2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Deep Lyapunov-Based Physics-Informed Neural Networks (DeLb-PINN) for Adaptive Control Design

Rebecca G. Hart, Omkar Sudhir Patil, Emily J. Griffis, and Warren E. Dixon

***Abstract*—Physics-informed learning is an emerging machine learning technique driven by the desire to leverage known physical principles in machine learning algorithms. Recent developments have produced physics-informed neural networks (PINNs) which are neural networks designed to be constrained by known physical principles. However, developing real-time adaptive control methods with stability guarantees for PINNs remains an open problem. This paper develops the first result for a deep Lyapunov-based physics-informed neural network (DeLb-PINN) architecture to adaptively control uncertain Euler-Lagrange systems. Lyapunov-derived weight adaptation laws provide continuous, online learning using the DeLb-PINN architecture without the need for offline training. A nonsmooth desired compensation adaptation law (DCAL) Lyapunov-based analysis is provided to guarantee global asymptotic tracking error convergence.**

I. Introduction

Deep neural network (DNN)-based techniques are gaining popularity to model and control practical engineering systems [1]. Yet, motivation still exists to use classical modeling techniques, because DNNs have typically been limited by availability and quality of training data, and the resulting outcomes may be difficult to correlate with physical insights and constraints. If improperly trained, or in the presence of unexpected dynamic behaviors, the use of DNNs can result in unpredictable and damaging behavior when connected with a physical system. However, if even feasible, the required effort and cost associated with obtaining an accurate physics-based model increases as system complexity increases, unlike black-box models such as DNNs.

Physics-inspired neural networks (PINNs) are an emerging tool that combine the approximation power of DNNs with the physical plausibility of classical models [2]–[6]. The resulting physics-inspired models are able to conform to the system's physics. Additionally, physical constraints can be used to reduce the possible solution space by eliminating invalid solutions resulting from noisy data, and provide additional training guidance for applications with sparse data [2]. Current PINN methods have proven useful for approximating the solution to differential equations, although limited results are available that use physics-inspired methods in feedback control algorithms [3], [6], [7].

Motivated by recent advances in physics-informed learning, results such as [5], [8]–[10] develop PINNs suitable for modeling and control of dynamical systems. Specifically, the result in [5] develops a deep Lagrangian Neural Network (DeLaN) which uses multiple DNNs based on the system's Lagrangian to form a model of the system. The DeLaN-based controller demonstrates lower tracking error for all training set sizes when compared to a single feedforward DNN-based estimate. However, the DeLaN architecture developed in [5] involves quasi-static weight updates based on training data (i.e., updates that are essentially offline) using optimization-based techniques, without stability guarantees.

Recent works in adaptive control literature have addressed the need for continuous learning without the requirement for training data [11]–[15]. Specifically, in [14] and [15], continuous-time adaptation laws for all layers of DNN-based controllers were derived from a constructive Lyapunov-based analysis. Unlike offline learning methods that require rich training datasets and involve static or quasi-static DNN weight estimates, Lyapunov-based adaptation laws provide continual learning of the system dynamics while providing stability guarantees [15]. Although these results eliminate the need for offline training data and allow for sustained learning, they do not take advantage of known information obtained from physics. Hence, motivation exists to develop adaptive controllers using PINN architectures that leverage physical knowledge of the system.

In the context of Euler-Lagrange dynamical systems, developing PINN-based adaptive controllers is challenging because PINN architectures require multiple DNN representations to approximate unknown matrix structures (i.e., inertia and centripetal-Coriolis matrices) that are coupled with acceleration and velocity-based terms. The result in [15] avoided these complexities by using DNNs to compensate for only uncoupled vector terms.

The authors are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville FL 32611-6250 USA. Email: {rebecca.hart, patilomkarsudhir, emilygriffis00, wdixon}@ufl.edu.

This research is supported in part by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1842473, AFRL grant FA8651-21-F-1027, Office of Naval Research Grant N00014-21-1-2481, and AFOSR grant FA9550-19-1-0169. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsoring agency.

This paper develops the first result for a deep Lyapunov-based physics-informed neural network (DeLb-PINN) architecture to adaptively control uncertain Euler-Lagrange systems. The mathematical challenges in developing Lyapunov-based adaptation laws are addressed by first using a vectorized representation of the inertia and centripetal-Coriolis matrices, and then invoking properties of vectorization and Kronecker product operators. A nonsmooth Lyapunov-based analysis is performed to constructively design the adaptation laws. By using a desired compensation adaptation law (DCAL) approach inspired by [16], the Lyapunov-based analysis yields global asymptotic tracking error convergence.

