LYAPUNOV-BASED PHYSICS-INFORMED LEARNING FOR CONTROL

By

REBECCA G. HART

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

2026

To Fred and Jill Hart

ACKNOWLEDGMENTS

This work was made possible by a host of mentors, colleagues, friends, and family who shaped my development and ultimately helped me become the person I am today. This dissertation would not have been possible without the guidance of my advisor, Dr. Warren Dixon. I am deeply grateful for his mentorship, patience, and support throughout this journey. I also thank my committee members, Dr. Yu Wang, Dr. Alina Zare, and Dr. Joel Harley for their constructive feedback and for challenging me to consider the broader implications of these findings. Within the NCR Lab, I am grateful to my lab mates for their support and the memories we shared. I would like to specifically thank Omkar Patil for his role as an invaluable mentor and his willingness to share his technical expertise. Finally, I want to thank my family, especially my parents and grandparents, for instilling in me the persistence required to complete this degree and for their unwavering support. To Cassie, thank you for your constant friendship and for providing the necessary balance to my life throughout this process. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-2236414. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

a.a.t	Almost All Time
a.e	Almost Everywhere
CL	Concurrent Learning
CLOE	Continuous Learning from Offline Experience
PD	Positive Definite
DeLaN	Deep Lagrangian Neural Network
DNN	Deep Neural Network
DOF	Degree of Freedom
FE	Finite Excitation
HNN	Hamiltonian Neural Network
Lb	Lyapunov-based
LIP	Linear-In-Parameter
LNN	Lagrangian Neural Network
LRE	Linear Regression Equation
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MBRL	Model-Based Reinforcement Learning
ML	Machine Learning
MPC	Model Predictive Control
MSE	Mean Squared Error
NIP	Nonlinear-In-Parameter
NN	Neural Network
NRE	Nonlinear Regression Equation
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PE	Persistence of Excitation

PI	Physics-Informed
PIML	Physics-informed Machine Learning
PINN	Physics-informed Neural Network
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RMS	Root Mean Square
SS	Skew Symmetric
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

LYAPUNOV-BASED PHYSICS-INFORMED LEARNING FOR CONTROL

By

Rebecca G. Hart

May 2026

Chair: Warren E. Dixon
Major: Mechanical Engineering

Accurate modeling of complex physical systems has garnered significant interest alongside the expansion of computational power. In recent years, Lyapunov-based neural networks (NNs) have emerged as powerful tools for real-time function approximation in closed-loop control. However, while traditional real-time adaptation of NNs compensates for unknown dynamics to achieve tracking, it often fails to identify the underlying system physics.

To address this, this dissertation shifts toward physics-informed learning. While incorporating physical constraints has shown empirical success in offline settings, developing real-time, physics-informed adaptation laws is an open research direction. Furthermore, providing formal guarantees on function approximation is difficult due to the nonlinear-in-parameter (NIP) structure of these learning algorithms, which complicates parameter convergence and identifiability analysis.

Chapter 1 provides a literature survey of physics-informed learning techniques and motivations, introduces the research section of the dissertation, and provides an outline of the remaining chapters. Chapter 2 develops two approaches for a physics-informed control framework using the Euler-Lagrange equation and designs real-time adaptation laws which incorporate the physics of the system. Chapter 3 establishes the identifiability conditions required to guarantee function approximation performance, bridging the gap between architectural identifiability and the excitation conditions

common in adaptive control. Chapter 4 develops a combined approach for identification and control leveraging historical information in the adaptation law. Chapter 5 concludes the dissertation by highlighting the contributions of each chapter. Additionally, future extensions based on the results in this dissertation are proposed.

CHAPTER 1
LITERATURE REVIEW AND DISSERTATION OUTLINE

To provide context for the contributions of this dissertation, this chapter surveys the existing literature across the three interconnected pillars of machine learning (ML), physics-informed learning, and adaptive control. The literature review begins with an overview of ML applications in control and establishes the motivation for physics-informed machine learning (PIML) architectures. This is followed by an overview of current PIML techniques, noting that while these methods offer a means to embed physical principles, they are predominantly relegated to offline applications which is unsuitable for many control applications. To bridge this gap, the chapter concludes with an exploration of adaptive control and real-time learning, focusing on Lyapunov-based techniques that provide formal stability guarantees and rigorous methods for system identification. Ultimately, this dissertation demonstrates how PIML methods can be used in Lyapunov-based control frameworks to enhance performance and system identification. After establishing the conditions necessary to guarantee approximation accuracy, subsequent chapters detail the development of a unified control and identification objective.

1.1 Machine Learning Background

The use of ML in engineering disciplines has shifted the way in which complex systems are modeled and controlled. Traditional engineering methods rely on white-box, mathematical, representations of systems derived from first principles. While these mathematical models offer high interpretability and guaranteed adherence to physical laws, they struggle to capture nuances of nonlinear or poorly understood systems without significant engineering effort. In contrast, ML methods have gained widespread popularity for their ability to generate predictions directly from raw data. By operating as black-box models, methods such as Neural Networks (NNs) can map inputs to outputs without requiring an explicit mathematical description of the systems internal physics.

This capability is particularly advantageous for complex systems where obtaining high-fidelity analytical models is computationally prohibitive or theoretically intractable.

1.1.1 Machine Learning in Control

The applications of learning methods, such as NNs, within control systems has evolved from static function approximation to dynamic, real-time, adaptation. Early works [3–11] established NN-based control, proving that NNs could approximate system dynamics when an analytical model was unavailable. Initial methods focused on offline optimization, where NN weights were determined by minimizing a loss function over a fixed dataset [5–8]. While useful in regression problems, the applicability of offline methods to control problems is limited due to their inability to handle out-of-distribution data and the requirement that the training set is sufficiently rich over the operating region. To address concept drift and improve generalization capabilities, the ML community has proposed various online methods and domain adaptation techniques [12–14]. However, these methods generally lack the formal guarantees required for safety-critical hardware. To solve this, adaptive NN-based control was developed which allows the weights in a NN to update in real-time based on a Lyapunov-based stability analysis allowing for guaranteed performance in unseen environments [9, 15–22]. Notably, the early approach in [9] applies to shallow NNs characterized by an input layer, a single hidden layer, and an output layer.

Deep Neural Networks (DNNs), in comparison to shallow NNs, are exponentially more efficient in terms of the number of neurons required to achieve similar function approximation performance [23]. Motivated by these performance improvements, subsequent research in [18–22, 24] leverages DNNs in an adaptive control framework.

1.1.2 Motivation for Physics-Informed Learning

Despite their efficiency, ML models, including NNs, remain inherently data-driven and are susceptible to producing outputs which are not interpretable, and in some cases, violate fundamental physical laws. The violation of physical principles and desire

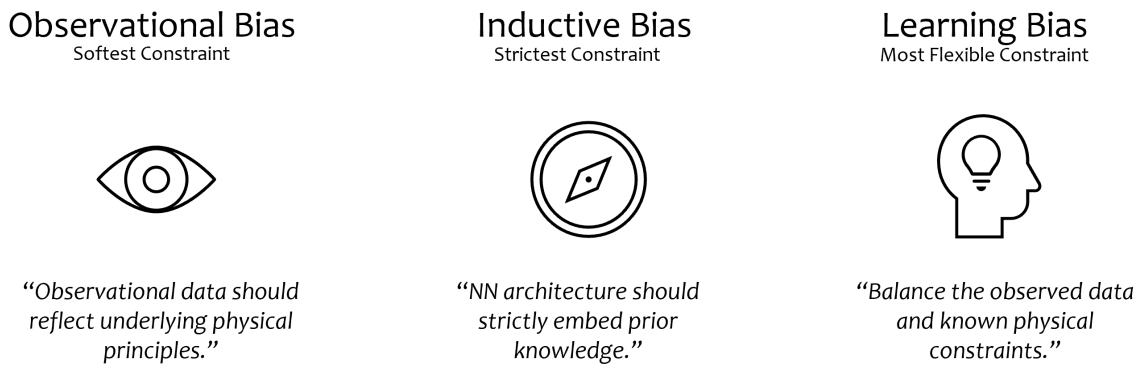

Figure 1-1. Different types of biases that can be used to incorporate physical principles into learning algorithms. Original figure adapted from [1]

to leverage known information has led to the development of PIML. PIML embeds properties of the system into the ML architecture or training regime with the goal of ensuring the ML-based solution is consistent with physical principles.

1.2 Physics-Informed Learning Background

Physics-informed learning leverages the function approximation capabilities of ML with established physics-based principles in a "gray-box" strategy [1, 25–27]. It has been shown that using prior knowledge or constraints in ML methods yields improved accuracy particularly for exploratory or generalization tasks where black-box methods often struggle [1]. Similarly, empirical results indicate that gray-box, physics-informed methods, lead to faster learning while requiring significantly less data and simpler models compared to black-box methods [25]. These gray-box approaches have been described by various terms in the literature including 'physics-informed', 'physics-based', 'physics-guided', and 'informed deep learning' [28].

1.2.1 How Physical Knowledge is incorporated

The incorporation of physical knowledge in PIML models can generally be viewed as enforcing biases in the form of an observational bias, inductive bias, or learning bias [1] (Figure 1-1). Observational bias assumes that physical principles are inherently captured within gathered data, yet this reliance demands vast datasets and carries

the risk of overfitting in complex systems. Inductive bias offers a stricter alternative by enforcing hard mathematical constraints which ensure physical laws are satisfied, though the mathematical complexity of encoding these constraints often limits their application to specific, well-defined laws. In contrast, learning bias provides a flexible middle ground by encouraging predictions to approximately respect physical principles through a soft constraint thus balancing traditional ML learning techniques with physical plausibility. Although the selection of biases is often task-dependent, the enforcement either hard or soft constraints has been shown to reduce the error of the variables that appear in the constraints [29]. These theoretical biases are realized through feature engineering, architecture design, or regularization strategies [28], with most frameworks adopting a combined approach.

Feature engineering leverages domain knowledge to transform input data and improve model performance [30]. In physics-informed learning, feature engineering can involve basing the transformations on known physical parameters such as invariance properties [31], Lie point symmetries [32], or scaling methods [33]. These methods can be supplemented by data augmentation methods [34], where high-fidelity simulations or digital twins are used to enhance the training dataset when data is scarce [35]. Within this context, physics-informed surrogate models have been shown to decrease computation time [36].

By embedding physical knowledge into the architectural design, the learning method is transformed from an observation-based regression to a structured series of operations where physical-consistency is enforced through hard constraints [31, 37–42]. This can be achieved by using known linear relationships [37], activation or basis function selection [31, 38, 42], or specific design of layers based on physical properties [39–42].

Physics-informed regularization incorporates physical laws as soft constraints within the loss function used during training and improves generalization in data-scarce

environments [25]. By penalizing violations of governing principles, this approach constrains the space of admissible solutions [25] and has been applied across various architectures including feedforward [25], graph [43], long-short term memory [44], and recurrent [40] neural networks. The general structure of the combined physics-informed loss follows

$$\mathcal{L} = \gamma_{\text{data}} \sum_{i}^{n} l_{i,\text{data}} + \gamma_{\text{physics}} \sum_{j}^{m} l_{j,\text{physics}}, \tag{1–1}$$

where $l_{i,\text{data}}$ represents n data-driven objective terms and $l_{j,\text{physics}}$ represents m physics-based regularization terms. The objectives are typically scaled by factors γ_{data} and γ_{physics} to balance the contributions of the data-driven and physics-based components in the overall loss function of the model. The terms $l_{i,\text{data}}$ and $l_{j,\text{physics}}$ could represent a variety of objective formulations such as mean squared error, mean absolute error, or cross entropy loss functions. The specific formulation of the physics loss can enforce a variety of requirements such as mass conservation [45], expected behavior at the boundary of the systems [46, 47], and the difference between the solution to the governing equation and the model [48–50].

1.2.2 Types of Physical Knowledge

Implementation strategies provide the framework for using physical data, but the specific attributes selected are determined by the available domain knowledge. These attributes generally appear in the form of governing equations, symmetry constraints, and conservation laws. By matching the learning bias to enforce the available domain knowledge, the model is conditioned to adhere to the known properties of the system and lead to improved accuracy [1].

Using NNs to solve ordinary differential equations (ODEs) and partial differential equations (PDEs) has been a topic of long-standing interest [51–54]. Within physics-informed methods, utilizing the governing equation has been widely used to guide the learning process. Originating from the PDE domain, researchers have employed initial and boundary constraints to refine the developed model [46, 53, 55–58]. These

implementations incorporate a variety of conditions such as periodic [57], Neumann and Dirichlet boundary conditions [56, 58], as well as specific initial state information [46, 53]. While many physics-informed methods originated from the fluid dynamics community in search of solving PDEs, their underlying principles are inherently applicable to a broad spectrum of dynamical systems. For instance, governing equations can be used as a regularization to learn system dynamics [48] or to approximate non-conservative forces [59] in the Euler-Lagrange equation. Furthermore, the governing equation can be embedded explicitly in the loss function [46] or used to provide a simulated output for feature engineering methods [50].

Beyond using the governing equations, symmetry constraints can significantly reduce the search space of learning algorithms [60]. While convolutional neural networks are the classic example of architectures designed to respect symmetry groups [61], their application to physical phenomena is relatively new [42, 62]. Symmetry constraints are often complemented by structural or temporal symmetries. For example, specialized hub layers have been used to enforce even/odd symmetry in noisy environments [39] while Lie point symmetries were used in [32]. Furthermore, for reversible classical systems, temporal consistency can be enforced via time-reversal symmetry ($t \rightarrow -t$), using the discrepancy between forward and backward predictions as a regularization metric [63].

Conservation laws offer a framework for grounding ML in physical properties. Beyond fluid dynamics applications where mass conservation can regularize flow field reconstruction [45], a significant body of research has focused on system energy conservation [26, 27, 49, 64–69]. Lagrangian NN (LNN) architectures like Deep Lagrangian Neural Networks (DeLaN) parameterize kinetic and potential energy by minimizing the squared residual of the Euler-Lagrange Equation [26]. While DeLaN typically operates using position and velocity, Hamiltonian Neural Networks (HNNs) utilize position and momentum. The choice between LNN and HNN models is dictated by the availability of coordinate data [64]. Recent refinements have extended these frameworks to handle

practical implementation challenges, such as utilizing differentiable ODE solvers to bypass the need for second order derivatives [49, 67], or providing a combined HNN and LNN approach to learn contact properties [68].

1.3 Physics Informed Learning in Control

There is strong motivation to use PIML in control due to its ability to ensure physical plausibility of the prediction and providing greater confidence in ML generated solutions to control engineers. In its simplest form, a physics informed model can replace the unknown dynamics in a controller. For instance, in [49] and [26] the physics-informed architecture outputs were incorporated in a PD controller. In [69], physics-informed architectures were applied to inverse dynamic control and energy control. Beyond its use as a feedforward component, PIML is also widely used in model-based methods, such as model predictive control (MPC) or model-based reinforcement learning (MBRL). This is typically accomplished by combining the physics-based priors with various regression methods to improve control performance.

1.3.1 Physics-informed Model Predictive Control

ML methods have been widely used in MPC control (for a full review see [70]). Building on recent advances in physics-informed learning, results such as [46, 47, 71, 72] incorporate physics-informed learning methods into the control architecture using MPC techniques. Early approaches utilized physics-based linear models to ensure safety of the system while a ML model was used to maximize control performance [73, 74]. Modern iterations, however, embed physical laws directly into the predictive element. This is achieved by using the solutions to governing ODEs as a regularization method [46] or by augmenting the physics-informed neural networks (PINNs) to accept initial states and control inputs as parameters [47]. To address temporal accuracy degradation common in PINNs, the ODE solution has been partitioned into valid sub-intervals, and applied to serial robot manipulators and quadrotor control [71, 72].

1.3.2 Physics-informed Reinforcement Learning and Control

MBRL has been developed to improve sample efficiency of reinforcement learning (RL) methods by using a learned model to make predictions about future behavior. By employing a model, MBRL avoids the high costs of trial-and-error learning which can be particularly detrimental when applied to mechanical systems [75]. However, MBRL can suffer from inaccurate modeling (i.e., model bias) which can lead to a lower success rate compared to model-free RL approaches [76, 77]. Incorporating physical properties into the RL architecture allows for regularization of the hypothesis space towards physically consistent solutions. Modern frameworks achieve this through varied strategies including directly encoding boundary conditions and conservation laws within auto-encoding recurrent networks to ensure structural consistency [75] or explicitly coupling deep RL agents with physically consistant NNs [78, 79].

1.3.3 Connections Between Physics-Informed Learning and Adaptive Control

Adaptive control methods are naturally suited for physics-informed learning due to their interconnection between system dynamics, performance metrics, and adaption laws that link learning to the systems behavior. Traditionally, adaptive control employs linear-in-the-parameters (LIP) formulations, commonly expressed as $f = Y\theta$, where f is an unknown function describing the system dynamics, Y is a known regression matrix derived from the system's dynamics, and θ is a set of unknown parameters which are being estimated. The adaptation of the parameter estimates $\hat{\theta}$ typically follow the form $\dot{\hat{\theta}} = \Gamma Y^\top e$, where $\dot{\hat{\theta}}$ denotes the designed time derivative of $\hat{\theta}$, Γ is a user-selected gain matrix, and e is an error term (e.g., position error, prediction error, etc.) in the system. A key feature of LIP adaptive control is that the adaptation is driven by a regression matrix, Y, derived from the system's dynamics thus directly incorporating properties of the physical system into the learning process.

Due to their versatility, adaptive control methods have been extended to use ML techniques such as NNs and DNNs [9, 15, 16, 20–22, 80–83]. In these approaches,

instead of the regression updating based off of a regression matrix derived from the system's dynamics, the regression is based on properties of the architecture. This leads to a black-box formulation that discards known system information. While such algorithms preform well, PIML has significant advantages including reduced data requirements, improved generalization, and increased interpretability [84]. These benefits are particularly important in control applications where reducing computational load and allowing the users to interpret the model are critical. Therefore, there is strong motivation to combine PIML with adaptive control methods. This would allow for real-time adaptation to changing system parameters while leveraging known physical laws of the system and simultaneously utilizing ML techniques to improve performance.

While many DNN formulations in PIML literature rely on offline optimization, adaptive control methods facilitate real-time learning or adaptation of NN weights. This capability addresses the challenge of obsolescence of pre-trained weights, particularly when the training set is not comprehensive. Early work on shallow NNs, such as the development in [16], enabled the real-time adaptation of the NN weights. Motivated by the desire to leverage the improved function approximation capabilities of deep networks, [19] introduced an approach where the inner layer weights are iteratively adjusted through batch updates, while the outer layer weights adapt in real-time. A similar iterative approach is used in [47] to counteract the degradation of the PINN over time. Recently, [20] developed a method which enables real-time adaptation of all weights within a DNN. This full adaptation approach is particularly advantageous as it allows the DNN to fully exploit its function approximation capabilities, enhancing its ability to respond to changing system dynamics.

1.4 Real-Time System Identification

The connection between System Identification and PIML efforts is defined by the shared transition from unconstrained function approximation to structured parameter

recovery. Historically, system identification has focused on determining physical constants from system data where early work focused on inertial parameters in rigid-body loads [85]. System identification techniques are applicable to a broad range of tasks including fault detection and accurate state estimation during intermittent feedback loss [86]. While PIML methods have demonstrated that embedding physical structure into the learning process enhances generalization capabilities and robustness to noise [60, 87], a fundamental challenge remains in translating these offline discovery techniques into real-time control applications.

Offline PIML results have demonstrated that when a learned mathematical model accurately captures the underlying equations, it can generalize beyond the training domain [60]. Recently, data-driven discovery of unknown governing equations using symbolic regression differences have been investigated to determine the underlying dynamic equations [88, 89]. Within the PIML community, [87] demonstrated that incorporating physics-based constraints can result in parameter identification and leads to fewer violations of physical laws compared to black-box NNs. Additionally, in the presence of noisy data the incorporation of the data-driven methods with known model properties showed significant improvement in mean squared error (MSE) performance compared to traditional modeling approaches [87].

The primary difficulty with using high-fidelity ML for real-time system identification lies in the distinction between closed-loop stability and parameter convergence. While an ML-based controller can achieve asymptotic tracking, the identification problem may remain ill-posed if the system's regressor lacks sufficient rank or excitation. In such cases, the weights of the DNN are not guaranteed to be unique or physically consistent. Therefore, establishing the mathematical conditions for well posedness is essential go guaranteed that the ML architecture is capable of true system identification rather than black-box error compensation.

Traditional adaptive control methods have embraced composite [90] and concurrent learning (CL) methods [91] as a regularization technique that guarantees convergence to the actual parameter values. A key aspect of these developments is the use of a prediction error in the adaptation law. When incorporated into the update law for real-time learning, prediction errors in the adaptation law are comparable to PIML methods using the solution to the differential equation alongside the ML model's estimate of that solution as seen in [48, 87]. However, in real-time learning, the solution to the dynamic function cannot be calculated *a prioi*, as it is often done in offline training for PIML methods. As a result, the prediction errors seen in traditional adaptive control are often reconstructed versions of the unknown system dynamics and their estimates [22,90–94].

While classical adaptive control methods can guarantee parameter convergence for LIP uncertainties assuming various exploration/excitation conditions, few works consider nonlinear-in-parameter (NIP) uncertainties [22, 91, 95–98]. Existing solutions often rely on simplifying assumptions such as convex/concave nonlinear parameterizations as in [99] or strong monotonicity as in [100] and [101]. However, the NIP structure of DNNs is neither convex/concave nor strongly monotonic.

Recent efforts to achieve parameter convergence such as [22, 95, 102, 103], use dynamic state derivative estimators and least-squares update laws. Such methods also rely on a Persistence of Excitation (PE) condition which cannot be verified online. Additionally, such stability results rely on excitation assumptions without addressing the fundamental well-posedness of the parameter estimation problem. While parameter convergence is guaranteed contingent on these assumptions, the structural conditions required to ensure unique parameter determination in NIP architectures remain unestablished.

This lack of a uniqueness guarantee underscores a critical gap in the literature regarding identifiability conditions for NIP systems and their relationship to the classical excitation definitions used in adaptive control. A key aspect of identifiability is whether

an architecture is sufficiently “rich” to capture necessary behaviors. For example, an architecture lacking a bias term would struggle with learning offsets. Deep learning architectures such as DNNs exhibit structural richness in the form of symmetries, where multiple weight configurations yield the same input-output map. While these symmetries may yield acceptable function approximation, they complicate the unique determination of parameters. Therefore, the pressing challenge is establishing properties that determine whether finite recorded data can uniquely identify the underlying system. Without these properties, the learned parameters may overfit the instantaneous data rather than capturing the local system dynamics.

1.5 Dissertation Outline

Chapter 2 and the work in [2] and [24] develops the first physics-informed real-time learning architecture which leverages the known structure of the Euler-Lagrange Equation in the architecture design. Section 2.1 introduces a physics-informed Long Short-Term Memory (PI-LSTM) controller, which leverages the memory retention capabilities of LSTMs to achieve stronger observational bias. The architecture is designed using the structure of the Euler-Lagrange equation, enabling the model to learn unknown system components separately and leverage known properties. This development employs a Lyapunov-based analysis to derive real-time weight adaptation laws for the PI-LSTM. Due to the architecture design being based on the governing equation, the PI-LSTM’s output must be post-multiplied by dynamic terms such as position or velocity. This results in the adaptation law being further informed by the dynamics of the system compared to a traditional adaptation law. A Lyapunov-based stability analysis was performed to ensure asymptotic tracking error convergence and boundedness of the developed PI-LSTM controller. Simulations were performed on a two-link robot manipulator yielding a RMS tracking error of 0.0185 rad and achieved 33.76% improvement over a physics-informed architecture which does not leverage memory capabilities of a LSTM.

Section 2.2 leverages the development in Section 2.1 and incorporates a physics-based regularization term in the adaptation. This additional term is motivated by the desire to leverage known model properties to guide the architecture's adaptation. Specifically, the regularization is applied to the approximation of the inertia and Coriolis matrices which are known to be skew-symmetric. Using this property, a skew-symmetric prediction error is developed based on the output of the estimates and incorporated into the real-time adaptation law. This chapter addresses challenges with compounding effects of uncertainties stemming from function approximation error and higher-order terms. This is the first result which uses a combination of architecture design and regularization to enforce physics-constraints in a real-time context. A Lyapunov-based stability analysis results in asymptotic convergence of the tracking and skew-symmetric prediction error. Simulations were performed on a two-link robot manipulator yielding a a 19.87% improvement in overall function approximation compared to methods which do not consider the symmetry constraint.

Chapter 3 addresses the fundamental question of when function approximation performance can be guaranteed and under what conditions it can be verified. Specifically, this chapter establishes the requirements for parameter identification in NIP systems and bridges the gap between theoretical identifiability and the verifiable finite excitation (FE) condition. By defining the conditions under which deep learning architectures, such as DNNs, become unidentifiable, this chapter provides a framework for assessing network convergence. Numerical analysis of the rank condition for DNNs validates the feasibility of the FE condition and demonstrates how these identifiability requirements can serve as a diagnostic metric for informed architectural design.

Chapter 4 and the work in [103] applies the results in Chapter 3 to the simultaneous system identification and control problem. Section 4.1 presents the first result enabling a CL adaptation law for all layers of a DNN-based controller. The developed CL framework achieves convergence of the DNN parameter estimates to a neighborhood of the values

which minimize the difference between the estimated and actual function, provided the DNN Jacobian satisfies the verifiable FE condition discussed in Chapter 3. A Lyapunov-based stability analysis is conducted to ensure the simultaneous convergence of the tracking error, weight estimation errors, and observer errors to a neighborhood of the origin. Simulation results demonstrate approximately 40% improvement in function approximation performance and 65% improvement in off trajectory evaluations compared to a baseline DNN controller which does not provide parameter convergence guarantees. Section 4.2 presents the first result which leverages Continuous Learning from Offline Experience (CLOE) which uses known function values to improve system estimates. Compared to Section 4.1, the developed method does not require an observer to be used and thus allows for robust disturbance rejection techniques such as sliding mode control without corrupting estimate.

Chapter 5 concludes the dissertation and comments on future research directions which can build on this dissertation.

1.6 Notation and Preliminaries

1.6.1 Notation

The space of essentially bounded Lebesgue measurable functions is denoted by $\mathcal{L}_\infty$. Given $A \triangleq [a_{j,i}] \in \mathbb{R}^{n\times m}$, $\text{vec}(A) \triangleq [a_{1,1}, \ldots, a_{n,1}, \ldots, a_{1,m}, \ldots, a_{n,m}]^\top$. The Kronecker product is denoted by $\otimes$. Given any $A \in \mathbb{R}^{n\times m}$, $B \in \mathbb{R}^{m\times p}$, and $C \in \mathbb{R}^{p\times r}$, $\text{vec}\,(ABC) = \left(C^\top \otimes A\right) \text{vec}\,(B)$. The identity matrix of size $n \times n$ is denoted by I_n. The notation $\lambda_{\text{min}}(A)$ and $\lambda_{\text{max}}(A)$ denote the minimum and maximum eigenvalues of a matrix, respectively. The notation $A \succ 0$ denotes that the matrix A is positive-definite (PD), such that all eigenvalues of A are strictly positive. The zero vector of size $q \times 1$ is given by 0_q. The signum function, denoted by $\text{sgn}(D)$ where $D \in \mathbb{R}^1$, is defined as $\text{sgn}(D) \triangleq 1$ if $D > 0$, $\text{sgn}(D) \triangleq 0$ if $D = 0$, and $\text{sgn}(D) \triangleq -1$ if $D < 0$. For a vector $\boldsymbol{D} \in \mathbb{R}^n$ defined as $\boldsymbol{D} \triangleq [D_1, D_2, \ldots, D_n]^\top$the sign function is applied element-wise such that $\text{sgn}(\boldsymbol{D}) \triangleq [\text{sgn}(D_1),\ \text{sgn}(D_2), \ldots,\ \text{sgn}(D_n)]^\top$. The notation $\overset{\text{a.e.}}{(\cdot)}$ denotes the relation

$(\cdot)$ holds almost everywhere (a.e.). The notation $\overset{\text{a.a.t}}{(\cdot)}$ denotes the relation $(\cdot)$ holds for almost all time (a.a.t). The notation K$[\cdot]$ denotes the Filippov's regularization $[\cdot]$ [104]. The right-to-left matrix product operator is represented by $\overset{\curvearrowleft}{\prod}$, i.e., $\overset{\curvearrowleft}{\prod_{p=1}^{m}} A_p = A_m \dots A_2 A_1$ and $\overset{\curvearrowleft}{\prod_{p=a}^{m}} A_p = I_n$, if $a > m$. Given some functions f and g function composition is denoted by $\circ$, where $(f \circ g)(x) \triangleq f(g(x))$.

1.6.2 Deep Neural Network (DNN) Model

Consider a collection of feedforward DNNs indexed by $i \in \mathcal{I}$ where $\mathcal{I}$ denotes the index set. In the case where only a single DNN is considered, the index i is omitted for brevity. For each $i \in \mathcal{I}$, a feedforward DNN $\Phi_i(X_i, \theta_i) \in \mathbb{R}^{n_i}$ with output size $n_i \in \mathbb{N}_{>0}$, can be modeled as [20]

$$\Phi_i(X_i, \theta_i) \triangleq (v_{k_i}^\top \phi_{k,i} \circ \dots \circ v_{1_i}^\top \phi_{1,i})(v_{0_i}^\top X_{a,i}), \tag{1–2}$$

where $\theta_i \triangleq \left[\text{vec}(v_{0_i})^\top, \dots, \text{vec}(v_{k_i})^\top\right]^\top \in \mathbb{R}^{\varkappa_i}$, where $\varkappa_i \triangleq \sum_{j=0}^{k_i} L_j L_{j+1}$, $j \in \{0, \dots, k_i\}$ and $k_i \in \mathbb{N}$ denotes the number of hidden layers within θ_i, $v_{j,i} \in \mathbb{R}^{L_{j,i} \times L_{j+1,i}}$ denotes the matrix of weights and biases in the j^{th} hidden layer, $L_{j,i} \in \mathbb{N}$ denotes the number of nodes within the j^{th} hidden layer for all $j \in \{0, \dots, k_i\}$, and $L_{0,i} \triangleq m_i + 1$, where m_i is the dimension of the $\mathbb{R}^{m_i}$ input vector to the DNN. Inputs to the DNN are denoted by $X_i \in \Omega_i$ and $\Omega_i \subset \mathbb{R}^m$ denotes a compact set, and the augmented input $X_{a,i} \in \mathbb{R}^{m+1}$ is defined as $X_{a,i} \triangleq \begin{bmatrix} X_i^\top & 1 \end{bmatrix}^\top$ for all $i \in \mathcal{I}$. The vector of smooth activation functions at the j^{th} layer is a function denoted by $\phi_{j,i}(\cdot) : \mathbb{R}^{L_{j,i}} \to \mathbb{R}^{L_{j,i}}$. It is defined for an input vector $\mathsf{z}_{j,i} \in \mathbb{R}^{L_{j,i}}$ as $\phi_{j,i}(\mathsf{z}_{j,i}) \triangleq \begin{bmatrix} \varsigma_{j,i,1}(\mathsf{z}_1) & \dots & \varsigma_{j,i,L_j-1}(\mathsf{z}_{L_{j-1,i}}) & 1 \end{bmatrix}^\top$, where $\varsigma_{j,i,y} : \mathbb{R} \to \mathbb{R}$ denotes the activation function at the y^{th} node of the j^{th} layer for all $j \in \{1, \dots, k_i\}$ and $i \in \mathcal{I}$. To incorporate a bias term into the DNN model in (1–2), the input X_i and the activation functions $\phi_{j,i}$ are augmented with a 1 for all $j \in \{1, \dots, k_i\}$ and $i \in \mathcal{I}$.

To facilitate the development of the online weight adaptation laws, the DNN model in (1–2) can also be represented recursively using the shorthand notation $\Phi_{j,i}$ for all

$j \in \{0, \ldots, k_i\}$ as [20]

$$\Phi_{j,i} \triangleq \begin{cases} v_{j,i}^\top \phi_{j,i}(\Phi_{j-1,i}), & j \in \{1, ..., k_{,i}\}, \\ v_{0,i}^\top x_{a,i} & j = 0, \end{cases} \tag{1–3}$$

where $\Phi_i(X_i, \theta_i) = \Phi_{k,i}$.

The Jacobian of the feedforward DNN, denoted $\Phi'(X_i, \theta_i)$, is represented here in a structured recursive form to support the subsequent stability proofs, this matrix is equivalent to the gradients calculated via standard backpropagation where the computational complexity of the DNN forward and backward pass is $\mathcal{O}(\varkappa)$. The analytical form of the Jacobian can be represented as $\Phi'(X_i, \theta_i) \triangleq \left[\Phi'_0(X_i, \theta_i), \ldots, \Phi'_j(X_i, \theta_i)\right] \in \mathbb{R}^{n_i \times \varkappa_i}$, where $\Phi'_j \triangleq \frac{\partial \Phi(X_i, \theta_i)}{\mathsf{vec}(v_{j,i})} \in \mathbb{R}^{n_i \times L_{j,i} L_{j+1,i}}$, for all $j \in \{0, \ldots, k_i\}$ and $i \in \mathcal{I}$. Using (1–3), the chain rule, and properties of the vectorization operator, the terms Φ'_0 and Φ'_j can be expressed as [20]

$$\begin{aligned} \Phi'_0(X_i, \theta_i) &\triangleq \left(\prod_{l=1}^{\overset{\curvearrowleft}{k_i}} v_{l,i}^\top \phi'_{l,i} \right) \left(I_{L_{1,i}} \otimes X_{a,i}^\top \right), \\ \Phi'_{j,i}(X_i, \theta_i) &\triangleq \left(\prod_{l=j+1,i}^{\overset{\curvearrowleft}{k_i}} v_{l,i}^\top \phi'_{l,i} \right) \left(I_{L_{j+1,i}} \otimes \phi_j^\top \right), \end{aligned} \tag{1–4}$$

for all $j \in \{1, \ldots, k_i\}$, where the activation function at the j^{th} layer and its Jacobian are expressed using the shorthand notations $\phi_{j,i} \triangleq \phi_{j,i}(\Phi_{j-1,i}(X_i, \theta_i))$ and $\phi'_{j,i} \triangleq \phi'_{j,i}(\Phi_{j-1,i}(X_i, \theta_i))$, respectively, and $\phi'_{j,i} : \mathbb{R}^{L_{j,i}} \to \mathbb{R}^{L_{j,i} \times L_{j,i}}$ is defined as $\phi'_{j,i}(\mathsf{z}_{j-1,i}) \triangleq \frac{\partial}{\partial \varrho_i} \phi_j(\varrho_i)|_{\varrho_i = \mathsf{z}_{j-1,i}}$, for all $y \in \mathbb{R}^{L_{j,i}}$ and $i \in \mathcal{I}$. To facilitate the subsequent development and analysis, the following assumption is made on the activation functions.

Assumption 1.1. For each $j \in \{0, \ldots, k_i\}$, the activation function $\phi_{j,i}$, its Jacobian $\phi'_{j,i}$, and Hessian $\phi''_{j,i}(y) \triangleq \frac{\partial^2}{\partial y^2} \phi_{j,i}(y)$ are bounded as

$$\begin{aligned} \|\phi_{j,i}(y)\| &\leq \mathfrak{a}_{1,i} \|y\| + \mathfrak{a}_{0,i}, \\ \left\|\phi'_{j,i}(y)\right\| &\leq \mathfrak{b}_{0,i}, \\ \left\|\phi''_{j,i}(y)\right\| &\leq \mathfrak{c}_{0,i}, \end{aligned} \tag{1–5}$$

where $\mathfrak{a}_{0,i}, \mathfrak{a}_{1,i}, \mathfrak{b}_{0,i}, \mathfrak{c}_{0,i} \in \mathbb{R}_{\geq 0}$ are known constants for $i \in \mathcal{I}$.

