive adaptation laws for the ResNet. For the ResNet in (3–2), a first-order Taylor series approximation-based error model is given by [71, Eq. 22]

$$\Phi^{\theta^*}(x_d) - \Phi^{\hat{\theta}}(x_d) = \Phi' \tilde{\theta} + \mathcal{O}^2\left(\left\|\tilde{\theta}\right\|\right), \tag{3–15}$$

where $\mathcal{O}^2\left(\left\|\tilde{\theta}\right\|\right)$ denotes higher-order terms. Since $\left\|\hat{\theta}\right\| \leq \bar{\theta}$ by the use of projection operator, it follows from Assumption 3.2 that $\left\|\tilde{\theta}\right\| \leq \|\theta^*\| + \left\|\hat{\theta}\right\| \leq 2\overline{\theta}$. Due to the facts that the ResNet is smooth, $x_d \in \Omega$, and $\left\|\tilde{\theta}\right\| \leq 2\overline{\theta}$, there exists a known constant $\overline{\Delta} \in \mathbb{R}_{>0}$ such that $\left\|\mathcal{O}^2\left(\left\|\tilde{\theta}\right\|\right)\right\| \leq \overline{\Delta}$ (cf., [13, Eq. 18]). Then, substituting (3–15) into (3–14), the closed-loop error system can be expressed as

$$\dot{e} = f_e + \Phi' \tilde{\theta} + \mathcal{O}^2\left(\left\|\tilde{\theta}\right\|\right) + \varepsilon(x_d) - \rho(\|e\|)e - \sigma_e e - \sigma_s \text{sgn}(e). \tag{3–16}$$

To facilitate the subsequent analysis, let $z \triangleq [e^T, \tilde{\theta}^T]^T \in \mathbb{R}^\Psi$ denote a concatenated state, where $\Psi \triangleq n + \Sigma_{p=1}^{m} \Sigma_{j=0}^{k_p} L_{p,j} L_{p,j+1}$. Then, using (3–5) and (3–16) yields

$$\dot{z} = h(z,t), \tag{3–17}$$

where $h : \mathbb{R}^\Psi \times \mathbb{R}_{\geq 0} \to \mathbb{R}^\Psi$ is defined as

$$h(z,t) \triangleq \begin{bmatrix} \begin{pmatrix} f_e + \Phi' \tilde{\theta} + \mathcal{O}^2\left(\left\|\tilde{\theta}\right\|\right) + \varepsilon(x_d) \\ -\rho(\|e\|)e - \sigma_e e - \sigma_s \text{sgn}(e) \end{pmatrix} \\ -\text{proj}\left(\Gamma \Phi'^T e\right) \end{bmatrix}. \tag{3–18}$$

3.2 Stability Analysis

Based on the nonsmooth analysis technique in [66], the following theorem establishes the invariance properties of Filippov solutions to (3–17) and provides guarantees of global asymptotic tracking error convergence for the system in (2–1).

Theorem 3.1. *For the dynamical system in* (2–1)*, the controller in* (3–13) *and the adaptation law in* (3–5) *ensure global asymptotic tracking error convergence in the sense that* $z, u, \dot{\hat{\theta}} \in \mathcal{L}_\infty$ *and* $\lim_{t\to\infty} \|e(t)\| = 0$*, provided Assumptions 3.1 and 3.2 hold, and the following gain condition is satisfied:*

$$\sigma_s \quad > \quad \overline{\varepsilon} + \overline{\Delta}. \tag{3–19}$$

Proof. Consider the candidate Lyapunov function $\mathcal{V}_L : \mathbb{R}^{\Psi} \to \mathbb{R}_{\geq 0}$ defined as

$$\mathcal{V}_L(z) \quad \triangleq \quad \frac{1}{2} e^T e + \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}, \tag{3–20}$$

which satisfies the inequality $\underline{\alpha} \|z\|^2 \leq \mathcal{V}_L(z) \leq \overline{\alpha} \|z\|^2$, where $\underline{\alpha}, \overline{\alpha} \in \mathbb{R}_{\geq 0}$ are known constants. Let $\partial \mathcal{V}_L$ denote the Clarke gradient of $\mathcal{V}_L$ defined in [75, p. 39]. Since $z \mapsto \mathcal{V}_L(z)$ is continuously differentiable, $\partial \mathcal{V}_L(z) = \{\nabla \mathcal{V}_L(z)\}$, where ∇ denotes the standard gradient operator. Based on (3–18) and the chain rule in [87, Thm 2.2], it can be verified that $t \to \mathcal{V}_L(z(t))$ satisfies the differential inclusion

$$\begin{aligned}
\dot{\mathcal{V}}_L \quad &\overset{a.a.t.}{\in} \quad \bigcap_{\xi \in \partial \mathcal{V}_L(z)} \xi^T K[h](z,t) \\
&= \quad \nabla \mathcal{V}_L(z)^T K[h](z,t) \\
&= \quad e^T \left(f_e + \mathcal{O}^2\left(\left\|\tilde{\theta}\right\|\right) + \varepsilon(x_d) - \rho(\|e\|) e \right) - \sigma_e \|e\|^2 - \sigma_s e^T K[\text{sgn}](e) \\
&\quad + e^T \Phi' \tilde{\theta} - \tilde{\theta}^T \Gamma^{-1} K[\text{proj}]\left(\Gamma \Phi'^T e\right).
\end{aligned} \tag{3–21}$$

Using [73, Lemma E.1.IV] and the fact that $K[\text{proj}](\cdot)$ is the set of convex combinations of $\text{proj}(\cdot)$ and $(\cdot)$, the term with the projection operator in (3–21) can be bounded as

$$-\tilde{\theta}^T \Gamma^{-1} K[\text{proj}]\left(\Gamma \Phi'^T e\right) \quad \leq \quad -\tilde{\theta}^T \Phi'^T e. \tag{3–22}$$

Using (3–22) and the facts that $e^T K\,[\text{sgn}]\,(e) = \|e\|_1$ and $e^T f_e \leq \rho(\|e\|)\,\|e\|^2$, (3–21) can be bounded as

$$\dot{\mathcal{V}}_L \overset{a.a.t.}{\leq} -\sigma_e \|e\|^2 + e^T \left(\mathcal{O}^2\left(\left\|\tilde{\theta}\right\|\right) + \varepsilon(x_d)\right) - \sigma_s \|e\|_1 . \tag{3–23}$$

Based on Holder's inequality, triangle inequality, and the fact that $\|e\| \leq \|e\|_1$, the following inequality can be obtained: $e^T \left(\mathcal{O}^2\left(\left\|\tilde{\theta}\right\|\right) + \varepsilon(x_d)\right) \leq \|e\|_1 \left(\left\|\mathcal{O}^2\left(\left\|\tilde{\theta}\right\|\right)\right\| + \|\varepsilon(x_d)\|\right) \leq \left(\bar{\varepsilon} + \overline{\Delta}\right)\|e\|_1$. Then, provided the gain condition in (3–19) is satisfied, the right-hand side of (3–23) can be upper-bounded as

$$\dot{\mathcal{V}}_L \overset{a.a.t.}{\leq} -\sigma_e \|e\|^2 . \tag{3–24}$$

Based on (3–24), invoking [66, Corollary 1] yields $z \in \mathcal{L}_\infty$ and $\lim_{t\to\infty} \|e(t)\| = 0$. Additionally, due to the facts that $(x_d, \hat{\theta}) \to \Phi^{\hat{\theta}}(x_d)$ is smooth, $x_d \in \Omega$, and $\hat{\theta} \in \mathcal{B}$, it follows that $\Phi^{\hat{\theta}}(x_d)$ is bounded. Since each term on the right-hand side of (3–13) is bounded, the control input $u \in \mathcal{L}_\infty$. Since $\phi_{p,j}$ and $\phi'_{p,j}$ are smooth for all $(p, j) \in \{1, \ldots, m\} \times \{0, \ldots, k_p\}$, it follows from (3–7)-(3–12) that Φ' is bounded. Then, every term on the right-hand side of (3–5) is bounded, and hence, $\dot{\hat{\theta}}$ is bounded. □

3.3 Simulations

Monte Carlo simulations are provided to demonstrate the performance of the developed ResNet-based adaptive controller, and the results are compared with a fully-connected DNN-based adaptive controller [13]. The system in (2–1) is considered with the state dimension $n = 10$. The unknown drift vector field in (2–1) is modeled as $f(x) = Ay(x)$, where $A \in \mathbb{R}^{n\times 6n}$ is a random matrix with all elements belonging to the uniform random distribution $U(0, 0.1)$, and $y(x) \triangleq [x^T, \tanh(x)^T, \sin(x)^T, \text{sech}(x)^T, (x \odot x)^T, (x \odot x \odot x)^T]$, where $\odot$ denotes the element-wise product operator. All elements of the initial state $x(0)$ are selected from the distribution $U(0, 2)$. The reference trajectory is selected as $x_d(t) = [0.5 + \sin(\omega_1 t), \ldots, 0.5 + \sin(\omega_n t)]$, where $\omega_1, \ldots, \omega_n \sim U(0, 20)$. The configuration

Table 3-1. Performance Comparison

Architecture	$\|e_{\text{rms}}\|$	$\left\|\tilde{f}_{\text{rms}}\right\|$	$\|u_{\text{rms}}\|$
ResNet	0.420	4.265	24.290
Fully-Connected	0.832	9.348	24.711

of the ResNet in (3–2) is selected with 20 hidden layers, a shortcut connection across each hidden layer, and 10 neurons in each layer. The hyperbolic tangent activation function is used in each node of the ResNet. The results are compared with an equivalent fully-connected DNN-based adaptive control, i.e., the same configuration as the ResNet but without shortcut connections. The control and adaptation gains are selected as $\sigma_e = 2$, $\sigma_s = 2$, and $\Gamma = I_{\Sigma_{p=1}^{m}\Sigma_{j=0}^{k_p} L_{p,j}L_{p,j+1}}$. The robust state-feedback term $-\rho(\|e\|)e$ is designed with $\rho(\|e\|) = 0.1\left(\|e\| + \|e\|^2\right)$. The bound on the projection operator is selected as $\bar{\theta} = 10,000$.

The performance of both the ResNet and the fully-connected DNN-based adaptive controller is sensitive to initial weights. To account for the sensitivity of performance to weight initialization, the initial weights for each method are obtained using a Monte Carlo method. In the Monte Carlo method, 10,000 simulations are performed, where the initial weights in each simulation are selected from $U(-0.05, 0.05)$, and the cost $J = \int_0^{10} \left(e^T(t)Qe(t) + u^T(t)Ru(t)\right) dt$ is evaluated in each simulation with $Q = I_{10}$ and $R = 0.01I_{10}$. For a fair comparison between the ResNet and the fully-connected DNN, the simulation results yielding the least J for each architecture are compared.

Table 3-1 provides the norm of the root mean square (RMS) tracking error, function approximation error, and control input given by $\|e_{\text{rms}}\|$, $\left\|\tilde{f}_{\text{rms}}\right\|$, and $\|u_{\text{rms}}\|$, respectively. In comparison to the fully-connected DNN, the ResNet shows 49.52% and 54.38% decrease in the norms of the tracking and function approximation errors, respectively. As shown in Figure 3-2, the fully-connected DNN exhibits a comparatively poor tracking and function approximation performance. As mentioned in Remark 3.3, fully-connected DNNs suffer from the vanishing gradient problem. Thus the fully-connected DNN

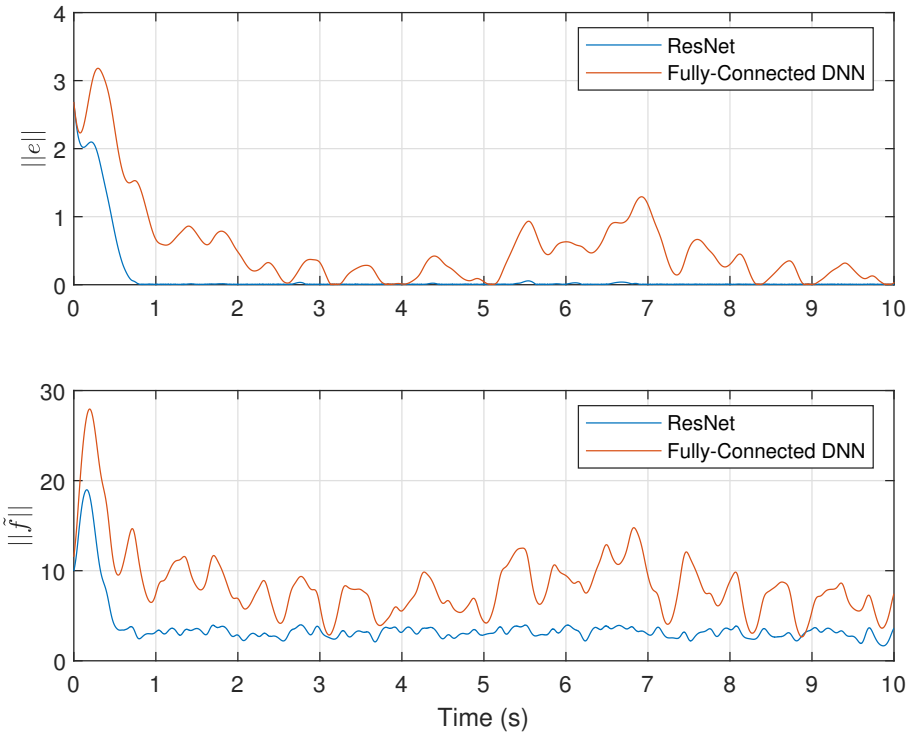

Figure 3-2. Plots of the tracking error norm $\|e\|$ and function approximation error norm $\left\|\tilde{f}\right\|$ with ResNet and fully-connected DNN-based adaptive controller.

weights remain approximately constant as shown in Figure 3-3. Consequently, the fully-connected DNN-based feedforward term fails to compensate for the uncertainty in the system which yields a relatively poor tracking and function approximation. In contrast to the fully-connected DNN, the presence of shortcut connections in the ResNet eliminates the vanishing gradient problem as mentioned in Remark 3.3. As a result, the ResNet weights are able to compensate for the system uncertainty as shown in Figure 3-3 which yields improved tracking and function approximation performance. Additionally, the ResNet requires approximately the same control effort as the fully-connected DNN. Therefore, the ResNet improves the tracking performance without requiring a higher control effort in comparison to the fully-connected DNN.

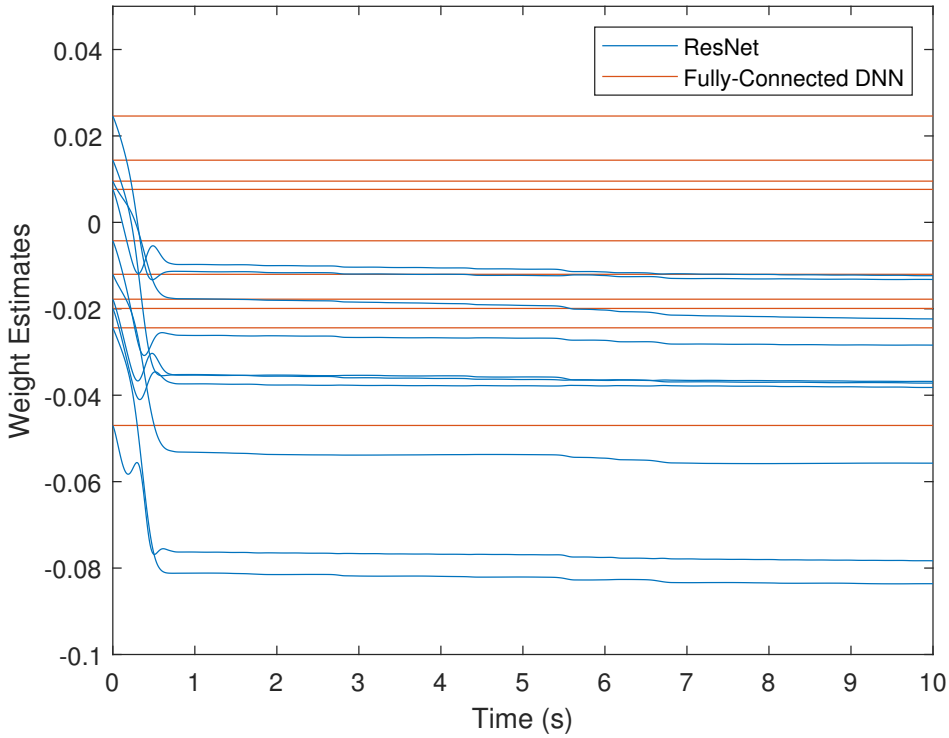

Figure 3-3. Plot of the weight estimates of the ResNet and fully-connected DNN. There are a total of 2,000 individual weights in each architecture. For better visualization, 10 arbitrarily selected weights are shown. The fully-connected DNN weights adapt slowly due to the problem of vanishing gradients. However, the ResNet weights are able to adapt faster since the ResNet does not have vanishing gradients.