Notation and Preliminaries: The vectorization operator is denoted by $\text{vec}(\cdot)$, i.e., given $A \triangleq [a_{j,i}] \in \mathbb{R}^{n\times m}$, $\text{vec}(A) \triangleq [a_{1,1}, ..., a_{1,m}, ..., a_{n,m}]^\top$. The notation $\overset{a.a.t.}{(\cdot)}$ denotes the relation $(\cdot)$ holds for almost all time (a.a.t.). The p-norm is denoted by $\|\cdot\|_p$, where the subscript is suppressed when $p = 2$. 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 $\overset{\curvearrowleft}{\prod}_{p=a}^{m} A_p = 1$ if $a > m$. The Kronecker product is defined by $\otimes$. Function composition is defined by $\circ$ where $(f \circ g)(x) \triangleq f(g(x))$. The $n \times n$ identity matrix is denoted by I_n. Given any $A \in \mathbb{R}^{p\times a}$, $B \in \mathbb{R}^{a\times h}$, and $C \in \mathbb{R}^{h\times s}$, $\text{vec}(ABC) = \left(C^\top \otimes A\right)\text{vec}(B)$ [17, Proposition 7.1.9].

II. Problem Formulation

A. Deep Neural Network (DNN) Background

Consider the family of feedforward DNNs $\Phi_i : \mathbb{R}^{m_i} \times \mathbb{R}^{\sum_{j=0}^{k_i} L_{j,i}L_{j+1,i}} \to \mathbb{R}^{L_{k+1,i}}$ defined as

$$\Phi_i(x_i, \theta_i) \triangleq (V_{k,i}^\top \phi_{k,i} \circ ... \circ V_{1,i}^\top \phi_{1,i})(V_{0,i}^\top x_{a,i}), \quad (1)$$

where $i \in \mathcal{I}$ denotes the index of a specific DNN within the family of DNNs denoted by the set $\mathcal{I}$, $V_{j,i} \in \mathbb{R}^{L_{j,i}\times L_{j+1,i}}$ is 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 $j \in \{0, ..., k_i\}$, $k_i \in \mathbb{N}$ denotes the number of hidden layers, and $L_{0,i} \triangleq m_i + 1$, where m is the size of the input to the DNN. All of the weights within Φ_i are represented by $\theta_i \in \mathbb{R}^{\sum_{j=0}^{k_i} L_{j,i}L_{j+1,i}}$, where $\theta_i \triangleq \left[\text{vec}(V_{0,i})^\top, ..., \text{vec}(V_{k,i})^\top\right]^\top$, the vector of smooth[1] activation functions at the j^{th} layer is denoted by $\phi_{j,i} : \mathbb{R}^{L_{j,i}} \to \mathbb{R}^{L_{j,i}}$ and is defined as $\phi_{j,i} \triangleq \begin{bmatrix}\varsigma_{j,i,1} & \dots & \varsigma_{j,i,L_j-1} & 1\end{bmatrix}^\top$, 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 $j \in \{1, ..., k_i\}$ and $i \in \mathcal{I}$. The augmented input $x_{a,i} \in \mathbb{R}^{m_i+1}$ is defined as $x_{a,i} \triangleq \begin{bmatrix}x_i^\top & 1\end{bmatrix}^\top$, where $x_i \in \Omega_i$ denotes the input to the DNN, where $\Omega_i \subseteq \mathbb{R}^{m_i}$ denotes a compact set for all $i \in \mathcal{I}$. To incorporate a bias term into the DNN model in (1), the input x_i and the activation functions $\phi_{j,i}$ are augmented with a 1 for all $j \in \{1, ..., k_i\}$ and $i \in \mathcal{I}$.

To facilitate the development of the weight adaptation laws, the DNN model in (1) can also be represented recursively using shorthand notation $\Phi_{j,i}(x_i, \theta_i)$ as

$$\Phi_{j,i} \triangleq \begin{cases} V_{j,i}^\top \phi_{j,i}(\Phi_{j-1,i}), & j \in \{1, ..., k_i\} \\ V_{0,i}^\top x_{a,i}, & j = 0, \end{cases} \quad (2)$$

for all $i \in \mathcal{I}$. The universal function approximation property states that the space of DNNs given by (2) is dense in $\mathcal{C}(\Omega_i)$, where $\mathcal{C}(\Omega_i)$ denotes a space of continuous functions over Ω_i [18, 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 $\sup_{x_i\in\Omega_i} \|f_i(x_i) - \Phi_i(x_i, \theta_i^*)\| \le \bar{\varepsilon}_i$, for all $i \in \mathcal{I}$.