Remark 1.1. Most activation functions used in practice satisfy Assumption 1.1. Specifically, sigmoidal activation functions (e.g., logistic function, hyperbolic tangent etc.) have $\|\phi_{j,i}(y)\|$, $\left\|\phi'_{j,i}(y)\right\|$, and $\left\|\phi''_{j,i}(y)\right\|$ bounded uniformly by constants. Smooth approximations of rectified linear unit (ReLUs) such as Swish grow linearly, and hence satisfy the bound $\|\phi_{j,i}(y)\| \leq \mathfrak{a}_{1,i}\|y\| + \mathfrak{a}_{0,i}$ of Assumption 1.1.

CHAPTER 2
EULER-LAGRANGE PHYSICS-INFORMED NEURAL NETWORKS FOR ADAPTIVE CONTROL

Motivated by the desire to leverage *a priori* knowledge of the system dynamics, the results presented in this chapter, and published in in [2] and [24], utilize the known structure of the Euler-Lagrange equation within a Lyapunov-based adaptive control framework. The key contribution of this chapter is the transition from data-driven estimation to a physics-informed architecture which estimates only the unknown matrices of the Euler-Lagrange equation. The chapter is organized into two primary investigations, the first section introduces a multi-model learning architecture, and demonstrates how various frameworks can be combined to generate an overall estimate. This modular approach can be leveraged for the subsequent developments, though they are presented with a single-algorithm implementation for clarity. The second section focuses on the enforcement of physical consistency through regularization, specifically targeting the skew-symmetric relationship between the inertia and Coriolis matrices.

2.1 Lyapunov-Based Physics-Informed Long Short-Term Memory (LSTM) Neural Network-Based Adaptive Control

Motivated by the desire to capture time-dependent relationships, LSTMs have gained popularity due to their ability to retain relevant information across multiple time steps [105]. Specifically, the LSTM cell includes an explicit memory which can be used to model internal dynamics and complex temporal behaviors. This chapter provides the first result on an adaptive physics-informed LSTM (PI-LSTM) control architecture, merging the function approximation capabilities of DNNs with LSTM's memory to model and compensate for unknown system dynamics using physical insights. Unlike the black-box DNN and LSTM architectures in previous results [17, 20], the developed PI-LSTM controller incorporates the known dynamic structure into the architecture, with real-time weight updates governed by Lyapunov-based adaptation laws. The Lyapunov-based PI-LSTM controller demonstrates a significant advancement in physics-informed

learning for control such as with real-time adaptability and physical consistancy. A Lyapunov-based stability analysis ensures asymptotic tracking error convergence and boundedness of the developed PI-LSTM controller. Simulations were performed on a two-link robot manipulator and yielded an root mean square (RMS) tracking error of 0.0185 rad and achieved a 33.76% improvement over the baseline method presented in [2]. The results of this section have been published in [24].

2.1.1 Problem Formulation

Model Dynamics and Control Objective

Consider an uncertain Euler-Lagrange system modeled as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d(t) = \tau(t), \tag{2–1}$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ denote the generalized position, velocity, and acceleration, respectively. The generalized inertia matrix, generalized centripetal-Coriolis effects, generalized potential forces, generalized dissipation effects, the time-varying disturbances, and the control input are denoted by $M \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times n}$, $G \in \mathbb{R}^n$, $F \in \mathbb{R}^n$, $\tau_d \in \mathbb{R}^n$, and $\tau \in \mathbb{R}^n$, respectively. The system disturbances are assumed to be bounded as $\|\tau_d(t)\| \leq \overline{d}$, where $\overline{d} \in \mathbb{R}_{>0}$ denotes a known constant. The system in (2–1) satisfies following properties described in [106, Sec. 2.3].

Property 1. The inertia matrix $M(q)$, satisfies $m_1\|\zeta\|^2 \leq \zeta^\top M(q)\zeta \leq m_2\|\zeta\|^2$ for all $\zeta, q \in \mathbb{R}^n$, where $m_1, m_2 \in \mathbb{R}_{>0}$ denote known constants.

Property 2. The time-derivative of the inertia matrix and centripetal-Coriolis matrix satisfy the skew-symmetry relation, $\zeta^\top(\dot{M}(q) - 2C(q,\dot{q}))\zeta = 0$, for all $q, \dot{q}, \zeta \in \mathbb{R}^n$.

The control objective is to design a PI-LSTM controller to asymptotically track a user-defined, time-varying desired trajectory, $q_d \in \mathbb{R}^n$, which is designed to be

sufficiently smooth such that $q_d(t), \dot{q}_d(t), \ddot{q}_d(t) \in \mathcal{Q},$ for all $t \in \mathbb{R}_{\geq 0}$, where $\mathcal{Q} \subseteq \mathbb{R}^n$ denotes a known compact set. To quantify the objective, the tracking error $e \in \mathbb{R}^n$ and auxiliary tracking error $r \in \mathbb{R}^n$ are defined as

$$e \triangleq q - q_d, \qquad r \triangleq \dot{e} + \alpha e, \tag{2–2}$$

respectively, where $\alpha \in \mathbb{R}_{>0}$ denotes a user-selected constant control gain. Quantitatively, the objective is to ensure $\|e(t)\| \to 0$ and $\|r(t)\| \to 0$ as $t \to \infty$. Using (2–1)-(2–2), the open-loop dynamics for r can be determined as

$$M(q)\dot{r} = \tau - M(q)\left(\ddot{q}_d - \alpha\dot{e}\right) - C(q,\dot{q})\left(r + \dot{q}_d - \alpha e\right) - G(q) - F\left(\dot{q}\right) - \tau_d. \tag{2–3}$$

2.1.2 Control Law Development

2.1.2.1 Adaptive PI-LSTM Architecture

Recent trends show an increased implementation of PINN architectures in control design, because they use known physical properties of systems to enhance the model accuracy and generalization [26]. Leveraging the memory capabilities and improved transient performance of LSTMs, the developed method combines LSTM cells which capture dynamic time dependencies with the function approximation power of feedforward DNN terms. Motivated by the known structure of Euler-Lagrange dynamics, the developed PI-LSTM structure is developed to individually approximate the contribution of the unknown terms $M(q)$, $C(q,\dot{q})$, $G(q)$, and $F(\dot{q})$ based on the model structure (e.g., functional dependencies, vectors that multiply by each uncertainty) thereby integrating the physics into the architecture. This differs from black box approaches that approximate the entirety of the dynamics.

2.1.2.2 Long Short-Term Memory Network (LSTM) Model

An LSTM cell can be modeled in continuous time as [17]

$$f\left(z_i, W_{f,i}\right) = \sigma_g \circ W_{f,i}^{\top} z_i, \quad p\left(z_i, W_{p,i}\right) = \sigma_g \circ W_{p,i}^{\top} z_i,$$

$$o\left(z_i, W_{o,i}\right) = \sigma_g \circ W_{o,i}^{\top} z_i, \quad \tilde{c}\left(z_i, W_{c,i}\right) = \sigma_c \circ W_{c,i}^{\top} z_i,$$
$$\dot{c}_i = -b_{c,i} c_i + b_{c,i} \Psi_c\left(z_i, c_i, \vartheta_i\right),$$
$$\dot{h}_i = -b_{h,i} h_i + b_{h,i} \Psi_h\left(z_i, c_i, \vartheta_i\right), \tag{2–4}$$

where user-selected constants are given by $b_{c,i}, b_{h,i} \in \mathbb{R}_{>0}$, the cell and hidden states are denoted by $c_i \in \mathbb{R}^{l_{2,i}}$ and $h_i \in \mathbb{R}^{l_{2,i}}$ respectively where $c_i(0) = h_i(0) = 0$ for all $i \in \mathcal{I}$ where $\mathcal{I} \triangleq \{M, C, G, F\}$, and $l_{2,i} \in \mathbb{R}_{>0}$ denotes the number of neurons within the LSTM. The concatenated state vector $z_i \in \mathbb{R}^{l_{1,i}}$ is defined as $z_i \triangleq \left[X_i^{\top}, h_i^{\top}, 1\right]^{\top}$, where $l_{1,i} \triangleq m_i + l_{2,i} + 1$. The state z_i is augmented with 1 to incorporate a bias term. The forget gate, input gate, cell gate, and output gate are represented by $f\left(z_i, W_{f,i}\right) \in \mathbb{R}^{l_{2,i}}$, $p\left(z_i, W_{p,i}\right) \in \mathbb{R}^{l_{2,i}}$, $\tilde{c}\left(z_i, W_{c,i}\right) \in \mathbb{R}^{l_{2,i}}$, and $o\left(z_i, W_{o,i}\right) \in \mathbb{R}^{l_{2,i}}$, respectively, and the vector sigmoid and tanh activation functions are denoted by $\sigma_g \in \mathbb{R}^{l_{2,i}}$ and $\sigma_c \in \mathbb{R}^{l_{2,i}}$, respectively. The weight matrices are given by $W_{f,i}^{\top}, W_{c,i}^{\top}, W_{p,i}^{\top}, W_{o,i}^{\top} \in \mathbb{R}^{l_{2,i} \times l_{1,i}}$, and $W_{h,i}^{\top} \in \mathbb{R}^{l_{3,i} \times l_{2,i}}$, where the size of the output is defined as $l_{3,i}$, and the collection of adaptive weight estimates is given by $\vartheta_i \triangleq \Big[\text{vec}\left(W_{c,i}\right)^{\top}, \text{vec}\left(W_{p,i}\right)^{\top}, \text{vec}\left(W_{f,i}\right)^{\top}, \text{vec}\left(W_{o,i}\right)^{\top}, \text{vec}\left(W_{h,i}\right)^{\top}\Big]^{\top} \in \mathbb{R}^{4 l_{2,i} l_{1,i} + l_{2,i} l_{3,i}}$ for $i \in \mathcal{I}$. The functions $\Psi_{c,i}\left(z_i, c_i, \vartheta_i\right) \in \mathbb{R}^{l_{2,i}}$ and $\Psi_{h,i}\left(z_i, c_i, \vartheta_i\right) \in \mathbb{R}^{l_{2,i}}$ in the cell and hidden state dynamics are defined as

$$\Psi_c\left(z_i, c_i, \vartheta_i\right) \triangleq f\left(z_i, W_{f,i}\right) \odot c_i + p\left(z_i, W_{p,i}\right) \odot \tilde{c}\left(z_i, W_{c,i}\right),$$
$$\Psi_h\left(z_i, c_i, \vartheta_i\right) \triangleq o\left(z_i, W_{o,i}\right) \odot \left(\sigma_c \circ \Psi_c\left(z_i, c_i, \vartheta_i\right)\right),$$

respectively. To ensure the output of the LSTM is of the appropriate dimension, a fully-connected layer with weight matrix W_h is added to the LSTM cell. Thus, the output of the i^{th} LSTM $\Xi_i\left(z_i, c_i, \vartheta_i\right) \in \mathbb{R}^{l_{3,i}}$ can be modeled as

$$\Xi_i\left(z_i, c_i, \vartheta_i\right) = W_{h,i}^{\top} \Psi_h\left(z_i, c_i, \vartheta_i\right), \tag{2–5}$$

for $i \in \mathcal{I}$.

2.1.2.3 Adaptive PI-LSTM Control Strategy

Using the developed PI-LSTM architecture, an adaptive estimate of the dynamics is developed and implemented in the subsequently designed controller. The universal function approximation property states that the space of DNNs given by (1–3) is dense in $\mathcal{C}(\Omega_i)$, where $\mathcal{C}(\Omega_i)$ denotes a space of continuous functions over Ω_i [107, Thm 3.2]. Therefore, for any given $f_i \in \mathcal{C}(\Omega_i)$ and prescribed $\bar{\varepsilon}_i \in \mathbb{R}_{>0}$, there exist some $k_i, L_{j,i} \in \mathbb{N}$, and corresponding ideal weights and biases $\theta_{j,i} \in \mathbb{R}^{L_{j,i} \times L_{j+1,i}}, \forall j \in \{0, ..., k_i\}$, such that $\sup_{X_i \in \Omega_i} \|f_i(X_i) - \chi_i(X_i, \theta_i)\| \leq \bar{\varepsilon}_i$, for all $i \in \mathcal{I}$. Based on this property, the unknown terms $M(q),\ V(q, \dot{q}),\ G(q)$, and $F(\dot{q})$ can be modeled as

$$\text{vec}\left(M\left(q\right)\right) = \chi_M\left(X_M, \theta_M, c_M, h_M, \vartheta_M\right) + \varepsilon_M\left(X_M\right), \tag{2–6}$$

$$\text{vec}(C(q, \dot{q})) = \chi_C\left(X_C, \theta_C, c_C, h_C, \vartheta_C\right) + \varepsilon_C\left(X_C\right), \tag{2–7}$$

$$G(q) = \chi_G\left(X_G, \theta_G, c_G, h_G, \vartheta_G\right) + \varepsilon_G\left(X_G\right), \tag{2–8}$$

$$F(\dot{q}) = \chi_F\left(X_F, \theta_F, c_F, h_F, \vartheta_F\right) + \varepsilon_F\left(X_F\right), \tag{2–9}$$

where the function χ_i defined as $\chi_i(X_i, \theta_i, c_i, h_i, \vartheta_i) \triangleq \Phi_i\left(X_i, \theta_i\right) + \Xi_i\left(X_i, c_i, h_i, \vartheta_i\right)$ represents the combined DNN-LSTM estimate, where $\chi_i \in \mathbb{R}^{l_{3,i}}$ for $i \in \mathcal{I}$. The unknown function approximation errors are denoted as $\varepsilon_i \in \mathbb{R}^{l_{3,i}}$ for $i \in \mathcal{I}$. The output of the DNN and the LSTM are of the same dimension such that $l_{3,i} = L_{k+1,i}$. Therefore the dimension of χ_i can be equivalently expressed as $\chi_i \in \mathbb{R}^{L_{k+1,i}}$ for all $i \in \mathcal{I}$. The inputs for each DNN-LSTM are denoted as $X_M \triangleq q$, $X_C \triangleq \left[q^\top, \dot{q}^\top\right]^\top$, $X_G \triangleq q$, and $X_F \triangleq \dot{q}$. The following assumption facilitates the subsequent development.

Assumption 2.1. There exists known constants $\bar{\theta}_i \in \mathbb{R}_{>0}$, $\overline{\vartheta}_i \in \mathbb{R}_{>0}$ such that the unknown ideal weights can be bounded as $\|\theta_i\| \leq \bar{\theta}_i$, $\|\vartheta_i\| \leq \bar{\vartheta}_i$ for all $i \in \mathcal{I}$ [10, Assumption 1].

To ensure an appropriate output dimension of the PI-LSTM estimate, the vectorization operator is applied to $M(q)$ and $C(q, \dot{q})$. Using properties of the vectorization

operator and using (2–6)-(2–9) yields

$$M(q)\dot{r} = \tau - \tau_d - \left((\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\right)(\chi_M + \varepsilon_M(X_M)) - C(q,\dot{q})r \\ - \left((\dot{q}_d - \alpha e)^\top \otimes I_n\right)(\chi_C + \varepsilon_C(X_C)) - (\chi_G + \varepsilon_G(X_G)) - (\chi_F + \varepsilon_F(X_F)). \tag{2–10}$$

Based on the subsequent stability analysis, an adaptive PI-LSTM control input is designed as

$$\tau(t) = \left((\dot{q}_d - \alpha e)^\top \otimes I_n\right)\widehat{\chi}_C + \widehat{\chi}_G + \widehat{\chi}_F - k_1 r - e + \left((\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\right)\widehat{\chi}_M \\ - \text{sgn}(r)\left(k_2 + k_3\|(\dot{q}_d - \alpha e)^\top \otimes I_n\| + k_4\|(\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\|\right), \tag{2–11}$$

where $k_1, k_2, k_3, k_4 \in \mathbb{R}_{>0}$ are user-defined control gains, and $\widehat{\chi}_i \triangleq \chi_i\left(X_i, \hat{\theta}_i, \hat{c}_i, \hat{h}_i, \hat{\vartheta}_i\right) \in \mathbb{R}^{l_{3,i}}$ denotes the combined DNN and LSTM estimate for all $i \in \mathcal{I}$. The sgn function in (2–11) is motivated to obtain an asymptotic result given the function approximation error and added disturbance terms. Let the adaptive DNN and LSTM weight estimates be denoted as $\hat{\theta}_i \triangleq \left[\text{vec}(\hat{v}_{0,i})^\top, \ ..., \ \text{vec}(\hat{v}_{k,i})^\top\right]^\top \in \mathbb{R}^{\varkappa_i}$ and $\hat{\vartheta}_i \triangleq \left[\text{vec}\left(\widehat{W}_{c,i}\right)^\top, \text{vec}\left(\widehat{W}_{p,i}\right)^\top, \text{vec}\left(\widehat{W}_{f,i}\right)^\top, \ \text{vec}\left(\widehat{W}_{o,i}\right)^\top, \text{vec}\left(\widehat{W}_{h,i}\right)^\top\right]^\top \in \mathbb{R}^{4l_{2,i}l_{1,i}+l_{2,i}l_{3,i}}$ for all $i \in \mathcal{I}$. Using the LSTM model in (2–4) and the adaptive weight estimates $\hat{\vartheta}_i$, the estimates of the cell state $\hat{c}_i$ and hidden state $\hat{h}_i$ dynamics are

$$\dot{\hat{c}}_i = -b_{c,i}\hat{c}_i + b_{c,i}\left(f\left(\hat{z}_i, \widehat{W}_{f,i}\right) \odot c_i + p\left(\hat{z}_i, \widehat{W}_{p,i}\right) \odot \tilde{c}_i\left(\hat{z}_i, \widehat{W}_{c,i}\right)\right), \tag{2–12}$$

$$\dot{\hat{h}}_i = -b_{h,i}\hat{h}_i + b_{h,i}\left(o\left(\hat{z}_i, \widehat{W}_{o,i}\right) \odot \sigma_{c,i} \circ \Psi_{c,i}\left(\hat{z}_i, \hat{c}_i, \hat{\vartheta}_i\right)\right), \tag{2–13}$$

respectively, for all $i \in \mathcal{I}$, where $\hat{z}_i \triangleq \left[X_i^\top, \hat{h}_i^\top, 1\right]^\top$ denotes the augmented input of the LSTM. Substituting (2–11) into (2–10) yields

$$M(q)\dot{r} = -\left((\dot{q}_d - \alpha e)^\top \otimes I_n\right)(\chi_C - \widehat{\chi}_C + \varepsilon_C) - \left((\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\right)(\chi_M - \widehat{\chi}_M + \varepsilon_M) - e \\ - \chi_G + \widehat{\chi}_G - \chi_F + \widehat{\chi}_F - k_1 r - \tau_d - C(q,\dot{q})r - \text{sgn}(r)\left(k_2 + k_3\|(\dot{q}_d - \alpha e)^\top \otimes I_n\|\right) \\ - k_4\text{sgn}(r)\|(\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\| - \varepsilon_G - \varepsilon_F. \tag{2–14}$$

2.1.3 PI-LSTM Weight Adaptation Laws

The development of Lyapunov-based adaptation laws for the PI-LSTM architecture allows for continuous real-time adaptation. Based on the subsequent stability analysis, the weight adaptation laws are designed as

$$\dot{\widehat{\mathcal{Z}}}_M = -\Gamma_M \widehat{\chi}_M'^{\top} \left((\ddot{q}_d - \alpha \dot{e})^{\top} \otimes I_n \right)^{\top} r, \tag{2–15}$$

$$\dot{\widehat{\mathcal{Z}}}_C = -\Gamma_C \widehat{\chi}_C'^{\top} \left((\dot{q}_d - \alpha e)^{\top} \otimes I_n \right)^{\top} r, \tag{2–16}$$

$$\dot{\widehat{\mathcal{Z}}}_F = -\Gamma_F \widehat{\chi}_F'^{\top} r, \tag{2–17}$$

$$\dot{\widehat{\mathcal{Z}}}_G = -\Gamma_G \widehat{\chi}_F'^{\top} r, \tag{2–18}$$

where $\widehat{\mathcal{Z}}_i \triangleq \left[\hat{\vartheta}_i^{\top}, \hat{\theta}_i^{\top}\right]^{\top}$, and $\Gamma_i \in \mathbb{R}^{l_{4,i} \times l_{4,i}}$ is a positive-definite adaptation gain matrix, where $l_{4,i} \triangleq 4l_{2,i}l_{1,i} + l_{2,i}l_{3,i} + \varkappa_i$ for $i \in \mathcal{I}$ and $j \in \{0, ..., k_i\}$. The shorthand notation $\widehat{\chi}_i'$ denotes the Jacobian $\widehat{\chi}_i' \triangleq \frac{\partial \chi_i\left(X_i, \hat{\theta}_i, \hat{c}_i, \hat{h}_i, \hat{\vartheta}_i\right)}{\partial \widehat{\mathcal{Z}}_i}$. Therefore, the Jacobian can be expressed as $\widehat{\chi}_i' \triangleq \left[\widehat{\Xi}_i',\ \widehat{\Phi}_i'\right]$, where $\widehat{\Xi}_i' \triangleq \frac{\partial \Xi_i(X_i, \hat{c}_i, \hat{h}_i, \hat{\vartheta}_i)}{\partial \hat{\vartheta}_i}$ and $\widehat{\Phi}_i' \triangleq \frac{\partial \Phi_i\left(X_i, \hat{\theta}_i\right)}{\partial \hat{\theta}_i}$ denote the Jacobians of the LSTM and feedforward DNN (given in Section 1.6.2) for $i \in \mathcal{I}$, respectively.

Jacobians of the LSTM Architecture

The Jacobian of the LSTMs $\widehat{\Xi}_i'$ can be represented as $\widehat{\Xi}_i' \triangleq [\widehat{\Xi}_{W_{c,i}}', \widehat{\Xi}_{W_{p,i}}', \widehat{\Xi}_{W_{f,i}}', \widehat{\Xi}_{W_{o,i}}', \widehat{\Xi}_{W_{h,i}}']$, where the Jacobian of each weight estimate can be expressed as $\widehat{\Xi}_{W_{j,i}}' \triangleq \frac{\partial \Xi_i(x_i, \hat{c}_i, \hat{h}_i, \hat{\vartheta}_i)}{\partial \text{vec}\left(\widehat{W}_{j,i}\right)}$ for $j \in \{c, p, f, o, h\}$ and $i \in \mathcal{I}$. Based on (2–5) and the chain rule, $\widehat{\Xi}_{W_{h,i}}'$ and $\widehat{\Xi}_{W_{j,i}}'$ can be written using $\widehat{\Xi}_{W_{h,i}}' = I_n \otimes \Psi_h^{\top}\left(\hat{z}_i, \hat{c}_i, \hat{\vartheta}_i\right)$, $\widehat{\Xi}_{W_{j,i}}' = \widehat{W}_{h,i}^{\top} \widehat{\Psi}_{h,W_{j,i}}'$, and $\widehat{\Psi}_{h,W_{j,i}}' \triangleq \frac{\partial \Psi_h\left(\hat{z}_i, \hat{c}_i, \hat{\vartheta}_i\right)}{\partial \text{vec}\left(\widehat{W}_{j,i}\right)}$ for all $j \in \{c, p, f, o\}$ and $i \in \mathcal{I}$. Using (2–4), (2–12)-(2–13), the chain rule, properties of the Hadamard product, and the vectorization operator, the terms $\widehat{\Psi}_{h,W_{j,i}}'$ and $\widehat{\Psi}_{h,W_{o,i}}'$ can be written as [17]

$$\widehat{\Psi}_{h,W_{j,i}}' = \text{diag}\left(\sigma_{g,i}\left(\widehat{W}_{o,i}^{\top}\hat{z}\right)\right) \sigma_{c,i}'\left(\Psi_c\left(\hat{z}_i, \hat{c}_i, \hat{\vartheta}_i\right)\right) \widehat{\Psi}_{c,W_{j,i}}',$$
$$\widehat{\Psi}_{h,W_{o,i}}' = \text{diag}\left(\sigma_{c,i}\left(\Psi_c\left(\hat{z}_i, \hat{c}_i, \hat{\vartheta}_i\right)\right)\right)\left(\sigma_{g,i}'\left(\widehat{W}_{o,i}^{\top}\hat{z}_i\right)\right)\left(I_{l_2,i} \otimes \hat{z}_i^{\top}\right),$$

respectively, for all $j \in \{c, p, f\}$ and $i \in \mathcal{I}$, where $\widehat{\Psi}'_{c,W_{j,i}} \triangleq \frac{\partial \Psi_{c,i}\left(\hat{z}_i, \hat{c}_i, \hat{\vartheta}_i\right)}{\partial \text{vec}\left(\widehat{W}_{j,i}\right)}$, and the gradient of the sigmoid and tanh activation functions are expressed using the shorthand notation $\sigma'_{g,i} \in \mathbb{R}^{l_{2,i} \times l_{2,i}}$ and $\sigma'_{c,i} \in \mathbb{R}^{l_{2,i} \times l_{2,i}}$, respectively. Similarly, the terms $\widehat{\Psi}'_{c,W_{c,i}}$, $\widehat{\Psi}'_{c,W_{p,i}}$, and $\widehat{\Psi}'_{c,W_{f,i}}$ can be expressed as

$$\begin{aligned}
\widehat{\Psi}'_{c,W_{c,i}} &= \text{diag}\left(\sigma_{g,i}\left(\widehat{W}_{p,i}^{\top}\hat{z}\right)\right)\sigma'_{c,i}\left(W_{c,i}^{\top}\hat{z}\right)\left(I_{l_2,i} \otimes \hat{z}_i^{\top}\right), \\
\widehat{\Psi}'_{c,W_{p,i}} &= \text{diag}\left(\sigma_{c,i}\left(\widehat{W}_{c,i}^{\top}\hat{z}\right)\right)\sigma'_{g,i}\left(W_{p,i}^{\top}\hat{z}\right)\left(I_{l_2,i} \otimes \hat{z}_i^{\top}\right), \\
\widehat{\Psi}'_{c,W_{f,i}} &= \text{diag}\left(\hat{c}_i\right)\sigma'_{g,i}\left(\widehat{W}_{f,i}^{\top}\hat{z}\right)\left(I_{l_2,i} \otimes \hat{z}_i^{\top}\right),
\end{aligned}$$

respectively, for all $i \in \mathcal{I}$.

2.1.4 Stability Analysis

To address the additional level of complexity due to the nonlinearity of the DNN and LSTM weight estimates, a first-order Taylor series approximation-based error model is given by [20, Eq. 22].

$$\chi_i(X_i, \theta_i, \hat{c}_i, \hat{h}_i, \vartheta_i) - \widehat{\chi}_i = \widehat{\chi}'_i \widetilde{\mathcal{Z}}_i + \mathcal{O}_i^2\left(\|\widetilde{\mathcal{Z}}_i\|\right), \tag{2–19}$$

where $\mathcal{O}_i^2\left(\|\widetilde{\mathcal{Z}}_i\|\right)$ denotes the higher-order terms, $\tilde{\vartheta} \triangleq \vartheta - \hat{\vartheta}$, $\tilde{\theta} \triangleq \theta - \hat{\theta}$ and $\widetilde{\mathcal{Z}}_i \triangleq \left[\tilde{\vartheta}_i^{\top}, \tilde{\theta}_i^{\top}\right]^{\top}$ for $i \in \mathcal{I}$. Adding and subtracting $\chi(X_i, \theta_i, \hat{c}_i, \hat{h}_i, \vartheta_i),\ \forall i \in \mathcal{I}$, and using (2–19), the closed loop error system can be written as

$$\begin{aligned}
M(q)\dot{r} = &-\left(\left(\dot{q}_d - \alpha e\right)^{\top} \otimes I_n\right)\left(\widehat{\chi}'_C \widetilde{\mathcal{Z}}_C + N_{1,C}\right) - \left(\left(\ddot{q}_d - \alpha \dot{e}\right)^{\top} \otimes I_n\right)\left(\widehat{\chi}'_M \widetilde{\mathcal{Z}}_M + N_{1,M}\right) \\
&- \text{sgn}(r)\left(k_2 + k_3\|\left(\dot{q}_d - \alpha e\right)^{\top} \otimes I_n\|\right) - \widehat{\chi}'_G \widetilde{\mathcal{Z}}_G - N_{1,G} - \widehat{\chi}'_F \widetilde{\mathcal{Z}}_F - N_{1,F} \\
&- k_4 \text{sgn}(r)\|\left(\ddot{q}_d - \alpha \dot{e}\right)^{\top} \otimes I_n\| - k_1 r - e - \tau_d - C(q, \dot{q})r,
\end{aligned} \tag{2–20}$$

where the auxiliary function $N_{1,i} \in \mathbb{R}^{l_{3,i}}$ is defined as $N_{1,i} \triangleq \chi_i(X_i, \theta_i, c_i, h_i, \vartheta_i) - \chi_i(X_i, \theta_i, \hat{c}_i, \hat{h}_i, \vartheta_i) + \varepsilon_i(X_i) + \mathcal{O}_i^2\left(\|\widetilde{\mathcal{Z}}_i\|\right)$ for all $i \in \mathcal{I}$, and $N_{1,i}$ can be bounded as $\|N_{1,i}\| \leq \overline{N_{1,i}}$, for a known constant $\overline{N_{1,i}} \in \mathbb{R}_{>0}$ [17, Lemma 1].

To facilitate the subsequent stability analysis, let the concatenated state vector $\zeta \in \mathbb{R}^{\psi}$ be defined as $\zeta \triangleq [e^{\top}, r^{\top}, \widetilde{\mathcal{Z}}_M^{\top}, \widetilde{\mathcal{Z}}_C^{\top}, \widetilde{\mathcal{Z}}_F^{\top}, \widetilde{\mathcal{Z}}_G^{\top}]^{\top}$, where $\psi \triangleq 2n + \sum_{i \in \mathcal{I}} l_{4,i}$, and let the open and connected set $\mathcal{B}_{\Lambda} \subset \mathbb{R}^{\psi}$ be defined as $\mathcal{B}_{\Lambda} \triangleq \left\{ \xi \in \mathbb{R}^{\psi} : \|\xi\| \leq \sqrt{\frac{\beta_1}{\beta_2}} \omega \right\}$, where $\omega \in \mathbb{R}_{>0}$ denotes a user selected constant. The following theorem establishes tracking error convergence using the developed PI-LSTM-based adaptive controller.

Theorem 2.1. *For the dynamical system in (2–1), the controller in (2–11) and the adaptation laws developed in (2–15)-(2–18) ensure asymptotic tracking in the sense that* $\|e(t)\| \to 0$ *and* $\|r(t)\| \to 0$ *as* $t \to \infty$*, provided Assumption 1 holds,* $\zeta(t_0) \in \mathcal{B}_{\Lambda}$*, and the gain conditions* $k_2 \geq \overline{d} + \overline{N_{1,F}} + \overline{N_{1,G}}$*,* $k_3 \geq \overline{N_{1,C}}$*, and* $k_4 \geq \overline{N_{1,M}}$ *are satisfied.*

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

$$\mathcal{V}_L(\zeta) \triangleq \frac{1}{2} e^{\top} e + \frac{1}{2} r^{\top} M r + \sum_{i \in \mathcal{I}} \frac{1}{2} \widetilde{\mathcal{Z}}_i \Gamma_i^{-1} \widetilde{\mathcal{Z}}_i. \tag{2–21}$$

The candidate Lyapunov function in (2–51) satisfies the inequality $\beta_1 \|\zeta\|^2 \leq \mathcal{V}_L(\zeta) \leq \beta_2 \|\zeta\|^2$, where $\beta_1 \triangleq \min\left\{ \frac{1}{2}, \frac{1}{2} m_1, \frac{1}{2} \min_{i \in \mathcal{I}} (\lambda_{\min}(\Gamma_i)) \right\}$ and $\beta_2 \triangleq \max\left\{ \frac{1}{2}, \frac{1}{2} m_2, \frac{1}{2} \max_{i \in \mathcal{I}} (\lambda_{\max}(\Gamma_i)) \right\}$. Taking the time-derivative of $\mathcal{V}_L(\zeta)$, applying the chain rule for differential inclusions in [108, Thm 2.2], and applying (2–14) and Property 2 yields

$$\begin{aligned} \dot{\mathcal{V}}_L(\zeta) \overset{a.a.t.}{\in} & -e^{\top} \alpha e + r^{\top} \Big(-\widehat{\chi}_G' \widetilde{\mathcal{Z}}_G - N_{1,G} - \widehat{\chi}_F' \widetilde{\mathcal{Z}}_F - N_{1,F} \\ & - \left((\ddot{q}_d - \alpha \dot{e})^{\top} \otimes I_n \right) \left(\widehat{\chi}_M' \widetilde{\mathcal{Z}}_M + N_{1,M} \right) - \left((\dot{q}_d - \alpha e)^{\top} \otimes I_n \right) \left(\widehat{\chi}_C' \widetilde{\mathcal{Z}}_C + N_{1,C} \right) \\ & - \varepsilon_G - \varepsilon_F - \tau_d - k_1 r - \mathsf{K}\,[\mathsf{sgn}]\,(r) \Big(k_2 + k_3 \|(\dot{q}_d - \alpha e)^{\top} \otimes I_n\| \\ & + k_4 \|(\ddot{q}_d - \alpha \dot{e})^{\top} \otimes I_n\| \Big) \Big) - \sum_{i \in \mathcal{I}} \left(\frac{1}{2} \dot{\widehat{\mathcal{Z}}}_i \Gamma_i^{-1} \widetilde{\mathcal{Z}}_i \right). \end{aligned} \tag{2–22}$$

Substituting (2–15)-(2–19), and combining like terms yields

$$\begin{aligned} \dot{\mathcal{V}}_L(\zeta) \overset{a.a.t.}{\leq} & -e^{\top} \alpha e - r^{\top} \Big(\left((\ddot{q}_d - \alpha \dot{e})^{\top} \otimes I_n \right) N_{1,M} + N_{1,G} + k_1 r \\ & + K\,[\mathsf{sgn}]\,(r) \left(k_2 + k_3 \|(\dot{q}_d - \alpha e)^{\top} \otimes I_n\| \right) + k_4 K\,[\mathsf{sgn}]\,(r) \|(\ddot{q}_d - \alpha \dot{e})^{\top} \otimes I_n\| + \tau_d \end{aligned}$$

$$+\left(\left(\dot{q}_d-\alpha e\right)^{\top}\otimes I_n\right)N_{1,C}+N_{1,F}\Big). \tag{2–23}$$

Provided the stated gain conditions are satisfied, (2–23) can be bounded as

$$\dot{\mathcal{V}}_L\left(\zeta\right)\overset{a.a.t.}{\leq}-\alpha\|e\|^2-k_1\|r\|^2. \tag{2–24}$$

Using (2–21) and (2–24) implies $e,r,\tilde{\vartheta}_i,\tilde{\theta}_i\in\mathcal{L}_\infty$. The fact that $q_d,\dot{q}_d,e,r\in\mathcal{L}_\infty$ implies $q,\dot{q}\in\mathcal{L}_\infty$. To show $X_i\in\Omega_i$, and therefore that the universal function approximation property holds for all $i\in\mathcal{I}$, let the open and connected sets $\Upsilon_i\subseteq\Omega_i$ be defined as $\Upsilon_M\triangleq\{\xi\in\Omega_M:\|\xi\|<\overline{q}_d+\omega\}$, $\Upsilon_C\triangleq\left\{\xi\in\Omega_C:\|\xi\|<\overline{q}_d+(2+\alpha)\omega+\overline{\dot{q}}_d\right\}$, $\Upsilon_F\triangleq\left\{\xi\in\Omega_F:\|\xi\|<(1+\alpha)\omega+\overline{\dot{q}}_d\right\}$, and $\Upsilon_G\triangleq\{\xi\in\Omega_G:\|\xi\|<\overline{q}_d+\omega\}$. The facts that $\dot{\mathcal{V}}_L\left(\zeta\left(t\right)\right)\overset{a.a.t.}{\leq}0$, and $\beta_1\left\|\zeta\right\|^2\leq\mathcal{V}_L\left(\zeta\right)\leq\beta_2\left\|\zeta\right\|^2$ imply that $\zeta\left(t\right)$ can be bounded as $\|\zeta(t)\|\leq\sqrt{\frac{\beta_2}{\beta_1}}\|\zeta\left(t_0\right)\|$. If $\|\zeta\left(t_0\right)\|\in\sqrt{\frac{\beta_1}{\beta_2}}\omega$, then $\|\zeta\left(t\right)\|\leq\omega$, and therefore $\|e\left(t\right)\|\leq\omega$ and $\|r\left(t\right)\|\leq\omega$. Therefore, if $\zeta\left(t_0\right)\in\mathcal{B}_\Lambda$, then $X_i\in\Upsilon_i\subseteq\Omega_i$. Then, using (2–21), (2–24), the extension of the LaSalle-Yoshizawa corollary in [109, Corollary 1] can be invoked to show that $\|e(t)\|\to0$ and $\|r(t)\|\to0$ as $t\to\infty$. □

2.1.5 Simulations

Simulation results are provided to demonstrate the performance of the developed method using a two-link planar revolute robot as modeled in [17]. Simulations were ran for 50 s with the selected desired trajectory $q_d(t)\triangleq[q_{d1},q_{d2}]^{\top}\in\mathbb{R}^2$ as $q_d\triangleq(1-\exp(-0.1))\begin{bmatrix}\frac{3\pi}{8}\sin\left(\frac{\pi}{2}t\right)\\ \frac{3\pi}{8}\sin\left(\frac{\pi}{2}t\right)\end{bmatrix}\in\mathbb{R}^2$ [rad] with sensor and process noise generated from a Gaussian distribution with standard deviation of 0.001 was injected into the simulation data. The simulation is initialized at $q(0)=[0.4,-0.3]^{\top}$ [rad] and $\dot{q}(0)=[0,0]^{\top}$ [rad/s]. To highlight the contribution of the combined estimation power of DNNs and LSTMs, the PI-LSTM was composed of 3 of DNNs with 4 layers and 7 neurons and tanh activation functions for the M,C, and F matrices while the LSTM model was used with $l_{2,i}=2$ neurons for $i\in\{M,C,F\}$. The resulting architecture has 817 individual weights, with

Table 2-1. Control gains

	α	k_1	k_2	k_3	k_4	M	C	F
Baseline Method [2]	2.2	9.6	0.7	0.1	0.4			
Γ_{DNN}						1.6	3.1	2.3
Developed Method	7.0	0.8	0.6	0.7	0.2			
Γ_{DNN}						1.2	3.6	1.9
Γ_{LSTM}						1.0	10.8	9.2
b_c						5.9	7.6	1.6
b_h						2.7	7.7	2.0

the DNN and LSTM having 661 and 156 individual weights, respectively. A comparative simulation was performed with the architecture developed in [2] as a baseline (i.e without the contribution of the memory properties gained from the LSTMs). The weights of the DNNs and LSTMs were randomly initialized from a uniform distribution $U(-1, 1)$. The gains were selected as shown in Table 2-1 with the learning gains Γ_i selected as $\Gamma_i = \text{diag}[\Gamma_{1,i}, \Gamma_{2,i}]$, where $\Gamma_{1,i} = \Gamma_{i,LSTM} \cdot I_{4l_{2,i}l_{1,i}+l_{2,i}l_{3,i}}$, and $\Gamma_{2,i} = \Gamma_{i,DNN} \cdot I_{\varkappa_i}$ for $i \in \{M, C, F\}$. The root mean squared (RMS) tracking error for the Lb-PINN and PI-LSTM controllers were of 0.0279 rad and 0.0185 rad, respectively. Although both architectures achieved rapid tracking error convergence as shown in Figure 2-1, the developed PI-LSTM converged faster and was more robust to the injected noise. The developed method yielded a 33.76% improvement compared to the developed architecture in [2].