3.4 Conclusion

A contribution of this chapter is Lyapunov-derived adaptation laws for the weights of each layer of a ResNet-based adaptive controller. A nonsmooth Lyapunov-based analysis is provided to guarantee global asymptotic tracking error convergence. Comparative Monte Carlo simulations are provided to demonstrate the performance of the developed ResNet-based adaptive controller. The developed ResNet-based adaptive controller provides 49.52% and 54.38% improvement in the tracking and function approximation performance, respectively, in comparison to an equivalent fully-connected DNN-based adaptive controller. Additionally, the ResNet overcomes the vanishing gradient problem present in the fully-connected DNN.

Future work may involve the application of ResNets for online system identification and state estimation problems. Additionally, more complicated architectures of ResNets such as recurrent ResNets can be explored for control applications.

CHAPTER 4
ADAPTIVE CONTROL OF TIME-VARYING PARAMETER SYSTEMS WITH ASYMPTOTIC TRACKING

A continuous adaptive controller is developed for nonlinear dynamical systems with linearly parameterizable uncertainty involving time-varying uncertain parameters. Through a unique stability analysis strategy, a new adaptive feedforward term is developed along with specialized feedback terms, to yield an asymptotic tracking error convergence result by compensating for the time-varying nature of the uncertain parameters. A Lyapunov-based stability analysis is shown for Euler-Lagrange systems, which ensures asymptotic tracking error convergence and boundedness of the closed-loop signals. Additionally, the time-varying uncertain function approximation error is shown to converge to zero. A simulation example of a two-link manipulator is provided to demonstrate the asymptotic tracking result.

4.1 Dynamic Model

The subsequent development is based on the general uncertain nonlinear Euler-Lagrange (EL) dynamics given by [88, Section 2.2]

$$M(q(t),t)\ddot{q}(t) + V_m(q(t),\dot{q}(t),t)\dot{q}(t) + G(q(t),t) + F(\dot{q}(t),t) + \tau_d(t) \quad = \quad \tau(t), \quad \text{(4–1)}$$

where $t \in [t_0, \infty)$ denotes time, $t_0 \in \mathbb{R}_{\geq 0}$ denotes the initial time, $q : [t_0, \infty) \to \mathbb{R}^n$ denotes a vector of generalized positions, $M : \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}^{n \times n}$ denotes a generalized inertia matrix, $V_m : \mathbb{R}^n \times \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}^{n \times n}$ denotes the Coriolis and centrifugal forces matrix, $G : \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}^n$ denotes a generalized vector of potential forces, $F : \mathbb{R}^n \times [t_0, \infty) \to \mathbb{R}^n$ denotes a generalized vector of dissipation, $\tau_d : [t_0, \infty) \to \mathbb{R}^n$ represents an exogenous disturbance acting on the system, and $\tau : [t_0, \infty) \to \mathbb{R}^n$ represents a generalized control input vector [88, Chapter 2]. The subsequent development is based on the assumption that only $q(t), \dot{q}(t)$ are measurable. The following assumptions about the EL system are made in the subsequent development [88, Sec. 2.3].

Assumption 4.1. The inertia matrix satisfies $m_1 \left\|\xi\right\|^2 \leq \xi^T M(q(t),t)\xi \leq \bar{m}(q) \left\|\xi\right\|^2 \; \forall \, \xi \in \mathbb{R}^n$, where $m_1 \in \mathbb{R}_{>0}$ is a known bounding constant, $\bar{m} : \mathbb{R}^n \to \mathbb{R}_{>0}$ is a known bounding function, and $\|\cdot\|$ denotes the Euclidean norm for a vector argument or the spectral norm for a matrix argument.

Assumption 4.2. The functions $M(q(t),t)$, $V_m(q(t),\dot{q}(t),t)$, $G(q(t),t)$ and $F(\dot{q}(t),t)$ are second order differentiable such that their second time derivatives are bounded if $q^{(i)} \in \mathcal{L}_\infty \; \forall \, i = 0,1,2,3$, where $\mathcal{L}_\infty$ denotes the space of essentially bounded Lebesgue measurable functions.

Assumption 4.3. The dynamics in (4–1) can be linearly parameterized[1] as

$$\begin{aligned} Y_p(q(t),\dot{q}(t),\ddot{q}(t),t)\theta_p(t) &= M(q(t),t)\ddot{q}(t) + F(\dot{q}(t),t) + G(q(t),t) \\ &\quad + V_m(q(t),\dot{q}(t),t)\dot{q}(t), \end{aligned} \tag{4–2}$$

where $Y_p : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times [t_0,\infty) \to \mathbb{R}^{n\times m}$ is a known regression matrix, and $\theta_p : [t_0,\infty) \to \mathbb{R}^m$ is a vector of time-varying unknown parameters.

The disturbance parameter vector $\tau_d(t)$ can be appended to the $\theta_p(t)$ vector, yielding an augmented parameter vector $\theta : [t_0,\infty) \to \mathbb{R}^{n+m}$ as

$$\theta(t) \triangleq \begin{bmatrix} \theta_p^T(t) & \tau_d^T(t) \end{bmatrix}^T, \tag{4–3}$$

and the augmented regressor $Y : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times [t_0,\infty) \to \mathbb{R}^{n\times(n+m)}$ can be designed as

$$Y(q(t),\dot{q}(t),\ddot{q}(t),t) \triangleq \begin{bmatrix} Y_p(q(t),\dot{q}(t),\ddot{q}(t),t) & I_n \end{bmatrix}. \tag{4–4}$$

[1] A linear parameterization is considered for simplicity. For systems that do not satisfy the linear-in-the-parameters assumption, the parameterization can yet be linearized according to [84, Equation 7], where the linearization error can be upper-bounded using [84, Lemma 1]. Subsequently, the adaptive design approach of this dissertation is then applicable.

Substituting the parameterization in (4–2)-(4–4) into (4–1) yields

$$M(q(t),t)\ddot{q}(t) + F(\dot{q}(t),t) + V_m(q(t),\dot{q}(t),t)\dot{q}(t) + G(q(t),t) + \tau_d(t) = Y(q(t),\dot{q}(t),\ddot{q}(t),t)\theta(t), \tag{4–5}$$

where $Y(q(t),\dot{q}(t),\ddot{q}(t),t)\theta(t) = \tau(t)$.

Assumption 4.4. The time-varying augmented parameter $\theta(t)$ and its time-derivatives, i.e., $\dot{\theta}(t)$, $\ddot{\theta}(t)$ are bounded by known constants, i.e., $\|\theta(t)\| \leq \zeta_0$, $\left\|\dot{\theta}(t)\right\| \leq \zeta_1$, and $\left\|\ddot{\theta}(t)\right\| \leq \zeta_2$, where $\zeta_0, \zeta_1, \zeta_2 \in \mathbb{R}_{>0}$ are known bounding constants.

Remark 4.1. For practical applications, it is often not difficult to develop sufficiently large bounds on uncertain parameters or their rate of change. For example, variation in a friction coefficient due to wear is difficult to model, but it is not difficult to obtain an upper-bound on the friction coefficient. Similarly, it is possible to develop an upper bound on the inertia and drag coefficient parameters of an aircraft. The reader is referred to the result in [89, Sec. 4] for an example of an aerospace system with bounded time-varying parameters. For systems with unknown bounds, robust adaptive control methods such as [32, Section IV] may provide insight for a solution, but such an extension is beyond the scope of the contributions of this dissertation.

4.2 Control Design

4.2.1 Control Objective

The objective is to design a controller such that the state tracks a smooth bounded reference trajectory, despite the time-varying nature of the uncertain parameters. The objective is quantified by defining the tracking error $e_1 : [t_0, \infty) \to \mathbb{R}^n$ as[2]

$$e_1 \triangleq q - q_d, \tag{4–6}$$

[2] Time-dependency is suppressed for the sake of brevity, except where explicit time-dependency adds clarity.

where $q_d : [t_0, \infty) \to \mathbb{R}^n$ is a desired trajectory. To facilitate the subsequent analysis, filtered tracking errors $e_2, r : [t_0, \infty) \to \mathbb{R}^n$ are defined as

$$e_2 \triangleq \dot{e}_1 + \alpha_1 e_1, \tag{4–7}$$

$$r \triangleq \dot{e}_2 + \alpha_2 e_2, \tag{4–8}$$

respectively, where $\alpha_1, \alpha_2 \in \mathbb{R}_{>0}$ are constant control gains. Substituting (4–6)-(4–8) into (4–5) yields the open-loop error system

$$M(q,t)r = \tau + S(t) - Y_d\theta(t), \tag{4–9}$$

where $S(t) \triangleq V_m(q_d, \dot{q}_d, t)\dot{q}_d - V_m(q, \dot{q}, t)\dot{q} + G(q_d, t) - G(q, t) + F(\dot{q}_d, t) - F(\dot{q}, t) + \left(M(q_d, t) - M(q, t)\right)\ddot{q}_d + M(q, t)\left(\alpha_1(e_2 - \alpha_1 e_1) + \alpha_2 e_2\right)$ and $Y_d \triangleq Y(q_d, \dot{q}_d, \ddot{q}_d, t)$ denotes the desired regression matrix.

Assumption 4.5. The desired trajectory $q_d(t)$ is bounded and smooth, such that $\|q_d(t)\| \leq \delta_0$, $\|\dot{q}_d(t)\| \leq \delta_1$, and $\|\ddot{q}_d(t)\| \leq \delta_2$, where $\delta_0, \delta_1, \delta_2 \in \mathbb{R}_{>0}$ are known bounding constants.

4.2.2 Control and Update Law Development

From the subsequent stability analysis, the continuous control input is designed as

$$\tau \triangleq Y_d\hat{\theta} - ke_2 + \mu, \tag{4–10}$$

where $k \in \mathbb{R}_{>0}$ is a constant control gain, $\mu : [t_0, \infty) \to \mathbb{R}^n$ is a subsequently defined auxiliary control term, and $\hat{\theta} : [t_0, \infty) \to \mathbb{R}^{n+m}$ denotes the parameter estimate of $\theta(t)$. Substituting the control input in (4–10) into the open-loop error system in (4–9) yields the following closed-loop error system

$$M(q,t)r = -Y_d\tilde{\theta}(t) + \mu - ke_2 + S(t), \tag{4–11}$$

where $\tilde{\theta} : [t_0, \infty) \to \mathbb{R}^{n+m}$ denotes the parameter estimation error, i.e., $\tilde{\theta}(t) \triangleq \theta(t) - \hat{\theta}(t)$. Taking the time-derivative of (4–11) yields

$$M(q,t)\dot{r} = -\dot{M}(q,t)r - \dot{Y}_d\tilde{\theta}(t) - Y_d\dot{\theta}(t) + Y_d\dot{\hat{\theta}} - k\dot{e}_2 + \dot{\mu} + \dot{S}(t). \tag{4–12}$$

The control variables $\dot{\hat{\theta}}(t)$ and $\dot{\mu}(t)$ now appear in the higher order dynamics in (4–12), and these control variables are designed with the use of a continuous projection algorithm [73, Appendix E]. The projection algorithm constrains $\hat{\theta}(t)$ to lie inside a bounded convex set $\mathcal{B} = \{\sigma \in \mathbb{R}^{(n+m)} | \, \|\sigma\| \leq \zeta_0\}$ by switching the adaptation law to its component tangential to the boundary of the set $\mathcal{B}$ when $\hat{\theta}(t)$ reaches the boundary. A continuously differentiable convex function $f : \mathbb{R}^{(n+m)} \to \mathbb{R}$ is used to describe the boundaries of the bounded convex set $\mathcal{B}$ such that $f(\sigma) < 0 \; \forall \; \|\sigma\| < \zeta_0$ and $f(\sigma) = 0 \; \forall \; \|\sigma\| = \zeta_0$. Based on the subsequent analysis, the continuous adaptation law is designed as

$$\begin{aligned} \dot{\hat{\theta}} &\triangleq \text{proj}(\Lambda_0) \\ &= \begin{cases} \Lambda_0, & ||\hat{\theta}|| < \zeta_0 \; \vee \; (\nabla f(\hat{\theta}))^T \Lambda_0 \leq 0, \\ \Lambda_1, & ||\hat{\theta}|| \geq \zeta_0 \; \wedge \; (\nabla f(\hat{\theta}))^T \Lambda_0 > 0, \end{cases} \end{aligned} \tag{4–13}$$

where $||\hat{\theta}(0)|| < \zeta_0$, $\vee$, $\wedge$ denote the logical 'or', 'and' operators, respectively, ∇ represents the gradient operator, i.e., $\nabla f(\hat{\theta}) = \left[\begin{array}{ccc} \frac{\partial f}{\partial \phi_1} & \cdots & \frac{\partial f}{\partial \phi_{n+m}} \end{array} \right]^T_{\phi = \hat{\theta}}$, and $\Lambda_0, \Lambda_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}^{n+m}$ are designed as[3]

$$\Lambda_0 \triangleq -\Gamma Y_d^T (Y_d \Gamma Y_d^T)^{-1} \left[k\alpha_2 e_2 + \beta \text{sgn}(e_2) \right], \tag{4–14}$$

$$\Lambda_1 \triangleq \left(I_{m+n} - \frac{(\nabla f(\hat{\theta}))(\nabla f(\hat{\theta}))^T}{||\nabla f(\hat{\theta})||^2} \right) \Lambda_0, \tag{4–15}$$

[3] Lemma 4.1 in the Appendix proves that $Y_d \Gamma Y_d^T$ is invertible.

respectively. In (4–14), $\beta \in \mathbb{R}_{>0}$ is a constant control gain, and $\Gamma \in \mathbb{R}^{(n+m)\times(n+m)}$ is a constant, positive-definite, control gain matrix with a block diagonal structure, i.e., $\Gamma \triangleq \begin{bmatrix} \Gamma_1 & 0_{m\times n} \\ 0_{n\times m} & \Gamma_2 \end{bmatrix}$, with $\Gamma_1 \in \mathbb{R}^{m\times m}$, $\Gamma_2 \in \mathbb{R}^{n\times n}$ and $I_{m+n} \in \mathbb{R}^{(n+m)\times(n+m)}$ is an identity matrix. The continuous auxiliary term $\mu(t)$ used in the control input in (4–10), is designed as a generalized solution to

$$\dot{\mu} \triangleq Y_d \left(\Lambda_0 - \text{proj}(\Lambda_0)\right), \tag{4–16}$$

where $\mu(t_0) = 0$. Substituting (4–13) and (4–16) into (4–12), the closed-loop error system can be obtained as

$$M(q,t)\dot{r} = -\dot{M}(q,t)r - \dot{Y}_d\tilde{\theta}(t) - Y_d\dot{\theta}(t) - \beta \text{sgn}(e_2) - kr + \dot{S}(t), \tag{4–17}$$

for both cases, i.e., when $||\hat{\theta}|| < \zeta_0 \vee (\nabla f(\hat{\theta}))^T\Lambda_0 \leq 0$ or $||\hat{\theta}|| \geq \zeta_0 \wedge (\nabla f(\hat{\theta}))^T\Lambda_0 > 0$. Let

$$z \triangleq \begin{bmatrix} e_1^T & e_2^T & r^T \end{bmatrix}^T \in \mathbb{R}^{3n} \tag{4–18}$$

denote a composite error vector. To facilitate the subsequent analysis, (4–17) can be rewritten as

$$M(q,t)\dot{r} = -\frac{1}{2}\dot{M}(q,t)r + \widetilde{N}(z,t) + N_B(\tilde{\theta},t) - \beta \text{sgn}(e_2) - kr - e_2, \tag{4–19}$$

where $\widetilde{N} : \mathbb{R}^{3n} \times [t_0,\infty) \to \mathbb{R}^n$ and $N_B : \mathbb{R}^{n+m} \times [t_0,\infty) \to \mathbb{R}^n$ are defined as $\widetilde{N}(z,t) \triangleq -\frac{1}{2}\dot{M}(q,t)r + \dot{S}(t) + e_2$ and $N_B(\tilde{\theta},t) \triangleq -\dot{Y}_d\tilde{\theta} - Y_d\dot{\theta}(t)$, respectively. The Mean Value Theorem (MVT) can be used to develop the following upper bound on the term $\widetilde{N}(z,t)$

$$||\widetilde{N}(z,t)|| \leq \rho(||z||)||z||, \tag{4–20}$$

where and $\rho : \mathbb{R}^{3n} \to \mathbb{R}$ is a positive, globally invertible and non-decreasing function. By Assumptions 4.4 and 4.5, Corollary 4.1 in the Appendix, and the bounding effect of projection algorithm on $\hat{\theta}(t)$, the term $N_B(\tilde{\theta},t)$ and its time-derivative $\dot{N}_B(\tilde{\theta},z,t)$ can be

upper bounded using known constants γ_1,γ_2, $\gamma_3 \in \mathbb{R}_{>0}$ as

$$||N_B(\tilde{\theta}, t)|| \leq \gamma_1, \qquad ||\dot{N}_B(\tilde{\theta}, z, t)|| \leq \gamma_2 + \gamma_3 \|e_2\|, \tag{4–21}$$

respectively.

4.3 Stability Analysis

To facilitate the subsequent analysis, let $y : [t_0, \infty) \to \mathbb{R}^{3n+1}$ be defined as

$$y \triangleq \begin{bmatrix} z^T & \sqrt{P} \end{bmatrix}^T, \tag{4–22}$$

where $P : [t_0, \infty) \to \mathbb{R}$ is a generalized solution to the differential equation

$$\dot{P} \triangleq -L. \tag{4–23}$$

In (4–23),

$$P(t_0) \triangleq \beta \|e_2(t_0)\|_1 - e_2(t_0)^T N_B(\tilde{\theta}(t_0), t_0), \tag{4–24}$$

and

$$L \triangleq r^T (N_B(\tilde{\theta}, t) - \beta \text{sgn}(e_2)) - \gamma_3 \|e_2\|^2. \tag{4–25}$$

In (4–24), $\|\cdot\|_1$ denotes the 1-norm. Provided that the gain condition

$$\beta > \gamma_1 + \frac{\gamma_2}{\alpha_2}, \tag{4–26}$$

is satisfied, $P(t) \geq 0$,[4] where the bounds γ_1, γ_2 and γ_3 are introduced in (4–21), and the control gain α_2 is introduced in (4–8). Therefore, it is valid to use $P(t)$ in the candidate Lyapunov function in the subsequent stability analysis. Furthermore, the auxiliary constant $\lambda_3 \triangleq \min\{\alpha_1 - \frac{1}{2}, \alpha_2 - \gamma_3 - \frac{1}{2}, \frac{k}{2}\}$ is introduced, where the control gains α_1 and k are introduced in (4–7) and (4–14), respectively. The gains α_1, α_2 and k are selected

[4] See the proof approach of Lemma 1 in [42] for details.

based on the sufficient gain condition

$$\lambda_3 > \frac{\rho^2\left(\sqrt{\frac{\lambda_2(q(t_0))}{\lambda_1}}\left\|y(t_0)\right\|\right)}{2k}, \tag{4–27}$$

with $\lambda_1 \triangleq \frac{1}{2}\min\{1, m_1\}$ and $\lambda_2(q) \triangleq \frac{1}{2}\max\{2, \bar{m}(q)\}$, where m_1 and $\bar{m}(q)$ are introduced in Assumption 4.1. From (4–7), (4–8), (4–19), (4–23) and (4–25), the differential equations describing the closed-loop system are

$$\dot{e}_1 = e_2 - \alpha_1 e_1, \tag{4–28}$$

$$\dot{e}_2 = r - \alpha_2 e_2, \tag{4–29}$$

$$\dot{r} = M^{-1}(q,t)(-\frac{1}{2}\dot{M}(q,t)r + \widetilde{N}(z,t) + N_B(\tilde{\theta},t) - \beta\text{sgn}(e_2) - kr - e_2), \tag{4–30}$$

$$\dot{P} = -r^T(N_B(t) - \beta\text{sgn}(e_2)) + \gamma_3\left\|e_2\right\|^2. \tag{4–31}$$

Theorem 4.1. *Given the Euler-Lagrange dynamic system in (4–1) along with Assumptions 4.1-4.5, for any arbitrary initial condition of the states $e_1(t_0)$, $e_2(t_0)$, and $r(t_0)$, selecting $P(t_0)$, $\alpha_1, \alpha_2, \beta$, and k according to (4–24), (4–26), and (4–27) ensures that $e_1, e_2, r, P \in \mathcal{L}_\infty$, and $\left\|e_1(t)\right\| \to 0$ as $t \to \infty$.*

Proof. Let $\mathcal{D} \subset \mathbb{R}^{3n+1}$ be the open and connected set defined as

$$\mathcal{D} \triangleq \left\{\sigma \in \mathbb{R}^{3n+1} \mid \left\|\sigma\right\| < \rho^{-1}\left(\sqrt{2\lambda_3 k}\right)\right\}, \tag{4–32}$$

and $V_L : \mathcal{D} \times [t_0, \infty) \to \mathbb{R}_{\geq 0}$ be a positive-definite candidate Lyapunov function defined as

$$V_L(y,t) \triangleq \frac{1}{2}r^T M(q,t)r + \frac{1}{2}e_2^T e_2 + \frac{1}{2}e_1^T e_1 + P. \tag{4–33}$$

The candidate Lyapunov function in (4–33) satisfies

$$\lambda_1\left\|y\right\|^2 \leq V_L \leq \lambda_2(q)\left\|y\right\|^2, \tag{4–34}$$

where λ_1 and $\lambda_2(q)$ are defined after (4–27). Let $\psi \triangleq \begin{bmatrix} e_1^T & e_2^T & r^T & P \end{bmatrix}^T$, and $\dot{\psi} \in K[g](\psi, t)$ denote the Filippov differential inclusion corresponding to (4–28)-(4–31),

where the operator $K[\cdot]$ is defined in [67, Equation 2b]. Note that $g : \mathbb{R}^{3n+1} \times [t_0, \infty) \to \mathbb{R}^{3n+1}$ is Lebesgue measurable and locally essentially bounded, since it is continuous except in the set with measure zero, $\{(\psi, t) \in \mathbb{R}^{3n+1} \times [0, \infty) | e_2 = 0\}$. Therefore, the existence of an absolutely continuous solution $t \mapsto \psi(t)$ to $\dot{\psi} \in K[g](\psi, t)$ is guaranteed[5] by [69, Proposition 3]. Let $\dot{\tilde{V}}_L(y,t) \triangleq \bigcap_{\xi \in \partial V_L(y,t)} \xi^T [K[g](\psi, t); 1]$ as defined in [87, Eq. 13], where $\partial V_L(y,t)$ denotes Clarke's generalized gradient [87, Eq. 7]. Since $(y,t) \mapsto V_L(y,t)$ is continuously differentiable, Clarke's gradient is the same as the standard gradient, i.e., $\partial V_L = \{\nabla V_L\}$. Using [87, Thm 2.2], $t \mapsto \dot{V}_L(y(t), t)$ exists almost everywhere[6] and $\dot{V}_L(y,t) \in \dot{\tilde{V}}_L(y,t)$ for almost all time (a.a.t.). Evaluating $\dot{\tilde{V}}_L(y,t)$ and (4–28)-(4–31) yields

$$\begin{aligned} \dot{\tilde{V}}_L \overset{a.a.t.}{\leq} & \; r^T(-\frac{1}{2}\dot{M}(q,t)r + \widetilde{N}(z,t) + N_B(\tilde{\theta}, t) - \beta K\left[\text{sgn}\right](e_2) - kr - e_2) + e_2^T(r - \alpha_2 e_2) \\ & + e_1^T(e_2 - \alpha_1 e_1) - r^T(N_B(t) - \beta K\left[\text{sgn}\right](e_2)) + \gamma_3 \left\|e_2\right\|^2 + \frac{1}{2} r^T \dot{M}(q,t) r. \end{aligned} \quad (4\text{–}35)$$

Using (4–20) and applying Young's inequality on $e_1^T e_2$ in (4–35), $\dot{V}_L$ can be upper bounded as

$$\dot{V}_L \overset{a.a.t.}{\leq} \rho(\left\|z\right\|) \left\|z\right\| \left\|r\right\| - k \left\|r\right\|^2 - (\alpha_2 - \gamma_3 - \frac{1}{2}) \left\|e_2\right\|^2 - (\alpha_1 - \frac{1}{2}) \left\|e_1\right\|^2.$$

The set of times $T \triangleq \{t \in [t_0, \infty) : r(t)^T \beta \text{SGN}(e_2(t)) - r(t)^T \beta \text{SGN}(e_2(t)) \neq \{0\}\} \subset \mathbb{R}_{\geq 0}$ is equal to the set of times $\{t : e_2(t) = 0 \wedge r(t) \neq 0\}$. Using $r = \dot{e}_2 + \alpha_2 e_2$, this set can

[5] The existing solution might have a finite escape time. This possibility is ruled out by proving the boundedness of Filippov trajectories under the aforementioned sufficient conditions using Lyapunov-based stability theory. Therefore, $\text{dom}\, \psi = [t_0, \infty)$, i.e., the solution is complete. The solution may not be unique; however, the results are applicable to all the trajectories, since a generalized Filippov solution is considered in the analysis.

[6] Since $\psi = \begin{bmatrix} z^T & P \end{bmatrix}^T$ and $y = \begin{bmatrix} z^T & \sqrt{P} \end{bmatrix}^T$, $y(t)$ can be evaluated along a Filippov trajectory $\psi(t)$ by a transformation which involves taking the square-root of $P(t)$, which is applicable since $P(t) \geq 0, \forall t \in [t_0, \infty)$.

also be represented by $\{t : e_2(t) = 0 \wedge \dot{e}_2(t) \neq 0\}$. Since e_2 is continuously differentiable because the right hand-side of (4–29) is continuous, [72, Lemma 2] can be used to show that the set of time instances $\{t : e_2(t) = 0 \wedge \dot{e}_2(t) \neq 0\}$ is isolated, and hence, measure zero; hence, T is measure zero. Therefore, $\dot{\tilde{V}}_L = \left\{\dot{V}_L\right\}$ a.e. in time, and an upper bound on $\dot{V}_L$ can be obtained a.e. in time, using the right-hand side of (4–35). Using Young's Inequality on $\rho(\|z\|)\|z\|\|r\|$ yields $\rho(\|z\|)\|z\|\|r\| \leq \frac{\rho^2(\|z\|)\|z\|^2}{2k} + \frac{1}{2}k\|r\|^2$. Therefore,

$$\begin{aligned}\dot{\tilde{V}}_L &\overset{a.a.t.}{\leq} \frac{\rho^2(\|z\|)\|z\|^2}{2k} - \frac{k}{2}\|r\|^2 - (\alpha_2 - \gamma_3 - \frac{1}{2})\|e_2\|^2 - (\alpha_1 - \frac{1}{2})\|e_1\|^2 \\ &\leq -\left(\lambda_3 - \frac{\rho^2(\|z\|)}{2k}\right)\|z\|^2. \end{aligned} \tag{4–36}$$

The expression in (4–36) can be rewritten as

$$\dot{V}_L \overset{a.a.t.}{\leq} -W(y) = -c\|z\|^2, \forall\, y \in \mathcal{D}, \tag{4–37}$$

with some constant $c \in \mathbb{R}_{>0}$, where $W : \mathbb{R}^{3n+1} \to \mathbb{R}$ is a continuous positive semi-definite function.

Whenever $y \in \mathcal{D}$, $\|y(t)\| < \rho^{-1}\left(\sqrt{2\lambda_3 k}\right)$ by definition of $\mathcal{D}$, which is sufficient to infer $\|z(t)\| < \rho^{-1}\left(\sqrt{2\lambda_3 k}\right)$ using (4–22). Therefore, if $y(t) \in \mathcal{D}$, $\lambda_3 > \frac{\rho^2(\|z\|)}{2k}$, which implies from (4–36) that there exists $c \in \mathbb{R}_{>0}$ which satisfies (4–37), and larger values of λ_3 expand the size of $\mathcal{D}$. Since V_L is non-increasing, which implies $\|y(t)\| \leq \sqrt{\frac{V_L(t)}{\lambda_1}} \leq \sqrt{\frac{V_L(t_0)}{\lambda_1}}$, it is sufficient to show that $\sqrt{\frac{V_L(t_0)}{\lambda_1}} < \rho^{-1}(\sqrt{2\lambda_3 k})$, to obtain $y(t) \in \mathcal{D}$. Since $V_L(t_0) \leq \lambda_2(q(t_0))\|y(t_0)\|^2$, the result $\sqrt{\frac{V_L(t_0)}{\lambda_1}} < \rho^{-1}(\sqrt{2\lambda_3 k})$ can be sufficiently obtained from $\sqrt{\frac{\lambda_2(q(t_0))}{\lambda_1}}\|y(t_0)\| < \rho^{-1}(\sqrt{2\lambda_3 k})$. Therefore, $\|y(t_0)\| < \sqrt{\frac{\lambda_1}{\lambda_2(q(t_0))}}\rho^{-1}(\sqrt{2\lambda_3 k})$, which implies that $\mathcal{S} \triangleq \left\{\sigma \in \mathcal{D} | \|\sigma\| < \sqrt{\frac{\lambda_1}{\lambda_2(q(t_0))}}\rho^{-1}(\sqrt{2\lambda_3 k})\right\}$ is the region where $y(t_0)$ should lie to guarantee that $y(t) \in \mathcal{D}$ for all $t \in [t_0, \infty)$. Though the sets $\mathcal{S}$ and $\mathcal{D}$ are defined to include $y(t)$ instead of $\psi(t)$, one can easily construct the bounded sets $\mathcal{S}_\psi$ and $\mathcal{D}_\psi$ such that $y(t_0) \in \mathcal{S}$ and $y(t) \in \mathcal{D}$ imply $\psi(t_0) \in \mathcal{S}_\psi$ and $\psi(t) \in \mathcal{D}_\psi$, respectively, to conclude the uniform boundedness of all Filippov trajectories $\psi(t)$ initializing in the set

$\mathcal{S}_\psi$. Using (4–32), (4–34) and (4–37), since g is Lebesgue measurable and essentially locally bounded, uniformly in time, the extension of the LaSalle-Yoshizawa corollary in [66, Corollary 1] can be invoked to show that e_1, e_2, r, $P \in \mathcal{L}_\infty$, and $\|z(t)\| \to 0$ as $t \to \infty$. Therefore, using the definition of z in (4–18), $\|e_1(t)\| \to 0$ as $t \to \infty$. The gain condition in (4–27) needs to be satisfied according to the initial condition, and the region of attraction can be made arbitrarily large to include any initial condition by increasing the gains α_1, α_2 and k accordingly, therefore, the result is semi-global.