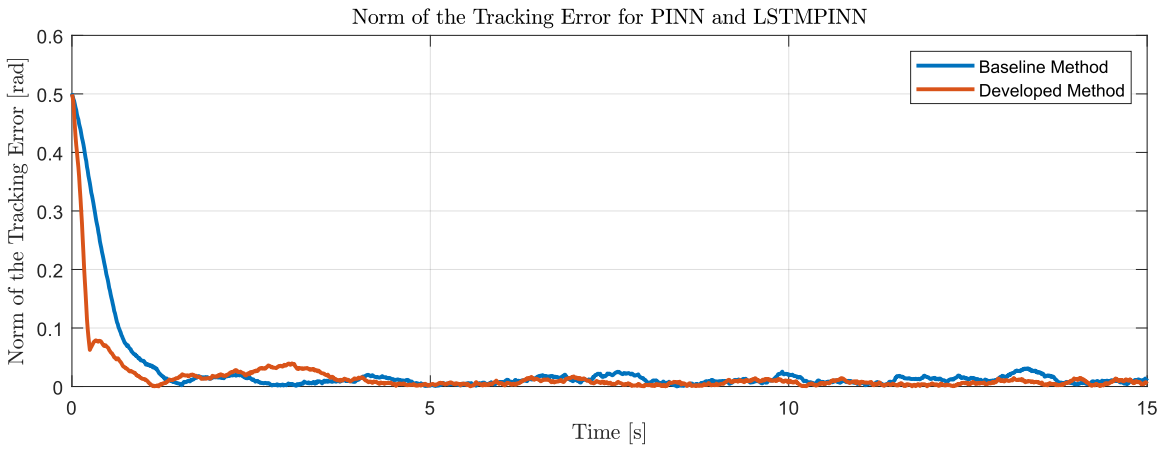

Figure 2-1. Comparison of tracking error norm for the baseline developed in [2] and the developed method.

2.1.6 Conclusions

This section provides the first result on Lyapunov-derived adaptation laws for the weights of a novel PI-LSTM architecture. The combination of the LSTM term within a physics-informed structure allows the architecture to effectively capture long term dynamic behavior. Comparative simulation results resulted in a 0.0185 rad RMS tracking error yielding a 33.76% improvement over the baseline. Future work would include constraining the output of the PI-LSTM to further respect other physical properties such as the positive definiteness of the inertia matrix.

2.2 Lyapunov-based Physics-Informed Deep Neural Networks with Skew Symmetry Considerations

For simplicity, the following development considers a fully-connected DNN described in Section 1.6.2, though the control and adaptation laws can be generalized to other architectures or used in a combined approach presented in the previous section of this chapter.

Motivated by the objective to incorporate system knowledge into the learning process, this section develops an update law which utilizes physics-informed regularization to promote convergence to outputs that satisfy known physical constraints in real-time. A Lyapunov-based stability analysis is provided to guarantee asymptotic convergence

of both the tracking error and the skew-symmetric prediction error. Simulation results demonstrate that the Skew-Symmetric Lyapunov-based Physics-informed Neural Network (SS-LbPINN) framework yields a 19.87% improvement in overall function approximation compared to methods which do not consider the symmetry constraint. Specifically, the method achieves 4.84%, 21.78%, and 2.52% improvements in the approximation of the M, C, and F matrices, respectively, while maintaining comparable tracking performance and control effort.

2.2.1 Time-Derivative of Deep Neural Network (DNN) Model

The subsequent chapter will utilize the time-derivative of the DNN described in Section 1.6.2. The time derivative of the DNN, represented by $\frac{d}{dt}\left(\Phi_i(X_i,\theta_i)\right) \in \mathbb{R}^{n_i}$, follows from the chain rule applied to (1–3), accounting for the change in both X_i and θ_i over time for all $i \in \mathcal{I}$. The time derivative of the Jacobian is represented as $\frac{d}{dt}\Phi_i' \triangleq \left[\frac{d}{dt}\Phi_{0,i}', \ldots, \frac{d}{dt}\Phi_{j,i}'\right] \in \mathbb{R}^{n_i \times \varkappa_i}$ where $\frac{d}{dt}\Phi_{j,i}' \triangleq \frac{d}{dt}\left(\frac{\partial \Phi_{j,i}(X_i,\hat{\theta}_i)}{\partial \hat{\theta}_i}\right) \in \mathbb{R}^{n_i \times L_{j+1,i}}$ where $\frac{d}{dt}\Phi_{0,i}'$ and $\frac{d}{dt}\Phi_{j,i}'$ can be obtained by applying chain rule to (1–4) for all $j \in \{0, \ldots k_i\}$ and $i \in \mathcal{I}$.

Additionally, Assumption 1.1 regarding the boundedness of the activation function must be modified as follows.

Assumption 2.2. For all $y \in \mathbb{R}^n$ and for each $j \in \{0, \ldots, k\}$, the activation function ϕ_j, its Jacobian ϕ_j', Hessian $\phi_j''(y) \triangleq \frac{\partial^2}{\partial y^2}\phi_j(y)$, and third-order tensor $\phi'''(y) \triangleq \frac{\partial^3}{\partial y^3}\phi_j(y)$ are bounded as

$$\begin{aligned} \|\phi_j(y)\| &\leq \mathfrak{a}_0, \qquad \left\|\phi_j''(y)\right\| \leq \mathfrak{c}_0, \\ \left\|\phi_j'(y)\right\| &\leq \mathfrak{b}_0, \qquad \left\|\phi_j'''(y)\right\| \leq \mathfrak{d}_0, \end{aligned} \tag{2–25}$$

where $\mathfrak{a}_0, \mathfrak{b}_0, \mathfrak{c}_0, \mathfrak{d}_0 \in \mathbb{R}_{\geq 0}$ are known constants.

Remark 2.1. Most activation functions used in practice satisfy Assumption 2.2. Specifically, sigmoidal activation functions (e.g., logistic function, hyperbolic tangent, etc.) have $\|\phi_j(y)\|$, $\left\|\phi_j'(y)\right\|$, $\left\|\phi_j''(y)\right\|$, and $\left\|\phi_j'''(y)\right\|$ bounded uniformly by constants.

2.2.2 Model Dynamics and Control Objective

Consider an uncertain Euler-Lagrange system modeled as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) = \tau(t), \tag{2–26}$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ denote the generalized position, velocity, and acceleration, respectively. The system dynamics are characterized by the unknown generalized inertia matrix, generalized centripetal-Coriolis effects, generalized potential forces, generalized dissipation effects, and the subsequently designed control input which are denoted by $M \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times n}$, $G \in \mathbb{R}^n$, $F \in \mathbb{R}^n$, and $\tau \in \mathbb{R}^n$, respectively. The system in (2–26) satisfies the properties given in Property 1 and 2.

The tracking control objective is to design a controller to asymptotically track a user-defined, time-varying desired trajectory, $q_d \in \mathbb{R}^n$, which is designed to be sufficiently smooth such that $q_d(t), \dot{q}_d(t), \ddot{q}_d(t) \in \mathcal{Q},$ for all $t \in \mathbb{R}_{\geq 0}$, where $\mathcal{Q} \subseteq \mathbb{R}^n$ denotes a known compact set. Quantitatively, the tracking objective is to ensure $\|e(t)\| \to 0$ and $\|r(t)\| \to 0$ as $t \to \infty$. Using (2–26) and (2–2), the open-loop dynamics for r can be given as

$$M(q)\dot{r} = \tau - M(q)\left(\ddot{q}_d - \alpha\dot{e}\right) - C(q,\dot{q})\left(\dot{q}_d - \alpha e\right) - G(q) - F(\dot{q}) - C(q,\dot{q})r. \tag{2–27}$$

Adaptive Physics-Informed Control Development

This physics-informed formulation described in Section 2.1 provides insight into the specific inputs to the DNNs, ensuring a more informed learning process and allows for the development of a skew-symmetric prediction error due to the availability of individual estimates of the skew-symmetric components. Consider the family of DNNs described in Section 1.6.2 where $\mathcal{I} \triangleq \{M, C, G, F\}$, and let $\mathcal{F} \triangleq \{\text{vec}(M(q)), \text{vec}(C(q,\dot{q})), G(q), F(\dot{q})\}$ where $\mathcal{F}_i$ represents the corresponding vector in $\mathcal{F}$ for the i^{th} element in $\mathcal{I}$. The universal function approximation theorem [107, Theorem 3.1] states that the function space of DNNs is dense in the space

of continuous functions $\mathcal{C}(\Omega_i)$ for $i \in \mathcal{I}$. As a result, there exists a DNN $\Phi_i(X_i, \theta_i^*)$ and corresponding parameters $\theta_i^* \in \mathbb{R}^{\varkappa_i}$ such that $\sup_{X_i \in \Omega_i} \|\mathcal{F}_i - \Phi_i(X_i, \theta_i^*)\| < \varepsilon_i$ for $i \in \mathcal{I}$.

Remark 2.2. It is not known how to obtain a bound $\bar{\theta}_i$ on such parameter θ_i^* for an arbitrary ε_i, which causes difficulties in constructing the bounded search space Θ_i. Therefore, we allow $\bar{\theta}_i$ to be arbitrarily selected in the above analysis, at the loss of guarantees on the approximation accuracy. Although the bound ε_i which bounds $\sup_{X_i \in \Omega_i} \|\mathcal{F}_i - \Phi_i(X_i, \theta_i^*)\|$ might no longer be arbitrary in this case, it would still exist due to the continuity of the function being estimated denoted by index i and Φ_i.

Based on this property, the unknown terms, $M(q),\ C(q, \dot{q}),\ G(q)$, and $F(\dot{q})$ can be modeled as

$$\text{vec}\left(M(q)\right) = \Phi_M(X_M, \theta_M^*) + \varepsilon_M(X_M), \tag{2–28}$$

$$\text{vec}\left(C(q, \dot{q})\right) = \Phi_C(X_C, \theta_C^*) + \varepsilon_C(X_C), \tag{2–29}$$

$$G(q) = \Phi_G(X_G, \theta_G^*) + \varepsilon_G(X_G), \tag{2–30}$$

$$F(\dot{q}) = \Phi_F(X_F, \theta_F^*) + \varepsilon_F(X_F), \tag{2–31}$$

respectively, where $\Phi_i(X_i, \theta_i^*) \in \mathbb{R}^{L_{out,i}}$ for $i \in \mathcal{I}$ are the DNNs with weights which yield the best approximation of $\mathcal{F}_i$ in the search space Θ_i, $L_{out,i} = n^2$ for $i \in \{M, C\}$, and $L_{out,i} = n$ for $i \in \{F, G\}$. The inputs to the individual DNNs are defined as $X_M \triangleq q$, $X_C \triangleq [q, \dot{q}]^\top$, $X_G \triangleq q$, $X_F \triangleq \dot{q}$. The unknown function approximation errors are denoted by $\varepsilon_M(X_M) : \mathbb{R}^n \to \mathbb{R}^{n^2}$, $\varepsilon_C(X_C) : \mathbb{R}^{2n} \to \mathbb{R}^{n^2}$, $\varepsilon_G(X_G) : \mathbb{R}^n \to \mathbb{R}^n$, and $\varepsilon_F(X_F) : \mathbb{R}^n \to \mathbb{R}^n$ for $\Phi_M(X_M, \theta_M),\ \Phi_C(X_C, \theta_C),\ \Phi_G(X_G, \theta_G)$, and $\Phi_F(X_F, \theta_F)$, respectively. The following assumption is made to facilitate the subsequent control development.

Assumption 2.3. Given any compact set Ω_i, there exist known constants $\overline{\varepsilon_i}, \overline{\varepsilon_i'} \in \mathbb{R}_{>0}$ such that the function approximation error and it's partial derivative with respect to the input can be bounded as $\varepsilon_i(X_i) \leq \overline{\varepsilon}_i$ and $\left\| \frac{\partial}{\partial X_i} \varepsilon_i(X_i) \right\| \leq \overline{\varepsilon_i'}$ for all $i \in \mathcal{I}$ and $X_i \in \Omega_i$. For an analysis of the function approximation properties of partial derivatives, see [110, Thm. 4.1].

Remark 2.3. Assumption 2.3 is reasonable because in practice the user can select $\overline{\theta}_i$ a priori and $\overline{\varepsilon}_i, \overline{\varepsilon_i'}$ can subsequently be prescribed for $i \in \mathcal{I}$ using a conservative estimate whose feasibility can be verified using heuristic methods (e.g., Monte Carlo search). Notably, DNN architectures which contain spectral normalization layers as in [111] inherently involve bounded weights since the weight matrices are normalized by their spectral norms.

Applying the vectorization operator and its properties on M and C, and substituting (2–28)-(2–31) into (2–27) yields the open loop error system

$$M(q)\dot{r} = \tau - \Phi_G(X_G, \theta_G^*) - \Phi_F(X_F, \theta_F^*) - \left((\dot{q}_d - \alpha e)^\top \otimes I_n\right)\left(\Phi_C(X_C, \theta_C^*) + \varepsilon_C(X_C)\right) - \left((\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\right)\left(\Phi_M(X_M, \theta_M^*) + \varepsilon_M(X_M)\right) - \varepsilon_G(X_G) - \varepsilon_F(X_F) - C(q, \dot{q})r. \tag{2–32}$$

Based on the subsequent stability analysis, the adaptive SS-LbPINN control input is defined as

$$\begin{aligned}\tau \triangleq & \left((\dot{q}_d - \alpha e)^\top \otimes I_n\right)\Phi_C(X_C, \hat{\theta}_C) + \Phi_G(X_G, \hat{\theta}_G) + \Phi_F(X_F, \hat{\theta}_F) \\ & - k_1 r - e + \left((\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\right)\Phi_M(X_M, \hat{\theta}_M) - \text{sgn}(r)\Big(k_2 \\ & + k_3 \left\|(\dot{q}_d - \alpha e)^\top \otimes I_n\right\| + k_4 \left\|(\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\right\|\Big),\end{aligned} \tag{2–33}$$

where $k_1, k_2, k_3, k_4 \in \mathbb{R}_{>0}$ are a user-defined control gains, $\Phi_i(X_i, \hat{\theta}_i) \in \mathbb{R}^{L_{out,i}}$ denotes the DNN estimate of the unknown matrices, and the individual adaptive DNN weight estimates are denoted as $\hat{\theta}_i \in \mathbb{R}^{\varkappa_i}$, for all $i \in \mathcal{I}$. To address the additional level of complexity due to the nonlinearity in the DNN, a first order Taylor series approximation-based error model is given by [20, Eq. 22]

$$\Phi_i(X_i, \theta_i^*) - \Phi_i(X_i, \hat{\theta}_i) = \Phi'(X_i, \hat{\theta}_i)\tilde{\theta}_i + R_i(X_i, \tilde{\theta}_i), \tag{2–34}$$

where $R_i : \mathbb{R}^{L_{in,i}} \times \mathbb{R}^{\varkappa_i} \to \mathbb{R}^{L_{out,i}}$ denotes the Lagrange remainder, $\Phi'(X_i, \hat{\theta}_i)$ denotes the Jacobian described in Section 2.2.1, and $\tilde{\theta}_i \triangleq \theta_i - \hat{\theta}_i$ denotes the parameter estimation

error for $i \in \mathcal{I}$. Using (2–32)-(2–34) the closed loop error system in (2–32) can be written as

$$M(q)\dot{r} = -k_1 r - e - C(q,\dot{q})r - \text{sgn}(r)\Big(k_2 + k_3 \left\|(\dot{q}_d - \alpha e)^\top \otimes I_n\right\| + k_4 \left\|(\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\right\|\Big) - \Big((\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\Big)\Big(\Phi'(X_M, \hat{\theta}_M)\tilde{\theta}_M + N_M\Big) - \Big((\dot{q}_d - \alpha e)^\top \otimes I_n\Big)\Big(\Phi'(X_C, \hat{\theta}_C)\tilde{\theta}_C + N_C\Big) - \Phi'(X_G, \hat{\theta}_G)\tilde{\theta}_G - N_G - \Phi'(X_F, \hat{\theta}_F)\tilde{\theta}_F - N_F, \tag{2–35}$$

where the auxiliary function $N_i \in \mathbb{R}^{L_{out,i}}$ is defined as $N_i \triangleq \varepsilon_i(X_i) + R_i(X_i, \tilde{\theta}_i)$ for all $i \in \mathcal{I}$.

2.2.3 Skew-Symmetric Prediction Error and Adaptive Update Laws

2.2.3.1 Skew-Symmetric Prediction Error Formulation

To incorporate known properties of the system, a skew-symmetric prediction error is developed to inform the update law. Specifically, the skew-symmetry relation stated in Property 2 describes an inherent structure in the system dynamics and is used to develop a penalty in the adaption. The developed skew-symmetric prediction error penalizes updates that violate the skew-symmetry constraint. To quantify the skew-symmetric objective, a skew-symmetric prediction error $\widetilde{E} \in \mathbb{R}$ is defined as

$$\widetilde{E}(t) \triangleq E - \widehat{E}, \tag{2–36}$$

where $\widehat{E}$ is the subsequently designed skew symmetric estimate, $E \triangleq \int_0^t \xi^\top \left(\frac{d}{d\varsigma} M(X_M(\varsigma)) - 2C(X_C(\varsigma))\right)\xi d\varsigma$, and $\xi \in \mathbb{R}^n$ denotes a vector of positive control gains.

Remark 2.4. The subsequently designed adaptation law may also utilize a sum of ξ vectors. In this case $E \triangleq \sum_{z=1}^{Z} \int_0^t \xi_z^\top \left(\frac{d}{d\varsigma} M(X_M(\varsigma)) - 2C(X_C(\varsigma))\right)\xi_z d\varsigma$ where Z denotes the number of vectors selected in a set of linearly independent vectors $\{\xi_z\}_{z=1}^{Z} \in \mathbb{R}^n$. This summation may be utilized to ensure that the set of vectors $\{\xi_z\}_{z=1}^{Z}$ sufficiently

spans the input space and ensures the estimates of $\frac{d}{dt}M(X_M(t)) - 2C(X_C(t))$ remain within the skew-symmetric subspace of $\mathbb{R}^{n\times n}$.

Using Property 2, and the fact that $E = 0$, indicates that $\widetilde{E}(t) = -\widehat{E}$. Quantitatively, the objective of the skew-symmetric constraint is to ensure $\left\|\widetilde{E}(t)\right\| \to 0$ as $t \to \infty$. The corresponding skew-symmetric estimate given by $\widehat{E} \in \mathbb{R}$ is defined as a Filippov solution of

$$\dot{\widehat{E}} \triangleq \left(\xi^\top \otimes \xi^\top\right)\left(\frac{d}{dt}\Big(\Phi(X_M, \hat{\theta}_M)\Big) - 2\Phi(X_C, \hat{\theta}_C) + \mu\right), \tag{2–37}$$

where $\mu(t) \in \mathbb{R}^n$ is given by

$$\mu(t) \triangleq -\widehat{\Phi}'_M\dot{\hat{\theta}}_M + \gamma_1\widetilde{E} + \left(\gamma_2 + \gamma_3 + \gamma_4\left\|\dot{X}_M\right\|\right)\text{sgn}(\widetilde{E}), \tag{2–38}$$

where $\gamma_1, \gamma_2, \gamma_3, \gamma_4 \in \mathbb{R}$ denote constant positive control gains.

To facilitate the subsequent stability analysis, it is desirable to obtain an analytical expression for $\widetilde{E}$ which allows the skew-symmetric prediction error to be explicitly related to the parameter estimation error of the skew-symmetric matrix estimates (i.e., $\tilde{\theta}_M$ and $\tilde{\theta}_C$). Therefore, an equivalent representation for E is developed using properties of vectorization and the Kronecker product which allows E to be expressed as $E = \int_0^t \left(\xi^\top \otimes \xi^\top\right)\left(\text{vec}\left(\frac{d}{d\varsigma}M(X_M(\varsigma))\right) - \text{vec}(2C(X_C(\varsigma)))\right)d\varsigma$. Using (2–29) and the time derivative of (2–28), and the fact that E is differentiable almost everywhere, then its time-derivative, where it exists, is given by

$$\dot{E} = \left(\xi^\top \otimes \xi^\top\right)\left(\frac{d}{dt}\left(\Phi(X_M, \theta^*_M) + \varepsilon(X_M)\right) - 2\left(\Phi(X_C, \theta^*_C) + \varepsilon(X_C)\right)\right). \tag{2–39}$$

Substituting (2–37) and (2–39) into (2–36) and applying (2–34) yields the skew symmetric prediction error $\widetilde{E}$ given by the Filippov solution of

$$\dot{\widetilde{E}} = \left(\xi^\top \otimes \xi^\top\right)\left(\frac{d}{dt}\left(\Phi'(X_M, \hat{\theta}_M)\tilde{\theta}_M + N_M\right) - 2\left(\Phi'(X_C, \hat{\theta}_C)\tilde{\theta}_C + N_C\right) - \mu\right). \tag{2–40}$$

Applying the chain rule to $\frac{d}{dt}(\Phi'(X_M, \hat{\theta}_M)\tilde{\theta}_M)$, using (2–38), and canceling cross terms, $\dot{\widetilde{E}}$ can be equivalently expressed as

$$\dot{\widetilde{E}} = \left(\xi^\top \otimes \xi^\top\right)\left(\frac{d}{dt}\left(\Phi'(X_M, \hat{\theta}_M)\right)\tilde{\theta}_M + \frac{d}{dt}\left(N_M\right) - \gamma_1 \widetilde{E}\right.$$
$$\left. - 2\left(\Phi'(X_C, \hat{\theta}_C)\tilde{\theta}_C + N_C\right) - \left(\gamma_2 + \gamma_3 + \gamma_4 \left\|\dot{X}_M\right\|\right)\text{sgn}(\widetilde{E})\right). \tag{2–41}$$

2.2.3.2 SS-LbPINN Weight Adaptation Laws

Lyapunov-based adaptation laws which incorporate skew-symmetric penalties enable continuous, real-time parameter updates which are guided by the performance of the system and the underlying known physical properties. Specifically, the subsequent adaptation laws for the inertial and centripetal-Coriolis estimates incorporate the skew-symmetric prediction error term developed in Section 2.2.3.1. The Lyapunov-based adaptation laws for the SS-LbPINN are defined as

$$\dot{\hat{\theta}}_M \triangleq \text{proj}\Big(\Gamma_M\Big(\frac{d}{dt}\left(\Phi'\left(X_M, \hat{\theta}_M\right)\right)^\top (\xi \otimes \xi)\,\widetilde{E}$$
$$- \Phi'^\top\left(X_M, \hat{\theta}_M\right)\left((\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\right)^\top r\Big), \Theta_M\Big), \tag{2–42}$$
$$\dot{\hat{\theta}}_C \triangleq \text{proj}\Big(\Gamma_C\Big(-2\Phi'^\top(X_C, \hat{\theta}_C)\,(\xi \otimes \xi)\,\widetilde{E}$$
$$- \Phi'^\top(X_C, \hat{\theta}_C)\left((\dot{q}_d - \alpha e)^\top \otimes I_n\right)^\top r\Big), \Theta_C\Big), \tag{2–43}$$
$$\dot{\hat{\theta}}_F \triangleq \text{proj}\Big(-\Gamma_F \Phi'^\top(X_F, \hat{\theta}_F) r, \Theta_F\Big), \tag{2–44}$$
$$\dot{\hat{\theta}}_G \triangleq \text{proj}\Big(-\Gamma_G \Phi'^\top(X_G, \hat{\theta}_G) r, \Theta_G\Big), \tag{2–45}$$

where $\Gamma_i \in \mathbb{R}^{\varkappa_i \times \varkappa_i}$ for $i \in \mathcal{I}$ is a positive definite user-defined adaptation gain matrix and $\text{proj}\left(\cdot\right)$ denotes a continuous projection operator (cf. [112, Appendix E]) which ensures $\hat{\theta}_i(t) \in \mathbb{R}^{\varkappa_i} \triangleq \left\{\theta_i \in \mathbb{R}^{\varkappa_i} : \|\theta_i\| \leq \overline{\theta}_i\right\}$ for all $t \in \mathbb{R}_{\geq 0}$ and $i \in \mathcal{I}$.

2.2.4 Stability Analysis

Let $z \triangleq \left[e^\top, r^\top, \widetilde{E}\right]^\top \in \mathbb{R}^{2n+1}$ denote the concatenated state. Using [113, Thm. 1] and the user-selected bound on the search space, the Lagrange remainder can be bounded as $\left\|R_i(X_i, \tilde{\theta}_i)\right\| \leq 4\rho_{0,i}\left(\|X_i\|\right)\overline{\theta}_i^2$ where $\rho_{0,i} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a non-decreasing

positive function of the form $\rho_{0,i}(\|X_i\|) = a_{2,i}\|X_i\|^2 + a_{1,i}\|X_i\| + a_{0,i}$ with some constants $a_{2,i}, a_{1,i}, a_{0,i} \in \mathbb{R}_{>0}$, for all $i \in \mathcal{I}$. Using the same process as [113], applying the chain rule to find the time derivative, and using the activation function bounds in (2–25), the time derivative of $R_M(X_M, \tilde{\theta}_M)$ can be bounded as $\left\|\frac{d}{dt}R_M(X_M, \tilde{\theta}_M)\right\| \leq 2\overline{\theta}_M^2\rho_{0,M}(\|X_M\|) + 4\overline{\theta}_M^2\rho_T(\|X_M\|, \left\|\dot{X}_M\right\|)$ where $\rho_T : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a non-decreasing positive function of the form $\rho_T(\|X_M\|, \left\|\dot{X}_M\right\|) = b_{h,i}\|X_i\|^h + b_{h,i}\left\|\dot{X}_i\right\|^h + b_{h-1,i}\|X_i\|^{h-1} + b_{h-1,i}\left\|\dot{X}_i\right\|^{h-1} + \cdots + b_{0,i}$ for some $b_{h,i}, \ldots, b_{0,i} \in \mathbb{R}_{>0}$. The terms N_i and $\frac{d}{dt}N_M$ can be bounded as $\|N_i\| \leq \left\|R_i(X_i, \tilde{\theta}_i) + \varepsilon_i(X_i)\right\|$ for $i \in \mathcal{I}$ and $\left\|\frac{d}{dt}N_M\right\| \leq \left\|\frac{d}{dt}R_M(X_M, \tilde{\theta}_M)\right\| + \left\|\frac{d}{dt}\varepsilon_M(X_M)\right\|$, respectively, where $\frac{d}{dt}\varepsilon_M(X_M) = \frac{\partial\varepsilon(X_M)}{\partial X_M}\dot{X}_M$. The inputs X_i can be upper-bounded as $\|X_i\| \leq \overline{z_i} + \overline{q}_{d,i}$ where, $\overline{z_M}, \overline{z_G} \triangleq \|z\|$, $\overline{z_C} \triangleq (\alpha+2)\|z\|$, $\overline{z_F} \triangleq (\alpha+1)\|z\|$, $\overline{q_{d,M}}, \overline{q_{d,G}} \triangleq \overline{q}_d$, $\overline{q_{d,C}} \triangleq \overline{q}_d + \overline{\dot{q}_d}$, and $\overline{q_{d,F}} \triangleq \overline{\dot{q}_d}$. The time derivative of the input X_M can be upper bounded as $\left\|\dot{X}_M\right\| \leq \overline{z}_{dt,M} + \overline{q}_{dt,M}$ where $\overline{z_{dt,M}} \triangleq (\alpha+1)\|z\|$ and $\overline{q_{dt,M}} \triangleq \overline{\dot{q}_d}$. Then, $\|N_i\| \leq 4\overline{\theta}_i^2\rho_{0,i}(\overline{z}_i + \overline{q}_{d,i}) + \overline{\varepsilon}_i$, and hence, from Assumption 2.3, $\left\|\frac{d}{dt}N_M\right\| \leq 2\overline{\theta}_M^2\rho_{0,M}(\overline{z}_M + \overline{q}_{d,M}) + 4\overline{\theta}_M^2\rho_T(\overline{z}_M + \overline{q}_{d,M} + \overline{z}_{dt,M} + \overline{q}_{dt,M}) + \overline{\varepsilon'_M}\left\|\dot{X}_M\right\|$, which can be further bounded as

$$\|N_i\| \leq 4\overline{\theta}_i^2\left(\rho_{1,i}(\overline{z_i}) + \rho_{2,i}(\overline{q_{d,i}})\right) + \overline{\varepsilon}_i \tag{2–46}$$

and

$$\begin{aligned}\left\|\frac{d}{dt}N_M\right\| \leq\ & 2\overline{\theta}_M^2\left(\rho_{1,M}(\overline{z}_M) + \rho_{2,M}(\overline{q_{d,M}})\right) + \overline{\varepsilon'_M}\left\|\dot{X}_M\right\| \\ & + 4\overline{\theta}_M^2\rho_{T1}(\overline{z_M} + \overline{z}_{dt,M}) + 4\overline{\theta}_M^2\rho_{T2}\left(\overline{q}_{d,M} + \overline{q}_{dt,M}\right)\end{aligned} \tag{2–47}$$

where $\rho_{1,i} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $\rho_{2,i} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, for all $i \in \mathcal{I}$, $\rho_{T1} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and $\rho_{T2} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ are non-decreasing positive functions. Let $\overline{\rho_{1,i}}(\|z\|) \triangleq 4\overline{\theta}_i^2\left(\rho_{1,i}(\overline{z_i}) - \rho_{1,i}(0)\right)$ for $i \in \mathcal{I}$ and $\overline{\rho_{T1}}(\|z\|) = 4\overline{\theta}_M\left(\rho_{T1}\left((\alpha_1+2)\|z\|\right) - \rho_{T1}(0)\right)$. Multiplying, $\overline{\rho_{1,i}}(\|z\|)$ and $\overline{\rho_{T1}}(\|z\|)$ by $\frac{\sqrt{c_{s,i}}}{\sqrt{c_{s,i}}}$ and $\frac{\sqrt{c_{sT}}}{\sqrt{c_{sT}}}$, respectively, where $c_{s,i}, c_{sT} \in \mathbb{R}_{>0}$ are user-defined constants, and applying the triangle-inequality, $\overline{\rho_{1,i}}$ and $\overline{\rho_{T1}}$ can be bounded as $\overline{\rho_{1,i}}(\|z\|) \leq \frac{\overline{\rho_{1,i}}^2(\|z\|)}{2c_{s,i}} + \frac{c_{s,i}}{2}$ and $\overline{\rho_{T1}}(\|z\|) \leq \frac{\overline{\rho_{T1}}^2(\|z\|)}{2c_{sT}} + \frac{c_{sT}}{2}$ where there exists ρ_i and ρ_{dt} such that $\frac{\overline{\rho_{1,i}}^2(\|z\|)}{2c_{s,i}} \leq$

$\rho_i(\|z\|)\,\|z\|$ for $i \in \mathcal{I}$ and $\frac{\overline{\rho_{T1}}^2(\|z\|)}{2c_{sT}} \leq \rho_{dt}(\|z\|)\,\|z\|$. Then, using (2–46), the developed bounds, and adding and subtracting $4\overline{\theta}_i^2\rho_{1,i}(0)$, then N_i can be bounded as

$$\|N_i\| \leq \rho_i(\|z\|)\,\|z\| + 4\overline{\theta}_i^2\left(\rho_{2,i}(\overline{q}_i) + \rho_{1,i}(0)\right) + \overline{\varepsilon_i} + \frac{1}{2}c_{s,i}, \tag{2–48}$$

for $i \in \mathcal{I}$. Similarly, using (2–46) and (2–47),

$$\begin{aligned}\left\|\frac{d}{dt}N_M\right\| \leq{}& \left(\rho(\|z\|) + \rho_{dt}(\|z\|)\right)\|z\| + 2\overline{\theta}_M^2\rho_{2,M}\left(\overline{q}_d\right) + 4\overline{\theta}_M^2\rho_{T2}(\overline{q_d} + \overline{\dot{q}_d}) \\ &+ \overline{\varepsilon'_M}\left\|\dot{X}_M\right\| + \frac{c_{sT}}{2} + 2\overline{\theta}_M^2\rho_{1,M}(0) + 4\overline{\theta}_M^2\rho_{T1}(0) + \frac{c_s}{2}.\end{aligned} \tag{2–49}$$

To facilitate the subsequent stability analysis, let the concatenated state vector $\zeta \in \mathbb{R}^{\psi}$ be defined as $\zeta \triangleq [e^\top, r^\top, \widetilde{E}, \sum_{i\in\mathcal{I}}\tilde{\theta}_i^\top]^\top$where $\psi \triangleq 2n + 1 + \sum_{i\in\mathcal{I}}\varkappa_i$. Since the approximation capabilities of DNNs stated in (2–28)-(2–31) hold only on the compact domains Ω_i, for $i \in \mathcal{I}$, the subsequent stability analysis requires ensuring $X_i(t) \in \Omega_i$ for all $t \in [t_0, \infty)$. This is achieved by demonstrating that ζ is constrained to a compact domain. To that end, consider the compact domain $\mathcal{D} \triangleq \left\{\sigma \in \mathbb{R}^{\psi} : \|\sigma\| < \chi\right\}$ in which ζ is supposed to lie. It follows that if $\|\zeta\| < \chi$, then $\|X_i\| < \overline{\chi}_i + \overline{q}_{d,i}$ for $i \in \mathcal{I}$ where $\overline{\chi_M}, \overline{\chi_G} \triangleq \chi$, $\overline{\chi_C} \triangleq (\alpha+2)\chi$, and $\overline{\chi_F} \triangleq (\alpha+1)\chi$. Therefore, if $\Omega_i \triangleq \left\{\sigma \in \mathbb{R}^{L_{in,i}} : \|\sigma\| < \overline{\chi}_i + \overline{q}_{d,i}\right\}$ for $i \in \mathcal{I}$, then $\zeta \in \mathcal{D}$ implies $X_i \in \Omega_i$ for all $i \in \mathcal{I}$. Let $\lambda_1 \triangleq \min\left\{\alpha, k_1, \min\left(\gamma_1\left(\xi^\top \otimes \xi^\top\right)\right)\right\}$, $\lambda_2 \in \mathbb{R}_{>0}$ be the desired convergence rate, $\rho_3 \triangleq \rho_{2,i}(\overline{q}_i) + \rho_{1,i}(0)$ for $i \in \mathcal{I}$, $P_1\left(\|z\|\right) = \sum_{i\in\mathcal{I}}\overline{Q_i}\rho_i(\|z\|) + \left\|\xi^\top \otimes \xi^\top\right\|\left(2\rho_V(\|z\|) + \rho_M(\|z\|) + \rho_{dt}(\|z\|)\right)$ where $\overline{Q_M} \triangleq \left\|(\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\right\|$, $\overline{Q_V} = \left\|(\dot{q}_d - \alpha e)^\top \otimes I_n\right\|$, and $\overline{Q_F}, \overline{Q_G} = 1$, and $P_1\left(\|z\|\right) \leq P\left(\|z\|\right)$ where P is a positive strictly increasing and invertible function. Let the set of stabilizing initial conditions $\mathcal{S} \subset \mathcal{D}$ be defined as

$$\mathcal{S} \triangleq \left\{\sigma \in \mathbb{R}^{\psi} : \|\sigma(t_0)\| < \sqrt{\frac{\beta_1}{\beta_2}}\chi\right\}, \tag{2–50}$$

where $\beta_1 \triangleq \min\left\{\frac{1}{2}, \frac{1}{2}m_1, \frac{1}{2}, \frac{1}{2}\min_{i\in\mathcal{I}}(\lambda_{\min}(\Gamma_i))\right\}$, $\beta_2 = \max\left\{\frac{1}{2}, \frac{1}{2}m_2, \frac{1}{2}, \frac{1}{2}\max_{i\in\mathcal{I}}(\lambda_{\max}(\Gamma_i))\right\}$[1] . The following analysis indicates that convergence starting from an arbitrary $\zeta(0)$ is ensured by selecting

$$\chi = P^{-1}\left(\lambda_1 - \lambda_2\right),$$

with a sufficiently large $\lambda_1 - \lambda_2$. The following theorem establishes the convergence of the tracking error and skew-symmetric prediction error using the developed SS-LbPINN adaptive controller, update laws, and prediction error formulation.