The parameter estimate $\hat{\theta} \in \mathcal{L}_\infty$ due to the projection operation, which implies $\tilde{\theta}(t) = \theta(t) - \hat{\theta}(t)$ is bounded, because the parameter $\theta \in \mathcal{L}_\infty$ by Assumption 4.4. Since $e_1, e_2, r \in \mathcal{L}_\infty$, and because $q_d, \dot{q}_d, \ddot{q}_d \in \mathcal{L}_\infty$ by Assumption 4.5, using (4–6)-(4–8) implies that $q, \dot{q}, \ddot{q} \in \mathcal{L}_\infty$. Furthermore, the regression matrix $Y_d \in \mathcal{L}_\infty$ by Assumption 4.5, because Y is locally bounded due to Properties 2 and 3. Therefore, by Corollary 4.1 in the Appendix, $\dot{\hat{\theta}} \in \mathcal{L}_\infty$. The expression in (4–11) indicates that $\mu \in \mathcal{L}_\infty$, because among the remaining terms in (4–11), $M(q)r$ and $Y_d\tilde{\theta}$ are comprised of bounded terms because M is locally bounded, and $S \in \mathcal{L}_\infty$ because its definition is comprised of terms that are locally bounded functions of the bounded errors and states due to Assumption 4.2. From the expression in (4–10), since $\hat{\theta}, Y_d, \mu \in \mathcal{L}_\infty$, $\tau \in \mathcal{L}_\infty$. Moreover, differentiating the right-hand side in (4–10) yields terms that are bounded, which implies $\dot{\tau} \in \mathcal{L}_\infty$; therefore, τ is continuous. Hence, all the closed-loop signals are bounded. □

4.4 Simulation Example

To demonstrate the performance and efficacy of the developed method, a simulation example of a horizontal two-link manipulator system is provided, and the results are compared with an e-modification (e-mod) based controller [43]. The dynamics of the manipulator system can be represented in the form of (4–1), with $M(q,t) = \begin{bmatrix} p_1(t) + 2p_3(t)c_2 & p_2(t) + p_3(t)c_2 \\ p_2(t) + p_3(t)c_2 & p_2(t) \end{bmatrix}$, $V_m(q,\dot{q},t) = \begin{bmatrix} -p_3(t)s_2\dot{q}_2 & -p_3(t)s_2(\dot{q}_1 + \dot{q}_2) \\ p_3(t)s_2\dot{q}_1 & 0 \end{bmatrix}$,

$F(\dot{q},t) = \begin{bmatrix} F_{d1}(t)\dot{q}_1 \\ F_{d2}(t)\dot{q}_2 \end{bmatrix}$ and $\tau_d(t) = \begin{bmatrix} \tau_{d1}(t) & \tau_{d2}(t) \end{bmatrix}^T$, where $c_2 \triangleq \cos(q_2)$, $s_2 \triangleq \sin(q_2)$ and $p_1, p_2, p_3, F_{d1}, F_{d2}, \tau_{d1}, \tau_{d2} : \mathbb{R}_{\geq 0} \to \mathbb{R}$, and the gravity term $G(q,t)$ is ignored for a horizontal manipulator. The augmented time-varying parameter vector for the manipulator system is given by $\theta(t) = \begin{bmatrix} p_1(t) & p_2(t) & p_3(t) & F_{d1}(t) & F_{d2}(t) & \tau_{d1}(t) & \tau_{d2}(t) \end{bmatrix}^T$. The control objective is to track a given reference trajectory $q_d(t) = \begin{bmatrix} \cos(0.5t) & 2\cos(t) \end{bmatrix}^T$. The time-varying parameters used in the simulation are $p_1(t) = 3.473 + 0.5\sin(3t)$, $p_2(t) = 0.196 + 0.2\exp(-\sin(t))$, $p_3(t) = 0.242 + 0.1\cos(10t)$, $F_{d1}(t) = 5.3 + 2\exp(-0.1t)$, $F_{d2}(t) = 1.1 + \cos(5t)$ and the disturbance terms $\tau_{d1}(t) = 0.5\cos(0.5t)$ and $\tau_{d2}(t) = \sin(t)$. The initial conditions used in the simulation are $q(0) = \begin{bmatrix} -1 & 1 \end{bmatrix}^T$, $\dot{q}(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ and $\hat{\theta}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. Note that in practice, the best guess estimates of the uncertain parameters or their approximate mean should be used for improved performance. The estimates were initialized to zero to illustrate adaptation with no prior knowledge.

The control gains for each method are obtained using a Monte-Carlo method; an appropriate range is qualitatively determined for each gain, and 10000 iterations are subsequently run with a uniform random gain sampling within those ranges in an attempt to minimize

$$J = \int_0^{10} \left(a \left\| e_1(t) \right\|^2 + b \left\| \tau(t) \right\|^2 \right) dt, \tag{4–38}$$

with $a = 1$ and $b = 0.01$. The gains that minimized (4–38) for the developed method are $K = 18.1502$, $\alpha_1 = 0.8982$, $\alpha_2 = 1.0552$, $\beta = 36.2946$ and $\Gamma = I_2$. This set of gains might not satisfy the gain conditions in (4–26) and (4–27), however, those conditions are not necessary, rather only sufficient. The gains were selected from the Monte-Carlo simulation to provide the best performance and an equal comparison with the e-mod method. For the projection algorithm, $\zeta_0 = 5000$ and the corresponding

function $f(\hat{\theta}) = \left\|\hat{\theta}\right\|^2 - \zeta_0^2$. For the e-mod update law, i.e. , $\dot{\hat{\theta}} = \Gamma_e Y_d^T r - \sigma \left\|e_1\right\| \hat{\theta}$ and the corresponding controller $\tau = Y_d\hat{\theta} - k_e r$, the gains are $\Gamma_e = 12.5$, $k_e = 9.7877$ and $\sigma = 9.7319$.

Figure 4-1 demonstrates the asymptotic convergence of the tracking error and the function estimation error ($Y\theta - Y_d\hat{\theta}$) to zero with the developed method in the simulation, as opposed to the UUB tracking with the e-mod scheme. From an applied perspective, if the upper bound used for projection algorithm, i.e., ζ_0 is selected to be sufficiently high such that the parameter estimates never reach the boundary of the set $\mathcal{B}$, then $\text{proj}(\Lambda_0(t)) = \Lambda_0(t)$, $\forall t \in [t_0, \infty)$, implying $\mu(t) = 0$, $\forall t \in [t_0, \infty)$. From (4–5), $Y\theta = \tau$, and $\tau - Y_d\hat{\theta} = \mu - ke_2$ using (4–10), therefore if $\mu(t) = 0 \forall t \in [t_0, \infty)$, then the function approximation error $Y\theta - Y_d\hat{\theta} = \mu - ke_2 = -ke_2 \to 0$ as $t \to \infty$. In case the parameter estimates reach the boundary of $\mathcal{B}$, $Y\theta - Y_d\hat{\theta}$ may not converge to zero, yet it is guaranteed to be bounded using the stability analysis since μ is bounded. Table 4-1 provides a quantitative comparison of the controllers, where e_{rms} is the root-mean-square (RMS) of e_1 (in deg) taken over the time interval $[0, 10]$, $e_{\text{rms,ss}}$ is the RMS of e_1 over the time interval $[5, 10]$ (i.e., after reaching the steady state), $e_{\text{max,ss}}$ is the maximum absolute value of the components of e_1 over the time-interval $[5, 10]$, $\tilde{Y}_{\text{rms}}$ denotes the RMS function estimation error (in Nm) over the interval $[0, 10]$ and τ_{rms} denotes the RMS simulated torque (in Nm) over the time interval $[0, 10]$. The developed method provides a significantly improved tracking and function estimation performance with less RMS control effort, upon comparison with e-mod.

Figure 4-2 demonstrates the tracking error performance in the presence of additive white Gaussian (AWG) noise with standard deviations of 2 deg and 2 deg/s in the q and $\dot{q}$ measurements, respectively. The RMS steady state tracking error norms in presence of measurement noise with the developed method and e-mod are 2.9427 and 4.5891, respectively.

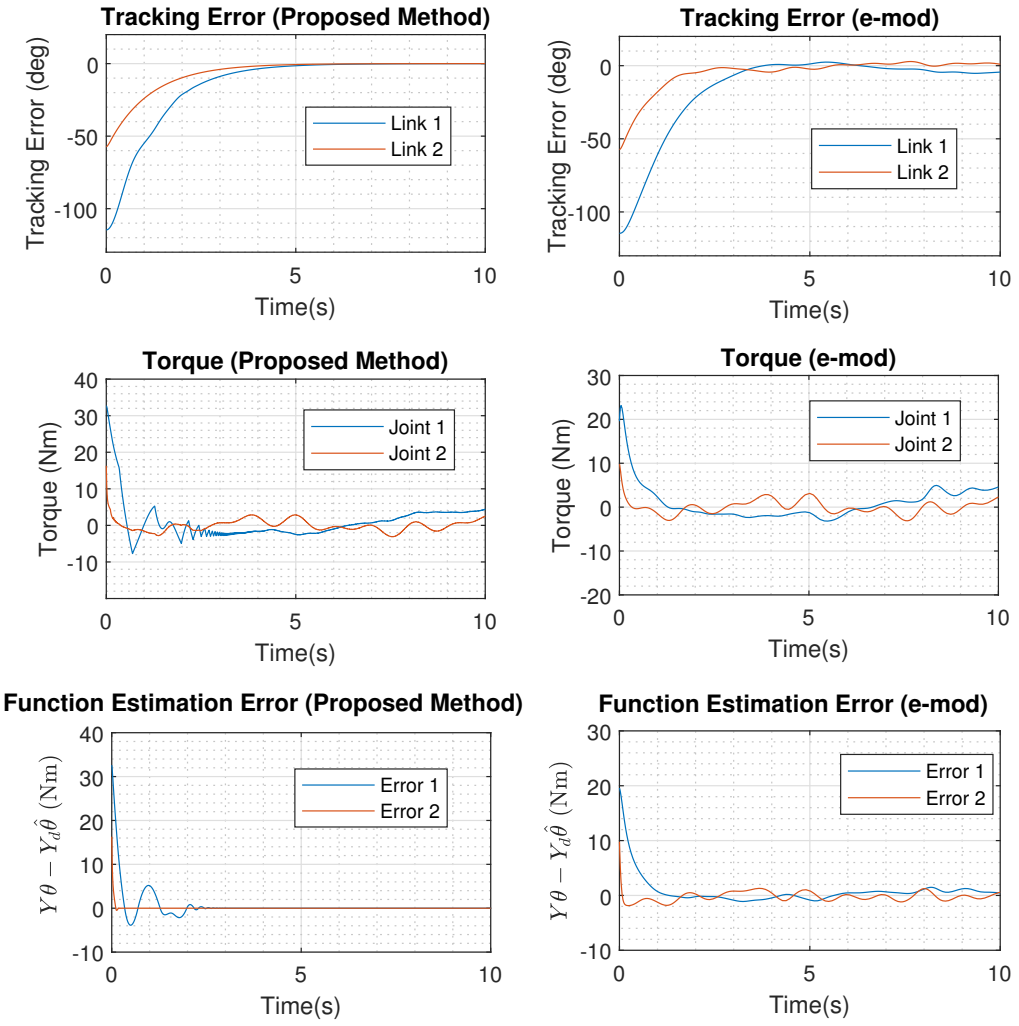

Figure 4-1. Plots of tracking error (deg), torque input (Nm) and function estimation error ($Y\theta - Y_d\hat{\theta}$) vs. time (s) with the proposed method and e-mod.

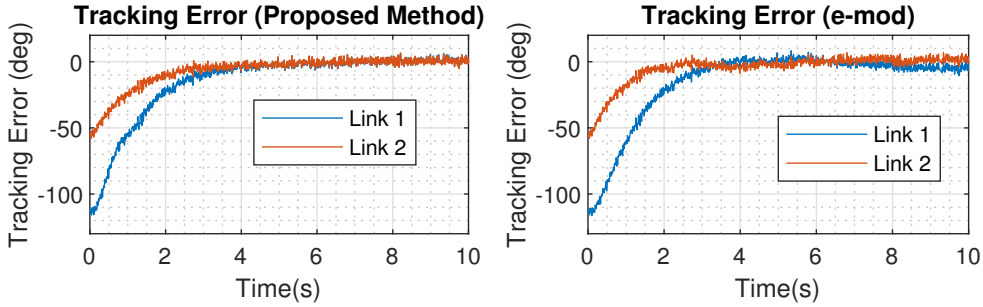

Figure 4-2. Plots of tracking error (deg) vs. time (s) in presence of AWG measurement noise with the proposed method and e-mod.

Table 4-1. Controller performance comparison

Method	$\|e_{\text{rms}}\|$	$\|e_{\text{rms,ss}}\|$	$e_{\text{max,ss}}$	$\left\|\tilde{Y}_{\text{rms}}\right\|$	$\|\tau_{\text{rms}}\|$
(4–10)	29.1340	0.5515	1.4666	1.3654	3.2726
e-mod	52.4838	3.5624	5.2500	6.1292	7.4722

4.5 Conclusion

A contribution of this chapter is the development of a continuous adaptive control design that achieves semi-global asymptotic tracking for linearly parameterizable nonlinear systems with time-varying uncertain parameters. Through a unique analysis strategy, an adaptive feedforward term is developed along with specialized feedback terms to compensate for the time-varying uncertainty. Asymptotic tracking error convergence is guaranteed via a Lyapunov-based stability analysis for an Euler-Lagrange system. Additionally, the time-varying uncertain function approximation error is shown to converge to zero. A simulation example of a two-link manipulator is provided to demonstrate the asymptotic tracking result, and a comparison with the e-mod scheme shows a better tracking performance with the proposed method. Future work may involve extension of the proposed approach to unstructured time-varying uncertainties using neural networks, compensation of time-varying uncertainty in presence of sensor noise, and delays in input and state measurements.

Appendix: Auxiliary Lemmas

Lemma 4.1. *Consider a positive-definite matrix* $\Gamma \in \mathbb{R}^{(n+m)\times(n+m)}$ *such that* Γ *has the block diagonal structure as* $\Gamma \triangleq \begin{bmatrix} \Gamma_1 & 0_{m\times n} \\ 0_{n\times m} & \Gamma_2 \end{bmatrix}$, *where* $\Gamma_1 \in \mathbb{R}^{m\times m}$ *and* $\Gamma_2 \in \mathbb{R}^{n\times n}$. *The matrix* $Y\Gamma Y^T$ *is positive-definite, and therefore invertible. Furthermore, the inverse of this matrix satisfies the property* $\left\|\left(Y\Gamma Y^T\right)^{-1}\right\|_2 \leq \frac{1}{\lambda_{min}\{\Gamma_2\}}$, *where* $\|\cdot\|_2$ *denotes the spectral norm and* $\lambda_{min}\{\cdot\}$ *denotes the minimum eigenvalue of* $\{\cdot\}$.

Proof. Substituting the definitions for Y and Γ in $Y\Gamma Y^T$ yields

$$\begin{aligned} Y\Gamma Y^T &= \begin{bmatrix} Y_p & I_n \end{bmatrix} \begin{bmatrix} \Gamma_1 & 0_{m\times n} \\ 0_{n\times m} & \Gamma_2 \end{bmatrix} \begin{bmatrix} Y_p^T \\ I_n \end{bmatrix} \\ &= Y_p\Gamma_1 Y_p^T + \Gamma_2. \end{aligned}$$

Since Γ is selected to be a positive-definite matrix, the block matrices Γ_1 and Γ_2 are both positive-definite, so $Y_p\Gamma_1 Y_p^T$ is positive semi-definite while the second term Γ_2 is positive-definite, hence the sum of these two terms, i.e., $Y\Gamma Y^T$ is positive-definite, and therefore invertible. Furthermore, the spectral norm satisfies the property, $\|A\|_2 = \sqrt{\lambda_{\max}\{A^TA\}}$ for some $A \in \mathbb{R}^{p\times q}$, where $\lambda_{\max}\{\cdot\}$ denotes the maximum eigenvalue of $\{\cdot\}$. Utilizing this property with $\left\|\left(Y\Gamma Y^T\right)^{-1}\right\|_2$ yields

$$\begin{aligned} \left\|\left(Y\Gamma Y^T\right)^{-1}\right\|_2 &= \sqrt{\lambda_{\max}\left\{\left(\left(Y\Gamma Y^T\right)^{-1}\right)^T\left(Y\Gamma Y^T\right)^{-1}\right\}} \\ &= \lambda_{\max}\left\{\left(Y\Gamma Y^T\right)^{-1}\right\}. \\ &= \frac{1}{\lambda_{\min}\{Y\Gamma Y^T\}} \leq \frac{1}{\lambda_{\min}\{\Gamma_2\}}. \end{aligned} \tag{4–39}$$

□

Corollary 4.1. *The norm of the time-derivative of the parameter estimate, $\left\|\dot{\hat{\theta}}\right\|$ can be upper bounded by as $\left\|\dot{\hat{\theta}}\right\| \leq \gamma_4 + \gamma_5\|e_2\|$, where $\gamma_4, \gamma_5 \in \mathbb{R}_{>0}$ are known bounding constants.*

Proof. Based on (4–13)

$$\begin{aligned} \left\|\dot{\hat{\theta}}\right\| &= \|\text{proj}(\Lambda_0)\| \leq \|\Lambda_0\| \\ &= \left\|\Gamma Y_d^T(Y_d\Gamma Y_d^T)^{-1}(\beta\text{sgn}(e_2) + k\alpha_2 e_2)\right\| \\ &\leq \left\|\Gamma Y_d^T(Y_d\Gamma Y_d^T)^{-1}\right\|(\beta + k\alpha_2\|e_2\|). \end{aligned} \tag{4–40}$$

Applying Holder's inequality to the right-hand side of (4–40) yields

$$\left\|\dot{\hat{\theta}}\right\| \leq \|\Gamma\|_2 \|Y_d\|_2 \left\|(Y_d \Gamma Y_d^T)^{-1}\right\|_2 (\beta + k\alpha_2 \|e_2\|) . \tag{4–41}$$

Using the result from Lemma 1 yields

$$\left\|\dot{\hat{\theta}}\right\| \leq \frac{\|\Gamma\|_2 \|Y_d\|_2}{\lambda_{\min}\{\Gamma_2\}} (\beta + k\alpha_2 \|e_2\|) .$$

Based on Assumption 4.5, the spectral norm of the desired regressor may be upper-bounded by a constant $\overline{Y}_d \in \mathbb{R}_{>0}$, i.e., $\|Y_d\|_2 \leq \overline{Y}_d$, because Y_d is a continuously differentiable function. Therefore, selecting $\gamma_4 = \frac{\beta \|\Gamma\|_2 \overline{Y}_d}{\lambda_{\min}\{\Gamma_2\}}$ and $\gamma_5 = \frac{k\alpha_2 \|\Gamma\|_2 \overline{Y}_d}{\lambda_{\min}\{\Gamma_2\}}$ yields

$$\begin{aligned} \left\|\dot{\hat{\theta}}\right\| &\leq \frac{\|\Gamma\|_2 \overline{Y}_d}{\lambda_{\min}\{\Gamma_2\}} (\beta + k\alpha_2 \|e_2\|) \\ &= \gamma_4 + \gamma_5 \|e_2\| . \end{aligned}$$

□

CHAPTER 5
EXPONENTIAL STABILITY WITH RISE CONTROLLERS

A class of continuous robust controllers termed RISE have been published over the past two decades as a means to yield asymptotic tracking error convergence and implicit asymptotic identification of time-varying uncertainties, for classes of nonlinear systems that are subject to sufficiently smooth bounded exogenous disturbances and/or modeling uncertainties. Despite the wide application of RISE-based techniques, an open question that has eluded researchers during this time-span is whether the asymptotic tracking error convergence is also uniform or exponential. This question has remained open due to certain limitations in the traditional construction of a Lyapunov function for RISE-based error systems. A contribution of this chapter and my work [23] is the use of new insights for the construction of a Lyapunov function that result in an exponential stability result for RISE-based controllers. As an outcome of this breakthrough, the implicit learning capability of RISE-based controllers is shown to yield exponential identification of state-dependent disturbances/uncertainty.

5.1 Control Design

5.1.1 Control Objective

Consider a control affine system with the nonlinear dynamics

$$\dot{x} = d(x, \nu, t) + u, \tag{5–1}$$

where $t \in \mathbb{R}_{\geq 0}$ denotes time, $x : \mathcal{I} \to \mathbb{R}^n$ denotes a Filippov solution to (5–1), with the interval of existence $\mathcal{I} = [t_0, t_1)$ for some $t_0, t_1 \in \mathbb{R}_{\geq 0}$ s.t. $t_1 > t_0$, $\nu : \mathbb{R}_{\geq 0} \to \mathbb{R}^m$ denotes an auxiliary function representing some external dynamic compensator-based terms (e.g., adaptive feedforward terms, observer-based terms), $d : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ represents $\mathcal{C}^2$ modeling uncertainty in the system, and $u : \mathcal{I} \to \mathbb{R}^n$ represents the control input. Let $\left[\dot{d}(x, \dot{x}, \nu, \dot{\nu}, t)\right]_i \triangleq \left[\nabla d^T(x, \nu, t)\right]_i [\dot{x}; \dot{\nu}; 1]$ and $\left[\ddot{d}(x, \dot{x}, \ddot{x}, \nu, \dot{\nu}, \ddot{\nu}, t)\right]_i = [\dot{x}; \dot{\nu}; 1]^T \left[\nabla^2 d(x, \nu, t)\right]_i [\dot{x}; \dot{\nu}; 1] + \nabla d^T(x, \nu, t) [\ddot{x}; \ddot{\nu}; 0]$, respectively, where ∇ and ∇^2 denote

the gradient and Hessian operators, respectively, and $[\cdot]_i$ denotes the i^{th} component of $[\cdot]$. It is assumed that for each $(a, b, p, v, w, s) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^m$, the mappings $t \mapsto d(a, v, t)$, $t \mapsto \dot{d}(a, b, v, w, t)$, and $t \mapsto \ddot{d}(a, b, p, v, w, s, t)$ are bounded. The objective is to design a controller such that the state tracks a smooth bounded reference trajectory. The objective is quantified by defining the tracking error according to (4–6) where $x_d : \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ is a $\mathcal{C}^2$ reference trajectory such that $x_d, \dot{x}_d \in \mathcal{L}_\infty$.

5.1.2 Control Law Development

To facilitate the subsequent analysis, a filtered tracking error $r : \mathcal{I} \to \mathbb{R}^n$ is defined as $r \triangleq d(x, \nu, t) + u - \dot{x}_d + \alpha e$, where $\alpha \in \mathbb{R}_{>0}$ is a constant control gain. To facilitate the subsequent analysis, the dynamics in terms of $\dot{e}$ can be rewritten using (5–1) and (4–6) as

$$\dot{e} = r - \alpha e. \tag{5–2}$$

Let $z : \mathcal{I} \to \mathbb{R}^{2n}$ denote the augmented tracking error, $z \triangleq \begin{bmatrix} e^T & r^T \end{bmatrix}^T$. From the subsequent stability analysis, a continuous RISE control input is designed as [42]

$$u \triangleq \dot{x}_d - \alpha e - \hat{d}, \tag{5–3}$$

where $\hat{d} : \mathcal{I} \to \mathbb{R}^n$ is an auxiliary term designed as a Filippov solution[1] to

$$\dot{\hat{d}} = kr + e + \beta \text{sgn}(e), \tag{5–4}$$

given any user-selected $\hat{d}(t_0) \in \mathbb{R}^n$. In (5–4), $k, \beta \in \mathbb{R}_{>0}$ are constant control gains. Using (5–1)-(5–3) yields

$$r = d(x, \nu, t) - \hat{d}. \tag{5–5}$$

[1] Since r may not be commonly available, $\hat{d}(t)$ is evaluated using $\hat{d}(t) = \hat{d}(t_0) + ke(t) - ke(t_0) + \int_{t_0}^{t} ((k\alpha + 1)e(\tau) + \beta \text{sgn}(e(\tau)))\, d\tau$ for closed-loop implementation. Note that $\beta \text{sgn}(e(\cdot))$ is Riemann integrable on $[t_0, t]$, $\forall t \in \mathcal{I}$ according to Lemma 5.2.

It follows from (5–4) and (5–5) that r is a Filippov solution to the closed-loop error system

$$\dot{r} = \widetilde{N} + N_B - kr - e - \beta \text{sgn}\,(e)\,, \tag{5–6}$$

where $\widetilde{N} \triangleq \dot{d}(x, \dot{x}, \nu, \dot{\nu}, t) - \dot{d}(x_d, \dot{x}_d, \nu, \dot{\nu}, t)$ and $N_B \triangleq \dot{d}(x_d, \dot{x}_d, \nu, \dot{\nu}, t)$.

Assumption 5.1. The function ν is a solution to some external dynamics such that there exist known constants, $\eta_1, \eta_2, \eta_3, \eta_4 \in \mathbb{R}_{\geq 0}$, and a known strictly increasing function, $\rho_{21} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, such that $\|\nu\| \leq \eta_1$, $\|\dot{\nu}\| \leq \eta_2$, and $\|\ddot{\nu}\| \leq \eta_3 + \eta_4 \|z\| + \rho_{21}(\|z\|)\|z\|$. Then, there exist known constants $\gamma_1, \gamma_3, \gamma_4 \in \mathbb{R}_{\geq 0}$ and a known strictly increasing function $\rho_{21} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that $\|N_B\| \leq \gamma_1$ and $\left\|\dot{N}_B\right\| \leq \gamma_3 + \gamma_4 \|z\| + \rho_2(\|z\|)\|z\|$, $\forall t \in \mathbb{R}_{\geq 0}$. Additionally, since ν is bounded and $t \mapsto \dot{d}(a, b, v, w, t)$ is bounded for each $(a, b, v, w) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^m$, $\left\|\widetilde{N}\right\| \leq \gamma_2 \|z\| + \rho_1(\|z\|)\|z\|$, $\forall t \in \mathbb{R}_{\geq 0}$, according to the Mean Value Theorem-based inequality in [72, Lemma 5], where $\gamma_2 \in \mathbb{R}_{\geq 0}$ is a known constant, and $\rho_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a known strictly increasing function. Note that the type of state-dependent bounds considered in Assumption 5.1 are general and often required in various applications where the RISE method is used (e.g., [49, 51] and [52]), typically as a consequence of augmenting adaptive feedforward controllers with a RISE term. In the case where ν represents adaptive feedforward terms, the developed approach offers modularity of design in the sense that $\hat{d}$ and ν can be designed independently, as long as ν satisfies Assumption 5.1. The following example illustrates a type of system satisfying Assumption 5.1.

Example 5.1. Consider a dynamic neural network given by

$$\dot{\nu} = \text{proj}\{W^T \sigma(V^T x), \nu\}, \tag{5–7}$$

where $\text{proj}\{\cdot,\cdot\}$ denotes the smooth projection operator in [90] that guarantees $\|\nu\| \leq \eta_1$, $\sigma : \mathbb{R}^L \to \mathbb{R}^L$ denotes a globally bounded continuous activation function, and $W \in \mathbb{R}^{m\times L}$ and $V \in \mathbb{R}^{L\times n}$ are constant[2] matrices of outer and inner-layer weights, respectively. Using [90, Property 3] and the fact that $\sigma(\cdot)$ is globally bounded, $\dot{\nu}$ can be bounded by a constant, i.e., $\|\dot{\nu}\| \leq \eta_2$. Taking the time-derivative of $\dot{\nu}$ yields $\ddot{\nu} = \frac{d}{dt}(\text{proj}\{W^T\sigma(V^Tx),\nu\}) = \frac{\partial}{\partial y}(\text{proj}\{y,\nu\})\Big|_{y=W^T\sigma(V^Tx)} \frac{d}{dt}(W^T\sigma(V^Tx)) + \frac{\partial}{\partial y}(\text{proj}\{W^T\sigma(V^Tx),y\})\Big|_{y=\nu} \dot{\nu}$. Based on the structure of the projection operator in [90, Eq. 7], the terms $\frac{\partial}{\partial y}(\text{proj}\{y,\nu\})\Big|_{y=W^T\sigma(V^Tx)}$ and $\frac{\partial}{\partial y}(\text{proj}\{W^T\sigma(V^Tx),y\})\Big|_{y=\nu}$ can be bounded by some known functions of x. Additionally, based on the right-hand-side of (5–2), the term $\frac{d}{dt}(W^T\sigma(V^Tx)) = W^T \frac{\partial}{\partial y}\sigma(y)\Big|_{y=V^Tx} V^T\dot{x} = W^T \frac{\partial}{\partial y}\sigma(y)\Big|_{y=V^Tx} V^T(\dot{x}_d + r - \alpha e)$ can be bounded by some known continuous function of z. Therefore, $\ddot{\nu}$ can be bounded as $\|\ddot{\nu}\| \leq \eta_3 + \eta_4\|z\| + \rho_{21}(\|z\|)\|z\|$. Thus, the dynamic neural network in (5–7) satisfies Assumption 5.1.

The structure of the closed-loop error system in (5–6) may appear similar to a higher order sliding-mode design (cf., [91]); however, there are some remarkable differences to highlight. Specifically, the $\beta\text{sgn}(e)$ term in (5–6) would need to be $\beta\text{sgn}(r)$ to facilitate the analysis for a standard continuous higher-order sliding-mode design. Since sensor measurements for the highest order derivative (e.g., $\dot{e}$ or r) may not be available for feedback, the controller in (5–3) is designed to depend only on state measurements. Additionally, the closed-loop error system in (5–2) and (5–6) is also different from a super-twisting system, since (5–2) would require an additional $-|e|^{1/2}\text{sgn}(e)$ term to facilitate a super-twisting design, which needs a different analysis approach [92].

Some supporting lemmas are now presented which facilitate the subsequent analysis. Proofs of all lemmas can be found in the Appendix.

[2] See [49] for continuous adaptive weight updates.

Lemma 5.1. *Given some Filippov solutions, e and r, to (5–2) and (5–6), respectively, the set of time-instants $T \triangleq \{t \in \mathcal{I} | \exists i \in \{1, 2, \ldots, n\} \, s.t. \, e_i(t) = 0 \ \wedge r_i(t) \neq 0\}$ has Lebesgue measure zero, where e_i and r_i denote the i^{th} element of e and r, respectively.*

Lemma 5.2. *Given some Filippov solution, e, to (5–2), $sgn(e(\cdot))$ is Riemann integrable on $[t_0, t_1]$, $\forall t_1 \in \mathcal{I}$.*

5.2 Stability Analysis

Following the development in Section 5.1, every Filippov solution to (5–1) and (5–4) with the controller in (5–3) corresponds to a Filippov solution of the transformed system in (5–2) and (5–6). Additionally, a P-function is introduced to facilitate the construction of a candidate Lyapunov function for analyzing the stability and convergence properties of z. The P-function is denoted by $P : \mathcal{I} \to \mathbb{R}$ and is defined as a Filippov solution to

$$\dot{P} = -\lambda_P P - L, \tag{5–8}$$

where $\lambda_P \in \mathbb{R}_{>0}$ is an auxiliary constant, and

$$\begin{aligned} L \triangleq \ & r^T N_B - r^T \beta \operatorname{sgn}(e) \\ & - (\gamma_4 + \rho_2(\|z\|)) \|z\| \|e\|_1, \end{aligned} \tag{5–9}$$

where $\|\cdot\|_1$ denotes the 1-norm, and

$$P(t_0) = \beta \|e(t_0)\|_1 - e^T(t_0) N_B(t_0). \tag{5–10}$$

The analytical solution to (5–8) is derived in Lemma 3. To facilitate the inclusion of the P-function in the candidate Lyapunov function, P is designed to be non-negative under certain gain conditions as described in Lemma 5.4.