Theorem 2.2. *Consider the Filippov regularization of the dynamical system in (2–26), the controller given in (2–33), and the adaptation laws in (2–42)-(2–45). Provided Assumptions 2.2-2.3 hold,* $\zeta(t_0) \in \mathcal{S}$*, and the following gain conditions are satisfied,* $\lambda_1 > \lambda_2 + P\left(\sqrt{\frac{\beta_2}{\beta_1}\left\|z(t_0)\right\|^2}\right)$, $k_2 > 4\sum_{j=\{F,G\}}\overline{\theta}_j^2\rho_{3,j} + \overline{\varepsilon_j} + \frac{1}{2}c_{s,j}$, $k_3 > 4\overline{\theta}_V^2\rho_{3,V} + \overline{\varepsilon_V} + \frac{1}{2}c_{s,V}$, $k_4 > 4\overline{\theta}_M^2\rho_{3,M} + \overline{\varepsilon_M} + \frac{1}{2}c_{s,M}$, $\gamma_2 > 2\overline{\theta}_M^2\rho_{3,M} + 4\overline{\theta}_M^2\rho_{T2}(\overline{q_d} + \overline{\dot{q}_d}) + 4\overline{\theta}_M^2\rho_{T1}(0) + \frac{c_{sT}}{2} + \frac{c_s}{2}$, $\gamma_3 > 8\overline{\theta}_V^2\rho_{3,V} + \overline{\varepsilon_V} + \frac{1}{2}c_{s,V}$, *and* $\gamma_4 > \overline{\varepsilon'_M}$ *, the tracking and skew-symmetric prediction errors asymptotically converge to zero. Specifically,* $\|e(t)\| \to 0$, $\|r(t)\| \to 0$, *and* $\left\|\widetilde{E}(t)\right\| \to 0$ *as* $t \to \infty$.

Proof. Consider the candidate Lyapunov function $\mathcal{V}_L \in \mathbb{R}_{\geq 0}$ which is a Lipschitz continuous positive definite function defined as

$$\mathcal{V}_L(\zeta, t) \triangleq \frac{1}{2}e^\top e + \frac{1}{2}r^\top M r^\top + \frac{1}{2}\widetilde{E}^2 + \frac{1}{2}\sum_{i\in\mathcal{I}}\tilde{\theta}_i^\top\Gamma_i^{-1}\tilde{\theta}_i. \tag{2–51}$$

The candidate Lyapunov function in (2–51) satisfies the inequality

$$\beta_1\left\|\zeta\right\|^2 \leq \mathcal{V}_L(\zeta, t) \leq \beta_2\left\|\zeta\right\|^2. \tag{2–52}$$

Let $\dot{\zeta} = \mathcal{H}(\zeta, t)$, where $\mathcal{H}(\zeta, t) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^\psi$ denotes the right-hand side of the closed-loop error signals. Using Filippov's theory of differential inclusions [114–117], the

[1] Recall, $m_1$and m_2 denote known constants defined in Property 1.

existence of solutions can be established for $\dot{\zeta} \in \mathsf{K}\left[\mathcal{H}(\zeta,t)\right](\zeta)$, where $\mathsf{K}[\mathcal{H}(\zeta,t)] \triangleq \bigcap_{\delta>0} \bigcap_{\mu L=0} \overline{co}\mathcal{H}(B(\zeta,\delta)\setminus L,t)$, the intersection over all sets L of Lebesgue measure zeros is denoted $\bigcap_{\mu L=0}$, $\overline{co}$ denotes convex closure, and $B(\zeta,\delta)=\left\{w\in R^{\psi}|\,\|\zeta-w\|<\delta\right\}$. The time derivative of (2–51) exists along the trajectories of $\dot{\zeta}=\mathsf{K}[\mathcal{H}(\zeta,t)]$ almost everywhere (a.e.), i.e., for almost all $t\in[t_0,t_f]$, and $\dot{\mathcal{V}}_L \overset{\text{a.e.}}{\in} \dot{\tilde{\mathcal{V}}}_L$ where $\dot{\tilde{\mathcal{V}}}=\nabla\mathcal{V}_L^{\top}\mathsf{K}[\mathcal{H}(\zeta,t)]^{\top}$. Applying Property 2 on (2–35), using (2–42)-(2–45), and canceling cross terms yields

$$\begin{aligned}\dot{\tilde{\mathcal{V}}}_L(\zeta,t)\subset\, & e^{\top}(-\alpha e)+r^{\top}\Big(-k_1 r-\mathsf{K}\left[\mathrm{sgn}(r)\right]\Big(k_2+k_3\left\|(\dot{q}_d-\alpha e)^{\top}\otimes I_n\right\|\\ & +k_4\left\|(\ddot{q}_d-\alpha\dot{e})^{\top}\otimes I_n\right\|\Big)-N_G-N_F-\Big((\ddot{q}_d-\alpha\dot{e})^{\top}\otimes I_n\Big)N_M\\ & -\Big((\dot{q}_d-\alpha e)^{\top}\otimes I_n\Big)N_C\Big)+\widetilde{E}\dot{\widetilde{E}}-\tilde{\theta}_M^{\top}\frac{d}{dt}\left(\widehat{\Phi}'(X_M,\hat{\theta}_M)\right)^{\top}\widetilde{E}\\ & +2\tilde{\theta}_C^{\top}\Phi'^{\top}(X_C,\hat{\theta}_C)\widetilde{E}. \end{aligned} \tag{2–53}$$

Injecting (2–41) into (2–53) and canceling cross terms yields

$$\begin{aligned}\dot{\mathcal{V}}_L(\zeta,t)\overset{\text{a.e}}{\in}\, & e^{\top}(-\alpha e)+r^{\top}\Big(-k_1 r-\mathsf{K}\left[\mathrm{sgn}(r)\right]\Big(k_2+k_3\left\|(\dot{q}_d-\alpha e)^{\top}\otimes I_n\right\|\\ & +k_4\left\|(\ddot{q}_d-\alpha\dot{e})^{\top}\otimes I_n\right\|\Big)-N_G-N_F-\Big((\ddot{q}_d-\alpha\dot{e})^{\top}\otimes I_n\Big)N_M\\ & -\Big((\dot{q}_d-\alpha e)^{\top}\otimes I_n\Big)N_C\Big)+\widetilde{E}\left(\xi^{\top}\otimes\xi^{\top}\right)\left(\frac{d}{dt}(N_M)-2N_C-\gamma_1\widetilde{E}\right.\\ & \left.-\left(\gamma_2+\gamma_3+\gamma_4\left\|\dot{X}_M\right\|\right)\mathsf{K}\left[\mathrm{sgn}(\widetilde{E})\right]\right). \end{aligned} \tag{2–54}$$

Applying the bounds developed in (2–48) and (2–49), and provided the stated gain conditions are satisfied, (2–54) can be bounded as

$$\begin{aligned}\dot{\mathcal{V}}_L(\zeta,t) &\overset{\text{a.e.}}{\leq} -\left(\lambda_1-P\left(\|z\|\right)\right)\|z\|^2,\\ &\overset{\text{a.e.}}{\leq} -\lambda_2\|z\|^2, \end{aligned} \tag{2–55}$$

when $\zeta\in\mathcal{D}$. Therefore, when ζ is initialized such that $\zeta(t_0)\in\mathcal{S}$, then using (2–51) and (2–55), the extension of the LaSalle-Yoshizawa corollary in [109, Corollary 1] can be invoked to show that $\lambda_2\|z\|^2\to 0$ as $t\to\infty$, for all $\zeta(t_0)\in\mathcal{S}$. Based on the definition of z, it follows that $\|e(t)\|\to 0$, $\|r(t)\|\to 0$, and $\left\|\widetilde{E}(t)\right\|\to 0$ for all $\zeta(t_0)\in\mathcal{S}$.

Table 2-2. Comparison of Performance (RMS)

RMS Values	$\left\|\tilde{f}\right\|$	$\left\|M-\widehat{\Phi}_M\right\|$	$\left\|C-\widehat{\Phi}_C\right\|$	$\left\|F-\widehat{\Phi}_F\right\|$
Baseline Method [2]	3.781 [N·m]	1.271[kg·m^2]	6.469 $[\text{kg}\cdot\frac{\text{m}^2}{\text{s}}]$	6.187 [N · m]
Developed Method	3.030 [N·m]	1.209[kg·m^2]	5.060 $[\text{kg}\cdot\frac{\text{m}^2}{\text{s}}]$	6.032 [N · m]

The fact that $\mathcal{V}_L(\zeta,t)$ is nonincreasing implies $\|\zeta(t)\| \leq \sqrt{\frac{\mathcal{V}_L(\zeta,t)}{\beta_1}} \leq \sqrt{\frac{\mathcal{V}_L(\zeta,t_0)}{\beta_1}}$ and the fact that $\zeta(t_0) \in \mathcal{S}$ guarantees that $\zeta(t) \in \mathcal{D}$ for all $t \in [0,\infty)$ and thus $X_i \in \Omega_i$ for all $t \in [0,\infty)$ and $i \in \mathcal{I}$. Using (2–51) and (2–55) implies $e, r, \widetilde{E}, \tilde{\theta}_M, \tilde{\theta}_C, \tilde{\theta}_F, \tilde{\theta}_G \in \mathcal{L}_\infty$. Using the fact that $\widetilde{E} \in \mathcal{L}_\infty$, the definition in (2–36), and the fact that $E = 0$, implies $\widehat{E} \in \mathcal{L}_\infty$. Due to the use of the projection operator, $\hat{\theta}_i \in \mathcal{L}_\infty$ for all $i \in \mathcal{I}$. The fact that $q_d, \dot{q}_d, e, r \in \mathcal{L}_\infty$ implies $q, \dot{q} \in \mathcal{L}_\infty$. Using (2–33), the fact that $e, r, q_d, \dot{q}_d, \ddot{q}_d,$ and $\hat{\theta}_i \in \mathcal{L}_\infty$ implies $X_i \in \mathcal{L}_\infty$ and thus $\Phi(X_i, \hat{\theta}_i)$ is bounded for $i \in \mathcal{I}$ and therefore τ is bounded. □

2.2.5 Simulations

Simulation results are provided to demonstrate the performance of the developed SS-LbPINN using a two-link planar revolute robot modeled in [118, Eqn. (24)-(26)]. Simulations were run for 50 s with the desired trajectory $q_d \triangleq (1-\exp(-0.1))\begin{bmatrix}\frac{3\pi}{8}\sin\left(\frac{\pi}{2}t\right)\\ \frac{3\pi}{8}\sin\left(\frac{\pi}{2}t\right)\end{bmatrix} \in \mathbb{R}^2$ [rad], initialized at $q(0) = [0.4, -0.3]^\top$ [rad] and $\dot{q}(0) = [0,0]^\top$ [rad/s]. To examine the effect of noise, white Gaussian noise with signal-to-noise-ratio (SNR) of 60 dB is added to the position and velocity measurements. To assess the impact of the skew-symmetric prediction error, a comparison was done between the developed method and the architecture in [2] which does not include the skew-symmetric prediction error. Both architectures were composed of 3 DNNs with 4 layers, 7 neurons in each hidden layer, and tanh activation functions for the $M, C,$ and F matrices. The gravitational effects represented by G were not considered due to the planar configuration of the robot. The simulations were performed using the same initial DNN weight estimates which were randomly initialized from a uniform distribution $U(-1,1)$. For a fair comparison between the baseline [2] and developed architectures, the gains shared between the architectures were

selected to be the same; because of this, it is expected that both architectures have similar tracking performance. The control gains are selected as $\alpha = 3.8$, $k_1 = 15.1$, $k_2 = 0.5$, $k_3 = 0.5$, $k_4 = 0.3$ with weight adaptation gains $\Gamma_M = 1.1 \cdot I_{\varkappa_M}$, $\Gamma_C = 11.5 \cdot I_{\varkappa_V}$, and $\Gamma_F = 9 \cdot I_{\varkappa_F}$. The gains associated with the skew-symmetric prediction error were selected as $\gamma_1 = 186.1$, $\gamma_2 = 1.2$, $\gamma_3 = 6.5$, $\gamma_4 = 2.7$, and $\xi = 0.4$. The developed method achieved an RMS value for $\left\| \widetilde{E} \right\|$ of 0.0048 [kg $\cdot \frac{\text{m}^2}{\text{s}}$]. Both architectures achieve similar tracking performance, with values of 0.0195 rad under 7.345 Nm of control effort, and 0.0194 rad under 7.350 Nm of control effort for the developed and baseline methods, respectively. However the SS-LbPINN was motivated by improving the accuracy of the individual estimates. As shown in Figure 2-2 and Table 2-2 the incorporation of a skew-symmetric prediction error resulted in a 4.84%, 21.78%, and 2.52% improvement for the individual matrix approximations for M, C, and F, respectively. The improvement in the individual matrix estimations is reflected in the overall function approximation error (Figure 2-3) represented as $\tilde{f} \triangleq \left(M - \widehat{\Phi}_M\right) \ddot{q} + \left(C - \widehat{\Phi}_C\right) \dot{q} + \left(F - \widehat{\Phi}_F\right)$ which exhibited a 19.87% improvement in the developed architecture compared to the baseline while requiring similar control effort.

2.2.6 Conclusions

A physics-informed controller is developed for general uncertain Euler-Lagrange systems. Leveraging the dynamic structure, the controller is composed of individual estimates of the unknown terms. Then, leveraging the individual estimates, an adaptive update law is developed which penalizes the skew-symmetric terms to exploit the known skew-symmetry of the system. Unlike traditional NN approaches, the developed SS-LbPINN control design and update laws are informed by the known physics of the system. Stability-driven weight adaption laws are developed for the SS-LbPINN weights in real-time, eliminating the need for offline pre-training. A Lyapunov-based stability analysis is performed and guarantees asymptotic convergence of the tracking errors and the skew-symmetric prediction errors. Simulations validate the SS-LbPINN, showing

a 19.87% improvement in function approximation and 4.84%, 21.78%, and 2.52% improvement in M, C, and F approximations, respectively, over [2], while maintaining similar tracking error and control effort.

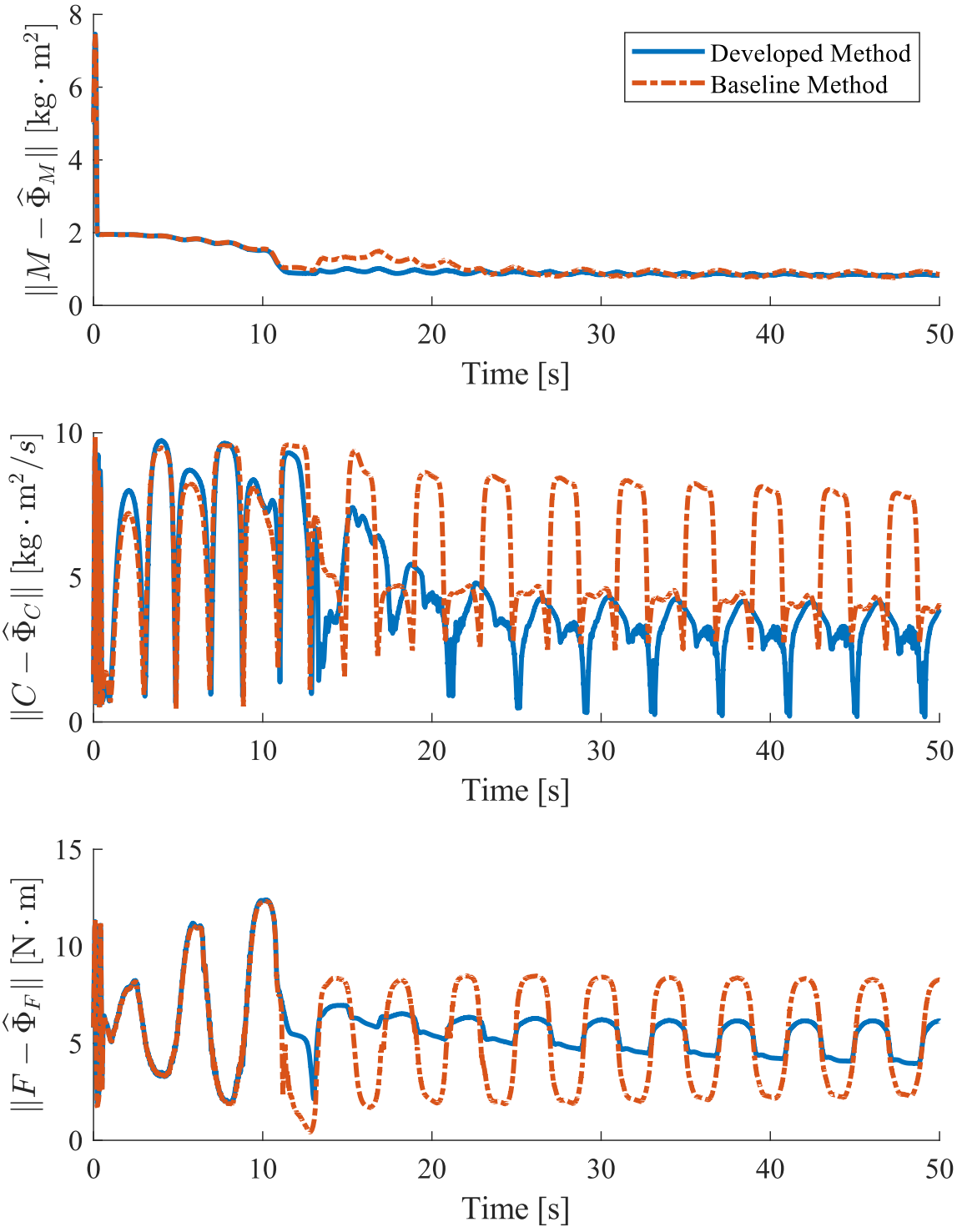

Figure 2-2. Individual function approximation errors over time for the developed method and the baseline method in [2].

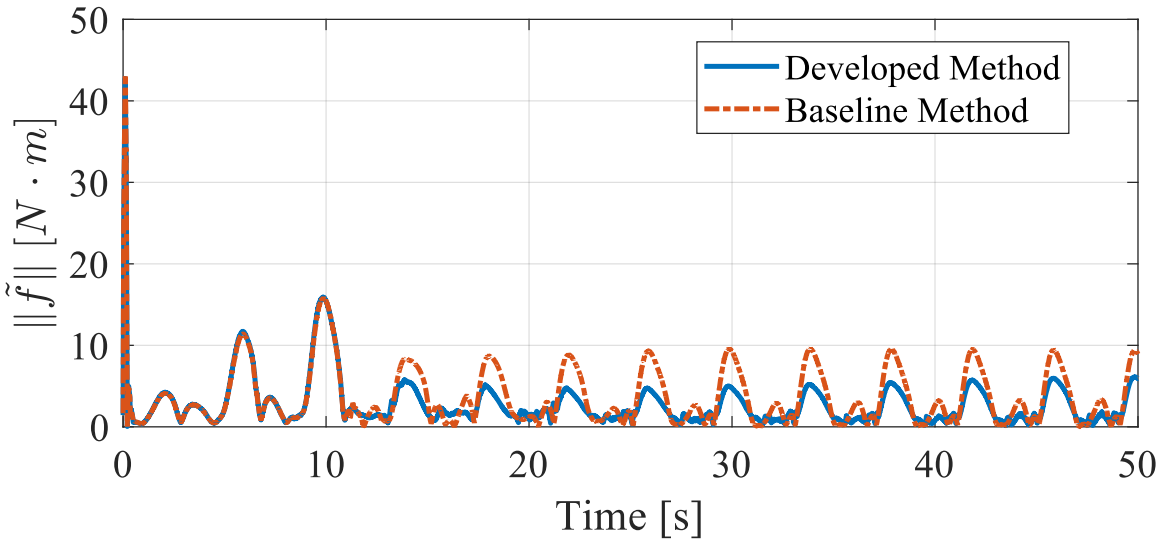

Figure 2-3. Overall function approximation error over time for the developed method and the baseline method in [2].

CHAPTER 3
ON THE IDENTIFIABILITY AND CONVERGENCE OF REAL-TIME DEEP NEURAL NETWORK REGRESSION

While many DNN-based control methods such as those presented in Chapter 2 focus primarily on tracking error convergence and can empirically demonstrate high function approximation capabilities. The challenge of uniquely identifying system dynamics is often neglected. This chapter addresses the identifiability of nonlinear-in-parameter (NIP) structures such as DNNs by deriving formal requirements for parameter convergence and demonstrates that the verifiable finite excitation (FE) condition is equivalent to parameter identifiability.

Leveraging this insight, a concurrent learning framework is developed for continuous nonlinear regression using all-layer adaptation. The developed adaptation law utilizes a history stack of recorded data to drive parameter estimates towards their desired values. A Lyapunov-based stability analysis is conducted to demonstrate that the weight estimation error converges to a neighborhood of the origin. Finally, numerical analysis validates the feasibility of the FE condition and demonstrates how these identifiability requirements can serve as a metric for informed architectural design.

3.1 Continuous-Time Nonlinear Regression

Various applications spanning across the domains of system identification, fault detection, financial time-series prediction, neuroscience, environmental predictions, medical diagnostics etc. require online identification of the underlying processes. Many such processes can be represented as a continuous-time nonlinear regression. Therefore, consider the nonlinear regression equation (NRE),

$$y(t) \quad = \quad f(X(t)) + \delta(t) \tag{3–1}$$

for all $t \in \mathbb{R}_{\geq 0}$, where $X(t) \in \Omega$ denotes a known input to the regression and lies in the compact set $\Omega \subset \mathbb{R}^m$, $y(t) \in \mathbb{R}^n$ denotes the output of the regression with available measurements, $f : \Omega \to \mathbb{R}^n$ denotes an unknown continuously differentiable function,

and $\delta(t) \in \mathbb{R}^n$ denotes a bounded disturbance with known bound $\bar{\delta} \in \mathbb{R}_{>0}$ such that $\|\delta(t)\| \leq \bar{\delta}$ for all $t \in \mathbb{R}_{\geq 0}$. Time-dependency notation is omitted in the subsequent development for brevity. Given the unknown structure of f, DNNs are employed to learn the NRE, leveraging their universal function approximation properties over compact sets [107]. Therefore, consider a DNN $\Phi : \mathbb{R}^m \times \mathbb{R}^\varkappa \to \mathbb{R}^n$, where a DNN-based approximation of (3–1) is given by

$$\hat{f} = \Phi\left(X, \hat{\theta}\right), \tag{3–2}$$

where $\hat{\theta} \in \mathbb{R}^\varkappa$ denotes the DNN parameter estimates that are designed based on the subsequent Lyapunov-based stability analysis. The objective is to find the best estimates of $\hat{\theta}$ such that $X \mapsto \Phi\left(X, \hat{\theta}\right)$ approximates $X \mapsto f(X)$ for any $X \in \Omega$. To quantify the DNN-based nonlinear regression objective, consider the loss function $\mathcal{L} : \mathbb{R}^\varkappa \to \mathbb{R}_{\geq 0}$ defined as

$$\mathcal{L}(\theta) \triangleq \int_\Omega \left(\|f(X) - \Phi(X, \theta)\|^2 + \sigma \|\theta\|^2\right) d\mu(X), \tag{3–3}$$

where μ denotes the Lebesgue measure, $\sigma \in \mathbb{R}_{>0}$ denotes a regularizing constant, and the term $\sigma \|\theta\|^2$ represents L_2 regularization (also popularly known as ridge regression) [119, Sec. 7.1.1]. Note that this loss function is defined independently of the time-dependent signal $t \mapsto X(t)$ because the objective is to ensure the approximation holds for any $X \in \Omega$ and not just the set of points traversed by $t \mapsto X(t)$. Additionally, consider a user-selected compact convex parameter search space Θ satisfying $0_\varkappa \in \Theta$ and having a smooth boundary and the bound $\bar{\theta} \triangleq \max_{\theta \in \Theta} \|\theta\| \in \mathbb{R}_{>0}$. The objective is to identify the vector of ideal DNN parameters $\theta^* \in \Theta$ defined as

$$\theta^* \triangleq \arg\min_{\theta \in \Theta} \mathcal{L}(\theta). \tag{3–4}$$

Furthermore, note that local minima for the $\|f(X) - \Phi(X, \theta)\|^2$ term are ubiquitous for various applications of DNNs. The important question is whether there are local minima of higher cost than the global minima. As noted in [119, Sec. 8.2.2.], this question remains open for general DNN architectures. However, for some DNN architectures, it has been established that every local minimum is a global minimum. For example, [120] established this property for deep residual neural networks (ResNets) with arbitrary non-linear activation functions. Results such as [121] and [122] have concluded the same for DNNs with linear activation functions, and [123] establishes the same for single-hidden-layer deep convolutional neural networks with ReLU activation functions. Furthermore, from a practical standpoint, it is acceptable to find a point in the parameter space that has low but not necessarily minimal cost [119, Sec. 8.2.2.].

Although using a bounded search space can restrict the optimality of the identified parameters to be local instead of global, it allows the subsequent development to be analyzed from a convex optimization perspective, which otherwise would be non-convex due to the nested NIP structure of DNNs. Specifically, due to the strict convexity of the regularizing term $\sigma \|\theta\|^2$ in (3–3), there exists $\sigma \in \mathbb{R}_{>0}$ which ensures $\mathcal{L}(\theta)$ is convex for all $\theta \in \Theta$. Additionally, the regularizing term has other advantages such as mitigation of overfitting [119, Sec. 7.1.1]. However, selecting very high values of σ can obscure the contribution of the $\|f(X) - \Phi(X, \theta)\|^2$ term to the loss function while also causing underfitting [119, Sec. 7.1.1]. While the regularization term in (3–3) facilitates a convex optimization perspective within Θ, it does not resolve the underlying identifiability issues inherent to the DNN architecture. DNNs are generally globally non-identifiable due to structural identifiability issues resulting from redundancies inherent in the network architecture. To illustrate this problem, consider the following example.

Example 3.1. Consider a scalar input-output NN with one hidden linear and one hidden neuron, $\Phi(X, \theta) = w_1 w_0 X$, where w_1 and w_0 are scalar NN weights. Then, for any $\alpha \in \mathbb{R}_{>0}$, defining the new weights $v_1 = \alpha w_1$ and $v_0 = \frac{w_0}{\alpha}$ yields an equivalent

parameterization $\Phi(X, \theta) = v_1 v_0 X$ with different weights. This scaling invariance renders the linear architecture non-identifiable. In contrast, consider a network with a bias term b and nonlinear activation $\phi(\cdot)$, modeled as $\Phi(X, \theta) = w_2 \phi(w_1 X + b)$. The presence of the bias and nonlinearity breaks the scaling symmetry as $\phi(\alpha w_1 X + \frac{b}{\alpha}) \neq \frac{1}{\alpha}\phi(w_1 X + b)$ for general nonlinearities. Consequently, the Jacobian becomes full rank on sufficiently rich trajectories, rendering the architecture locally identifiable.

Acknowledging that global identifiability is unattainable for DNNs due to remaining symmetries (e.g., node permutations), we consider local identifiability and seek to identify a single parameter vector from the set of equivalent parameters that acts as a unique local representative of θ^*. A model is locally identifiable at θ^* if there exists a neighborhood $U(\theta^*)$ such that no other $\theta \in U$ produces the same output mapping.

3.2 Identifiability Conditions

To define the conditions under which the problem of identifying θ^* is well-posed, we provide a definition of least squares identifiability, which is a modified version of [124, Definition 3] tailored to the regression problem in (3–1).

Definition 3.1. The parameter vector θ^* is identifiable over the set Θ if and only if it is a unique minimizer of $\mathcal{L}$ over Θ. If θ^* is only an isolated local minimizer (i.e., if there exists an arbitrarily small neighborhood of θ^* where it is a unique minimizer), then θ^* is termed locally identifiable. If θ^* is identifiable over the entire Euclidean space $\mathbb{R}^{\varkappa}$, then it is globally identifiable.

For general deep learning tasks, achieving global identifiability can be intractable due to the aforementioned reasons for why we consider a bounded search space; therefore, we examine conditions such that θ^* is locally identifiable or identifiable over Θ. To obtain the conditions under which θ^* is identifiable over Θ, strict convexity is imposed on $\mathcal{L}$. Because Θ is a convex set, imposing strict convexity on $\mathcal{L}$ ensures unique solutions to (3–4). A sufficient condition for ensuring strict convexity is ensuring the Hessian $\frac{\partial^2 \mathcal{L}}{\partial \theta^2}$ is positive definite, i.e., $\frac{\partial^2 \mathcal{L}}{\partial \theta^2} \succ 0$ for all $\theta \in \Theta$ [125, Sec. 3.1.4]. Imposing

positive-definiteness on $\frac{\partial^2 \mathcal{L}}{\partial \theta^2}$ yields the condition

$$\int_{\Omega} \left(\left(\Phi'^{\top}(X,\theta)\,\Phi'(X,\theta) + \sigma I_{\varkappa} \right) - \sum_{i=1}^{n} \left(\frac{\partial^2 \Phi_i(X,\theta)}{\partial \theta^2} \left(f_i(X) - \Phi_i(X,\theta) \right) \right) \right) d\mu(X) \succ 0, \tag{3–5}$$

where i denotes the index of the individual output elements from vectors f and Φ for all $\theta \in \Theta$. Although the inequality in (3–5) offers an identifiability condition, this condition is challenging to verify because it involves a partial integro-differential inequality consisting of the unknown function approximation error $f(X) - \Phi(X,\theta)$ and the second-derivative of the DNN $\frac{\partial^2 \Phi(X,\theta)}{\partial \theta^2}$ which is often computationally intractable. However, stricter conditions can be imposed to obtain tractable sufficient conditions for identifiability. To derive these stricter conditions, notice the term $\frac{\partial^2 \Phi(X,\theta)}{\partial \theta^2} \left(f(X) - \Phi(X,\theta) \right)$ is bounded for all $X \in \Omega$ and $\theta \in \Theta$; this fact can be established by noticing the terms $\frac{\partial^2 \Phi(X,\theta)}{\partial \theta^2}$ and $f(X) - \Phi(X,\theta)$ are bounded for all $X \in \Omega$ and $\theta \in \Theta$ because Φ is $\mathcal{C}^2$ and f is continuous. Therefore, let $\bar{\sigma} \in \mathbb{R}_{>0}$ denote the bound $\bar{\sigma} \triangleq \sup_{X \in \Omega, \theta \in \Theta} \left\| \sum_{i=1}^{n} \left(\frac{\partial^2 \Phi_i(X,\theta)}{\partial \theta^2} \left(f_i(X) - \Phi_i(X,\theta) \right) \right) \right\|$. Then, a stricter condition for identifiability is obtained as

$$\int_{\Omega} \left(\Phi'^{\top}(X,\theta)\,\Phi'(X,\theta) + \sigma I_{\varkappa} - \bar{\sigma} I_{\varkappa} \right) d\mu(X) \succ 0. \tag{3–6}$$

Remark 3.1. Selecting a large regularizing constant σ can trivially ensure the condition in (3–6). However, doing so is undesirable as it obscures the DNN's contribution to the loss function in (3–3). Instead, it is desirable that the term $\int_{\Omega} \Phi'^{\top}(X,\theta)\,\Phi'(X,\theta)\,d\mu(X)$ contributes to achieving the identifiability condition in (3–6), which yields the condition $\int_{\Omega} \Phi'^{\top}(X,\theta)\,\Phi'(X,\theta)\,d\mu(X) \succ \bar{\sigma} I_{\varkappa} \mu(\Omega)$. Note that this condition, although still computationally intensive, is relatively easier to verify using a discrete approximation of the integral as compared to (3–5). Alternatively, a more conservative approach can be taken where σ is selected to be equal to $\bar{\sigma}$ while requiring only positive-definiteness to be verified for $\int_{\Omega} \Phi'^{\top}(X,\theta)\,\Phi'(X,\theta)\,d\mu(X)$.

Remark 3.2. The sufficient condition in (3–6) highlights the intrinsic relationship between NN architecture and the required regularization. Architectures with high structural redundancy (e.g., deep linear networks or symmetric layers) often possess a Hessian spectrum with eigenvalues near zero. In such cases, a larger magnitude of σ is mathematically necessary to enforce the positive definiteness required for unique parameterization. Conversely, architectures designed with linear independence among hidden units are structurally identifiable on sufficiently rich trajectories and allow for a significantly smaller regularization bound $\overline{\sigma}$.

The following lemma provides equivalent conditions for verifying if $\int_{\Omega} \Phi'^{\top}(X,\theta)\,\Phi'(X,\theta)\,d\mu(X)$ is positive definite to check for identifiability.