Lemma 5.3. *Given some Filippov solutions, e and r, to (5–2) and (5–6), respectively,*

$$\begin{aligned} P = \ & \beta \|e\|_1 - e^T N_B \\ & + \mathrm{e}^{-\lambda_P t} * \left((\alpha - \lambda_P) \left(\beta \|e\|_1 - e^T N_B \right) + e^T \dot{N}_B \right) \end{aligned}$$

$$+\mathrm{e}^{-\lambda_P t} * \left(\left(\gamma_4 + \rho_2 \left(\|z\| \right) \right) \|z\| \, \|e\|_1 \right), \tag{5–11}$$

is the unique Filippov solution to the differential equation in (5–8) initialized according to (5–10), where '$*$' denotes the convolution operator, i.e., $p(t) * q(t) = \int_{t_0}^{t} p(t-\tau)q(\tau)d\tau$, for any given $p, q : [t_0, \infty) \to \mathbb{R}$.

Lemma 5.4. *Given any pair of Filippov solutions, e and r, to (5–2) and (5–6), respectively, provided that P is initialized according to (5–10), and the gain conditions*

$$\alpha > \lambda_P, \tag{5–12}$$

$$\beta > \gamma_1 + \frac{\gamma_3}{\alpha - \lambda_P}, \tag{5–13}$$

are satisfied, $P(t) \geq 0, \forall t \in \mathcal{I}$, where the gains α and β are introduced in (5–2) and (5–4), respectively, and γ_1, γ_4 are introduced in Assumption 1.

Let $\psi \triangleq \begin{bmatrix} e^T & r^T & P \end{bmatrix}^T$, and $\dot{\psi} = g(\psi, t)$ denote the differential equations in (5–2), (5–6) and (5–8), where $g : \mathbb{R}^{2n+1} \times [t_0, \infty) \to \mathbb{R}^{2n+1}$ is Lebesgue measurable and locally essentially bounded (i.e., bounded on a neighborhood of every point, excluding sets of measure zero), since it is continuous except in the set $\{(\psi, t) \in \mathbb{R}^{2n+1} \times [t_0, \infty) | e = 0\}$. To facilitate the stability analysis, let $V_L : \mathbb{R}^{2n+1} \to \mathbb{R}_{\geq 0}$ be defined as

$$V_L(\psi) \triangleq \frac{1}{2} e^T e + \frac{1}{2} r^T r + P. \tag{5–14}$$

Let $c \triangleq \min\{k - \gamma_2 - n\gamma_4, \alpha - \gamma_2 - n\gamma_4, \frac{\lambda_P}{2}\}$, $\rho(\cdot) \triangleq \rho_1(\cdot) + n\rho_2(\cdot)$, and consider the regions, $\mathcal{D} \triangleq \{\sigma \in \mathbb{R}^{2n+1} | V_L(\sigma) < \frac{(\rho^{-1}(c-\lambda_V))^2}{2}\}$ and $\mathcal{S} \triangleq \{\sigma \in \mathbb{R}^{2n} | \, \|\sigma\| < c - \lambda_V\}$, where $\lambda_V \in \mathbb{R}_{>0}$ is a user-defined constant.

Theorem 5.1. *Let $W : \mathbb{R}^{2n} \to \mathbb{R}_{\geq 0}$ be defined as $W(z(t_0)) \triangleq \sqrt{\|z(t_0)\|^2 + 2(\beta + \gamma_1)\|z(t_0)\|_1}$. Given any initial condition $z(t_0) \in \mathbb{R}^{2n}$, every maximal solution to (5–2), (5–6), and (5–8) with $P(t_0)$ initialized according to (5–10) is complete, and the zero solution to (5–2) and (5–6), $(e(t), r(t)) \equiv (0, 0)$, is semi-globally exponentially stable in the sense that $\|z(t)\| \leq W(z(t_0)) \exp(-\lambda_V (t - t_0))$,*

$\forall (z(t_0), t) \in \mathbb{R}^{2n} \times [t_0, \infty)$*, provided that the gains* α, β, k *and* λ_P *are selected according to the gain conditions in (5–12), (5–13), and*

$$c > \lambda_V + \rho\left(W(z(t_0))\right). \tag{5–15}$$

Proof. The existence of a Filippov solution, $\psi : \mathcal{I} \to \mathbb{R}^{2n+1}$, to $\dot{\psi} = g(\psi, t)$ is guaranteed[3] by [69, Proposition 3]. The time-derivative of V_L along ψ, starting from the specified initial conditions, exists a.e., and $\dot{V}_L(\psi, t) \overset{a.e.}{\in} \dot{\tilde{V}}_L(\psi, t)$ [87, Equations 12 and 13], where the notation $\overset{a.e.}{(\cdot)}$ implies that the relation holds for almost all time $t \in \mathcal{I}$, and

$$\begin{aligned} \dot{\tilde{V}}_L(\psi, t) &= \bigcap_{\xi \in \partial V_L(\psi)} \xi^T K\left[g\right](\psi, t) \\ &= \nabla V_L^T K\left[g\right](\psi, t) \\ &= \begin{bmatrix} z^T & 1 \end{bmatrix}^T K\left[g\right](\psi, t). \end{aligned} \tag{5–16}$$

In (5–16), $\partial V_L(\psi)$ denotes Clarke's generalized gradient [87, Equation 7], and $K[\cdot]$ is defined in [67, Equation 2b]. Since $\psi \mapsto V_L(\psi)$ is continuously differentiable, $\partial V_L = \{\nabla V_L\}$ using [69, Proposition 6]. Substituting (5–2), (5–6), and (5–8) into (5–16), utilizing $\|\cdot\|_1 \leq \sqrt{n}\,\|\cdot\|$, and applying Young's inequality on $\|z\|\,\|e\|_1$ yields

$$\begin{aligned} \dot{\tilde{V}}_L &= r^T(\widetilde{N} + N_B - kr - e - \beta K\left[\operatorname{sgn}\right](e)) + e^T(r - \alpha e) - \lambda_P P \\ &\quad -r^T N_B + r^T \beta K\left[\operatorname{sgn}\right](e) + \left(\gamma_4 + \rho_2\left(\|z\|\right)\right)\|z\|\,\|e\|_1 \\ &\leq -k\,\|r\|^2 - \alpha\,\|e\|^2 + \left(\gamma_2 + \rho_1\left(\|z\|\right)\right)\|r\|\,\|z\| \\ &\quad -\lambda_P P + \left(\gamma_4 + \rho_2\left(\|z\|\right)\right)\|z\|\,\|e\|_1 \\ &\leq -2\left(c - \rho\left(\|z\|\right)\right) V_L \end{aligned}$$

[3] The solution ψ may not be unique; however, P is unique according to Lemma 5.3 for a given pair (e, r). Moreover, the results in this dissertation are applicable to all the trajectories even when ψ is non-unique, since a generalized Filippov solution is considered in the analysis.

$$\leq \; -2\left(c - \rho\left(\sqrt{2V_L}\right)\right) V_L, \tag{5–17}$$

for almost all $t \in \mathcal{I}$, where c and ρ are introduced before the theorem statement, the term $t \mapsto r^T(t)\beta K\left[\text{sgn}\right](e(t))$ is set-valued only for the set of time instants $T = \{t \in [t_0, \infty) | \exists i \in \{1, 2, \ldots, n\} \text{ s.t. } e_i(t) = 0 \wedge r_i(t) \neq 0\}$. According to Lemma 5.1, the set T has Lebesgue measure zero. It follows from (5–17) that V_L is non-increasing along all trajectories initialized such that $\psi(t_0) \in \mathcal{D}$. Selecting c according to (5–15) and using (5–10) yields $c > \lambda_V + \|z(t_0)\| = \lambda_V + \rho(\sqrt{2V_L(\psi(t_0))})$. Then, $V_L(\psi(t))$ is non-increasing, implying $\psi(t) \in \mathcal{D}$, $\forall t \in \mathcal{I}$. It follows from (5–17) and (5–15) that $\dot{V}_L$ can be upper-bounded as

$$\dot{V}_L \leq -2\lambda_V V_L, \tag{5–18}$$

for almost all $t \in \mathcal{I}$. Using the comparison principle [93, Lemma 4.4] in (5–18) yields

$$V_L(\psi(t)) \leq V_L(\psi(t_0)) \exp\left(-2\lambda_V (t - t_0)\right), \tag{5–19}$$

$\forall (\psi(t_0), t) \in \mathcal{D} \times \mathcal{I}$. Since $(\psi, t) \mapsto K\left[g\right](\psi, t)$ is locally bounded over $\mathbb{R}^{2n+1} \times [t_0, \infty)$ and (5–19) implies that ψ is precompact, then [94, Lemma 3.3 and Remark 3.4] can be invoked to show that every maximal solution ψ with $P(t_0)$ initialized according to (5–10) is complete, i.e., $\mathcal{I} = [t_0, \infty)$. Using the definition of z, (5–19) and Lemma 5.4, $V_L(\psi(t)) = \frac{1}{2}\|z(t)\|^2 + P(t) \geq \frac{1}{2}\|z(t)\|^2$, $\forall t \in [t_0, \infty)$. Therefore,

$$\|z(t)\| \leq \sqrt{2V_L(\psi(t))}, \; \forall t \in [t_0, \infty). \tag{5–20}$$

Using (5–19) and (5–20), $\|z(t)\|$ can further be upper-bounded as

$$\|z(t)\| \leq \sqrt{2V_L(\psi(t_0))} \exp\left(-\lambda_V (t - t_0)\right), \tag{5–21}$$

$\forall (\psi(t_0), t) \in \mathcal{D} \times [t_0, \infty)$. Moreover, substituting (5–10) in the expression for $V_L(\psi(t_0))$ yields $V_L(\psi(t_0)) = \frac{1}{2}\|z(t_0)\|^2 + P(t_0)$. Consequently, $\psi(t_0) \in \mathcal{D}$ implies $z(t_0) \in \mathcal{S}$. Using

(5–21) yields

$$\|z(t)\| \leq W(z(t_0)) \exp\left(-\lambda_V (t - t_0)\right), \tag{5–22}$$

$\forall (z(t_0), t) \in \mathcal{S} \times [t_0, \infty)$, implying the zero solution to (5–2) and (5–6), $(e(t), r(t)) \equiv (0, 0)$, is semi-globally exponentially stable. Note that the exponential stability result is semi-global (cf., [95, Remark 2]) because the size of the set $\mathcal{S}$ can be arbitrarily increased using (5–15) to include any $z(t_0) \in \mathbb{R}^{2n}$. Moreover, $x \in \mathcal{L}_\infty$ since $e, x_d \in \mathcal{L}_\infty$. Since $d \in \mathcal{L}_\infty$ by Assumption 5.1, it follows from (5–5) that $\hat{d} \in \mathcal{L}_\infty$. Therefore, since all the terms on the right hand side of (5–3) are bounded and continuous, $u \in \mathcal{L}_\infty$ and is continuous. Moreover, since $e(t) = e(t_0) + \int_{t_0}^{t} (r(\tau) - \alpha e(\tau))\, d\tau$ for all $t \in [t_0, \infty)$ using (5–2), the continuity of $r - \alpha e$ implies that the Filippov solution e is also continuously differentiable. □

Remark 5.1. The relation in (5–5) indicates that r is the estimation error between the RISE term $\hat{d}(t)$ and the uncertainty $d(x, \nu, t)$. Therefore, (5–22) implies that the RISE term is an exponentially convergent estimator of the uncertainty, i.e., $\hat{d}(t) \to d(x, \nu, t)$ with a uniform and exponential convergence as $t \to \infty$.

Remark 5.2. For the special case when $\dot{d}(t)$ and $\ddot{d}(t)$ are bounded by known constants, the analysis approach in [55] can also be considered.

Remark 5.3. The exponential stability result is global when the bounds on $\widetilde{N}$ and $\dot{N}_B$ are linear in $\|z\|$, i.e., $\rho_1 = \rho_2 = 0$.

5.3 Conclusion

A contribution of this chapter is the development of new insights for the construction of a P-function to yield exponential stability with RISE-based controllers. As an outcome of this breakthrough, the inherent learning capability of RISE-based controllers is shown to yield exponential identification of disturbances/uncertainty, as compared to all previous asymptotic results. Future work could involve extension of the proposed stability analysis methodology for RISE-based error systems with sensor noise, and delays in input and state measurements.

Appendix: Proofs of Lemmas

Proof of Lemma 5.1. The set T can also be represented as $T = \{t \in \mathcal{I} | \exists i \in \{1,2,\ldots,n\} \text{ s.t. } e_i(t) = 0 \wedge r_i(t) - \alpha e_i(t) \neq 0\} = \{t \in \mathcal{I} | \exists i \in \{1,2,\ldots,n\} \text{ s.t. } e_i(t) = 0 \wedge \dot{e}_i(t) \neq 0\}$ to facilitate the subsequent analysis using the dynamics in (5–6). Select $a \in T$, which yields $e_i(a) = 0$ using the definition of T. Given $\dot{e}_i(a) \neq 0$, one can assume without loss of generality that $\dot{e}_i(a) > 0$; the proof easily extends for $\dot{e}_i(a) < 0$. Since e and r are absolutely continuous, it follows from (5–2) that $\dot{e}$ is continuous. Given $\dot{e}_i(a) > 0$ and continuity of $\dot{e}$, there exists $\delta > 0$ such that $\dot{e}_i(t) > 0$ for all $t \in (a-\delta, a+\delta)$. Based on $\dot{e}_i(t) > 0$, it follows that $\int_a^t \dot{e}_i(\tau) d\tau > 0$ for all $t \in (a, a+\delta)$. Then, using $e_i(a) = 0$ yields $\int_a^t \dot{e}_i(\tau) d\tau = e_i(t) - e_i(a) = e_i(t) > 0$. Similarly, $-\int_t^a \dot{e}_i(\tau) d\tau = e_i(t) - e_i(a) = e_i(t) < 0$ for all $t \in (a-\delta, a)$, implying $e_i(t) \neq 0$ for all $t \in (a-\delta, a+\delta) \setminus \{a\}$. If more than one component has $e_i(a) = 0$, the intersection of each neighborhood found above can be selected and represented by $U(a)$. Therefore, there exists a neighborhood, $U(a)$, for any $a \in T$, s.t. $e_i(t) \neq 0$ for all time-instants $t \in U(a) \setminus \{a\}$ and $i \in \{1,2,\ldots,n\}$. When $t \in T$, $e_i(t) = 0$ for some $i \in \{1,2,\ldots,n\}$, which implies $U(a) \cap T = \{a\}$ for all $a \in T$; therefore, T is discrete and consequently has measure zero. □

Proof of Lemma 5.2. The function $\text{sgn}(e(\cdot))$ is discontinuous only at time-instants where it changes sign, i.e., the set $\{t \in \mathcal{I} | \exists i \in \{1,2,\ldots,n\} \text{ s.t. } e_i(t) = 0 \wedge \dot{e}_i(t) \neq 0\} = \{t \in \mathcal{I} | \exists i \in \{1,2,\ldots,n\} \text{ s.t. } e_i(t) = 0 \wedge r_i(t) - \alpha e_i(t) \neq 0\} = T$. Since T has Lebesgue measure zero according to Lemma 5.1, $\text{sgn}(e(\cdot))$ is continuous a.e., implying it is Riemann integrable [96, Theorem 11.33] on $[t_0, t_1]$, $\forall t_1 \in \mathcal{I}$. □

Proof of Lemma 5.3. The right hand side (RHS) of (5–11) is almost everywhere (a.e.) differentiable with respect to time, because every term on the RHS is absolutely continuous, including $\|e\|_1$, since $\|\cdot\|_1$ is globally Lipschitz and e is absolutely continuous. The time-derivative of $\|e\|_1$, whenever it exists, is $\dot{e}^T \text{sgn}(e)$, using the chain rule. Therefore, taking the time-derivative of both the sides of (5–11) at points where P is differentiable

yields

$$\dot{P} \overset{a.e.}{=} \beta \dot{e}^T \text{sgn}(e) - \dot{e}^T N_B - e^T \dot{N}_B + \frac{d}{dt}(\mathrm{e}^{-\lambda_P t} * ((\alpha - \lambda_P)(\beta \|e\|_1 - e^T N_B) + e^T \dot{N}_B)) + \frac{d}{dt}\left(\mathrm{e}^{-\lambda_P t} * ((\gamma_4 + \rho_2(\|z\|))\|z\|\,\|e\|_1)\right). \tag{5–23}$$

Based on the Leibniz rule, for any given $q : [t_0, \infty) \to \mathbb{R}$, the function $\mathrm{e}^{-\lambda_P t}$ satisfies $\frac{d}{dt}\left(\mathrm{e}^{-\lambda_P t} * q\right) = \frac{d}{dt}\left(\int_{t_0}^{t} \mathrm{e}^{-\lambda_P (t-\tau)} q(\tau) d\tau\right) = q(t) - \lambda_P \int_{t_0}^{t} \mathrm{e}^{-\lambda_P (t-\tau)} q(\tau) d\tau = -\lambda_P \mathrm{e}^{-\lambda_P t} * q + q$. Additionally, $L = \dot{e}^T N_B + \alpha e^T N_B - \beta \dot{e}^T \text{sgn}(e) - \alpha\beta \|e\|_1 - (\gamma_4 + \rho_2(\|z\|))\|z\|\,\|e\|_1$ is obtained after substituting (5–2) into (5–9). Therefore, the expression for $\dot{P}$ in (5–23) can be rewritten as

$$\begin{aligned} \dot{P} &\overset{a.e.}{=} -\lambda_P \mathrm{e}^{-\lambda_P t} * ((\gamma_4 + \rho_2(\|z\|))\|z\|\,\|e\|_1) \\ &\quad -\lambda_P \mathrm{e}^{-\lambda_P t} * ((\alpha - \lambda_P)(\beta \|e\|_1 - e^T N_B) + e^T \dot{N}_B) \\ &\quad -\lambda_P \beta \|e\|_1 + \lambda_P e^T N_B + \beta \dot{e}^T \text{sgn}(e) - \dot{e}^T N_B + \alpha\beta \|e\|_1 - \alpha e^T N_B \\ &\quad + (\gamma_4 + \rho_2(\|z\|))\|z\|\,\|e\|_1 \\ &= -\lambda_P P - L. \end{aligned} \tag{5–24}$$

Filippov's differential inclusion for (5–8) is given by

$$\dot{P} \in -\lambda_P P - K[L]. \tag{5–25}$$

To prove the uniqueness of the solution to (5–25), consider any two solutions with the same initial conditions, i.e., P_1 and P_2, with $P_1(t_0) = P_2(t_0) = 0$, implying

$$\dot{P}_1 \overset{a.e.}{\in} -\lambda_P P_1 - K[L], \tag{5–26}$$

$$\dot{P}_2 \overset{a.e.}{\in} -\lambda_P P_2 - K[L]. \tag{5–27}$$

Based on (5–9), $t \mapsto K[L](\psi(t))$ is set-valued only when there exists some $i \in \{1, 2, \ldots, n\}$ such that $t \to K[\text{sgn}](e_i(t))$ is set-valued and $r_i(t) \neq 0$. Using Lemma 5.1,

$t \mapsto K\left[L\right]\left(\psi(t)\right)$ is set-valued only for a set of time-instants of measure zero. Therefore, defining $\Delta(t) = P_2(t) - P_1(t)$, and using (5–26) and (5–27) yields

$$\dot{\Delta} \overset{a.e.}{=} -\lambda_P \Delta, \tag{5–28}$$

with $\Delta(t_0) = 0$. Since $\Delta \equiv 0$ is an equilibrium point of (5–28), $\Delta(t_0) = 0$ implies $\|\Delta(t)\| = 0$, $\forall t \in \mathcal{I}$, therefore $P_1(t) = P_2(t)$, $\forall t \in [t_0, \infty)$, i.e., any two solutions are equal, implying the solution is unique. □

Proof of Lemma 5.4. Using Holder's inequality and Assumption 1 yields lower bounds on $-e^T N_B$ and $e^T \dot{N}_B$,

$$-e^T N_B \geq -\|e\|_1 \|N_B\|_1 \geq -\gamma_1 \|e\|_1, \tag{5–29}$$

and

$$\begin{aligned} e^T \dot{N}_B &\geq -\|e\|_1 \left\|\dot{N}_B\right\|_1 \\ &\geq -\left(\gamma_3 + \gamma_4 \|z\| + \rho_2\left(\|z\|\right)\|z\|\right)\|e\|_1, \end{aligned} \tag{5–30}$$

$\forall t \in \mathcal{I}$. Substituting the bounds in (5–29) and (5–30) into the expression for P in (5–11) yields

$$P \geq \beta \|e\|_1 - \gamma_1 \|e\|_1 + \mathrm{e}^{-\lambda_P t} * \left(\left(\left(\alpha - \lambda_P\right)\left(\beta - \gamma_1\right) - \gamma_3\right)\|e\|_1\right). \tag{5–31}$$

Selecting $P(t_0)$ according to (5–10), and α and β according to the gain conditions (5–12) and (5–13) yields $P(t) \geq 0$, $\forall t \in \mathcal{I}$ using (5–31). □

CHAPTER 6
CONCLUSIONS AND FUTURE WORK

Adaptive and learning-based control strategies are a powerful tool to compensate for nonlinear modeling uncertainties in numerous practical engineering systems in real-time. These strategies are designed by incorporating an approximate feedforward model into the controller, or by implicitly learning the model through the feedback structure. This dissertation focused on developing such adaptive and learning-based controllers. Since the dissertation concentrates on general nonlinear systems, a constructive Lyapunov-based design and analysis approach is employed.

In Chapter 2, a fully-connected feedforward DNN-based adaptive controller was developed where Lyapunov-based real-time weight update laws were designed for each layer of a feedforward DNN. Additionally, the developed method also allows nonsmooth activation functions to be used in the DNN architecture. A nonsmooth Lyapunov-based stability analysis was provided to guarantee global asymptotic tracking error convergence. Simulation results were provided for a nonlinear system using DNNs involving leaky ReLU and hyperbolic tangent activation functions to demonstrate the efficacy of the developed method. Although adapting for more layers might cause initial overshoot in the tracking error, the developed method provides tenfold and twofold improvement in steady-state function estimation as compared to offline pre-training and output-layer adaptation, respectively.

Chapter 3 provided the first result on Lyapunov-derived weight adaptation for a ResNet-based adaptive controller. A nonsmooth Lyapunov-based analysis was provided to guarantee global asymptotic tracking error convergence. Comparative Monte Carlo simulations were provided to demonstrate the performance of the developed ResNet-based adaptive controller. The ResNet-based adaptive controller showed a 49.52% and 54.38% improvement in the tracking and function approximation performance, respectively, in comparison to a fully-connected DNN-based adaptive controller.

In Chapter 4, the problem of adaptive control of systems with uncertain time-varying parameters was addressed. Through a unique analysis strategy, an adaptive feedforward term was developed along with specialized feedback terms to compensate for the time-varying uncertainty. Asymptotic tracking error convergence was guaranteed via a Lyapunov-based stability analysis for an Euler-Lagrange system. Additionally, the time-varying uncertain function approximation error was shown to converge to zero. A simulation example of a two-link manipulator was provided to demonstrate the asymptotic tracking result, and a comparison with the e-mod scheme shows a better tracking performance with the proposed method.

Chapter 5 provided new stability results for a class of implicit learning controllers called Robust Integral of the Sign of the Error (RISE) controllers. RISE controllers have been published over the past two decades as a means to yield asymptotic tracking error convergence and implicit asymptotic identification of time-varying uncertainties, for classes of nonlinear systems that are subject to sufficiently smooth bounded exogenous disturbances and/or modeling uncertainties. Despite the wide application of RISE-based techniques, an open question that had eluded researchers during this time-span is whether the asymptotic tracking error convergence is also uniform or exponential. This question was open due to certain limitations in the traditional construction of a Lyapunov function for RISE-based error systems. In this dissertation, new insights on the construction of a Lyapunov function were used that resulted in an exponential stability result for RISE-based controllers. As an outcome of this breakthrough, the inherent learning capability of RISE-based controllers is now known to yield exponential identification of state-dependent disturbances/uncertainty.

All of these chapters provided promising results on compensating for nonlinear modeling uncertainties using adaptive and learning-based control techniques. Based on these developments, the following open problems can be explored in the future. Although Chapter 2 and Chapter 3 provide results for DNN-based adaptive controllers,

they do not provide results on how to select the configuration for the DNN architecture. Thus, neural architecture search methods can be explored in the future to select the optimal DNN architecture that yields the best performance.

Additionally, the results in Chapter 2 and Chapter 3 only guarantee asymptotic tracking error convergence, without providing any guarantees on the weight estimation performance. Recent results in [97] and [98] develop an adaptation mechanism called concurrent learning, which yields parameter estimation error convergence, provided the system is sufficiently excited for finite time. However, these results are limited to linearly parameterized systems, and applying this technique for nonlinear parameterizations such as DNNs remains an open problem. In future work, the mathematical ideas used in Chapter 2 and Chapter 3 can potentially be leveraged to develop a concurrent learning-based adaptation algorithm for DNNs, which can yield results on weight estimation performance.

Moreover, the results in Chapter 2 and Chapter 3 only consider feedforward architectures such as fully-connected DNN and ResNet, and cannot be applied for recurrent neural network (RNN) architectures. Unlike feedforward DNNs, RNNs are a dynamic mapping, i.e., they involve a feedback loop and thus have an internal memory. This internal memory allows RNNs to capture time-varying, accumulative effects exhibited in some dynamical systems that feedforward DNNs cannot. Specifically, one type of RNN called long short-term memory (LSTM) neural network is known to provide enhanced approximation performance by retaining relevant information and forget irrelevant information in the memory. Thus, the development of Lyapunov-based adaptation laws can be explored for RNN architectures such as the LSTM network in future work.

Additionally, the results in Chapter 2 and Chapter 3 involve deterministic uncertainties in the system; thus, the results do not apply for stochastic systems. The stochastic

stability results developed in [99] can potentially be used along with the technique developed in Chapter 2 and Chapter 3 to develop DNN-based adaptive controllers that compensate for stochastic uncertainties in the system.

Furthermore, the results in Chapter 2 and Chapter 3 do not provide any optimality guarantees on the developed adaptive controller. An approximate optimal control technique called adaptive dynamic programming (ADP) provides a solution to the optimal control problem using adaptive actor-critic neural networks (cf. [100–102]). However, existing ADP results only use single-layer actor-critic neural networks. The development of ADP-based controllers with deep actor-critic networks with real-time adaptation for inner layers remains an open problem that can be addressed in the future.

Although Chapter 5 provides exponential stability results for general control-affine nonlinear systems, the development does not account for actuator saturation. However, most practical engineering systems involve actuator saturation. Thus, the development of a RISE controller that yields exponential stability guarantees despite the actuator saturation can be addressed in future work.

REFERENCES

[1] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, pp. 251–257, 1991.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] D. Rolnick and M. Tegmark, "The power of deeper networks for expressing natural functions," in *Int. Conf. Learn. Represent.*, 2018.

[4] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*. MIT press Cambridge, 2016, vol. 1.

[5] B. Karg and S. Lucia, "Efficient representation and approximation of model predictive control laws via deep learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3866–3878, 2020.

[6] M. Hertneck, J. Köhler, S. Trimpe, and F. Allgöwer, "Learning an approximate model predictive controller with guarantees," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 543–548, 2018.

[7] J. Nubert, J. Köhler, V. Berenz, F. Allgöwer, and S. Trimpe, "Safe and fast tracking on a robot manipulator: Robust MPC and neural network control," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3050–3057, 2020.

[8] F. L. Lewis, "Nonlinear network structures for feedback control," *Asian J. Control*, vol. 1, no. 4, pp. 205–228, 1999.

[9] G. Joshi and G. Chowdhary, "Deep model reference adaptive control," in *Proc. IEEE Conf. Decis. Control*. IEEE, 2019, pp. 4601–4608.

[10] G. Joshi, J. Virdi, and G. Chowdhary, "Asynchronous deep model reference adaptive control," in *Conf. Robot Learn.*, 2020.

[11] R. Sun, M. Greene, D. Le, Z. Bell, G. Chowdhary, and W. E. Dixon, "Lyapunov-based real-time and iterative adjustment of deep neural networks," *IEEE Control Syst. Lett.*, vol. 6, pp. 193–198, 2022.

[12] D. Le, M. Greene, W. Makumi, and W. E. Dixon, "Real-time modular deep neural network-based adaptive control of nonlinear systems," *IEEE Control Syst. Lett.*, vol. 6, pp. 476–481, 2022.

[13] O. Patil, D. Le, M. Greene, and W. E. Dixon, "Lyapunov-derived control and adaptive update laws for inner and outer layer weights of a deep neural network," *IEEE Control Syst Lett.*, vol. 6, pp. 1855–1860, 2022.

[14] A. L. Maas, A. Y. Hannun, A. Y. Ng *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Int. Conf. Mach. Learn.*, vol. 30, no. 1. Citeseer, 2013, p. 3.

[15] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," in *Int. Conf. Mach. Learn.* PMLR, 2013, pp. 1319–1327.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[17] M. Hardt and T. Ma, "Identity matters in deep learning," *Int. Conf. Learn. Represent.*, 2017.

[18] K. Nar and S. Sastry, "Residual networks: Lyapunov stability and convex decomposition," *arXiv preprint arXiv:1803.08203*, 2018.

[19] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3147–3155.

[20] J. Li, F. Fang, K. Mei, and G. Zhang, "Multi-scale residual network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 517–532.

[21] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 5, pp. 1181–1193, 2018.

[22] T. Tan, Y. Qian, H. Hu, Y. Zhou, W. Ding, and K. Yu, "Adaptive very deep convolutional residual network for noise robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 8, pp. 1393–1405, 2018.

[23] O. S. Patil, D. M. Le, E. Griffis, and W. E. Dixon, "Deep residual neural network (ResNet)-based adaptive control: A Lyapunov-based approach," in *Proc. IEEE Conf. Decis. Control*, 2022.

[24] G. Kreisselmeier, "Adaptive control of a class of slowly time-varying plants," *Sys. control lett.*, vol. 8, no. 2, pp. 97–103, 1986.

[25] R. H. Middleton and G. C. Goodwin, "Adaptive control of time-varying linear systems," *IEEE Trans. Autom. Control*, vol. 33, no. 2, pp. 150–155, 1988.

[26] K. Tsakalis and P. Ioannou, "Adaptive control of linear time-varying plants," *Automatica*, vol. 23, no. 4, pp. 459–468, 1987.

[27] J. E. Gaudio, A. M. Annaswamy, E. Lavretsky, and M. Bolender, "Parameter estimation in adaptive control of time-varying systems under a range of excitation conditions," *IEEE Trans. Autom. Control*, 2021.

[28] E. Arabi and T. Yucelen, “Set-theoretic model reference adaptive control with time-varying performance bounds,” *Int. J. Control*, vol. 92, no. 11, pp. 2509–2520, 2019.