Lemma 3.1. *Let $\sigma = \bar{\sigma}$. Then θ^* is identifiable over Θ if any of the following equivalent conditions are satisfied,*

i) $\int_{\Omega} \Phi'^{\top}(X,\theta)\,\Phi'(X,\theta)\,d\mu(X) \succ 0$ *for all* $\theta \in \Theta$,

ii) there exists $X_1, \ldots, X_{\varkappa} \in \Omega$ *such that* $\sum_{i=1}^{\varkappa} \Phi'^{\top}(X_i,\theta)\,\Phi'(X_i,\theta) \succ 0$ *for all* $\theta \in \Theta$,

iii) there exists $X_1, \ldots, X_{\varkappa} \in \Omega$ *such that* $rank\left(\left[\Phi'^{\top}(X_1,\theta), \ldots, \Phi'^{\top}(X_p,\theta)\right]\right) = \varkappa$ *for all* $\theta \in \Theta$.

Proof. Condition i) follows from (3–6). By the definition of positive-definiteness of a matrix, Condition i) is equivalent to stating $\int_{\Omega} v^{\top}\Phi'^{\top}(X,\theta)\,\Phi'(X,\theta)\,v d\mu(X) = \int_{\Omega} \|\Phi'(X,\theta)\,v\|^2 d\mu(X) > 0$ for all $v \neq 0_{\varkappa}$. Because the integrand is nonnegative, the set $E(v) \triangleq \{X \in \Omega : \|\Phi'(X,\theta)v\|^2 > 0\}$ has positive measure, and hence, is nonempty for all $v \neq 0_{\varkappa}$. The subsequent analysis will establish i) implies ii) using an inductive kernel shrinking argument. To this end, let $K_0 \triangleq \mathbb{R}^{\varkappa}$. For each integer $j \geq 1$, provided $K_{j-1} \neq \{0\}$, define the set K_j, and points X_j and v_j using the recursive algorithm

$$\begin{aligned} v_j &\in K_{j-1} \setminus \{0\} \\ X_j &\in E(v_j) \end{aligned}$$

$$K_j \triangleq K_{j-1} \cap \ker\left(\Phi'\left(X_j, \theta\right)\right).$$

The algorithm selects X_j such that $v_j \notin \ker(\Phi'\left(X_j, \theta\right))$. Hence, by the definition of K_j, it follows that $v_j \notin K_j$, implying K_j is a strict subspace of K_{j-1}. It is shown by induction that whenever K_j is defined, $\dim\left(K_j\right) \leq \varkappa - j$. For the base case $j = 0$, the hypothesis holds because $\dim\left(K_0\right) = \varkappa$. For the inductive step, assume the hypothesis holds for $j-1$ and that $K_{j-1} \neq \{0_\varkappa\}$, so that K_j is defined. Since K_j is a strict subspace of K_{j-1}, it follows that $\dim(K_j) \leq \dim(K_{j-1}) - 1 \leq (\varkappa - (j-1)) - 1 = \varkappa - j$. This completes the induction. Therefore, the construction terminates by step $\varkappa$ (since $\dim(K_\varkappa) \leq 0$ implies $K_\varkappa = \{0\}$). As a result, $\bigcap_{i=1}^{\varkappa} \ker\left(\Phi'\left(X_i, \theta\right)\right) = \{0_\varkappa\}$. Hence, for any $v \neq 0_\varkappa$, it holds that $v \notin \bigcap_{i=1}^{\varkappa} \ker\left(\Phi'\left(X_i, \theta\right)\right)$, so there exists i such that $\Phi'\left(X_i, \theta\right) v \neq 0$. Hence, $v^\top \left(\sum_{i=1}^{\varkappa} \Phi'^\top\left(X_i, \theta\right) \Phi'\left(X_i, \theta\right)\right) v > 0$. Because this result holds for all $v \neq 0_\varkappa$, it follows that $\sum_{i=1}^{\varkappa} \Phi'^\top\left(X_i, \theta\right) \Phi'\left(X_i, \theta\right) \succ 0$, establishing ii).

To establish that conditions i) and ii) are equivalent, we now establish that condition i) is also necessary to satisfy condition ii). We establish this by the method of contradiction. Thus, assume for the sake of contradiction that condition i) is violated and condition ii) is satisfied. Due to the violation of condition i), $\int_\Omega \Phi'^\top\left(X, \theta\right) \Phi'\left(X, \theta\right) d\mu\left(X\right)$ is not positive-definite but only positive semi-definite. In this case, there exists $v \neq 0_\varkappa$ such that $\int_\Omega \left\|\Phi'\left(X, \theta\right) v\right\|^2 d\mu\left(X\right) = 0$. Furthermore, notice that $\int_\Omega \left\|\Phi'\left(X, \theta\right) v\right\|^2 d\mu\left(X\right)$ can be lower-bounded as $0 \leq \sum_{i=1}^{\varkappa} \left(\inf_{X \in \Omega_i} \left\|\Phi'\left(X, \theta\right) v\right\|^2\right) \mu\left(\Omega_i\right) \leq \int_\Omega \left\|\Phi'\left(X, \theta\right) v\right\|^2 d\mu\left(X\right)$. Therefore, if $\int_\Omega \left\|\Phi'\left(X, \theta\right) v\right\|^2 d\mu\left(X\right) = 0$, then $\sum_{i=1}^{\varkappa} \left(\inf_{X \in \Omega_i} \left\|\Phi'\left(X, \theta\right) v\right\|^2\right) \mu\left(\Omega_i\right) = 0$, which is only possible when $\inf_{X \in \Omega_i} \left\|\Phi'\left(X, \theta\right) v\right\|^2 = 0$ for all $i \in \{1, \ldots, \varkappa\}$. Since the partitions are arbitrary, this condition must apply for all possible partitions of Ω, i.e., there does not exist a method to partition Ω_i such that $\inf_{X \in \Omega_i} \left\|\Phi'\left(X, \theta\right) v\right\|^2 > 0$ for any $i \in \{1, \ldots, \varkappa\}$. Since condition ii) is assumed to be satisfied, there exists $X_1, \ldots, X_\varkappa \in \Omega$ such that $\sum_{i=1}^{\varkappa} \left\|\Phi'\left(X_i, \theta\right) v\right\|^2 > 0$, implying $\left\|\Phi'\left(X_i, \theta\right) v\right\|^2 > 0$ for at least one $i \in \{1, \ldots, \varkappa\}$. For such an i, due to the continuity of $\left\|\Phi'\left(X_i, \theta\right) v\right\|^2$, there exists a

neighborhood of X_i of radius $\epsilon \in \mathbb{R}_{>0}$ given by $\mathcal{N}_i \triangleq \{X \in \Omega : \|X - X_i\| \leq \epsilon\}$ such that $\|\Phi'(X, \theta) v\|^2 > 0$ for all $X \in \mathcal{N}_i$. Additionally, note that $\mu(\mathcal{N}_i) > 0$. Then partitioning Ω such that $\Omega_i = \mathcal{N}_i$ yields $\inf_{X \in \Omega_i} \|\Phi'(X, \theta) v\|^2 > 0$, thus leading to a contradiction with the implications of condition i). Therefore, one cannot violate condition i) and satisfy condition ii) simultaneously. Hence, condition i) is necessary and sufficient for condition ii).

Now we establish the equivalence between conditions ii) and iii). Notice that condition ii) holds if and only if $\operatorname{rank}\left(\sum_{i=1}^{\varkappa} \Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta)\right) = \varkappa$, which is equivalent to the statement $\ker\left(\sum_{i=1}^{\varkappa} \Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta)\right) = \{0_\varkappa\}$ due to the rank-nullity theorem [126, Thm. 2.3]. Additionally, due to the property that $\ker(A + B) \supseteq \ker(A) \cap \ker(B)$ for any given matrices A and B, the condition $\ker\left(\sum_{i=1}^{\varkappa} \Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta)\right) = \{0_\varkappa\}$ holds if and only if $\bigcap_{i=1}^{\varkappa} \ker\left(\Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta)\right) = \{0_\varkappa\}$. Furthermore, note that $\ker\left(\Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta)\right) = \ker(\Phi'(X_i, \theta))$ which is established as follows. For any vector $v \in \mathbb{R}^{\varkappa}$, if $\Phi'(X_i, \theta) v = 0_n$, then multiplying by $\Phi'^{\top}(X_i, \theta)$ on both sides yields $\Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta) v = 0_\varkappa$. Additionally, if $\Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta) v = 0_\varkappa$, then multiplying by $v^{\top}$ on both sides yields $v^{\top} \Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta) v = 0$, equivalent to $\|\Phi'(X_i, \theta) v\| = 0$ which holds if and only if $\Phi'(X_i, \theta) v = 0_n$. Therefore, $\Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta) v = 0_\varkappa$ if and only if $\Phi'(X_i, \theta) v = 0_\varkappa$, thus implying $\ker\left(\Phi'^{\top}(X_i, \theta) \Phi'(X_i, \theta)\right) = \ker(\Phi'(X_i, \theta))$. Therefore, condition ii) is equivalent to the condition $\bigcap_{i=1}^{\varkappa} \ker(\Phi'(X_i, \theta)) = \{0_\varkappa\}$, which holds if and only if there exists no vector $v \neq 0_\varkappa$ such that $\Phi'(X_i, \theta) v = 0_n$ for all $i \in \{1, \ldots, \varkappa\}$. This condition can hold if and only if the matrix $\left[\Phi'^{\top}(X_1, \theta), \ldots, \Phi'^{\top}(X_\varkappa, \theta)\right]^{\top}$ has $\varkappa$ linearly independent columns, which is equivalent to the statement in condition iii) that $\operatorname{rank}\left(\left[\Phi'^{\top}(X_1, \theta), \ldots, \Phi'^{\top}(X_\varkappa, \theta)\right]\right) = \varkappa$. Therefore, conditions ii) and iii) are equivalent, thus establishing the equivalence between conditions i)-iii). □

Remark 3.3. To determine local identifiability (i.e., identifiability in an arbitrarily small neighborhood of θ^* as defined in Definition 3.1) instead of identifiability over Θ, the conditions i)-iii) stated in Lemma 3.1 can be relaxed to hold

only at θ^* instead of holding for all $\theta \in \Theta$. Furthermore, $\bar{\sigma}$ can be redefined as $\bar{\sigma} = \sup_{X\in\Omega} \left\| \sum_{i=1}^{n} \frac{\partial^2 \Phi_i(X,\theta^*)}{\partial \theta^2} \left(f_i(X) - \Phi_i(X,\theta^*) \right) \right\|$ to check for local identifiability. This redefinition of $\bar{\sigma}$ is expected to yield a significantly smaller value than its previous definition because it does not involve the supremum to be computed over the entire Θ but only at θ^*.

Remark 3.4. Notice that the universal function approximation theorem for DNNs was not invoked in the definition of θ^* or in the derivation of the identifiability conditions. The universal function approximation theorem [107, Theorem 3.1] states that the function space of DNNs is dense in the space of continuous functions $\mathcal{C}(\Omega)$. As a result, for any prescribed $\varepsilon > 0$, there exists a DNN Φ and a corresponding parameter θ such that $\sup_{X\in\Omega} \|f(X) - \Phi(X,\theta)\| < \varepsilon$, and therefore $\int_\Omega \|f(X) - \Phi(X,\theta)\|^2 d\mu(X) < \varepsilon^2 \mu(\Omega)$. However, it is not known how to obtain a bound $\bar{\theta}$ on such parameter θ for an arbitrary ε, which causes difficulties in constructing the bounded search space Θ. Therefore, we allow the user-defined search space Θ to be arbitrarily selected in the above analysis, at the loss of guarantees on the approximation accuracy. Although the constant ε that bounds $\sup_{X\in\Omega} \|f(X) - \Phi(X,\theta^*)\|$ might no longer be arbitrary in this case, it would still be finite due to the continuity of f and Φ, where minimizing (3–3) would yield the best regularized approximation of f. Therefore, the unknown nonlinear function can be modeled as

$$f(X) = \Phi(X,\theta^*) + \varepsilon(X), \tag{3–7}$$

where $\varepsilon(X) \triangleq f(X) - \Phi(X,\theta^*)$ is bounded by $\bar{\varepsilon}$ as $\|\varepsilon(X)\| \leq \bar{\varepsilon}$ for all $X \in \Omega$.

Recent work [100] provided identifiability conditions for linear regression equations (LREs) with a LIP model, and they showed that it is equivalent to FE or interval excitation (IE) conditions. The IE/FE condition is known to be strictly weaker than the persistence of excitation (PE) condition. Therefore, adaptive methods such as CL (and other methods discussed in the Introduction) that achieve parameter identification under only the IE/FE conditions can be interpreted to leverage only the identifiability of the

parameters. In this context, the conditions in Lemma 3.1 can be viewed as a generalization of the identifiability conditions in [100] to NREs where there is an NIP model. This observation raises the question whether an adaptive estimation technique can be developed to identify θ^* under the sufficient identifiability conditions stated in Lemma 3.1, as opposed to the more stringent PE condition. This chapter answers this question with an affirmative by developing a CL-based adaptive estimator for θ^*.

Before providing the CL-based estimator development, we provide a brief overview of the gradient-based estimators for DNNs for better context. For the loss function in (3–3), the corresponding loss density is given by $\mathfrak{L}\left(\hat{\theta}\right) = \left\| f\left(X\right) - \Phi\left(X, \hat{\theta}\right)\right\|^2 + \sigma \left\|\hat{\theta}\right\|^2$. Then the gradient-based estimator is derived by formulating the negative gradient flow of $\mathfrak{L}\left(\hat{\theta}\right)$ given by

$$\dot{\hat{\theta}} = -\Gamma \nabla_{\hat{\theta}} \mathfrak{L}\left(\hat{\theta}\right) = -\Gamma\sigma\hat{\theta} - \Gamma \Phi'^{\top}\left(X, \hat{\theta}\right)\tilde{y}, \tag{3–8}$$

where $\Gamma \in \mathbb{R}^{\varkappa \times \varkappa}$ denotes the adaptation gain, and

$$\tilde{y} \triangleq y\left(t\right) - \Phi\left(X, \hat{\theta}\right) \tag{3–9}$$

denotes the regression error. Using a similar approach as in [22] and [95], $\hat{\theta}$ converges to a neighborhood of θ^* provided the PE condition is satisfied. The PE condition requires that there exist some $T \in \mathbb{R}_{>0}$ such that $\int_{\underline{t}}^{\underline{t}+T} \Phi'^{\top}\left(X(\tau), \hat{\theta}(\tau)\right)\Phi'\left(X(\tau), \hat{\theta}(\tau)\right) d\tau \succ 0$ for all $\underline{t} \in \mathbb{R}_{\geq 0}$, which is restrictive because the positive definiteness has to hold on a moving time window for all time and cannot be verified online. In contrast, Lemma 3.1 suggests that θ^* is identifiable if $t \to X(t)$ traverses through points $X_1, \ldots, X_\varkappa \in \Omega$ such that conditions ii) or iii) of Lemma 3.1 are satisfied. Specifically, if $t \to X(t)$ is shaped to traverse through such points $X_1, \ldots, X_\varkappa \in \Omega$ in a finite-time interval, it must be possible to develop an adaptive estimator to identify θ^* without requiring PE. Based on this insight, we provide CL-based adaptive update laws in the following section.

3.3 Concurrent Learning Algorithm for Continuous-Time Regression Problems

Consider the NRE in (3–1). Using (1–2), Remark 3.4, and (3–7), the regression error in (3–9) can be represented as

$$\tilde{y}(X) = \Phi(X, \theta^*) + \varepsilon(X) + \delta(t) - \Phi(X, \hat{\theta}). \tag{3–10}$$

Therefore, the parameter identification objective is to minimize the parameter estimation error, defined as

$$\tilde{\theta} \triangleq \theta^* - \hat{\theta}, \tag{3–11}$$

where θ^* is defined in (3–4) and $\hat{\theta}$ represents the parameter estimates. Using the definitions of $\Phi(X, \theta^*)$ and $\Phi(X, \hat{\theta})$, a first-order Taylor series approximation-based model of the estimation error is used to obtain [20]

$$\Phi(X, \theta^*) - \Phi(X, \hat{\theta}) = \Phi'(X, \hat{\theta})\tilde{\theta} + R(X, \tilde{\theta}), \tag{3–12}$$

where $R(X, \tilde{\theta})$ represents the Lagrange remainder term.

3.3.1 Concurrent Learning (CL) Adaptation Laws for Nonlinear Regression

The adaptation law employs established CL methodologies, utilizing the known input and the system dynamics estimated by the DNN. The discrepancy between the known input and the DNN estimate is then stored in a history stack which allows previous information to be incorporated into the update law. The Jacobian of the DNN is denoted $\Phi'(X, \hat{\theta})$ and the Jacobian of the DNN at a specified point X_i and the current state estimates is denoted $\Phi'(X_i, \hat{\theta})$. The core idea is to leverage the recorded input-output data to refine the parameter estimates. The implementable form of the CL-DNN update law is designed as

$$\dot{\hat{\theta}} = \text{proj}\left(\gamma_1 \sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta}) \left(y_i - \Phi\left(X_i, \hat{\theta}\right)\right) - \gamma_2 \hat{\theta}\right), \tag{3–13}$$

where $\gamma_1, \gamma_2 \in \mathbb{R}$ denote positive user-selected adaptation gains, $N \in \mathbb{N}_{>0}$ is the user-selected number of input-output pairs residing in the history stack, and the projection

operator is a smooth projection operator defined in [112, Appendix E] which ensures that the adaptive estimate remains within the bounded parameter search space Θ. Data management strategies such as those in [91] and [127] could be leveraged to maximize the history stack's rank.

An analytical form of the implementable weight update law can be obtained using (3–9), (3–10), and (3–12) as

$$\dot{\hat{\theta}} = \text{proj}\Big(\gamma_1 \sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta}) \left(\Phi'(X_i, \hat{\theta})\tilde{\theta}\right) - \gamma_2 \hat{\theta} + \gamma_1 \sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta}) \left(R(X_i, \tilde{\theta}) + \varepsilon(X_i) + \delta(t)\right)\Big). \tag{3–14}$$

3.3.2 Stability Analysis

Using the structure of the DNN, the following bounds can be established where the Lagrange remainder in (3–12) is bounded as

$$\|R(X, \tilde{\theta})\| \leq \rho_0(\|X\|) \left\|\tilde{\theta}\right\|^2, \tag{3–15}$$

where $\rho_0(\|X\|) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a strictly increasing quadratic polynomial of the form $\rho_0(\|X\|) = a_2\|X\|^2 + a_1\|X\| + a_0$ with some constants $a_2, a_1, a_0 \in \mathbb{R}_{>0}$. Similarly, the Jacobian of the DNN can be bounded as

$$\left\|\Phi'(X, \hat{\theta})\right\| \leq \rho_1 \left(\|X\|\right), \tag{3–16}$$

where $\rho_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes a strictly-increasing function. Due to the fact that $R(X, \tilde{\theta})$, $\varepsilon(X)$, and $\delta(t)$, are bounded, the bounds $\left\|\Phi'^{\top}(X_i, \hat{\theta})R(X_i, \tilde{\theta}))\right\| \leq \rho_a\left(\|X_i\|\right) \left\|\tilde{\theta}\right\|^2$ and $\left\|\Phi'^{\top}(X_i, \hat{\theta})\left(\varepsilon(X_i) + \delta(t)\right)\right\| \leq \rho_b\left(\|X_i\|\right)$ hold where ρ_a, ρ_b: $\mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denote strictly-increasing bounding functions.

Let $\lambda_1 \triangleq \frac{\gamma_2}{2} - \gamma_1 + \gamma_1 \left(\lambda_{\min}\left\{\sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta})\right\}\right)$, $\lambda_{1d} \in \mathbb{R}_{\geq 0}$ be the desired convergence rate, and constant $\iota_1 \in \mathbb{R}_{>0}$ be defined as $\iota_1 \triangleq \frac{\gamma_2}{2}\overline{\theta}^2 + \frac{\gamma_1}{2}\left(\sum_{i=1}^{N} \rho_b\left(\|X_i\|\right)\right)^2$.

Assumption 3.1. There exists $\lambda_e > 0$ and there exists $T > 0$ for all $t \geq T$ such that $\lambda_{\text{min}} \left\{ \sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta}) \Phi'(X_i, \hat{\theta}) \right\} \geq \lambda_e$.

Remark 3.5. The derived stability conditions reveal a direct trade-off between the level of excitation and the magnitude of the ultimate bound. Specifically, since $\lambda_{\text{min}} \left\{ \sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta}) \Phi'(X_i, \hat{\theta}) \right\}$ is positive under Assumption 2 and is lower-bounded by λ_e, an increase in λ_e results in a larger value for λ_1 being obtained, and thus the user can select a larger desired convergence rate λ_{1d}. If Assumption 2 does not hold, the gain γ_2, which is based on the sigma modification technique in [128, Sec. 8.4.1], assists in demonstrating the convergence of $\tilde{\theta}$. Because the ultimate bound, $\sqrt{\frac{\iota_1}{\lambda_{1d}}}$, is inversely proportional to $\sqrt{\lambda_{1d}}$, which is upperbounded by λ_1, higher excitation directly tightens the ultimate bound obtained in the theorem allowing for the final error to be smaller than the bound on the parameter space $\overline{\theta}$.

Remark 3.6. The subsequent theorem establishes ultimate boundedness for the error signal, rather than the stronger property of uniform ultimate boundedness (UUB). This distinction, where the characteristics of the ultimate bound depend on the initial conditions, is a known consequence of relaxing the PE condition, as observed in the related literature such as [98] and [129]. In this result, the value of the history stack is determined by the value of $\hat{\theta}$ which evolves as a function of the initial condition. This dependence prevents the ultimate bound from being uniform across all initial conditions.

To facilitate the stability analysis, let the candidate Lyapunov function $V : \mathbb{R}^{\varkappa} \to \mathbb{R}_{\geq 0}$ be defined as

$$V(\tilde{\theta}) = \frac{1}{2} \tilde{\theta}^{\top} \tilde{\theta}. \tag{3–17}$$

Taking the time derivative of $V(\tilde{\theta})$ yields

$$\dot{V}(\tilde{\theta}) = \tilde{\theta}^{\top} \dot{\tilde{\theta}}. \tag{3–18}$$

Theorem 3.1 provides convergence guarantees for the parameter estimation errors for the update law in (3–14).

Theorem 3.1. *For the parameter identification objective defined in (3–11), the adaptation law developed in (3–14) ensures that the parameter estimation error $\tilde{\theta}$ is bounded in the sense that $\left\|\tilde{\theta}(t)\right\| \leq \sqrt{\left\|\tilde{\theta}(t_0)\right\|^2 e^{-2\lambda_{1d}(t-t_0)} + \frac{\iota_1}{\lambda_{1d}}\left(1 - e^{-2\lambda_{1d}t}\right)}$, for all $t \in \mathbb{R}_{\geq 0}$, provided $\lambda_1 > \lambda_{1d} + \frac{\gamma_1}{2}\rho_a^2\left(\|X_i\|\right)\left\|\tilde{\theta}(t_0)\right\|^2$.*

Proof. Consider the candidate Lyapunov function in (3–17). From (3–14) and (3–18),

$$\dot{V}(\tilde{\theta}) = -\tilde{\theta}^\top\Bigg(\gamma_1 \sum_{i=1}^{N} \Phi'^\top(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta})\tilde{\theta} + \gamma_2\left(\tilde{\theta} - \theta^*\right) + \gamma_1 \sum_{i=1}^{N} \Phi'^\top(X_i, \hat{\theta})\left(R(X_i, \tilde{\theta}) + \varepsilon(X_i) + \delta(t)\right)\Bigg). \tag{3–19}$$

Upper bounding (3–19) yields

$$\dot{V}(\tilde{\theta}) \leq -\left(\lambda_1 - \frac{\gamma_1}{2}\sum_{i=1}^{N}\rho_a^2\left(\|X_i\|\right)\left\|\tilde{\theta}\right\|^2\right)\|\tilde{\theta}\|^2 + \frac{\gamma_1}{2}\left(\sum_{i=1}^{N}\rho_b\left(\|X_i\|\right)\right) + \frac{\gamma_2}{2}\overline{\theta}^2. \tag{3–20}$$

When $\lambda_1 > \lambda_{1d} + \frac{\gamma_1}{2}\rho_a^2\left(\|X_i\|\right)\left\|\tilde{\theta}(t_0)\right\|^2$,

$$\dot{V}(\tilde{\theta}) \leq -\lambda_{1d}\left\|\tilde{\theta}\right\|^2 + \iota_1. \tag{3–21}$$

The inequality in (3–21) can be further bounded as $V(\tilde{\theta}(t)) \leq V(\tilde{\theta}(t_0))e^{-\lambda_{1d}(t-t_0)} + \frac{\iota_1}{\lambda_{1d}}\left(1 - e^{-\lambda_{1d}t}\right)$. Then, [130, Def. 4.6] can be invoked to conclude that $\tilde{\theta}$ is bounded such that $\left\|\tilde{\theta}(t)\right\| \leq \sqrt{\left\|\tilde{\theta}(t_0)\right\|^2 e^{-2\lambda_{1d}(t-t_0)} + \frac{\iota_1}{\lambda_{1d}}\left(1 - e^{-2\lambda_{1d}t}\right)}$. Using (3–17) and (3–21) implies $\tilde{\theta} \in \mathcal{L}_\infty$. Additionally, due to the use of the projection operator, $\hat{\theta} \in \mathcal{L}_\infty$. □

3.4 Numerical Analysis of Identifiability

The theoretical development in Section 3.2 established that for NIP systems, the FE condition is equivalent to parameter identifiability. Consequently, confirming the structural validity of a network architecture reduces to verifying whether its Jacobian can satisfy the rank requirement in Assumption 3.1. This analysis serves as a diagnostic tool to screen architectures for structural identifiability prior to deployment. To empirically

validate this capability, a numerical analysis of the rank conditions was performed on various LIP and NIP network structures.

Four network architectures were evaluated, each constrained to have exactly $\varkappa = 13$ parameters to decouple structural properties from approximation capacity. The shallow architectures were designed with a single hidden layer containing 4 neurons while the deep architectures were designed to have 2 hidden layers with 2 neurons each. Both linear and nonlinear activation functions were employed, with the hyperbolic tangent chosen for the nonlinear case. For each trial the network was initialized using Xavier initialization [131] with scaling parameter 4.0, while biases were initialized from a normal distribution with a standard deviation of 0.75.

The analysis was performed using double-precision floating-point arithmetic to distinguish between structural singularity and numerical ill-conditioning. In every trial, the network Jacobian $\Phi'(X, \hat{\theta})$ was evaluated over a discrete set of $N = 200$ inputs. This set was obtained by sampling the function $X(t) \triangleq 10\,(\sin(t) + \sin(0.5t))$ uniformly across the domain $t \in [0, 20]$.

Remark 3.7. Utilizing a sinusoidal trajectory in this offline analysis serves as a baseline validation to isolate structural properties.

To quantify the impact of regularization on the condition number, the relationship between the regularization magnitude and the resulting condition number, computed as $\kappa(\sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta}) + \sigma I)$, was analyzed and is illustrated in Figure 3-1. As shown in Figure 3-1, there is a critical distinction between the stabilization requirements of architectures with nonlinear activation functions which can achieve a stable condition number ($\kappa < 10^5$) with minimal regularization, while the linear baselines require a regularization parameter orders of magnitude larger to reach the same threshold. This confirms the assertion in Remark 3.2 that architectures with high structural redundancy require artificial regularization to enforce uniqueness, whereas nonlinear structures inherently satisfy the identifiability condition.

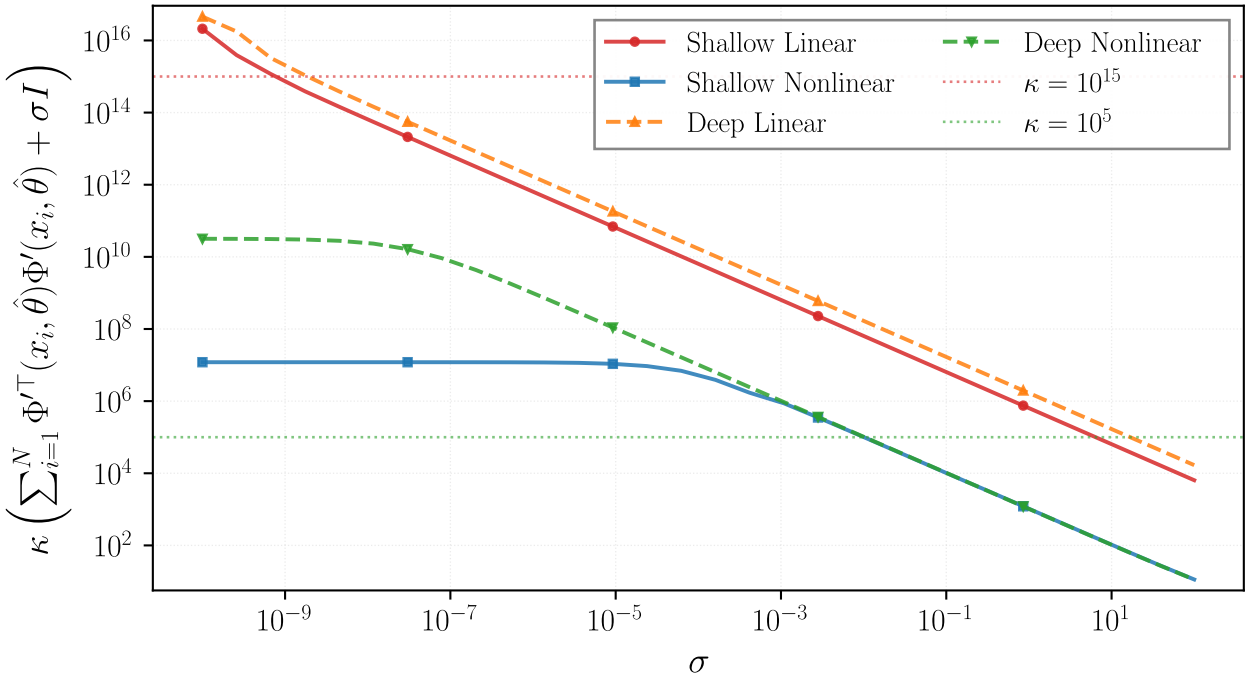

Figure 3-1. Condition number vs. regularization parameter

3.5 Conclusions

A framework for the online identification of DNNs characterized by a NIP structure is presented. Addressing the lack of formal connections between excitation and unique parameterization in existing literature, generalized identifiability conditions were derived to demonstrate that the verifiable FE condition is equivalent to parameter identifiability. Furthermore, numerical analysis validates that this rank condition serves as an effective diagnostic tool for architectural design, confirming that structural redundancies in linear networks lead to singularity while appropriate nonlinearities restore identifiability. Leveraging this theoretical foundation, a continuous nonlinear regression framework was developed using all-layer adaptation. Finally, a Lyapunov-based stability analysis guarantees the ultimately bounded convergence of weight estimation errors to a neighborhood of the origin.

CHAPTER 4
COMBINED SYSTEM IDENTIFICATION AND CONTROL USING LYAPUNOV-BASED DEEP NEURAL NETWORKS

Building on the theoretical foundations of DNN identifiability established in the previous chapter, this chapter and the work in [103] presents two approaches to the combined problem of system identification and control, unified by the use of system dynamics as a regularization mechanism within the adaptation law. The first section develops a concurrent learning (CL) framework that utilizes a history stack of recorded data to satisfy parameter convergence requirements. This history stack is constructed by leveraging a reconstructed solution to the system dynamics, formulated through the DNN output and an acceleration observer. The second section introduces a continuous learning from offline experience (CLOE) adaptation law, which leverages prior knowledge of dynamic behavior under specific operating conditions to enhance estimation in novel environments. By reformulating the learning objective to operate without the need for acceleration observers, this method offers increased robustness to disturbances and permits the use of non-smooth control inputs, such as sliding mode terms. Together, these results demonstrate how leveraging historical data as a regularization term can significantly enhance the performance and robustness of real-time control applications.

4.1 Concurrent Learning for System Identification and Control Using Lyapunov-Based Deep Neural Networks

This section presents the first result that enables the use of a CL adaptation law for the wights of all the layers of a DNN-based controller applied to a class of second-order control-affine systems allowing for simultaneous system identification and control. The developed CL-based adaptation achieves convergence of the DNN's parameter estimates to a neighborhood of their ideal values, provided the DNN's Jacobian satisfies a finite excitation condition. A Lyapunov-based stability analysis is conducted to ensure convergence of the tracking error, weight estimation errors, and observer errors to a neighborhood of the origin. Simulations demonstrated approximately 40% improvement

in function approximation performance and over 65% improvement on off-trajectory evaluations compared to a baseline without CL.The results of this section have been published in [103].

4.1.1 Control Design

Consider a control-affine nonlinear dynamic system modeled as

$$\ddot{x} = f(x, \dot{x}) + u, \tag{4–1}$$

where $x, \dot{x} \in \mathbb{R}^n$ denotes the available generalized position and velocity, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes an unknown continuously differentiable function, and $u \in \mathbb{R}^n$ denotes a control input. The tracking control objective is to simultaneously track a user-defined reference trajectory $x_d \in \mathbb{R}^n$ and learn the unknown function $f(x, \dot{x})$ online using a DNN starting from the initial time t_0. Therefore, $f(x, \dot{x})$ can be modeled as

$$f(x, \dot{x}) = \Phi(X, \theta^*) + \varepsilon(X), \tag{4–2}$$

where the input to the DNN is denoted $X \triangleq \left[x^\top, \dot{x}^\top\right]^\top$. Due to the continuity of $f(x, \dot{x})$ and Φ and the constraint of a bounded search space, the function approximation error ε which bounds $\sup_{x \in \Omega} \|f(x, \dot{x}) - \Phi(X, \theta)\|$ exists but may not be prescribable as stated in the universal function approximation theorem [107, Theorem 3.1].

The reference trajectory is assumed to satisfy $\|x_d\| \leq \overline{x_d}$, $\|\dot{x}_d\| \leq \overline{\dot{x}_d}$, and $\|\ddot{x}_d\| \leq \overline{\ddot{x}_d}$, where $\overline{x_d}, \overline{\dot{x}_d}, \overline{\ddot{x}_d} \in \mathbb{R}_{>0}$ are known constants. To quantify the tracking objective, the trajectory and auxiliary tracking errors $e, r \in \mathbb{R}^n$ are defined as

$$e \triangleq x - x_d, \quad r \triangleq \dot{e} + \alpha_1 e, \tag{4–3}$$

where $\alpha_1 \in \mathbb{R}_{>0}$ is a user-selected control gain. Taking the time derivative of (4–3), applying (4–1), and (4–2) yields the open-loop error system

$$\dot{r} = \Phi(X, \theta^*) + \varepsilon(X) + u - \ddot{x}_d + \alpha_1 \dot{e}. \tag{4–4}$$

The subsequent development considers the concurrent tracking control and system identification problem. The parameter identification objective is quantified using the parameter estimation error defined as $\tilde{\theta} \triangleq \theta^* - \hat{\theta}$, where $\hat{\theta} \in \mathbb{R}^{\varkappa}$ represents the parameter estimates. Using the definitions of $\Phi(X, \theta^*)$ and $\Phi(X, \hat{\theta})$, a first-order Taylor series approximation-based model of the estimation error is used to obtain [20]

$$\Phi(X, \theta^*) - \Phi(X, \hat{\theta}) = \Phi'(X, \hat{\theta})\tilde{\theta} + R(X, \tilde{\theta}), \tag{4–5}$$

where $R(X, \tilde{\theta})$ represents the Lagrange remainder term.

4.1.1.1 Closed-loop Error System and Control Law Development

Since $f(x, \dot{x})$ in (4–4) is unknown, we are motivated to develop an approximation that can be used as a feedforward control element. Based on the subsequent stability analysis, the control input is designed as

$$u = \ddot{x}_d - \Phi(X, \hat{\theta}) - k_1 r - e - \alpha_1 \dot{e}, \tag{4–6}$$

where $k_1 \in \mathbb{R}_{>0}$ is a user-defined control gain. Substituting (4–2) and (4–6) into (4–4), applying (4–5), and canceling cross terms yields

$$\dot{r} = \Phi'(X, \hat{\theta})\tilde{\theta} + R(x, \tilde{\theta}) + \varepsilon(X) - k_1 r - e. \tag{4–7}$$

4.1.1.2 Dynamic State-Derivative Observer

Motivated by the desire to incorporate previous state information into the update law in a CL style, the update law is augmented with a history stack containing the error between the calculated control input and a reconstructed version of the control input. To reconstruct (4–1), an observer is developed to estimate the unmeasurable state $\ddot{x}$. The dynamic state-derivative observer is designed as

$$\dot{\hat{r}} = \hat{\Delta} - \ddot{x}_d + \alpha_1 \dot{e} + \alpha_2 \tilde{r}, \tag{4–8}$$

$$\dot{\hat{\Delta}} = \tilde{r} + k_{\Delta}\tilde{\Delta} - \dot{u}, \tag{4–9}$$

where $\hat{r}, \hat{\Delta} \in \mathbb{R}^n$ denote the observer estimates for r and $\ddot{x}$, respectively, $\tilde{r}, \tilde{\Delta} \in \mathbb{R}^n$ denote the observer errors, $\tilde{r} = r - \hat{r}$ and $\tilde{\Delta} = \ddot{x} - \hat{\Delta}$, respectively $\alpha_2, k_\Delta \in \mathbb{R}_{>0}$ denote constant observer gains, and $\dot{u}$ is the unmeasurable derivative of the control input. Taking the time derivative of $\tilde{r}$ and $\tilde{\Delta}$ and applying (4–8) and (4–9) yields

$$\dot{\tilde{r}} = \tilde{\Delta} - \alpha_2 \tilde{r}, \tag{4–10}$$

$$\dot{\tilde{\Delta}} = \dot{f}(x, \dot{x}) + \dot{u} - \tilde{r} - k_\Delta \tilde{\Delta} - \dot{u}, \tag{4–11}$$

where $\dot{f}(x, \dot{x}) \triangleq \frac{d}{dt} f(x, \dot{x}) = \frac{\partial f}{\partial x}\dot{x} + \frac{\partial f}{\partial \dot{x}}\ddot{x}$ and $\dot{u} \triangleq \frac{d}{dt}u$. The observer error $\tilde{r}$ is known because r and $\hat{r}$ are known. Because $\tilde{\Delta} = \dot{\tilde{r}} + \alpha_2 \tilde{r}$, the implementable form of (4–9) can be obtained by integrating on both sides and using the relation $\int_{t_0}^{t} \dot{\tilde{r}} = \tilde{r}(t) - \tilde{r}(t_0)$, which yields $\hat{\Delta}(t) = \hat{\Delta}(t_0) + k_\Delta(\tilde{r}(t) - \tilde{r}(t_0)) - (u(t) - u(t_0)) + \int_{t_0}^{t} (k_\Delta \alpha_2 + 1)\tilde{r}(\tau) d\tau$.

Let the concatenated error vectors $z \in \mathbb{R}^{2n}$ and $\zeta \in \mathbb{R}^{\psi}$ be defined as $z \triangleq \left[\tilde{r}^\top, \tilde{\Delta}^\top\right]^\top$ and $\zeta \triangleq \left[r^\top, e^\top, \tilde{\theta}^\top\right]^\top$, respectively, where $\psi \triangleq 2n + \varkappa$. Additionally, let the open and connected set $\mathcal{D} \subset \mathbb{R}^{\psi}$ be defined as $\mathcal{D} \triangleq \left\{\zeta \in \mathbb{R}^{\psi} : \|\zeta\| < \chi\right\}$, where $\chi \in \mathbb{R}_{>0}$ denotes a subsequently defined known upper bound. The following lemma establishes a bound on $\dot{f}(x, \dot{x})$ when $\zeta \in \mathcal{D}$ to facilitate the convergence analysis for the observer.

Lemma 4.1. *For all $\zeta \in \mathcal{D}$, there exists a constant $\delta_f \in \mathbb{R}_{>0}$ such that the bound $\left\|\dot{f}(x, \dot{x})\right\| \leq \delta_f$ holds.*

Proof. For all $\zeta \in \mathcal{D}$, $\|\zeta\| < \chi$, and hence $\|r\|, \|e\|, \left\|\tilde{\theta}\right\| < \chi$. Since $\|x\| = \|e + x_d\| \leq \|e\| + \|x_d\| \leq \|e\| + \overline{x_d}$ and $\|\dot{x}\| = \|r - \alpha_1 e + \dot{x}_d\| \leq \|r\| + \alpha_1 \|e\| + \overline{\dot{x}_d}$, the bounds $\|x\| \leq \chi + \overline{x_d}$ and $\|\dot{x}\| \leq (\alpha_1 + 1)\chi + \overline{\dot{x}_d}$ hold for all $\zeta \in \mathcal{D}$. Moreover, using (4–1) and (4–6), $\ddot{x} = f(x, \dot{x}) - \Phi(X, \hat{\theta}) - k_1 r - e - \alpha_1 \dot{e} + \ddot{x}_d$. Using the structure of the DNN in (1–2), and because $\hat{\theta}, \theta^* \in \Theta$, it follows that there exists $\bar{\Phi} \in \mathbb{R}_{>0}$ such that $\|\Phi(X, \theta^*)\|, \left\|\Phi(X, \hat{\theta})\right\| \leq \bar{\Phi}$. Hence, using (4–2) yields $\left\|f(x, \dot{x}) - \Phi(X, \hat{\theta})\right\| \leq 2\bar{\Phi} + \bar{\varepsilon}$. Therefore, the definition (4–3) and bound $\|\ddot{x}_d\| \leq \overline{\ddot{x}_d}$ yields $\|\ddot{x}\| \leq \overline{\ddot{x}} \triangleq 2\bar{\Phi} + \bar{\varepsilon} + ((k_1 + \alpha_1) + (\alpha_1^2 + 1))\chi + \overline{\ddot{x}_d}$ for all $\zeta \in \mathcal{D}$. Since $f(x, \dot{x})$ is continuously differentiable, there exist constants $\varrho_1, \varrho_2 \in \mathbb{R}_{>0}$ such that $\left\|\frac{\partial f}{\partial x}\right\| \leq \varrho_1$ and $\left\|\frac{\partial f}{\partial \dot{x}}\right\| \leq \varrho_2$ for all $\zeta \in \mathcal{D}$. Therefore, $\left\|\dot{f}(x, \dot{x})\right\| \leq \left\|\frac{\partial f}{\partial x}\right\| \|\dot{x}\| + \left\|\frac{\partial f}{\partial \dot{x}}\right\| \|\ddot{x}\|$

$\leq \varrho_1 \left((\alpha_1 + 1)\chi + \overline{\dot{x}_d} \right) + \varrho_2 \overline{\ddot{x}}$. Thus, selecting $\delta_f \triangleq \varrho_1 \left((\alpha_1 + 1)\chi + \overline{\dot{x}_d} \right) + \varrho_2 \overline{\ddot{x}}$ yields $\left\| \dot{f}(x, \dot{x}) \right\| \leq \delta_f$ for all $\zeta \in \mathcal{D}$. □

Let $\Lambda_1 \triangleq \min\{k_\Delta, 2\alpha_2\}$. The following lemma establishes the convergence properties of the observer error system in (4–10) and (4–11).

Remark 4.1. When the observer is activated with the system at rest, the initial velocity and acceleration are both zero, simplifying the calculation and eliminating the need for an upper bound. If the initial acceleration is not exactly known, a conservative upperbound can be used and will result in a longer settling time t_Δ.

Lemma 4.2. *Consider the observer given by (4–8) and (4–9). The observer error is bounded in the sense that* $\|z(t)\| \leq \sqrt{\|z(t_0)\|^2 e^{-\Lambda_1 (t - t_0)} + \frac{\delta_f^2}{\Lambda_1} \left(1 - e^{-\Lambda_1 (t - t_0)}\right)}$ *for all* $t \in \mathbb{R}_{\geq 0}$*, provided* $\zeta \in \mathcal{D}$*. Furthermore, suppose the initial acceleration error is bounded by a known constant,* $\left\| \tilde{\Delta}(t_0) \right\| \leq \bar{\Delta}_0$ *where* $\bar{\Delta}_0 \in \mathbb{R}_{>0}$*. The observer can achieve a prescribed accuracy* $\delta_\Delta \in \mathbb{R}_{>0}$ *with the settling time* $t_\Delta \triangleq t_0 + \frac{1}{\Lambda_1} \ln \left(\frac{k_\Delta \Lambda_1 \bar{\Delta}_0^2 - \delta_f^2}{k_\Delta \Lambda_1 \delta_\Delta^2 - \delta_f^2} \right)$*, provided the feasibility gain condition* $k_\Delta \Lambda_1 > \frac{\delta_f^2}{\delta_\Delta^2}$ *is satisfied.*

Proof. Consider the candidate Lyapunov function $\mathcal{V}_\Delta(z) \triangleq \frac{1}{2} \tilde{\Delta}^\top \tilde{\Delta} + \frac{1}{2} \tilde{r}^\top \tilde{r}$. Taking the derivative, using (4–10), (4–11), and canceling cross terms yields $\dot{\mathcal{V}}_\Delta(z) \leq -\tilde{\Delta}^\top k_\Delta \tilde{\Delta} - \tilde{r}^\top \alpha_2 \tilde{r} + \tilde{\Delta}^\top \dot{f}(x, \dot{x})$. Using Young's inequality and Lemma 4.1, $\dot{\mathcal{V}}_\Delta(z)$ can be further upper-bounded as $\dot{\mathcal{V}}_\Delta \leq -\Lambda_1 \mathcal{V}_\Delta + \frac{\delta_f^2}{2k_\Delta}$ provided $\zeta \in \mathcal{D}$. Therefore, $\mathcal{V}_\Delta(z(t)) \leq \mathcal{V}_\Delta(z(t_0)) e^{-\Lambda_1 (t - t_0)} + \frac{\delta_f^2}{2k_\Delta \Lambda_1} \left(1 - e^{-\Lambda_1 (t - t_0)}\right)$ and $\|z\| \leq \sqrt{\|z(t_0)\|^2 e^{-\Lambda_1 (t - t_0)} + \frac{\delta_f^2}{k_\Delta \Lambda_1} \left(1 - e^{-\Lambda_1 (t - t_0)}\right)}$, provided $\zeta \in \mathcal{D}$. Furthermore, if $\left\| \tilde{\Delta}(t_0) \right\| \leq \bar{\Delta}_0$ and $\hat{r}(t_0) = r(t_0)$ (because $r(t_0)$ can be measured), then $\|z(t_0)\| \leq \bar{\Delta}_0$. Then $\|z\| \leq \sqrt{\bar{\Delta}_0^2 e^{-\Lambda_1 (t - t_0)} + \frac{\delta_f^2}{k_\Delta \Lambda_1} \left(1 - e^{-\Lambda_1 (t - t_0)}\right)}$. For the prescribed accuracy δ_Δ, using the differential inequality, the settling time $t_\Delta = t_0 + \frac{1}{\Lambda_1} \ln \left(\frac{k_\Delta \Lambda_1 \bar{\Delta}_0^2 - \delta_f^2}{k_\Delta \Lambda_1 \delta_\Delta^2 - \delta_f^2} \right)$ is obtained after imposing $\delta_\Delta \geq \sqrt{\bar{\Delta}_0^2 e^{-\Lambda_1 (t - t_0)} + \frac{\delta_f^2}{k_\Delta \Lambda_1} \left(1 - e^{-\Lambda_1 (t - t_0)}\right)}$, provided $\zeta \in \mathcal{D}$. For the settling time to be feasible, the argument of the natural logarithm needs to be positive; imposing this condition yields the feasibility gain condition $k_\Delta \Lambda_1 > \frac{\delta_f^2}{\delta_\Delta^2}$. □

Remark 4.2. Using Lemma 4.2, the observer errors, $\tilde{\Delta}$ and $\tilde{r}$ will converge to the prescribed ultimate bound δ_Δ after the time t_Δ has passed.

4.1.2 CL Adaptation Laws For Adaptive Control

The adaptation law is developed by extending established CL techniques to the NIP structure of the DNN. The shorthand notation $\Phi'(X, \hat{\theta})$ in (4–12) denotes the Jacobian of the DNN at the current state and weight estimates, and $\Phi'(X_i, \hat{\theta})$ represents the Jacobian of the DNN using the current weight estimates and the previous state X_i, where $X_i \triangleq X(t_i)$ and $t_i \in [t_\Delta, t]$, represents states within this time interval. Compared to other CL approaches which begin gathering data immediately, the history stack is only constructed after $t \geq t_\Delta$ where the observer settling time t_Δ is defined in Lemma 4.2. This time-based condition ensures that the user-defined number of data points $N \in \mathbb{N}$ which construct the history stack are gathered only after the observer has reached its ultimate bound which is required to ensure accuracy of the stored data. The implementable form of the CL-DNN update law is designed as

$$\dot{\hat{\theta}} = \text{proj}\Big(\Gamma\Big(\Phi'^{\top}(X, \hat{\theta})r - \gamma_1 \sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta})\,(u_i - \hat{u}_i) - \gamma_2 \hat{\theta}\Big)\Big), \tag{4–12}$$

where $\gamma_1, \gamma_2 \in \mathbb{R}_{>0}$ denote user-selected adaptation gains, and $\Gamma \in \mathbb{R}^{\varkappa \times \varkappa}$ denotes a positive-definite time-varying least squares adaptation gain matrix where

$$\frac{d}{dt}\Gamma^{-1} = \begin{cases} -\beta\Gamma^{-1} + \gamma_1 \left(\sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta})\right), \\ \quad \text{if } \lambda_{\Gamma,\text{min}} < \lambda_{\text{min}}(\Gamma) \text{ and } \lambda_{\text{max}}(\Gamma) < \lambda_{\Gamma,\text{max}} \\ 0, \qquad \text{otherwise} \end{cases} \tag{4–13}$$

where β represents a user selected forgetting factor $\beta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and $\lambda_{\Gamma,\text{min}}$, $\lambda_{\Gamma,\text{max}}$ are user-selected bounds for the minimum and maximum eigenvalues of Γ, respectively. The adaptation gain in (4–13) is initialized to be positive-definite (PD) and it can be shown that $\Gamma(t)$ remains PD for all $t \in \mathbb{R}_{\geq 0}$ [90]. A reconstructed estimate of the

calculated control input can be determined as

$$\hat{u}_i = \hat{\Delta}_i - \Phi(X_i, \hat{\theta}). \tag{4–14}$$

An analytical form of the implementable weight adaptation law in (4–12) can be obtained from (4–1), (4–2), (4–5), (4–12), and (4–14) as

$$\dot{\hat{\theta}} = \text{proj}\Big(\Gamma \Big(\Phi'^{\top}(X, \hat{\theta}) r - \gamma_2 \hat{\theta} + \gamma_1 \sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta}) \Phi'(X_i, \hat{\theta}) \tilde{\theta} + \gamma_1 \sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta}) \left(R(X, \tilde{\theta}) + \varepsilon(X_i) - \tilde{\Delta}_i \right) \Big) \Big). \tag{4–15}$$

4.1.3 Stability Analysis

Using the structure of the DNN, the following bounds can be established

$$\begin{aligned} \left\| R(X, \tilde{\theta}) \right\| &\leq \rho_1 \left(\|\zeta\| \right) \|\zeta\|^2, \\ \left\| \widehat{\Phi}'(X_i, \hat{\theta}) \right\| &\leq \rho_2 \left(\|X_i\| \right), \end{aligned} \tag{4–16}$$

where $\rho_1, \rho_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denote strictly-increasing functions. Due to the fact that $R(X, \tilde{\theta})$, $\Phi'(X_i, \hat{\theta})$, and ε are bounded, then the following bounds can be established

$$\begin{aligned} \left\| \Phi'^{\top}(X_i, \hat{\theta}) R(X_i, \tilde{\theta}) \right\| &\leq \rho_3 \left(\|\zeta_i\| \right) \|\zeta\|^2, \\ \left\| \Phi'^{\top}(X_i, \hat{\theta}) \varepsilon(X_i) \right\| &\leq \rho_4 \left(\|X_i\| \right), \end{aligned} \tag{4–17}$$

where $\rho_3, \rho_4 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denote strictly-increasing functions. Let $\rho_\Delta \left(\|\zeta\| \right) \geq \frac{1}{2k_1} \rho_1^2 \left(\|\zeta\| \right) \|\zeta\|^2 + \frac{\gamma_1^2}{2} \sum_{i=1}^{N} \rho_3^2 \left(\|\zeta_i\| \right) \|\zeta\|^2$, where $\rho_\Delta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes an invertible strictly-increasing function. Since the approximation capabilities of DNNs holds on a compact domain Ω, the subsequent stability analysis requires ensuring $X(t) \in \Omega$ for all $t \in [t_0, \infty)$. This is achieved by yielding a stability result which constrains ζ in a compact domain. Therefore, consider the compact domain $\mathcal{D} \triangleq \left\{ \sigma \in \mathbb{R}^{\psi} : \|\sigma\| \leq \chi \right\}$ in which ζ is supposed to lie to develop Theorem 4.1. It follows that if $\|\zeta\| \leq \chi$ then X can be bounded as $\|X\| \leq (\alpha + 2)\chi + \overline{x_d} + \overline{\dot{x}_d}$. Therefore,

select $\Omega \triangleq \{\sigma \in \mathbb{R}^{2n} : \|\sigma\| \leq (\alpha_1 + 2)\chi + \overline{x_d} + \overline{\dot{x}_d}\}$. Then $\zeta \in \mathcal{D}$ implies $X \in \Omega$. Let $\lambda \triangleq \min\left(\frac{k_1}{2} - \frac{1}{2}, \alpha_1, \frac{\gamma_1}{2}\left(\lambda_{\min}\left(\sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta})\right) - 1 - N\delta_{\Delta}\right) - \frac{1}{2} + \frac{\gamma_2}{2}\right)$ and $\lambda_d \in \mathbb{R}_{>0}$ denote the desired convergence rate. To ensure that arbitrary initial conditions are always included, the user-selected constant χ is selected as $\chi \triangleq \rho_{\Delta}^{-1}(\lambda - \lambda_d)$. Then, it follows that $z(t_0) \in \mathcal{D} = \{\zeta \in \mathbb{R}^{2n+p} : \|\zeta\| \leq \chi\}$ is always satisfied. Because the solution $t \mapsto \zeta(t)$ is continuous due to the Caratheodory existence conditions [132, Ch.2, Theorem 1.1], there exists a time-interval $\mathcal{I} \triangleq [t_0, t_1)$ such that $\|\zeta(t)\| < \mathcal{D}$ for all $t \in \mathcal{I}$. It follows that $X(t) \in \Omega$ for all $t \in \mathcal{I}$, therefore the universal function approximation property holds over this time interval. In the subsequent stability analysis, we analyze the convergence properties of the solutions and also establish that $\mathcal{I}$ can be extended to $[t_0, \infty)$. Let the set $\mathcal{S} \subset \mathcal{D}$ be the set of stabilizing initial conditions defined as

$$\mathcal{S} \triangleq \left\{\sigma \in \mathbb{R}^{\psi} : \|\zeta(t_0)\| < \sqrt{\frac{\beta_1}{\beta_2}\rho_{\Delta}^{-1}(\lambda - \lambda_d)^2 - \frac{\iota}{\lambda_d}}\right\},$$

where $\beta_1 \triangleq \min\left\{\frac{1}{2}, \frac{1}{2}\frac{1}{\lambda_{\Gamma,\max}}\right\}$, $\beta_2 \triangleq \max\left\{\frac{1}{2}, \frac{1}{2}\frac{1}{\lambda_{\Gamma,\min}}\right\}$, and $\iota \triangleq \frac{1}{2}\overline{\varepsilon}^2 + \frac{\gamma_2}{2}\overline{\theta}^2 + \frac{\gamma_1}{2}\sum_{i=1}^{N}\rho_4^2(\|X_i\|) + \frac{\gamma_1 N\delta_{\Delta}}{2}\sum_{i=1}^{N}\rho_2^2(\|X_i\|)$.

To guarantee that the adaptive update law in Theorem 4.1 achieves the parameter identification, i.e., $\|\tilde{\theta}\|$ is minimized, the history stack needs to be sufficiently rich, i.e., the system needs to be excited over a finite duration of time as specified in the following assumption.

Assumption 4.1. There exists $\lambda_e > 0$ and there exists $T > t_{\Delta}$ for all $t \geq T$ such that $\lambda_{\min}\left\{\sum_{i=1}^{N} \Phi'^{\top}(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta})\right\} \geq \lambda_e$.

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

$$\mathcal{V}(\zeta) = \frac{1}{2}r^{\top}r + \frac{1}{2}e^{\top}e + \frac{1}{2}\tilde{\theta}^{\top}\Gamma^{-1}\tilde{\theta}, \tag{4–18}$$

which satisfies the inequality $\beta_1 \|\zeta\|^2 \leq \mathcal{V}(\zeta) \leq \beta_2 \|\zeta\|^2$. Taking the time derivative of $\mathcal{V}(\zeta)$, and applying (4–3) and (4–7) yields

$$\dot{\mathcal{V}} = r^\top \left(\Phi'(X, \hat{\theta})\tilde{\theta} + R(X, \tilde{\theta}) + \varepsilon(X) - k_1 r - e \right) - e^\top \alpha_1 e + \tilde{\theta}^\top \Gamma^{-1} \dot{\tilde{\theta}} + \frac{1}{2}\tilde{\theta}^\top \left(\frac{d}{dt}\Gamma^{-1} \right) \tilde{\theta}. \tag{4–19}$$

Using (4–13), the last term in (4–19) can be bounded as

$$\frac{1}{2}\tilde{\theta}^\top \left(\frac{d}{dt}\Gamma^{-1} \right) \tilde{\theta} \leq \frac{1}{2}\tilde{\theta}^\top \gamma_1 \left(\sum_{i=1}^{N} \Phi'^\top(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta}) \right) \tilde{\theta}. \tag{4–20}$$

Theorem 4.1 provides convergence guarantees for the tracking and parameter estimation errors using the update law in (4–15).

Theorem 4.1. *Let the gain conditions $k_\Delta \Lambda_1 > \frac{\delta_f^2}{\delta_\Delta^2}$ and $\lambda_d > 0$ be satisfied, and $\|\zeta(0)\| \in \mathcal{S}$. For the dynamical system in (4–1), the controller in (4–6) and the adaptation law developed in (4–15) ensures the concatenated error vector ζ is bounded in the sense that $\|\zeta(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 e^{-\frac{\lambda_d}{\beta_2}(t-t_0)} + \frac{\beta_2 \iota}{\beta_1 \lambda_d}\left(1 - e^{-\frac{\lambda_d}{\beta_2}(t-t_0)}\right)}$, for all $t \in \mathbb{R}_{\geq 0}$.*

Proof. Consider the candidate Lyapunov function in (4–18). From (4–15), (4–19), and using (4–20), $\dot{\mathcal{V}}$ can be upper bounded as

$$\dot{\mathcal{V}} \leq -r^\top k_1 r - e^\top \alpha_1 e + r^\top \left(R(X, \tilde{\theta}) + \varepsilon(X) \right) - \tilde{\theta}^\top \frac{\gamma_1}{2} \sum_{i=1}^{N} \Phi'^\top(X_i, \hat{\theta})\Phi'(X_i, \hat{\theta})\tilde{\theta} - \tilde{\theta}^\top \gamma_2 \tilde{\theta} + \tilde{\theta}^\top \gamma_2 \theta^* - \tilde{\theta}^\top \gamma_1 \sum_{i=1}^{N} \Phi'^\top(X_i, \hat{\theta}) \left(R(X_i, \tilde{\theta}) + \varepsilon(X_i) - \tilde{\Delta}_i \right).$$

Using Lemma 4.2, the bound $\left\| \tilde{\Delta}_i \right\| \leq \delta_\Delta$ holds for all $i \in \{1, \ldots, N\}$, provided $\zeta \in \mathcal{D}$, the data is collected after the settling time t_Δ, and the feasibility gain condition $k_\Delta \Lambda_1 > \frac{\delta_f^2}{\delta_\Delta^2}$ is satisfied. Therefore, using the bounds in (4–16) and (4–17), $\dot{\mathcal{V}}$ can be upper bounded as $\dot{\mathcal{V}} \leq -\left(\lambda - \rho_\Delta\left(\|\zeta\|\right)\right)\|\zeta\|^2 + \iota$, when $\zeta \in \mathcal{D}$. Therefore, when ζ is initialized such that $\zeta(t_0) \in \mathcal{S}$, then, from the definition of $\mathcal{S}$, $\lambda > \lambda_d + \rho_\Delta\left(\sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2 \iota}{\beta_1 \lambda_d}}\right)$. Recall that, because the solution $t \mapsto \zeta(t)$ is continuous, ζ cannot instantaneously

escape $\mathcal{S}$ at t_0, therefore there exists a time-interval $\mathcal{I}$ such that $\zeta(t) \in \mathcal{S}$ for all $t \in \mathcal{I}$, implying $\|\zeta(t)\| < \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2 \iota}{\beta_1 \lambda_d}}$ for all $t \in \mathcal{I}$. Because ρ_Δ is strictly increasing, $\rho_\Delta(\|\zeta(t)\|) < \rho_\Delta\left(\sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2 \iota}{\beta_1 \lambda_d}}\right)$ for all $t \in \mathcal{I}$. As a result,

$$\dot{\mathcal{V}} \leq -\lambda_d \|\zeta\|^2 + \iota \tag{4–21}$$

for all $t \in \mathcal{I}$. Using $\beta_1 \|\zeta\|^2 \leq \mathcal{V}(\zeta) \leq \beta_2 \|\zeta\|^2$, $\dot{\mathcal{V}}$, and solving the differential inequality over $\mathcal{I}$ yields $\mathcal{V}(\zeta(t)) \leq \mathcal{V}(\zeta(t_0))e^{-\frac{\lambda_d}{\beta_2}(t-t_0)} + \frac{\beta_2 \iota}{\lambda_d}\left(1 - e^{-\frac{\lambda_d}{\beta_2}(t-t_0)}\right)$. Using [130, Thm. 3.3], $\mathcal{I}$ can be extended to the interval $[t_0, \infty)$ and $\|\zeta(t)\| < \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2 \iota}{\beta_1 \lambda_d}}$ for all $t \in [t_0, \infty)$. Under more excitation $\lambda_{\min}\left\{\sum_{i=1}^{N} \widehat{\Phi}'^{\top}(X_i, \hat{\theta})\widehat{\Phi}'(X_i, \hat{\theta})\right\}$ grows, thus increasing λ_e and λ_d, tightening the ultimate bound of ζ. Hence, if $\zeta(t_0) \in \mathcal{S}$, then $\zeta(t) \in \mathcal{S} \subset \mathcal{D}$ and therefore $X \in \Omega$ for all $t \geq 0$. Using (4–18) and (4–21) implies $e, r, \tilde{\theta} \in \mathcal{L}_\infty$. The fact that $x_d, \dot{x}_d, \ddot{x}_d, e, r \in \mathcal{L}_\infty$ implies $x, \dot{x} \in \mathcal{L}_\infty$, using this and the fact $\hat{\theta} \in \mathcal{L}_\infty$ is bounded by the use of the projection operator implies u is bounded. □

4.1.4 Simulations

Comparative simulation results are provided to demonstrate the performance of the developed method with a baseline which is tracking error based (i.e., $\gamma_1 = 0$). Simulations were performed on a 6-DOF rigid body vehicle with state vector $x = [\mathrm{x}, \mathrm{y}, \mathrm{z}, \varphi, \vartheta, \psi]^\top \in \mathbb{R}^6$. The system dynamics are represented by $f(x, \dot{x}) = \left[-\frac{k_L}{m}\dot{\mathrm{x}}|\dot{\mathrm{x}}|\left[\frac{m}{s^2}\right], -\frac{k_L}{m}\dot{\mathrm{y}}|\dot{\mathrm{y}}|\left[\frac{m}{s^2}\right], -g - \frac{k_L}{m}\dot{\mathrm{z}}|\dot{\mathrm{z}}|\left[\frac{m}{s^2}\right], \frac{I_{yy}-I_{zz}}{I_{zz}}\dot{\vartheta}\dot{\psi} - k_A\dot{\varphi}|\dot{\varphi}|\left[\frac{\mathrm{rad}}{s^2}\right], \frac{I_{zz}-I_{xx}}{I_{yy}}\dot{\varphi}\dot{\psi} - k_A\dot{\vartheta}|\dot{\vartheta}|\left[\frac{\mathrm{rad}}{s^2}\right], \frac{I_{xx}-I_{yy}}{I_{zz}}\dot{\varphi}\dot{\vartheta} - k_A\dot{\psi}|\dot{\psi}|\left[\frac{\mathrm{rad}}{s^2}\right]\right]^\top$ where $k_L = 0.25$, $k_A = 0.03$, $m = 1.5\mathrm{kg}$, $g = 9.81\frac{m}{s^2}$, $I_{xx} = 0.0348\mathrm{kg} \cdot \mathrm{m}^2$, $I_{yy} = 0.0348\mathrm{kg} \cdot \mathrm{m}^2$, $I_{yy} = 0.0977\mathrm{kg} \cdot \mathrm{m}^2$, represent the linear and angular drag coefficients, mass, gravity, X-axis, Y-axis, and Z-axis inertia, respectively. Simulations were run for 100s with an update preformed every 0.01s to demonstrate the impact of the real-time adaptation. The settling time t_Δ was selected as 5s and the history stack was constructed using a sliding window of the previous 100 data points (1s of data) with the stack being updated every 5 new data points gathered or 0.05s. The tracking objective

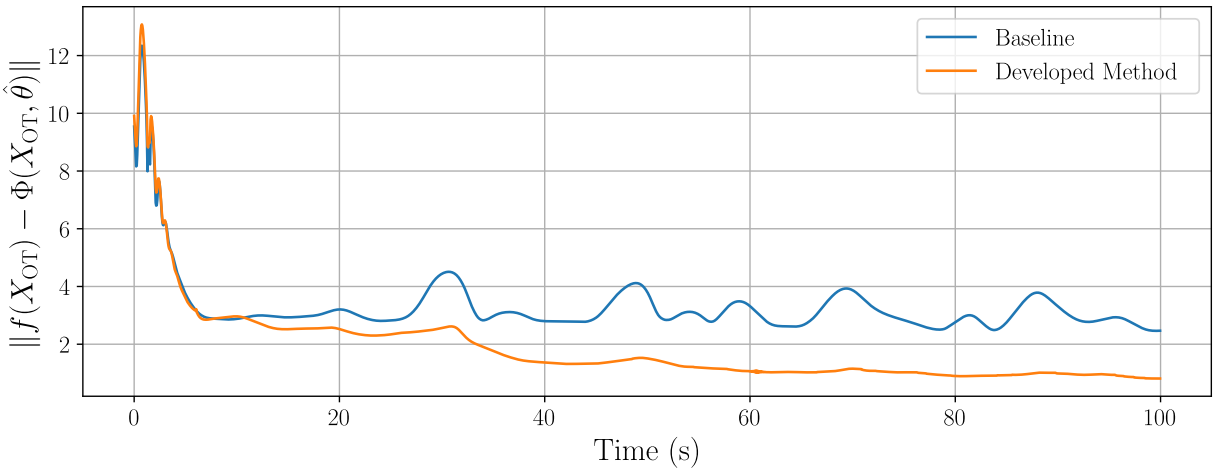

Figure 4-1. Function approximation errors evaluated at off trajectory points.

Table 4-1. Tracking and Function Approximation Metrics for the Developed Methods vs. Baseline Method

RMS	$\|e_R\|$ [deg]	$\|e_p\|$ [mm]	$\left\|\tilde{f}\right\|$	$\left\|\tilde{f}_{OT}\right\|$
Baseline	0.0524	8.48	0.163	3.03
CL-DNN	0.0624	9.33	0.097	1.03

was to track the desired trajectory $x_d = \left[2\sin(0.5t),\ 1.5\cos(0.7t),\ 1 + 0.8\sin(0.3t),\ 0.2\sin(0.4t),\ 0.15\cos(0.6t),\ 0.3\sin(0.2t)\right]^\top \in \mathbb{R}^6$ and tracking performance $e = \left[e_p^\top, e_R^\top\right]$ where $e_p \in \mathbb{R}^3$ and $e_R \in \mathbb{R}^3$ are the errors for the position and rotation metrics, respectively. The simulation is initialized at $x(0) = [0.1, -0.1, 0.05, 0.02, -0.02, 0.01]^\top$ and $\dot{x}(0) = [0, 0, 0, 0, 0, 0]$. The DNNs are composed of 4 layers, 12 neurons, and tanh activation functions. The weights of the DNN were randomly initialized from a uniform distribution $U(-0.25, 0.25)$. The gains were selected as $\alpha_1 = 15$, $\alpha_2 = 50$, $k_1 = 3$, $k_\Delta = 20$, $\beta = 0.1$, $\gamma_1 = 0.5$, $\gamma_2 = 0.001$, and $\Gamma(0) = 0.25$. The same randomly selected initial weights and control gains were used to demonstrate the performance of the adaptation under the same conditions.

The developed method improved function approximation while maintaining baseline-level control effort of 10.1 N and tracking performance. The minimal tracking difference is consistent with the function approximation capabilities of DNNs. The developed method's guaranteed parameter convergence led to a 40.54% reduction in the function

approximation error which is defined as $\tilde{f} \triangleq f(X) - \Phi(X, \hat{\theta})$. To evaluate off-trajectory performance, a test set of 100 inputs denoted by X_{OT} was constructed from a uniform distribution, $U(-1, 1)$. The function approximation error for this unseen data is denoted $\tilde{f}_{OT} \triangleq f(X_{OT}) - \Phi(X_{OT}, \hat{\theta})$. The mean function approximation error for the off-trajectory points shown in Figure 4-1 and in Table 4-1, which demonstrates a 65.75% improvement in off-trajectory function approximation compared to the baseline.

4.1.5 Conclusions

This work overcomes challenges in parameter convergence for continuous all-layer DNN adaptation, arising from the NIP nature of the inner layers by introducing a Lb-CL-DNN update law. A Lyapunov-based stability analysis guarantees ultimately bounded error convergence for both the tracking and the weight estimation errors. Simulations conducted on a 6-DOF rigid body vehicle demonstrated improvements in function approximation of 40.54% compared to the baseline method and off-trajectory simulations demonstrated 65.75% improvement for the developed method.

4.2 Continuous Learning with Offline Experience (CLOE): A Lyapunov-Based Deep Neural Network Approach for Real-Time Control and System Identification

This section introduces a framework which unifies offline and online approaches and presents the first method for continuous, all-layer DNN weight adaptation that is formally informed by offline-trained data. Compared with the previous section, this approach allows the data gathered offline to inform the adaptation and thus, does not require the use of an observer which allows for the use of discontinuous control laws and thus, allows for robust disturbance rejection. The developed adaptation law guarantees parameter estimation error convergence to a neighborhood about the origin provided the Jacobian evaluated using the offline experience meets a FE condition. A Lyapunov-based analysis ensures that the tracking error, parameter estimation error, and offline function approximation errors are ultimately bounded. Simulations

demonstrate 45.5% improvement in function approximation capabilities compared to the baseline.

4.2.1 Unknown System Dynamics, Objective Definitions, and Control Design

To illustrate the contribution of the CLOE approach, we consider a second-order, control-affine, nonlinear dynamic system described by

$$\ddot{x} = f(x, \dot{x}) + d(t) + u, \tag{4–22}$$

where $\ddot{x} \in \mathbb{R}^n$ denotes generalized acceleration, $x, \dot{x} \in \mathbb{R}^n$ denote the available generalized position and velocity, $f : \mathbb{R}^n \to \mathbb{R}^n$ denotes an unknown continuously differentiable function, $d(t) \in \mathbb{R}^n$ denotes an unknown time varying disturbance which can be upper bounded by $\|d(t)\| \leq \overline{d}$ where $\overline{d} \in \mathbb{R}_{>0}$ denotes a known constant, and $u \in \mathbb{R}^n$ denotes a control input. The objective of the developed controller and update law is to achieve identification of the unknown function $f(x, \dot{x})$ despite the presence of an unknown disturbance, $d(t)$, while simultaneously tracking a desired trajectory.