[29] R. Marino and P. Tomei, “Robust adaptive regulation of linear time-varying systems,” *IEEE Trans. Autom. Control*, vol. 45, no. 7, pp. 1301–1311, 2000.

[30] ——, “Adaptive control of linear time-varying systems,” *Automatica*, vol. 39, no. 4, pp. 651–659, 2003.

[31] J.-X. Xu, “A new periodic adaptive control approach for time-varying parameters with known periodicity,” *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 579–583, Apr. 2004.

[32] Z. Qu and J. X. Xu, “Model-based learning controls and their comparisons using Lyapunov direct method,” in *Asian Journal of Control*, vol. 4, No. 1, no. No. 1, Mar. 2002, pp. 99–110.

[33] Z. Zhang, X.-J. Xie, and S. S. Ge, “Adaptive tracking for uncertain mimo nonlinear systems with time-varying parameters and bounded disturbance,” *IEEE Trans. Syst., Man, Cybern., Syst.*, 2019.

[34] K. Chen and A. Astolfi, “Adaptive control for systems with time-varying parameters,” *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 1986–2001, 2020.

[35] H. Ríos, D. Efimov, J. A. Moreno, W. Perruquetti, and J. G. Rueda-Escobedo, “Time-varying parameter identification algorithms: Finite and fixed-time convergence,” *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3671–3678, 2017.

[36] S. Dahliwal and M. Guay, “Set-based adaptive estimation for a class of nonlinear systems with time-varying parameters,” *J. Process Control*, vol. 24, no. 2, pp. 479–486, 2014.

[37] E. Moshksar and M. Guay, “Almost invariant manifold approach for adaptive estimation of periodic and aperiodic unknown time-varying parameters,” *Int. J. Adapt. Control Signal Process.*, vol. 30, no. 1, pp. 76–92, 2016.

[38] R. Ortega, S. Aranovskiy, A. A. Pyrkin, A. Astolfi, and A. A. Bobtsov, “New results on parameter estimation via dynamic regressor extension and mixing: Continuous and discrete-time cases,” *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2265–2272, 2021.

[39] G. Tao, S. M. Joshi, and X. Ma, “Adaptive state feedback and tracking control of systems with actuator failures,” *IEEE Trans. Autom. Control*, vol. 46, no. 1, pp. 78–95, 2001.

[40] J. E. Gaudio, A. M. Annaswamy, E. Lavretsky, and M. A. Bolender, “Parameter estimation in adaptive control of time-varying systems under a range of excitation conditions,” *arXiv preprint arXiv:1911.03810*, 2019.

[41] O. S. Patil, R. Sun, S. Bhasin, and W. E. Dixon, “Adaptive control of time-varying parameter systems with asymptotic tracking,” *arXiv preprint arXiv:2007.11801*, 2020.

[42] B. Xian, D. M. Dawson, M. S. de Queiroz, and J. Chen, “A continuous asymptotic tracking control strategy for uncertain nonlinear systems,” *IEEE Trans. Autom. Control*, vol. 49, no. 7, pp. 1206–1211, 2004.

[43] K. S. Narendra and A. M. Annaswamy, “A new adaptive law for robust adaptive control without persistent excitation,” *IEEE Trans. Autom. Control*, vol. 32, pp. 134–145, 1987.

[44] W. E. Dixon, Y. Fang, D. M. Dawson, and T. J. Flynn, “Range identification for perspective vision systems,” *IEEE Trans. Autom. Control*, vol. 48, pp. 2232–2238, 2003.

[45] Z. Cai, M. S. de Queiroz, and D. M. Dawson, “Robust adaptive asymptotic tracking of nonlinear systems with additive disturbance,” *IEEE Trans. Autom. Control*, vol. 51, pp. 524–529, 2006.

[46] C. Makkar, G. Hu, W. G. Sawyer, and W. E. Dixon, “Lyapunov-based tracking control in the presence of uncertain nonlinear parameterizable friction,” *IEEE Trans. Autom. Control*, vol. 52, pp. 1988–1994, 2007.

[47] B. Xian, M. S. de Queiroz, D. M. Dawson, and M. McIntyre, “A discontinuous output feedback controller and velocity observer for nonlinear mechanical systems,” *Automatica*, vol. 40, no. 4, pp. 695–700, 2004.

[48] P. M. Patre, W. Mackunis, C. Makkar, and W. E. Dixon, “Asymptotic tracking for systems with structured and unstructured uncertainties,” *IEEE Trans. Control Syst. Technol.*, vol. 16, pp. 373–379, 2008.

[49] P. M. Patre, W. MacKunis, K. Kaiser, and W. E. Dixon, “Asymptotic tracking for uncertain dynamic systems via a multilayer neural network feedforward and RISE feedback control structure,” *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2180–2185, 2008.

[50] P. Patre, W. Mackunis, M. Johnson, and W. E. Dixon, “Composite adaptive control for Euler-Lagrange systems with additive disturbances,” *Automatica*, vol. 46, no. 1, pp. 140–147, 2010.

[51] P. Patre, S. Bhasin, Z. D. Wilcox, and W. E. Dixon, “Composite adaptation for neural network-based controllers,” *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 944–950, 2010.

[52] P. Patre, W. Mackunis, K. Dupree, and W. E. Dixon, “Modular adaptive control of uncertain Euler-Lagrange systems with additive disturbances,” *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 155–160, 2011.

[53] N. Fischer, Z. Kan, R. Kamalapurkar, and W. E. Dixon, "Saturated RISE feedback control for a class of second-order nonlinear systems," *IEEE Trans. Autom. Control*, vol. 59, no. 4, pp. 1094–1099, Apr. 2014.

[54] N. Sharma, S. Bhasin, Q. Wang, and W. E. Dixon, "RISE-based adaptive control of a control affine uncertain nonlinear system with unknown state delays," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 255–259, Jan. 2012.

[55] B. Zhao, B. Xian, Y. Zhang, and X. Zhang, "Nonlinear robust adaptive tracking control of a quadrotor uav via immersion and invariance methodology," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 2891–2902, 2014.

[56] N. Fischer, D. Hughes, P. Walters, E. Schwartz, and W. E. Dixon, "Nonlinear RISE-based control of an autonomous underwater vehicle," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 845–852, Aug. 2014.

[57] J. Yao, W. Deng, and Z. Jiao, "RISE-based adaptive control of hydraulic systems with asymptotic tracking," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 3, pp. 1524–1531, 2015.

[58] Z. Yao, J. Yao, and W. Sun, "Adaptive RISE control of hydraulic systems with multilayer neural-networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8638–8647, 2018.

[59] T. Dierks and S. Jagannathan, "Neural network control of mobile robot formations using RISE feedback," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 39, no. 2, pp. 332–347, 2008.

[60] B. Bidikli, E. Tatlicioglu, A. Bayrak, and E. Zergeroglu, "A new Robust 'Integral of Sign of Error' feedback controller with adaptive compensation gain," in *Proc. IEEE Conf. Decis. Control*, 2013, pp. 3782–3787.

[61] B. Bidikli, E. Tatlicioglu, and E. Zergeroglu, "A self tuning RISE controller formulation," in *Proc. Am. Control Conf.*, 2014, pp. 5608–5613.

[62] S. Bhasin, R. Kamalapurkar, H. T. Dinh, and W. Dixon, "Robust identification-based state derivative estimation for nonlinear systems," *IEEE Trans. Autom. Control*, vol. 58, no. 1, pp. 187–192, Jan. 2013.

[63] H. T. Dinh, R. Kamalapurkar, S. Bhasin, and W. E. Dixon, "Dynamic neural network-based robust observers for uncertain nonlinear systems," *Neural Netw.*, vol. 60, pp. 44–52, Dec. 2014.

[64] C. Sun, M. Ye, and G. Hu, "Distributed time-varying quadratic optimization for multiple agents under undirected graphs," *IEEE Trans. Autom. Control*, vol. 62, no. 7, pp. 3687–3694, 2017.

[65] O. Patil, A. Isaly, B. Xian, and W. E. Dixon, "Exponential stability with RISE controllers," *IEEE Control Syst. Lett.*, vol. 6, pp. 1592–1597, 2022.

[66] N. Fischer, R. Kamalapurkar, and W. E. Dixon, "LaSalle-Yoshizawa corollaries for nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2333–2338, Sep. 2013.

[67] B. E. Paden and S. S. Sastry, "A calculus for computing Filippov's differential inclusion with application to the variable structure control of robot manipulators," *IEEE Trans. Circuits Syst.*, vol. 34, no. 1, pp. 73–82, Jan. 1987.

[68] D. S. Bernstein, *Matrix mathematics*. Princeton university press, 2009.

[69] J. Cortes, "Discontinuous dynamical systems," *IEEE Control Sys.*, vol. 28, no. 3, pp. 36–73, 2008.

[70] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Conf. Learn. Theory*, 2020, pp. 2306–2327.

[71] F. L. Lewis, A. Yegildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 388–399, Mar. 1996.

[72] R. Kamalapurkar, J. A. Rosenfeld, J. Klotz, R. J. Downey, and W. E. Dixon. (2014) Supporting lemmas for RISE-based control methods. arXiv:1306.3432.

[73] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. New York, NY, USA: John Wiley & Sons, 1995.

[74] R. Kamalapurkar, J. A. Rosenfeld, A. Parikh, A. R. Teel, and W. E. Dixon, "Invariance-like results for nonautonomous switched systems," *IEEE Trans. Autom. Control*, vol. 64, no. 2, pp. 614–627, Feb. 2019.

[75] F. H. Clarke, *Optimization and nonsmooth analysis*. SIAM, 1990.

[76] H. Lin and S. Jegelka, "ResNet with one-neuron hidden layers is a universal approximator," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[77] P. Tabuada and B. Gharesifard, "Universal approximation power of deep residual neural networks via nonlinear control theory," in *Int. Conf. Learn. Represent.*, 2020.

[78] M. de Queiroz, J. Hu, D. Dawson, T. Burg, and S. Donepudi, "Adaptive position/force control of robot manipulators without velocity measurements: Theory and experimentation," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 27-B, no. 5, pp. 796–809, 1997.

[79] A. M. Annaswamy, F. P. Skantze, and A.-P. Loh, "Adaptive control of continuous time systems with convex/concave parametrization," *Automatica*, vol. 34, no. 1, pp. 33–49, 1998.

[80] A. Kojić, A. M. Annaswamy, A.-P. Loh, and R. Lozano, “Adaptive control of a class of nonlinear systems with convex/concave parameterization,” *Syst. Control Lett.*, vol. 37, no. 5, pp. 267–274, 1999.

[81] W. Lin and C. Qian, “Adaptive control of nonlinearly parameterized systems: the smooth feedback case,” *IEEE Trans. Autom. Control*, vol. 47, no. 8, pp. 1249–1266, 2002.

[82] ——, “Adaptive control of nonlinearly parameterized systems: a nonsmooth feedback framework,” *IEEE Trans. Autom. Control*, vol. 47, no. 5, pp. 757–774, May 2002.

[83] Z. Qu, R. A. Hull, and J. Wang, “Globally stabilizing adaptive control design for nonlinearly-parameterized systems,” *IEEE Trans. Autom. Control*, vol. 51, no. 6, pp. 1073–1079, Jun. 2006.

[84] S. B. Roy, S. Bhasin, and I. N. Kar, “Robust gradient-based adaptive control of nonlinearly parametrized plants,” *IEEE Control Syst. Lett.*, vol. 1, no. 2, pp. 352–357, 2017.

[85] F. Lewis, A. Yesildirek, and K. Liu, “Multilayer neural net robot controller: structure and stability proofs,” *IEEE Trans. Neural Netw.*, vol. 7, no. 2, pp. 388–399, 1996.

[86] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*. Boston, MA: Kluwer Academic Publishers, 2002.

[87] D. Shevitz and B. Paden, “Lyapunov stability theory of nonsmooth systems,” *IEEE Trans. Autom. Control*, vol. 39 no. 9, pp. 1910–1914, 1994.

[88] R. Ortega, A. Loría, P. J. Nicklasson, and H. J. Sira-Ramirez, *Passivity-based Control of Euler-Lagrange Systems: Mechanical, Electrical and Electromechanical Applications*. Springer, 1998.

[89] C. Riano-Rios, R. Sun, R. Bevilacqua, and W. E. Dixon, “Aerodynamic and gravity gradient attitude control for cubesats in the presence of environmental and spacecraft uncertainties,” *Acta Astronautica*, vol. 180, pp. 439–450, 2021.

[90] Z. Cai, M. S. de Queiroz, and D. M. Dawson, “A sufficiently smooth projection operator,” *IEEE Trans. Autom. Control*, vol. 51, no. 1, pp. 135–139, Jan. 2006.

[91] A. Levant, “Sliding order and sliding accuracy in sliding mode control,” *Int. J. Control*, vol. 58, no. 6, pp. 1247–1263, 1993.

[92] T. Gonzalez, J. A. Moreno, and L. Fridman, “Variable gain super-twisting sliding mode control,” *IEEE Trans. Autom. Control*, vol. 57, no. 8, pp. 2100–2105, 2011.

[93] Y. Lin, E. D. Sontag, and Y. Wang, “A smooth converse Lyapunov theorem for robust stability,” *SIAM J. Control Optim.*, vol. 34, no. 1, pp. 124–160, 1996.

[94] R. Kamalapurkar, W. E. Dixon, and A. Teel, "On reduction of differential inclusions and lyapunov stability," *ESAIM: Control, Optim. Calc. of Var.*, vol. 26, no. 24, pp. 1–16, 2020.

[95] K. Y. Pettersen, "Lyapunov sufficient conditions for uniform semiglobal exponential stability," *Automatica*, vol. 78, pp. 97–102, 2017.

[96] W. Rudin, *Principles of Mathematical Analysis*. McGraw-Hill, 1976.

[97] G. Chowdhary, T. Yucelen, M. Mühlegg, and E. N. Johnson, "Concurrent learning adaptive control of linear systems with exponentially convergent bounds," *Int. J. Adapt. Control Signal Process.*, vol. 27, no. 4, pp. 280–301, 2013.

[98] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence using finite excitation," *Int J Adapt Control Signal Process*, vol. 33, no. 12, pp. 1775–1787, Dec. 2019.

[99] H. Deng, M. Krstic, and R. J. Williams, "Stabilization of stochastic nonlinear systems driven by noise of unknown covariance," *IEEE Trans. Autom. Control*, vol. 46, no. 8, pp. 1237–1253, 2001.

[100] S. Bhasin, R. Kamalapurkar, M. Johnson, K. G. Vamvoudakis, F. L. Lewis, and W. E. Dixon, "A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems," *Automatica*, vol. 49, no. 1, pp. 89–92, Jan. 2013.

[101] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," *Automatica*, vol. 64, pp. 94–104, 2016.

[102] M. Greene, "Nonsmooth data-based reinforcement learning for online approximate optimal control," Ph.D. dissertation, University of Florida, 2022.

BIOGRAPHICAL SKETCH

Omkar Sudhir Patil was born in 1996, and he has since pursued a career in engineering. He completed his Bachelor of Technology (B.Tech.) degree in production and industrial engineering from the esteemed Indian Institute of Technology (IIT) Delhi in 2018, where he was honored with the BOSS award for his outstanding bachelor's thesis project. Following his graduation, Omkar worked as a Project Associate at the Control & Automation Lab of IIT Delhi on a sponsored project entitled "Aerial Manipulation - Modeling, Planning and Control". In August 2019, he joined the Nonlinear Controls and Robotics (NCR) Laboratory at the University of Florida under the guidance of Dr. Warren Dixon to pursue his doctoral studies. Omkar received his Master of Science (M.S.) degree in mechanical engineering in August 2022 and Ph.D. in mechanical engineering in May 2023 from the University of Florida. His research focuses on the development and application of innovative Lyapunov-based nonlinear, robust, and adaptive control techniques.