The subsequent development considers the concurrent tracking control and system identification problem. The parameter identification objective is quantified using the parameter estimation error defined as $\tilde{\theta} \triangleq \theta^* - \hat{\theta}$, where $\hat{\theta} \in \mathbb{R}^{\varkappa}$ represents the parameter estimates. The unknown function $f(x, \dot{x})$ can be modeled as

$$f(x, \dot{x}) = \Phi(X, \theta^*) + \varepsilon(X), \tag{4–23}$$

where the input to the DNN is denoted by $X \triangleq \left[x^\top, \dot{x}^\top\right]^\top \in \mathbb{R}^{2n}$, the DNN estimate which results in the best approximation of $f(x, \dot{x})$ is denoted by $\Phi(X, \theta^*) \in \mathbb{R}^n$, and $\varepsilon(X) \triangleq f(x, \dot{x}) - \Phi(X, \theta^*)$ is bounded by $\overline{\varepsilon}$ as $\|\varepsilon(x)\| \leq \overline{\varepsilon}$ for all $x \in \Omega$. To quantify the system identification objective, the parameter estimation error is defined as

$$\tilde{\theta} \triangleq \theta^* - \hat{\theta}, \tag{4–24}$$

where $\hat{\theta} \in \mathbb{R}^{\nu}$ represents the current parameter estimates. Using the definitions of $\Phi(X, \theta^*)$, $\Phi(X, \hat{\theta})$, and a first-order Taylor series approximation-based model in [20], the following relation is obtained

$$\Phi(X, \theta^*) - \Phi(X, \hat{\theta}) = \Phi'(X, \hat{\theta})\tilde{\theta} + R(X, \tilde{\theta}), \tag{4–25}$$

where $R(X, \tilde{\theta})$ represents the Lagrange remainder term.

Tracking Objective Definition and Control Law Development

The tracking objective is to ensure the system state x tracks a user-defined reference trajectory $x_d \in \mathbb{R}^n$ which is sufficiently smooth such that $\|x_d\| \leq \overline{x_d}$, $\|\dot{x}_d\| \leq \overline{\dot{x}_d}$, and $\|\ddot{x}_d\| \leq \overline{\ddot{x}_d}$, where $\overline{x_d}, \overline{\dot{x}_d}, \overline{\ddot{x}_d} \in \mathbb{R}_{>0}$ denote known constants. Taking the time derivative of (4–3), applying (4–22), and using the DNN representation for $f(x, \dot{x})$ from (4–23) yields the open-loop error system

$$\dot{r} = \Phi(X, \theta^*) + \varepsilon(X) + u - \ddot{x}_d + \alpha_1 \dot{e} + d(t). \tag{4–26}$$

The control input is designed as

$$u = \ddot{x}_d - \Phi(X, \hat{\theta}) - k_1 r - k_2 \text{sgn}(r) - e - \alpha_1 \dot{e}, \tag{4–27}$$

where $k_1, k_2 \in \mathbb{R}_{>0}$ are a user-defined control gains. Using the definitions of $\Phi(X, \theta^*)$ and $\Phi(X, \hat{\theta})$, substituting (4–25) and (4–27) into (4–26), and canceling cross terms yields

$$\dot{r} = \Phi'(X, \hat{\theta})\tilde{\theta} + R(X, \tilde{\theta}) + \varepsilon(X) - k_1 r - k_2 \text{sgn}(r) - e + d(t). \tag{4–28}$$

4.2.2 CLOE Prediction Error and Weight Adaptation Laws

4.2.2.1 Offline Experience Prediction Error Formulation

To incorporate knowledge gathered from offline experience, the adaptation law includes a prediction error-based penalty. This term discourages parameter updates that cause the function estimate to deviate from its known, offline-trained value at specific points within the training dataset. The motivation for the offline learning penalty is to

ensure that the output of the DNN in real-time continues to satisfy the value of the function at known points. The use of offline experience in the real-time adaptation law allows the DNN to adapt in unexplored regions and enables the subsequent parameter convergence guarantee. To quantify the offline-experience prediction error, $\widetilde{\mathcal{Y}} \in \mathbb{R}$ is defined as

$$\widetilde{\mathcal{Y}}(X_o, t) \triangleq \kappa_1 \int_{t_o}^{t} \left(f(X_o) - \widehat{\mathcal{Y}}(X_o, \tau) \right) d\tau, \tag{4–29}$$

where κ_1 is a user-selected constant scaling factor, $X_o \in \mathbb{R}^{2n}$ denotes a specific offline training point, $\widehat{\mathcal{Y}}(X_o, \tau)$ is the subsequently designed estimate of the offline trained function, and $f(X_o)$ is the known function which has been learned using offline-learning methods. Therefore, the offline learned function can be equivalently expressed using (4–23) as

$$f(X_o) = \Phi(X_o, \theta^*) + \varepsilon(X_o), \tag{4–30}$$

where $\varepsilon(X_o)$ is bounded as $\|\varepsilon(X_o)\| \leq \overline{\varepsilon_o}$.

Remark 4.3. The bound $\overline{\varepsilon_o}$ encompasses the function approximation error and the offline learning error. The assertion that the offline learning error is bounded has been widely adopted in learning-based control methods e.g., [111, 133].

The corresponding offline-learning estimate given by $\widehat{\mathcal{Y}}(X_o, t) \in \mathbb{R}^n$ is defined as

$$\widehat{\mathcal{Y}}(X_o, t) = \Phi(X_o, \hat{\theta}) + \kappa_2 \widetilde{\mathcal{Y}}(X_o, t), \tag{4–31}$$

where $\kappa_2 \in \mathbb{R}$ denotes a constant positive user-selected control gain. To facilitate the subsequent stability analysis it is desirable to obtain an analytical expression for $\widetilde{\mathcal{Y}}(X_o, t)$ which allows the offline-based prediction error to be explicitly related to the parameter estimation error of the DNN estimate. Using (4–25), (4–30), (4–31), and the fact that $\widetilde{\mathcal{Y}}(X_o, t)$ is differentiable, the time-derivative of (4–29) is given by $\dot{\widetilde{\mathcal{Y}}} = \kappa_1 \left(f(X_o) - \widehat{\mathcal{Y}}(X_o, t) \right)$. Equivalently, the analytical form can be written as

$$\dot{\widetilde{\mathcal{Y}}} = \kappa_1 \left(\Phi'(X_o, \hat{\theta})\tilde{\theta} + R(X_o, \tilde{\theta}) + \varepsilon(X_o) - \kappa_2 \widetilde{\mathcal{Y}}(X_o, t) \right). \tag{4–32}$$

4.2.2.2 CLOE-Lb-DNN Weight Adaptation Laws

To leverage *a priori* knowledge, the adaptation law is regularized using a penalty term based on the offline prediction error developed in Section 4.2.2.1. The Lyapunov-based real-time adaptation law for the CLOE-Lb-DNN is defined as

$$\dot{\hat{\theta}} \triangleq \text{proj}\left(\Gamma\left(\Phi'^{\top}(X,\hat{\theta})r + \sum_{i\in\mathcal{O}} \Phi'^{\top}(X_{o,i},\hat{\theta})\widetilde{\mathcal{Y}}_i - \gamma_2\hat{\theta} + \gamma_1 \sum_{i\in\mathcal{O}} \Phi'^{\top}(X_{o,i},\hat{\theta})\left(\dot{\widetilde{\mathcal{Y}}}_i + \kappa_1\kappa_2\widetilde{\mathcal{Y}}_i\right)\right)\right). \tag{4–33}$$

The set $\mathcal{O}$ denotes the set of points where the value of the function is known based on the offline experience. User-selected constant adaptation gains are denoted by $\gamma_1, \gamma_2 \in \mathbb{R}_{>0}$ and $\Gamma \in \mathbb{R}^{\varkappa\times\varkappa}$ is a positive definite time-varying least squares adaptation gain matrix where

$$\frac{d}{dt}\Gamma^{-1} = \begin{cases} -\beta\Gamma^{-1} + \gamma_1\kappa_1 \sum_{i\in\mathcal{O}} \Phi'^{\top}(X_{o,i},\hat{\theta})\Phi'(X_{o,i},\hat{\theta}), \\ \qquad \text{if } \lambda_{\Gamma,\min} < \lambda_{\min}(\Gamma) \text{ and } \lambda_{\max}(\Gamma) < \lambda_{\Gamma,\max} \\ 0, \qquad \text{otherwise} \end{cases} \tag{4–34}$$

where β represents a user selected forgetting factor $\beta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, and $\lambda_{\Gamma,\min}$, $\lambda_{\Gamma,\max}$ are user-selected bounds for the minimum and maximum eigenvalues of Γ, respectively. The adaptation gain in (4–34) is initialized to be PD, and it can be shown that $\Gamma(t)$ remains PD for all $t \in \mathbb{R}_{\geq 0}$ [90].

The analytical form of the offline experience prediction error given by (4–32) can be used in (4–33) to obtain the equivalant analytical form of the update law given as

$$\begin{aligned} \dot{\hat{\theta}} = \text{proj}\Bigg(&\Gamma\left(\Phi'^{\top}(X,\hat{\theta})r + \sum_{i\in\mathcal{O}} \Phi'^{\top}(X_{o,i},\hat{\theta})\widetilde{\mathcal{Y}}_i - \gamma_2\hat{\theta}\right) \\ &+ \gamma_1 \sum_{i\in\mathcal{O}} \Phi'^{\top}(X_{o,i},\hat{\theta})\kappa_1\left(R(X_{o,i},\tilde{\theta}) + \varepsilon(X_{o,i})\right) + \gamma_1 \sum_{i\in\mathcal{O}} \Phi'^{\top}(X_{o,i},\hat{\theta})\kappa_1\Phi'(X_{o,i},\hat{\theta})\tilde{\theta}. \end{aligned} \tag{4–35}$$

4.2.3 Stability Analysis

Let $\zeta \triangleq \left[e^\top, r^\top, \sum_{i\in\mathcal{O}} \widetilde{\mathcal{Y}}_i^\top, \tilde{\theta}^\top\right] \in \mathbb{R}^\psi$ denote the concatenated state, where $\psi \triangleq 3n + \varkappa$. Using (4–34), $\frac{d}{dt}\Gamma^{-1}$ can be bounded as

$$\frac{d}{dt}\Gamma^{-1} \leq \gamma_1\kappa_1 \left(\sum_{i\in\mathcal{O}} \widehat{\Phi}'^\top(X_{o,i},\hat{\theta})\widehat{\Phi}'(X_{o,i},\hat{\theta})\right). \tag{4–36}$$

The Lagrange remainder in (4–25) is bounded as [113]

$$\|R(X,\tilde{\theta})\| \leq \rho_0(\|X\|)\left\|\tilde{\theta}\right\|^2, \tag{4–37}$$

where $\rho_0(\|X\|) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a strictly increasing quadratic polynomial. Similarly, using [113], the Jacobian of the DNN can be bounded as

$$\left\|\Phi'(X,\hat{\theta})\right\| \leq \rho_1\left(\|X\|\right), \tag{4–38}$$

where $\rho_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes a strictly-increasing function. Due to the fact that $R(X,\tilde{\theta})$ and $\varepsilon(x)$ are bounded, the following bounds can be established

$$\begin{aligned}
\left\|R(X,\tilde{\theta})\right\| &\leq \rho_a\left(\|X\|\right)\|\zeta\|^2 \\
\left\|R(X_{o,i},\tilde{\theta})\right\| &\leq \rho_b(\|X_{o,i}\|)\|\zeta\|^2 \\
\left\|\Phi'^\top(X_{o,i},\hat{\theta})R(X_{o,i},\tilde{\theta})\right\| &\leq \rho_c\left(\|X_{o,i}\|\right)\|\zeta\|^2 \\
\left\|\Phi'^\top(X_{o,i},\hat{\theta})\varepsilon(X_{o,i})\right\| &\leq \rho_d\left(\|X_{o,i}\|\right)
\end{aligned} \tag{4–39}$$

where $\rho_a, \rho_b, \rho_c, \rho_d : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denote strictly increasing functions. Let $\rho\left(\|\zeta\|\right) \triangleq \frac{1}{2}\rho_a^2\left(\|X\|\right)\|\zeta\|^2 + \frac{\kappa_1}{2}\sum_{i\in\mathcal{O}}\rho_b^2(\|X_{o,i}\|)\|\zeta\|^2 + \frac{\gamma_1}{2}\kappa_1\sum_{i\in\mathcal{O}}\rho_c^2\left(\|X_{o,i}\|\right)\|\zeta\|^2$, where $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes an invertible strictly-increasing function. Since the approximation capabilities of the DNN stated in (4–23) holds only on the compact domain Ω, the subsequent stability analysis requires ensuring $X(t) \in \Omega$ for all $t \in [t_0, \infty)$. This goal is achieved by demonstrating that ζ is constrained to a compact domain. To that end, consider the compact domain $\mathcal{D} \triangleq \left\{\varsigma \in \mathbb{R}^\psi : \|\varsigma\| < \mathcal{B}\right\}$ in which ζ is supposed

to lie. It follows that if $\|\zeta\| < \mathcal{B}$, then $\|X\| < (\alpha+2)\mathcal{B} + \overline{x}_d + \overline{\dot{x}_d}$. Therefore, if $\Omega \triangleq \left\{\varsigma \in \mathbb{R}^{2n} : \|\varsigma\| < (\alpha+2)\mathcal{B} + \overline{x}_d + \overline{\dot{x}_d}\right\}$, then $\zeta \in \mathcal{D}$ implies $X \in \Omega$. Furthermore, to ensure that arbitrary initial conditions are included, let user-selected constant $\mathcal{B}$ be selected as

$$\mathcal{B} \triangleq \rho^{-1}(\lambda - \lambda_d). \tag{4–40}$$

Let $\lambda \triangleq \min\left\{k_1 - 1, \alpha, \kappa_1\left(\kappa_2 - \frac{1}{2}\right), \frac{\gamma_1}{2}\kappa_1\lambda_{\min}\left(\sum_{i\in\mathcal{O}} \Phi'^{\top}(X_{o,i},\hat{\theta})\Phi'(X_{o,i},\hat{\theta})\right) + \frac{\gamma_2}{2} - \gamma_1\kappa_1\right\}$, $\lambda_d \in \mathbb{R}_{\geq 0}$ be the desired convergence rate, and the constant $\iota \in \mathbb{R}_{>0}$ be defined as $\iota \triangleq \frac{\gamma_1}{2}\kappa_1 \sum_{i\in\mathcal{O}} \left(\rho_d\left(\|X_{o,i}\|\right)\right)^2 + \frac{1}{2}\overline{\varepsilon}^2 + \frac{\kappa_1}{2}\sum_{i\in\mathcal{O}}\overline{\varepsilon_{o,i}} + \frac{\gamma_2}{2}\overline{\theta}^2$. Let the set $\mathcal{I} \subset \mathcal{D}$ be the set of stabilizing initial conditions defined as

$$\mathcal{I} \triangleq \left\{\varsigma \in \mathbb{R}^{\psi} : \|\zeta(t_0)\| < \sqrt{\rho^{-1}(\lambda - \lambda_d) - \frac{\iota}{\lambda_d}}\right\}.$$

To guarantee that the adaptive update law in (4–33) achieves the parameter identification objective given in (4–24), the generated data record needs to be sufficiently rich, i.e., the system needs to be excited on points in $\mathcal{O}$ according to the following assumption.

Assumption 4.2. There exists $\lambda_e > 0$ and there exists $t \geq t_0$ such that $\lambda_{\min}\left\{\sum_{i\in\mathcal{O}} \Phi'^{\top}(X_{o,i},\hat{\theta})\Phi'(X_{o,i},\hat{\theta})\right\} \geq \lambda_e$.

Remark 4.4. An important advantage of the developed method is that the excitation condition noted in Assumption 4.2 can be satisfied from inputs explored during offline training and does not have to be executed by the physical system as in [102]. This provides the practical advantage that the system is not required to operate through trajectories which are unsafe or detrimental to performance in order to achieve sufficient excitation. As a result, the theoretical guarantee of parameter convergence is established without compromising the operational safety or performance of the closed-loop system.

Theorem 4.2. *Let the gain conditions $\lambda_d > 0$ and $k_2 \geq \overline{d}$ be satisfied and $\|\zeta(0)\| \in \mathcal{I}$. For the Filippov regularization of the dynamical system in (4–22), the controller in*

(4–27), the offline prediction error in (4–29), and the adaptation law developed in (4–35) ensure the concatenated error vector ζ *is bounded in the sense that* $\|\zeta(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 e^{-\frac{\lambda_d}{\beta_2}(t-t_0)} + \frac{\beta_2 \iota}{\beta_1 \lambda_d}\left(1 - e^{-\frac{\lambda_d}{\beta_2}t}\right)}$*, for all* $t \in \mathbb{R}_{\geq 0}$.

Proof. Consider the candidate Lyapunov function $\mathcal{V} : \mathbb{R}^{\psi} \to \mathbb{R}_{\geq 0}$ which is a Lipzchitz continuous positive definite function defined as

$$\mathcal{V}(\zeta, t) \triangleq \frac{1}{2} r^{\top} r + \frac{1}{2} e^{\top} e + \frac{1}{2} \sum_{i \in \mathcal{O}} \left(\widetilde{\mathcal{Y}}_i^{\top} \widetilde{\mathcal{Y}}_i\right) + \frac{1}{2} \tilde{\theta}^{\top} \Gamma^{-1} \tilde{\theta}, \tag{4–41}$$

which satisfies the inequality

$$\beta_1 \|\zeta\|^2 \leq \mathcal{V}(\zeta, t) \leq \beta_2 \|\zeta\|^2 \tag{4–42}$$

where $\beta_1 \triangleq \min\left\{\frac{1}{2}, \frac{1}{\lambda_{\Gamma,\max}}\right\}$ and $\beta_2 \triangleq \max\left\{\frac{1}{2}, \frac{1}{\lambda_{\Gamma,\min}}\right\}$. Let $\dot{\zeta} = \mathcal{H}(\zeta, t)$, where $\mathcal{H}(\zeta, t) \in \mathbb{R}_{\geq 0} \times \mathbb{R}^{\psi}$ denotes the right-hand side of the closed-loop error signals. Using Filippov's theory of differential inclusions [114–117], the existence of solutions can be established for $\dot{\zeta} \in \mathsf{K}\left[\mathcal{H}(\zeta, t)\right](\zeta)$, where $\mathsf{K}[\mathcal{H}(\zeta, t)] \triangleq \bigcap\limits_{\delta > 0} \bigcap\limits_{\mu L = 0} \overline{co}\mathcal{H}(B(\zeta, \delta) \setminus L, t)$, the intersection over all sets L of Lebesgue measure zeros is denoted by $\bigcap\limits_{\mu L = 0}$, $\overline{co}$ denotes convex closure, and $B(\zeta, \delta) = \left\{w \in R^{\psi} | \, \|\zeta - w\| < \delta\right\}$. Let $t \mapsto \zeta(t)$ be a maximal solution to the differential inclusion $\dot{\zeta} \overset{\text{a.e.}}{\in} \mathcal{H}(\zeta, t)$ with initial condition $\zeta(t_0) \in \mathcal{I}$ defined on the interval of existence $\mathcal{T}_e = [t_0, t_{max})$. Define $t_E \triangleq \inf\{t \in \mathcal{T}_e : \zeta(t) \notin \mathcal{D}\}$ and consider the time interval $\mathcal{T} \triangleq [t_0, t_E)$. By continuity and the fact that $\mathcal{D}$ is closed, we have $\zeta(t_E) \in \partial\mathcal{D}$ where $\partial\mathcal{D}$ denotes the boundary of $\mathcal{D}$. Since $\zeta(t_0) \in \mathcal{I} \subset \mathcal{D}$, then the time interval $\mathcal{T}$ exists such that $\|\zeta(t)\| < \mathcal{D}$, for all $t \in \mathcal{T}$. It follows that $X(t) \in \Omega$ for all $t \in \mathcal{T}$, therefore the universal function approximation property holds over this time interval. In the subsequent stability analysis, we analyze the convergence properties of the solutions and also establish that $\mathcal{T}$ can be extended to $\mathcal{T}_e$, which in turn can be extended to $[t_0, \infty)$. The time derivative of (4–41) exists along the trajectories of $\dot{\zeta} = \mathsf{K}[\mathcal{H}(\zeta, t)]$ almost everywhere (a.e.), i.e., for almost all $t \in \mathcal{T}$, and $\dot{\mathcal{V}}_L \overset{\text{a.e.}}{\in} \dot{\tilde{\mathcal{V}}}_L$ where $\dot{\tilde{\mathcal{V}}} = \nabla \mathcal{V}_L^{\top} \mathsf{K}[\mathcal{H}(\zeta, t)]^{\top}$. Applying (4–3), (4–28), and

canceling cross terms yields

$$\dot{\tilde{\mathcal{V}}}(\zeta,t) \subset r^\top\Big(\Phi'(X,\hat{\theta})\tilde{\theta} + R(x,\tilde{\theta}) + \varepsilon(X) - k_1 r - k_2 \mathsf{K}\left[\mathsf{sgn}(r)\right] + d(t)\Big) - e^\top \alpha e + \sum_{i\in\mathcal{O}} \widetilde{\mathcal{Y}}_i^\top \dot{\widetilde{\mathcal{Y}}}_i + \tilde{\theta}^\top \Gamma^{-1}\dot{\tilde{\theta}} + \frac{1}{2}\tilde{\theta}^\top\left(\frac{d}{dt}\Gamma^{-1}\right)\tilde{\theta}. \tag{4–43}$$

From (4–32), (4–35), (4–36), and (4–43), $\dot{\mathcal{V}}$ can be upper bounded as

$$\begin{aligned}\dot{\tilde{\mathcal{V}}}(\zeta,t) \overset{\text{a.e.}}{\leq}\ & -r^\top k_1 r - e^\top \alpha e - \sum_{i\in\mathcal{O}} \widetilde{\mathcal{Y}}_i^\top \kappa_2\kappa_1 \widetilde{\mathcal{Y}}_i - \tilde{\theta}^\top\gamma_2\tilde{\theta} - \tilde{\theta}^\top\frac{\gamma_1}{2}\kappa_1\Big(\sum_{i\in\mathcal{O}} \widehat{\Phi}'^\top(X_{o,i},\hat{\theta})\Phi'(X_{o,i},\hat{\theta})\tilde{\theta} \\ & + r^\top\Big(R(x,\tilde{\theta}) + \varepsilon(X) - k_2\mathsf{K}\left[\mathsf{sgn}(r)\right] + d(t)\Big) + \sum_{i\in\mathcal{O}} \widetilde{\mathcal{Y}}_i^\top\kappa_1\Big(R(X_{o,i},\tilde{\theta}) + \varepsilon(X_{o,i})\Big) \\ & + 2\sum_{i\in\mathcal{O}} \Phi'^\top(X_{o,i},\hat{\theta})\Big(R(X_{o,i},\tilde{\theta}) + \varepsilon(X_{o,i})\Big)\Big) + \tilde{\theta}^\top\gamma_2\theta^*.\end{aligned} \tag{4–44}$$

Therefore applying the gain condition $k_2 \geq \overline{d}$ and using the bounds in (4–39), $\dot{\mathcal{V}}$ can be upper bounded as

$$\dot{\tilde{\mathcal{V}}}(\zeta,t) \overset{\text{a.e.}}{\leq} -\left(\lambda - \rho(\|\zeta\|)\right)\|\zeta\|^2 + \iota, \tag{4–45}$$

when $\zeta \in \mathcal{D}$. Therefore, when ζ is initialized such that $\zeta(t_0) \in \mathcal{I}$, then it follows that $\lambda > \lambda_d + \rho\left(\sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2\iota}{\beta_1\lambda_d}}\right)$. Let $t \mapsto \zeta$ be a maximal solution to the differential inclusion $\dot{\zeta} \overset{\text{a.e.}}{\in} \mathcal{H}(\zeta,t)$, defined on its interval of existence and assume for contradiction that $t_E < \infty$. Applying $\lambda > \lambda_d + \rho\left(\sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2\iota}{\beta_1\lambda_d}}\right)$ yields

$$\dot{\tilde{\mathcal{V}}}(\zeta,t) \overset{\text{a.e.}}{\leq} -\lambda_d\|\zeta\|^2 + \iota \tag{4–46}$$

for $t \in \mathcal{T}$. Using (4–42) and the fact that $\dot{\mathcal{V}}(\zeta,t) \overset{\text{a.e.}}{\in} \dot{\tilde{\mathcal{V}}}(\zeta,t)$ yields $\dot{\mathcal{V}}(\zeta,t) \leq -\frac{\lambda_d}{\beta_2}\mathcal{V} + \iota$ for all $t \in \mathcal{T}$. Solving the differential inequality over $\mathcal{T}$ yields $\mathcal{V}(\zeta,t) \leq \mathcal{V}(\zeta(t_0))e^{-\frac{\lambda_d}{\beta_2}(t-t_0)} + \frac{\beta_2\iota}{\lambda_d}\left(1 - e^{-\frac{\lambda_d}{\beta_2}t}\right)$ for all $t \in \mathcal{T}$. Then, [130, Def. 4.6] can be invoked to conclude that ζ is bounded such that $\|\zeta(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 e^{-\frac{\lambda_d}{\beta_2}(t-t_0)} + \frac{\beta_2\iota}{\beta_1\lambda_d}\left(1 - e^{-\frac{\lambda_d}{\beta_2}t}\right)}$ for all $t \in \mathcal{T}$. Further taking the upper bound yields $\|\zeta(t)\| < \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2\iota}{\beta_1\lambda_d}}$ for $t \in \mathcal{T}$. Since $\zeta(t_0) \in \mathcal{I}$, we have $\|\zeta(t_0)\| < \sqrt{\rho^{-1}(\lambda - \lambda_d) - \frac{\iota}{\lambda_d}}$ for all $t \in \mathcal{T}$. Therefore, $\|\zeta(t)\| < \frac{\beta_2}{\beta_1}\rho^{-1}(\lambda - \lambda_d)$ for all $t \in \mathcal{T}$. Taking the limit as $t \to t_E$ yields

$\|\zeta(t_E)\| < \frac{\beta_2}{\beta_1}\rho^{-1}(\lambda - \lambda_d)$ which implies $\zeta(t_E) \in \text{int}(\mathcal{D})$, violating the assumption made for contradiction. Therefore, $t_E = \infty$ and $\zeta(t) \in \mathcal{D}$ for all $t \in \mathcal{T}_e$ and thus $\zeta(t)$ is bounded on $\mathcal{T}_e$.

It remains to be shown that $\mathcal{T}_e$ can be extended to $[t_0, \infty)$. Based on the preceding analysis, for any compact subinterval $\mathcal{J} \subset \mathcal{T}_e$ we have $\|\zeta(t)\| < \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 + \frac{\beta_2 \iota}{\beta_1 \lambda_d}}$. By definition, $(\zeta, t) \mapsto \mathcal{H}(\zeta, t)$ is locally bounded when ζ is bounded. Since $\zeta(t)$ is bounded on $\mathcal{J}$, the Filippov solution $t \mapsto \zeta(t)$ is precompact. Furthermore since the map $(\zeta, t) \mapsto \mathcal{H}(\zeta, t)$ is locally bounded it follows from [134, Remark 3.4] that $\bigcup_{t \in \mathcal{J}} \mathcal{H}(\zeta, t)$ is bounded for every compact subinterval $\mathcal{J} \subset \mathcal{T}_e$. Therefore [134, Lemma 3.3] is satisfied, guaranteeing that $t \mapsto \zeta(t)$ is complete, i.e., $\mathcal{T}_e = [t_0, \infty)$. Thus, all trajectories satisfy $\|\zeta(t)\| \leq \sqrt{\frac{\beta_2}{\beta_1}\|\zeta(t_0)\|^2 e^{-\frac{\lambda_d}{\beta_2}(t-t_0)} + \frac{\beta_2 \iota}{\beta_1 \lambda_d}\left(1 - e^{-\frac{\lambda_d}{\beta_2}t}\right)}$ for all $t \in [t_0, \infty)$. Hence, if $\zeta(t_0) \in \mathcal{I}$, then $\zeta(t) \in \mathcal{I} \subset \mathcal{D}$, and therefore $X \in \Omega$ for all $t \geq 0$. Using (4–41) and (4–46) implies $e, r, \sum_{i \in \mathcal{O}} \widetilde{\mathcal{Y}}_i, \tilde{\theta} \in \mathcal{L}_\infty$, and thus $\widetilde{\mathcal{Y}} \in \mathcal{L}_\infty$. The fact that $x_d, \dot{x}_d, \ddot{x}_d, e, r \in \mathcal{L}_\infty$ implies $x, \dot{x} \in \mathcal{L}_\infty$. Due to the use of the projection operator, $\hat{\theta} \in \mathcal{L}_\infty$. Using the fact that X_o are user selected points from a bounded training set, and $\hat{\theta}, \widetilde{\mathcal{Y}} \in \mathcal{L}_\infty$ implies $\Phi(X_o, \hat{\theta})$ is also bounded, and therefore $\widehat{\mathcal{Y}}$ is bounded. Using the fact that $e, r, x_d, \dot{x}_d, \ddot{x}_d$ and $\hat{\theta} \in \mathcal{L}_\infty$ implies $X \in \mathcal{L}_\infty$, and thus $\Phi(X, \hat{\theta})$ is bounded, and therefore u is bounded. □

4.2.4 Simulations

To validate the performance of the developed method, simulations were conducted on a nonlinear two-dimensional Duffing oscillator system. The total dynamics $f(x, \dot{x}) = [f_{x1}, f_{x2}]^\top$ are defined as the sum of linear and nonlinear components $f_{xi} \triangleq f_{\text{lin},i} + f_{\text{nl},i}$ for each dimension $i \in \{1, 2\}$ where the linear and nonlinear components are given by

$$f_{\text{lin},i}(x_i, \dot{x}_i) = -0.35x_i - 0.5\dot{x}_i,$$
$$f_{\text{nl},i}(x_i, \dot{x}_i) = -0.015x_i^3 - 0.35\dot{x}_i^2. \tag{4–47}$$

This composition is motivated to showcase a scenario where offline data may be restricted to a subspace of the operating region (e.g., data collected from an air foil in a

wind tunnel that has confined operating conditions). The tracking objective is designed to force the system to track the desired trajectory $x_d(t) \triangleq \begin{bmatrix} 8cos(\frac{\pi}{16}t) \\ 8sin(\frac{\pi}{16}t) \end{bmatrix}$, which leads to conditions where nonlinear dynamics are more prevalent. To evaluate robustness, simulations were run both with and without additive Gaussian noise $d(t) \sim \mathcal{N}(0, 1)$.

4.2.4.1 Generation of Offline Experience Data

To analyze the method's performance under realistic availability of data, we simulate a scenario where the offline experience is limited to a low-energy region which mimics practical constraints where the data is only available from a safe or restricted region. To generate the offline experience traditional offline machine learning methods were used. However, the developed method is agnostic to the means by which the offline experience is obtained. An offline training dataset containing 20,000 samples was generated by uniformly sampling the state variables $(x_i, \dot{x}_i)$ for $i \in \{1, 2\}$ from the restricted domain $X_o \in [0, 0.25]$. Sampling within this domain ensures that the nonlinearities are not the dominating dynamics and result in less than 12% of the total. The corresponding true function values were computed using (4–47). Prior to training, input data was scaled to a range of [0,1], and output data was scaled to [–1,1]. The pre-processed data was subsequently partitioned into training and validation sets with an 85% to 15% split.

The offline DNN model employs a fully-connected feedforward architecture consisting of three hidden layers with 128, 128, and 64 neurons, respectively where ReLU activation functions were applied to all hidden layers. The output layer was composed of 2 neurons with a linear activation function. The model was trained using the Adam optimizer and the MSE as the loss function. The mean absolute error (MAE) was concurrently monitored with the values reported in Table 4-2. Training utilized a mini-batch size of 32 samples and the Adam optimizer's default learning rate (0.001). A learning rate reduction strategy was implemented with a patience of 50 epochs. If no improvement in validation loss was observed, the learning rate would be reduced by a factor of

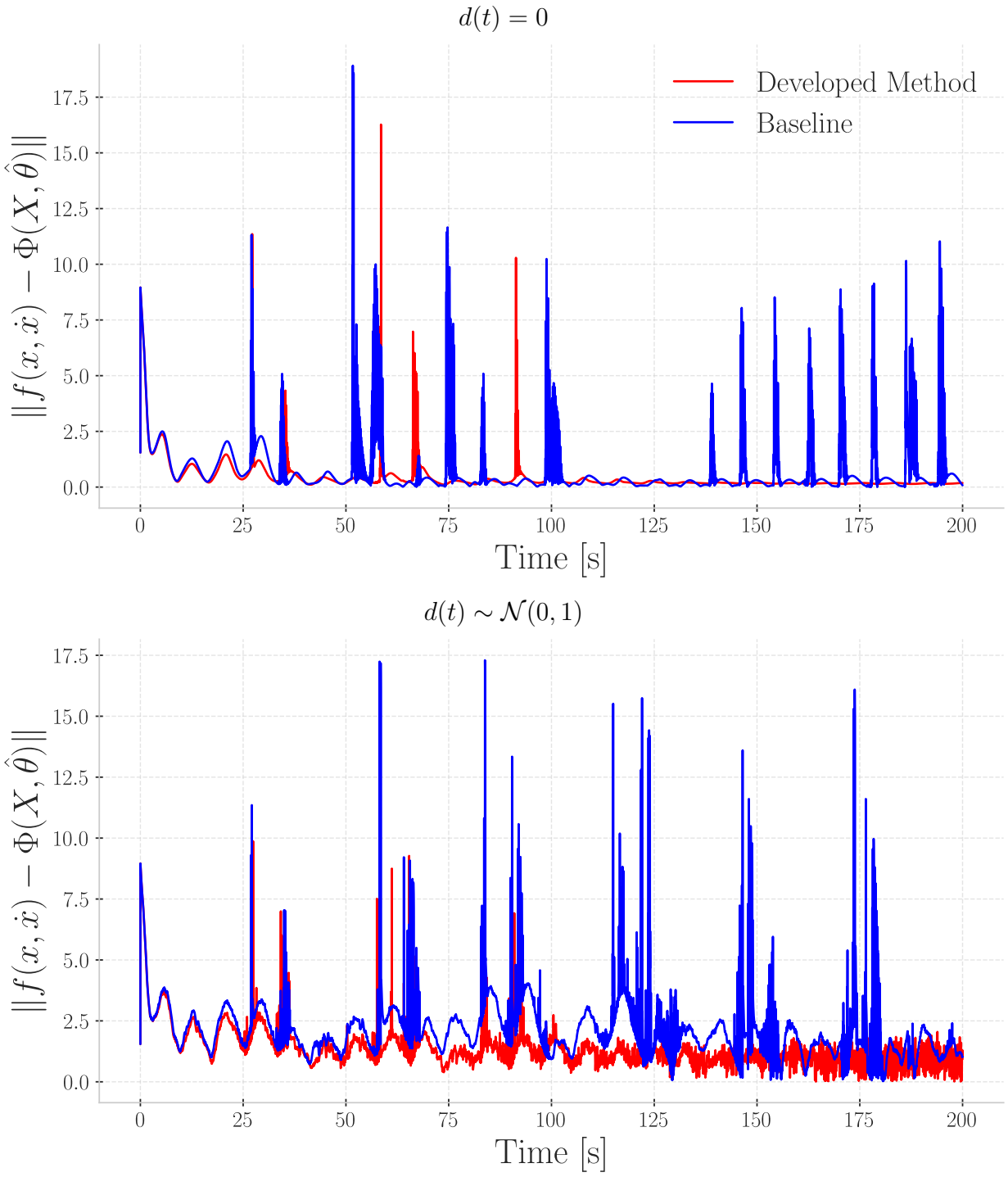

Figure 4-2. Function approximation performance compared to learning without offline experience.

Table 4-2. Evaluation of Offline Training Results on Training and Test Sets

	Train	Test
MSE	7×10^{-6}	8×10^{-6}
MAE	2.175×10^{-3}	2.214×10^{-3}
RMSE	2.705×10^{-3}	2.765×10^{-3}
# Epochs	237	-

0.5, with a minimum allowable learning rate of 1×10^{-7}. Training ceased via early stopping if the validation loss failed to improve by a minimum of 1×10^{-5} for 200 consecutive epochs. Upon stopping, the model weights corresponding to the epoch with the best validation loss were automatically restored.

The final offline experience dataset, $\mathcal{O}$, for subsequent online adaptation was constructed using a K-Means diversity sampling method to select 50 representative samples from the original training domain. The input states and their corresponding predicted values from the offline-trained model were saved and subsequently used as the offline experience for the real-time adaptation of a DNN.

4.2.4.2 Online Adaptation and Results

Simulations were run for 200s with a time step of $t = 0.01$s to track the desired trajectory, which operates outside the offline experience dataset domain, X_o. A baseline comparison was performed with an adaptation law which does not leverage offline experience (i.e., $\kappa_1, \kappa_2, \beta, \gamma_1 = 0$) . The states were initialized at $x(0) = [7.0472, -0.5236]^\top$ and $\dot{x}(0) = [0, 0]^\top$ [rad/s]. The DNN that adapts in real-time, and is responsible for achieving the control and identification objective, was constructed with 4 layers, with 2 neurons in each layer, and tanh activation functions, with weights initialized using Xavier initialization [131]. The gains for the respective controllers and adaptation laws were selected as $k_1 = 10$, $\alpha = 1$, $\gamma_2 = 0.0001$, and $\Gamma(0) = 1$, for both the developed and baseline methods. When disturbances were considered $k_2 = 1$, otherwise $k_2 = 0$. The developed method utilized the additional gains $\beta = 0.1$, $\kappa_1 = 0.0001$, $\kappa_2 = 1$, and

Table 4-3. RMS Tracking, Control Effort, and Function Approximation Metrics for the Developed vs. Baseline Method.

RMS Values	$\|e\|$[rad]	$\|\tau\|$[N]	$\left\|\widetilde{\mathcal{Y}}\right\|$[mN]	$\left\|\tilde{f}\right\|$[N]
$d(t) = 0$				
Baseline Method	0.0728	6.057	-	1.189
Developed Method	0.0701	5.979	30.02	0.6476
$d(t) \sim \mathcal{N}(0, 1)$				
Baseline Method	0.0841	6.033	-	1.776
Developed Method	0.0671	6.038	27.95	1.117

$\gamma_1 = 0.01$. The offline experience updates to account for the new $\hat{\theta}$ parameters every 5 time steps or 0.05 seconds.

The results presented in Table 4-3 demonstrate that the most notable improvement lies in the function approximation error $\tilde{f} \triangleq \left\|f(x, \dot{x}) - \Phi(X, \hat{\theta})\right\|$. Qualitatively, as demonstrated in Figure 4-2, the developed method achieves improved performance compared to the baseline with fewer and less severe oscillations during steady state operation. Quantitatively, the developed method demonstrates a 45.5% improvement compared to the baseline in the absence of disturbances, and a 37.1% improvement in the presence of Gaussian noise.

4.2.5 Conclusions

A CLOE-based adaptation law was developed to guide online adaptation of a Lb-DNN. The developed method results in simultaneous system identification and control where parameter estimation convergence is facilitated by excitation from offline-explored trajectories. The practical implication of using offline-explored trajectories is that exploration can be performed in the offline domain which balances exploration and exploitation and mitigates hardware damage and system behavior risks inherent in online exploration. A Lyapunov-based stability analysis results in ultimately bounded error convergence of the tracking, parameter estimation, and offline prediction errors. Simulation results demonstrated CLOE improved system identification by 45.5%

compared to the baseline in the absence of disturbances and by 37.1% in the presence of Gaussian noise.

CHAPTER 5
CONCLUSIONS AND FUTURE WORK

This dissertation has contributed to the development of physics-informed adaptation and guaranteed identifiability in adaptive control frameworks. While traditional deep learning methods prioritize function approximation at the expense of physical consistency and stability, the research presented in this dissertation demonstrates that using *a priori* structural knowledge with Lyapunov-based adaptation yields improved tracking and identification metrics. The technical contributions of this dissertation are unified through leveraging additional system information and mathematical identifiability guarantees.

5.1 Conclusions

Chapter 2 leverages the known structure of Euler-Lagrange systems to transition from black-box estimation to a physics-informed architecture. The physics-informed design was initially leveraged in a combined learning architecture framework and demonstrated that the structuring process and combining different architectures resulted in performance gains. Subsequently, the structure of the physics-informed architecture also facilitates additional constraints to be incorporated including the skew-symmetric relationship between the inertia and Coriolis matrices in a regularization approach.

While Chapter 2 focuses on architecture designs to improve function approximation capabilities, Chapter 3 focuses on whether the ML model parameters can be uniquely determined and system identification can be guaranteed. By deriving generalized identifiability conditions for NIP DNN structures, this chapter establishes the formal requirements for parameter convergence and links them with the verifiable FE condition. This theoretical foundation allows the black-box DNN approximations to be transformed into a reliable system identification tool with guaranteed convergence.

Building on the foundations in Chapter 3, Chapter 4 develops CL and CLOE adaptation laws. By utilizing system dynamics as a regularization mechanism, the

developed frameworks achieved simultaneous system identification and control. The use of historical data in the adaptation law as a regularization method that mirrors the principles of physics-informed learning, providing the necessary excitation for parameter convergence without compromising real-time stability.

Ultimately, this dissertation develops frameworks which will bridge the gap between deep learning and control theory while providing a methodology for designing adaptive systems that are physically grounded and verifiable.

5.2 Future Work

Future work would include extending the physics-informed work to include additional constraints including conservation properties and positive definiteness of the inertia matrix, and using physics-informed adaptive control methods in PDE-based control.

5.2.1 Positive Definite Constraints

While Section 2.2 investigated how to incorporate a skew-symmetric penalty into the update law, another key property of Euler-Lagrange systems is the positive definiteness of the inertia matrix, $M(q)$. While current results do not guarantee the estimate of $M(q)$ is positive definite, future research should explore architectural constraints that enforce this property by construction.

Cholesky Decomposition Architectures

The decomposition $M(q) = L^\top(q)L(q)$ ensures that $M(q)$ is at least positive semi-definite. However, the use of this decomposition into a Lyapunov-based framework introduces significant analytical complexity. Specifically, the Taylor Series expansion of the product $\Phi^\top(X,\theta)\Phi(X,\theta)$ results in high-order cross-terms involving the function approximation error $\varepsilon(X)$ and Lagrange Remainder $R(X,\tilde{\theta})$.

Future approaches may have success in the use of strictly triangular architectures where the diagonal elements of $L(q)$ are constrained to be strictly positive (e.g.,

through activation function selection). This would allow the estimate to be strictly positive definite. However, the complexity within the Lyapunov-framework would remain. Additionally, the decomposition of the inertia matrix poses a major challenge for system identification. If we learn $L(q)$ instead of $M(q)$ the relationship between the physical parameters and network weights becomes more complex and may result in challenges in guaranteeing identifiability.

Control Barrier Function (CBF) on the Output Set

A promising direction is to treat the positive definiteness as a safety constraint using a control barrier function (CBF) approach. While traditional CBFs ensure that the system states $q(t)$ remain within the safe set, they could be reformulated to ensure that the estimated inertia matrix (i.e., the output of the DNN) remains within the strictly positive definite cone $\mathcal{C} = \{M \in \mathbb{R}^{n\times n} | M > \delta I_n\}$. Where the candidate barrier function is based on the minimum eigenvalue. This direction may raise questions with feasibility of the safe set particularly when balancing a tracking error and PD constraint.

Extension to Manifold-Valued Learning

Beyond the inertia matrix, many robotic properties lie on specific Lie Groups (e.g., $SO(3)$ or $SE(3)$). A promising future direction is the development of Geometric Physics-Informed Neural Networks for Adaptive control. Rather than treating the weights as Euclidean vectors, future adaptation laws could be formulated as the gradient flow along the manifold of positive definite matrices ensuring that the weight update moves along a path which preserves the structural properties of $M(q)$. This would require bridging the gap between Riemannian Geometry, Lyapunov Stability, and Real-time Learning with adaptive control techniques.

Implications in other areas

While ensuring positive definiteness is a physical requirement for learning the inertia matrix, the ability to enforce such constraints has significant implications for other

areas of control design. Specifically, learning approaches with guaranteed positive definite outputs may be useful for areas such as learning Lyapunov functions or CBFs.

5.2.2 Extension to Partial Differential Equations (PDEs)

As established in Chapter 1, existing physics-informed architectures are predominately limited by their offline nature and often decoupled from control objectives. The focus of this dissertation has been linking concepts of physics-informed learning with adaptive control methods however the focus has predominately been ordinary differential equation systems. Future work should investigate physics-informed PDE-based control which is guided by a Lyapunov-based stability framework. To achieve this, the Lyapunov-based analysis must account for the infinite-dimensional state space of PDEs and the designed controller must remain robust to function approximation errors. Due to the infinite dimensionality of PDEs, control is often restricted to boundary or sparse interior actuation. To ensure physical feasibility and to facilitate many of the analysis methods, future control laws should maintain smoothness and avoid the use of discontinuous sliding mode terms (i.e., $\text{sgn}(\cdot)$). Furthermore, given that UUB or bounded results can manifest as instabilities in an infinite-dimensional context, pursuing asymptotic stability is essential. This objective requires changes to Lyapunov analysis to ensure that the disturbances from real-time learning and PDE approximations vanish over time.

Lyapunov-Based PDE Analysis

As the real-time learning framework is extended from lumped-parameter systems to PDEs, the Lyapunov-based stability analysis must be adjusted to account for the infinite-dimensional spaces. One approach involves defining a Lyapunov candidate which represents the total energy of the distributed system. While these energy-based methods are traditionally suitable for stabilization tasks, adapting them for trajectory tracking remains a significant challenge. For systems lacking a natural energy interpretation, the PDE backstepping framework offers a principled alternative. This

method utilizes a Volterra-type integral transformation to map an uncertain, unstable PDE into a stable target system.

Operator Learning for Infinite-Dimensional Mapping

While standard architectures presented in this dissertation are designed to approximate functions between finite-dimensional Euclidean spaces, Operator Networks such as DeepONets and Fourier Neural Operators are capable of learning the underlying operators that map between infinite-dimensional function spaces. This shift offers the transformative potential required for spatially distributed parameters; however, previous results in operator learning have been primarily limited to offline training. For these architectures to be effective in autonomous and safety-critical PDE systems, real-time adaptation is required to handle time-varying environments and parameter drift that static, offline-trained models cannot capture. In practical applications like flexible-link robotics or fluid-structure interactions, the physical properties of the medium may evolve, necessitating a "living model" that updates its internal representation online. Furthermore, the discretization-invariant nature of operator networks allows a real-time adaptive controller to remain robust even as sensor configurations change or mobile agents move across different spatial coordinates. This evolution from offline estimation to real-time operator adaptation represents a shift from data-driven curve fitting toward the online identification of the fundamental physics of the operators themselves, providing a pathway for certifiably stable control of infinite-dimensional systems.

Physics-informed Learning in PDEs

While physics-informed learning has primarily been driven by the desire to solve PDEs in offline settings, this dissertation has focused on the intersection of online system identification and real-time adaptive control using physics-informed methods. There is a significant opportunity to extend the identification methods presented here to the PDE domain. By leverage the theoretical foundations of all-layer adaptation and identifiability established in the previous chapters, future research would move toward

identifying the underlying operators of PDEs in real-time. Furthermore, a PDE-based analysis offers an opportunity to refine the adaptive control results by incorporating more traditional physics-informed techniques to enhance the fidelity of the learned models in high-dimensional state spaces with applications in flexible structure or fluid-structure interaction.

REFERENCES

[1] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, pp. 422–440, 2021.

[2] R. Hart, O. Patil, E. Griffis, and W. E. Dixon, "Deep Lyapunov-based physics-informed neural networks (DeLb-PINN) for adaptive control design," in *Proc. IEEE Conf. Decis. Control*, pp. 1511–1516, 2023.

[3] S. S. Ge, C. C. Hang, and T. Zhang, "Adaptive neural network control of nonlinear systems by state and output feedback," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, pp. 818–828, December 1999.

[4] S. Ge, C. Hang, and T. Zhang, "Nonlinear adaptive control using neural networks and its application to CSTR systems," *J. Process Control*, vol. 9, no. 4, pp. 313–323, 1999.

[5] A. Yesildirek and F. L. Lewis, "A neural net controller for robots with Hebbian tuning and guaranteed tracking," in *Proc. Am. Control Conf.*, (Seattle, WA), pp. 2784–2789, 1995.

[6] H. Gomi and M. Kawato, "Neural network control for a closed-loop system using feedback-error-learning," *Neural Networks*, vol. 6, no. 7, pp. 933–946, 1993.

[7] J. Noriega and H. Wang, "A direct adaptive neural-network control for unknown nonlinear systems and its application," *IEEE Transactions on Neural Networks*, vol. 9, pp. 27–34, 1998.

[8] F. Chen and H. Khalil, "Adaptive control of a class of nonlinear discrete-time systemsusing neural networks," *IEEE Trans. Autom. Control*, vol. 40, no. 5, pp. 791–801, 1995.

[9] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 703–715, 1995.

[10] F. L. Lewis, A. Yegildirek, and K. Liu, "Multilayer neural-net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 7, pp. 388–399, Mar. 1996.

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

[12] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Int. Conf. Mach. Learn.*, pp. 1126–1135, PMLR, 2017.

[13] M. OConnell, G. Shi, X. Shi, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, "Neural-fly enables rapid learning for agile flight in strong winds," *Sci. Robotics*, vol. 7, no. 66, p. 6597, 2022.

[14] S. Sahoo, C. Lampert, and G. Martius, "Learning equations for extrapolation and control," in *Proc. Int. Conf. Mach. Learn.*, pp. 4442–4450, PMLR, July 2018.

[15] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," in *Proc. IEEE Int. Symp. Intell. Control*, (Chicago, Illinois), pp. 225–231, 1993.

[16] F. L. Lewis, A. Yesildirek, and K. Liu, "Neural net robot controller: structure and stability proofs," in *Proc. IEEE Conf. Decis. Control*, pp. 2785–2791, Dec. 1993.

[17] E. Griffis, O. Patil, Z. Bell, and W. E. Dixon, "Lyapunov-based long short-term memory (Lb-LSTM) neural network-based control," *IEEE Control Syst. Lett.*, vol. 7, pp. 2976–2981, 2023.

[18] G. Joshi, J. Virdi, and G. Chowdhary, "Design and flight evaluation of deep model reference adaptive controller," in *AIAA Scitech 2020 Forum*, p. 1336, 2020.

[19] 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.

[20] 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.

[21] 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*, pp. 3487–3492, 2022.

[22] O. S. Patil, E. J. Griffis, W. A. Makumi, and W. E. Dixon, "Composite adaptive Lyapunov-based deep neural network (Lb-DNN) controller," *arXiv preprint arXiv:1306.3432*, 2023.

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

[24] R. Hart, E. Griffis, O. Patil, and W. E. Dixon, "Lyapunov-based physics-informed long short-term memory (LSTM) neural network-based adaptive control," *IEEE Control Syst. Lett.*, vol. 8, pp. 13–18, 2024.

[25] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, 2019.

[26] M. Lutter, C. Ritter, and J. Peters, "Deep Lagrangian networks: Using physics as model prior for deep learning," in *Int. Conf. Learn. Represent.*, 2019.

[27] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," *arXiv preprint arXiv:2003.04630*, 2020.

[28] S. Kim, I. Kim, J. Lee, and S. Lee, "Knowledge integration into deep learning in dynamical systems: an overview and taxonomy," *J. Mech. Sci. Technol.*, vol. 35, pp. 1331–1342, Apr. 2021.

[29] T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, and P. Gentine, "Enforcing analytic constraints in neural-networks emulating physical systems," 2021.

[30] T. Verdonck, B. Baesens, M. Óskarsdóttir, and S. vanden Broucke, "Special issue on feature engineering editorial," *Mach. Learn.*, vol. 113, pp. 3917–3928, July 2024.

[31] J. Ling, R. Jones, and J. Templeton, "Machine learning strategies for systems with invariance properties," *J. Comput. Phys.*, vol. 318, pp. 22–35, Aug. 2016.

[32] J. Brandstetter, M. Welling, and D. E. Worrall, "Lie point symmetry data augmentation for neural PDE solvers," in *Int. Conf. Mach. Learn. (ICML)*, pp. 2241–2256, June 2022.

[33] L. Lu, M. Dao, P. Kumar, U. Ramamurty, G. E. Karniadakis, and S. Suresh, "Extraction of mechanical properties of materials through deep learning from instrumented indentation," vol. 117, pp. 7052–7062, Mar. 2020.

[34] P. Simard, D. Steinkraus, and J. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Int. Conf. Doc. Anal. Recognit.*, pp. 958–963, Aug. 2003.

[35] S. A. Niederer, M. S. Sacks, M. Girolami, and K. Willcox, "Scaling digital twins from the artisanal to the industrial," *Nat. Comput. Sci.*, vol. 1, pp. 313–320, May 2021.

[36] C. Rodwell and P. Tallapragada, "Physics-informed reinforcement learning for motion control of a fish-like swimming robot," *Sci. Rep.*, vol. 13, p. 10754, July 2023.

[37] Y. Lu, M. Rajora, P. Zou, and S. Y. Liang, "Physics-embedded machine learning: Case study with electrochemical micro-machining," *Machines*, vol. 5, p. 4, Mar. 2017.

[38] Y. Wang, Q. Gao, and M. Pajic, "Learning monotone dynamics by neural networks," in *Am. Control Conf.*, pp. 1485–1490, 2022.

[39] M. Mattheakis, P. Protopapas, D. Sondak, M. Di Giovanni, and E. Kaxiras, "Physical symmetries embedded in neural networks," *arXiv preprint arXiv:1904.08991*, Jan. 2020.

[40] R. G. Nascimento and F. A. C. Viana, “Fleet prognosis with physics-informed recurrent neural networks,” *arXiv*, Jan. 2019. arXiv:1901.05512.

[41] M. Sadoughi and C. Hu, “Physics-based convolutional neural network for fault diagnosis of rolling element bearings,” *IEEE Sens. J.*, vol. 19, pp. 4181–4192, June 2019.

[42] R. Wang, R. Walters, and R. Yu, “Incorporating symmetry into deep dynamics models for improved generalization,” in *Int. Conf. Learn. Represent.*, 2021.

[43] S. Seo and Y. Liu, “Differentiable physics-informed graph networks,” *arXiv*, Feb. 2019. arXiv:1902.02950.

[44] S. K. Singh, R. Yang, A. Behjat, R. Rai, S. Chowdhury, and I. Matei, “PI-LSTM: Physics-infused long short-term memory network,” in *IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, pp. 34–41, Dec. 2019.

[45] Y. Lebedev, M. W. Lee, and A. Zare, “Construction of a fluid flow field from discrete point data using machine learning,” in *AIAA Scitech 2023 Forum*, p. 1186, 2023.

[46] Y. Zheng, C. Hu, X. Wang, and Z. Wu, “Physics-informed recurrent neural network modeling for predictive control of nonlinear processes,” *J. Process Control*, vol. 128, pp. 2804–2818, Aug. 2023.

[47] E. A. Antonelo, E. Camponogara, L. O. Seman, E. R. de Souza, J. P. Jordanou, and J. F. Hubner, “Physics-informed neural nets for control of dynamical systems,” 2022.

[48] R. Nathasarma and B. Roy, “Physics-informed long-short-term memory neural network for parameters estimation of nonlinear systems,” *IEEE T. Ind. Appl.*, pp. 1–9, 2023.

[49] Y. D. Zhong, B. Dey, and A. Chakraborty, “Symplectic ODE-net: Learning hamiltonian dynamics with control,” in *Int. Conf. Learn. Represent.*, 2020.

[50] R. Götte and J. Timmermann, “Composed physics- and data-driven system identification for non-autonomous systems in control engineering,” in *Int. Conf. Artif. Intell. Robot. Control*, pp. 67–76, May 2022.

[51] H. Lee and I. S. Kang, “Neural algorithm for solving differential equations,” *J. Comput. Phys.*, vol. 91, pp. 110–131, Nov. 1990.

[52] A. J. Meade and A. A. Fernandez, “Solution of nonlinear ordinary differential equations by feedforward neural networks,” *Math. Comput. Model.*, vol. 20, pp. 19–44, Nov. 1994.

[53] I. Lagaris, A. Likas, and D. Fotiadis, “Artificial neural networks for solving ordinary and partial differential equations,” *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 987–1000, 1998.

[54] I. Lagaris, A. Likas, and D. Papageorgiou, “Neural-network methods for boundary value problems with irregular boundaries,” *IEEE Trans. Neural Netw.*, vol. 11, pp. 1041–1049, Sept. 2000.

[55] H. Sheng and C. Yang, “PFNN: A penalty-free neural network method for solving a class of second-order boundary-value problems on complex geometries,” *J. Comput. Phys.*, vol. 428, p. 110085, 2021.

[56] K. McFall and J. Mahan, “Artificial neural network method for solution of boundary value problems with exact satisfaction of arbitrary boundary conditions,” *IEEE Trans. Neural Netw.*, vol. 20, no. 8, pp. 1221–1233, 2009.

[57] S. Dong and N. Ni, “A method for representing periodic functions and enforcing exactly periodic boundary conditions with deep neural networks,” *Journal of Computational Physics*, vol. 435, p. 110242, June 2021.

[58] R. Shekari Beidokhti and A. Malek, “Solving initial-boundary value problems for systems of partial differential equations using neural networks and optimization techniques,” *J. Frankl. Inst.*, vol. 346, pp. 898–913, Nov. 2009.

[59] M. Roehrl, T. Runkler, V. Brandtstetter, M. Tokic, and S. Obermayer, “Modeling system dynamics with physics-informed neural networks based on lagrangian mechanics,” *IFAC-PapersOnLine*, vol. 53, pp. 9195–9200, Jan. 2020.

[60] J. Willard, X. Jia, S. Xu, M. Steinbach, and V. Kumar, “Integrating physics-based modeling with machine learning: A survey,” *ArXiv*, Mar. 2020.

[61] Y. LeCun and Y. Bengio, “Convolutional networks for images, speech, and time series,” *Handb. Brain Theory Neural Netw.*, vol. 3361, no. 10, p. 1995, 1995.

[62] B. Anderson, T. S. Hy, and R. Kondor, “Cormorant: Covariant molecular neural networks,” in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019.

[63] I. Huh, E. Yang, S. J. Hwang, and J. Shin, “Time-reversal symmetric ode network,” in *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 19016–19027, 2020.

[64] S. Greydanus, M. Dzamba, and J. Yosinski, “Hamiltonian neural networks,” in *NeurIPS*, vol. 32, 2019.

[65] M. Finzi, K. A. Wang, and A. G. Wilson, “Simplifying Hamiltonian and Lagrangian neural networks via explicit constraints,” in *NeurIPS*, vol. 33, pp. 13880–13889, 2020.

[66] M. Lutter, K. Listmann, and J. Peters, “Deep Lagrangian networks for end-to-end learning of energy-based control for under-actuated systems,” in *IEEE Int. Conf. Intell. Robots Syst.*, pp. 7718–7725, 2019.

[67] Y. D. Zhong, B. Dey, and A. Chakraborty, “Dissipative symODEN: Encoding hamiltonian dynamics with dissipation and control into deep learning,” in *Int. Conf. Learn. Represent.*, 2020.

[68] Y. D. Zhong, B. Dey, and A. Chakraborty, “Extending lagrangian and hamiltonian neural networks with differentiable contact models,” *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 21910–21922, 2021.

[69] M. Lutter and J. Peters, “Combining physics and deep learning to learn continuous-time dynamics models,”

[70] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-based model predictive control: Toward safe learning in control,” *Annu. Rev. Control Robot. Auton. Syst.*, vol. 3, no. 1, pp. 269–296, 2020.

[71] J. Nicodemus, J. Kneifl, J. Fehr, and B. Unger, “Physics-informed neural networks-based model predictive control for multi-link manipulators,” in *Proc. Int. Conf. Math. Model.*, vol. 55, pp. 331–336, Jan. 2022.

[72] S. Sanyal and K. Roy, “RAMP-net: A robust adaptive MPC for quadrotors via physics-informed neural network,” in *IEEE Int. Conf. Robot. Autom.*, pp. 1019–1025, May 2023.

[73] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, “Provably safe and robust learning-based model predictive control,” *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[74] P. Bouffard, A. Aswani, and C. Tomlin, “Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results,” in *IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 279–284, 2012.

[75] X.-Y. Liu and J.-X. Wang, “Physics-informed dyna-style model-based deep reinforcement learning for dynamic control,” *Proc. R. Soc. A*, vol. 477, p. 20210618, Nov. 2021.

[76] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, “Model-based reinforcement learning: A survey,” *arXiv*, Mar. 2022. arXiv:2006.16712.

[77] A. Plaat, W. Kosters, and M. Preuss, “Deep model-based reinforcement learning for high-dimensional problems, a survey,” *arXiv*, Dec. 2020. arXiv:2008.05598.

[78] L. Di Natale, B. Svetozarevic, P. Heer, and C. N. Jones, “Physically consistent neural networks for building thermal modeling: Theory and analysis,” *Appl. Energy*, vol. 325, p. 119806, Nov. 2022.

[79] L. Di Natale, B. Svetozarevic, P. Heer, and C. N. Jones, "Near-optimal deep reinforcement learning policies from data for zone temperature control," in *IEEE Int. Conf. Control Autom.*, pp. 698–703, June 2022.

[80] S. Bhasin, K. Dupree, P. M. Patre, and W. E. Dixon, "Neural network control of a robot interacting with an uncertain viscoelastic environment," *IEEE Trans. Control Syst. Technol.*, vol. 19, no. 4, pp. 947–955, 2011.

[81] D. M. Le, O. S. Patil, C. Nino, and W. E. Dixon, "Accelerated gradient approach for neural network-based adaptive control of nonlinear systems," in *Proc. IEEE Conf. Decis. Control*, pp. 3475–3480, 2022.

[82] E. Griffis, D. Le, K. Stubbs, and W. E. Dixon, "Closed-loop deep neural network-based FES control for human limb tracking," in *Proc. IEEE Conf. Decis. Control*, pp. 1263–1268, 2021.

[83] E. Griffis, A. Isaly, D. M. Le, and W. E. Dixon, "Deep neural network-based adaptive FES-cycling control: Part II, a hybrid systems approach," in *Proc. IEEE Conf. Decis. Control*, pp. 3262–3267, 2022.

[84] T. X. Nghiem, J. Drgoňa, C. Jones, Z. Nagy, R. Schwan, B. Dey, A. Chakrabarty, S. Di Cairano, J. A. Paulson, A. Carron, M. N. Zeilinger, W. Shaw Cortez, and D. L. Vrabie, "Physics-informed machine learning for modeling and control of dynamical systems," in *Am. Control Conf. (ACC)*, pp. 3735–3750, May 2023.

[85] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *Int. J. Robot. Res.*, vol. 5, pp. 101–119, Sept. 1986.

[86] A. Pulido, K. Volle, K. Waters, Z. I. Bell, P. Ganesh, and J. Shin, "Uncertainty-aware guidance for target tracking subject to intermittent measurements using motion model learning," in *Proc. Am. Control Conf.*, pp. 3140–3145, 2025.

[87] R. S. Patel, S. Bhartiya, and R. D. Gudi, "Physics constrained learning in neural network based modeling," in *IFAC Symp. Dyn. Control Process Syst. Biosyst. (DYCOPS)*, vol. 55, pp. 79–85, Jan. 2022.

[88] J. Bongard and H. Lipson, "Automated reverse engineering of nonlinear dynamical systems," *Proc. Natl. Acad. Sci. USA*, vol. 104, pp. 9943–9948, June 2007.

[89] M. Schmidt and H. Lipson, "Distilling free-form natural laws from experimental data," *Science*, vol. 324, pp. 81–85, Apr. 2009.

[90] J. J. Slotine and W. Li, "Composite adaptive control of robot manipulators," *Automatica*, vol. 25, pp. 509–519, July 1989.

[91] G. Chowdhary, M. Mühlegg, J. How, and F. Holzapfel, "Concurrent learning adaptive model predictive control," in *Advances in Aerospace Guidance, Navigation and Control*, pp. 29–47, Springer Berlin Heidelberg, 2013.

[92] 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, pp. 1775–1787, Dec. 2019.

[93] Z. Bell, A. Parikh, J. Nezvadovitz, and W. E. Dixon, "Adaptive control of a surface marine craft with parameter identification using integral concurrent learning," in *Proc. IEEE Conf. Decis. Control*, pp. 389–394, 2016.

[94] C. Riano-Rios, R. Bevilacqua, and W. E. Dixon, "Differential drag-based multiple spacecraft maneuvering and on-line parameter estimation using integral concurrent learning," *Acta Astronautica*, vol. 174, pp. 189–203, Sept. 2020.

[95] H. M. Sweatland, O. S. Patil, and W. E. Dixon, "Adaptive deep neural network-based control barrier functions," *arXiv preprint arXiv:2406.14430*, 2025.

[96] Y. Pan and H. Yu, "Composite learning robot control with guaranteed parameter convergence," *Automatica*, vol. 89, pp. 415–419, Mar. 2018.

[97] S. Basyal, J. Ting, K. Mishra, and B. C. Allen, "Augmentation of a Lyapunov-based deep neural network controller with concurrent learning for control-affine nonlinear systems," *Proc. Am. Control Conf.*, pp. 2870–2875, 2024.

[98] 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.

[99] C. Cao, A. M. Annaswamy, and A. Kojic, "Parameter convergence in nonlinearly parameterized systems," *IEEE Trans. Autom. Control*, vol. 48, no. 3, pp. 397–412, 2003.

[100] L. Wang, R. Ortega, A. Bobtsov, J. G. Romero, and B. Yi, "Identifiability implies robust, globally exponentially convergent on-line parameter estimation: Application to model reference adaptive control," *arXiv preprint arXiv:2108.08436*, 2021.

[101] R. Ortega, A. Bobtsov, N. Nikolaev, and R. Costa-Castelló, "Parameter estimation of two classes of nonlinear systems with non-separable nonlinear parameterizations," *Automatica*, vol. 163, p. 111559, 2024.

[102] W. Makumi, O. S. Patil, and W. E. Dixon, "Lyapunov-based adaptive deep system identification for approximate dynamic programming," *Automatica*, vol. 180, p. 112462, 2025.

[103] R. G. Hart, O. S. Patil, Z. I. Bell, and W. E. Dixon, "System identification and control using lyapunov-based deep neural networks without persistent excitation: A concurrent learning approach," *arXiv preprint arXiv:2505.10678*, 2025.

[104] 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, pp. 73–82, Jan. 1987.

[105] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[106] 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.

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

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

[109] N. Fischer, R. Kamalapurkar, and W. E. Dixon, “LaSalle-Yoshizawa corollaries for nonsmooth systems,” *IEEE Trans. Autom. Control*, vol. 58, pp. 2333–2338, Sep. 2013.

[110] A. Pinkus, “Approximation theory of the MLP model in neural networks,” *Acta Numer.*, vol. 8, pp. 143–195, 1999.

[111] G. Shi, X. Shi, M. OConnell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, “Neural lander: Stable drone landing control using learned dynamics,” in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 9784–9790, 2019.

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

[113] O. S. Patil, B. C. Fallin, C. F. Nino, R. G. Hart, and W. E. Dixon, “Bounds on deep neural network partial derivatives with respect to parameters,” *arXiv preprint arXiv:2503.21007*, 2025.

[114] A. F. Filippov, “Differential equations with discontinuous right-hand side,” in *Fifteen papers on differential equations*, vol. 42 of *American Mathematical Society Translations - Series 2*, pp. 199–231, American Mathematical Society, 1964.

[115] A. F. Filippov, *Differential equations with discontinuous right-hand side*. Kluwer Academic Publishers, 1988.

[116] G. V. Smirnov, *Introduction to the theory of differential inclusions*. American Mathematical Society, 2002.

[117] J. P. Aubin and H. Frankowska, *Set-valued analysis*. Birkhäuser, 2008.

[118] D. M. Le, O. S. Patil, P. Amy, and W. E. Dixon, “Integral concurrent learning-based accelerated gradient adaptive control of uncertain euler-lagrange systems,” in *Proc. Am. Control Conf.*, pp. 806–811, June 2022.

[119] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[120] K. Kawaguchi and Y. Bengio, “Depth with nonlinearity creates no bad local minima in ResNets,” *Neural Netw.*, vol. 118, pp. 167–174, 2019.

[121] K. Kawaguchi, “Deep learning without poor local minima,” *NeurIPS*, vol. 29, 2016.

[122] H. Lu and K. Kawaguchi, “Depth creates no bad local minima,” *arXiv preprint arXiv:1702.08580*, 2017.

[123] S. Du, J. Lee, Y. Tian, A. Singh, and B. Poczos, “Gradient descent learns one-hidden-layer CNN: Dont be afraid of spurious local minima,” in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, pp. 1339–1348, 2018.

[124] M. Grewal and K. Glover, “Identifiability of linear and nonlinear dynamical systems,” *IEEE Trans. Autom. Control*, vol. 21, no. 6, pp. 833–837, 1976.

[125] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004.

[126] S. Friedberg, A. Insel, and L. Spence, *Linear Algebra*, vol. 4. 2003.

[127] R. Kamalapurkar, B. Reish, G. Chowdhary, and W. E. Dixon, “Concurrent learning for parameter estimation using dynamic state-derivative estimators,” *IEEE Trans. Autom. Control*, vol. 62, pp. 3594–3601, July 2017.

[128] P. Ioannou and J. Sun, *Robust Adaptive Control*. Prentice Hall, 1996.

[129] R. Ortega, J. G. Romero, and S. Aranovskiy, “A new least squares parameter estimator for nonlinear regression equations with relaxed excitation conditions and forgetting factor,” *Syst. Control Lett.*, vol. 169, p. 105377, 2022.

[130] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 3 ed., 2002.

[131] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. 13th Int. Conf. Artif. Intell. Stat.*, vol. 9, pp. 249–256, 13–15 May 2010.

[132] E. A. Coddington and N. Levinson, *Theory of ordinary differential equations*. McGraw-Hill, 1955.

[133] R. Zheng, T. Chen, X. Zhang, Z. Zhang, X. Jing, and H. Yin, “Offline-to-online learning enabled robust control for uncertain robotic systems pursuing constraint-following,” *IEEE Trans. Ind. Electron.*, vol. 72, no. 6, pp. 6432–6442, 2025.

[134] 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.

BIOGRAPHICAL SKETCH

Rebecca Hart was born in Grapevine, Texas, and raised in Hendersonville, North Carolina. In 2022, she graduated *summa cum laude* from North Carolina State University with a Bachelor of Science in mechanical engineering. During her undergraduate career, she conducted research in the Intelligent Structures and Systems Research Lab (iSSRL) and was recognized for her work with the North Carolina Space Grant Research Scholarship. Following her undergraduate studies, she joined the Nonlinear Controls and Robotics (NCR) Laboratory at the University of Florida, where she earned her Master of Science in mechanical engineering with a minor in electrical engineering in 2023. Rebecca completed her PhD in mechanical engineering at the University of Florida under the supervision of Dr. Warren E. Dixon in 2025. Her doctoral research focused on Lyapunov-based methods for nonlinear and adaptive control, with a specific emphasis on physics-informed learning and real-time system identification. Her doctoral work was supported by the National Science Foundation Graduate Research Fellowship Program (NSF GRFP). During her time at the University of Florida she has served as Chair, Vice Chair, and Safety Chair of the Mechanical and Aerospace Engineering Graduate Student Council.