B. Model Dynamics

Consider an uncertain Euler-Lagrange system modeled as

$$M(q)\ddot{q} + V_m(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d(t) = \tau(t), \quad (3)$$

where $q, \dot{q}, \ddot{q} : \mathbb{R}_{\ge 0} \to \mathbb{R}^n$ are the generalized position, velocity, and acceleration, respectively. The inertia matrix, centripetal-Coriolis effects, generalized gravitational forces, generalized dissipation effects, the time varying disturbances, and the control input are denoted by $M : \mathbb{R}^n \to \mathbb{R}^{n\times n}$, $V_m : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n}$, $G : \mathbb{R}^n \to \mathbb{R}^n$, $F : \mathbb{R}^n \to \mathbb{R}^n$, $\tau_d : \mathbb{R}_{\ge 0} \to \mathbb{R}^n$ and $\tau : \mathbb{R}_{\ge 0} \to \mathbb{R}^n$, respectively. It is assumed the disturbance τ_d can be bounded as $\|\tau_d(t)\| \le \overline{d}$ where $\overline{d} \in \mathbb{R}_{>0}$ denotes a known constant. The system in (3) satisfies properties in [19, Sec. 2.3].

Property 1. The inertia matrix $M(q)$, satisfies $m_1\|\zeta\|^2 \le \zeta^\top M(q)\zeta \le m_2\|\zeta\|^2$ 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, $\zeta^\top(\dot{M}(q) - 2V_m(q, \dot{q}))\zeta = 0$, for all $q, \dot{q}, \zeta \in \mathbb{R}^n$.

III. Control Law Development

A. Control Objective

The control objective is to design a DeLb-PINN control architecture to asymptotically track a user-defined, time-varying desired trajectory $q_d : \mathbb{R}_{\ge 0} \to \mathbb{R}^n$. To quantify the control objective, let the tracking error $e : \mathbb{R}_{\ge 0} \to \mathbb{R}^n$ and auxiliary tracking error $r : \mathbb{R}_{\ge 0} \to \mathbb{R}^n$ be defined as

$$e \triangleq q - q_d, \quad (4)$$

$$r \triangleq \dot{e} + \alpha e, \quad (5)$$

respectively, where $\alpha \in \mathbb{R}_{>0}$ denotes a user-selected constant control gain. Quantitatively, the objective is to ensure

[1]For notational simplicity, the DNN model in (1) only considers smooth activation functions. To consider nonsmooth activation functions, the switched systems analysis in [15] can be used with the subsequent control development.

$\|e(t)\| \to 0$ as $t \to \infty$. The desired trajectory and its time-derivative are designed to be continuously differentiable 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.

Taking the time-derivative of r, pre-multiplying by $M(q)$, and using (3)-(5) yields

$$M(q)\dot{r} = \tau(t) - V_m(q,\dot{q})\dot{q} - G(q) - F(\dot{q}) - \tau_d(t) - M(q)\ddot{q}_d + M(q)\alpha\dot{e}. \tag{6}$$

After some algebraic manipulation, (6) can be re-written in the advantageous form

$$\begin{aligned} M(q)\dot{r} = {} & \tau(t) - \tau_d(t) - V_m(q,\dot{q})\, r \\ & + (M_e - M(q_d))(\ddot{q}_d - \alpha\dot{e}) \\ & + (V_{me} - V_m(q_d,\dot{q}_d))(\dot{q}_d - \alpha e) \\ & + G_e - G(q_d) + F_e - F(\dot{q}_d), \end{aligned} \tag{7}$$

\left(x_{F}\right)where $V_{me},\ M_e,\ G_e,$ and F_e are defined as $V_{me} \triangleq V_m(q_d,\dot{q}_d) - V_m(q,\dot{q})$, $M_e \triangleq M(q_d) - M(q)$, $G_e \triangleq G(q_d) - G(q)$, and $F_e \triangleq F(\dot{q}_d) - F(\dot{q})$, respectively.

B. DeLb-PINN Neural Network Architecture

PINN architectures have grown in recent popularity due to their ability to leverage known physical properties of the system to improve model accuracy and the ability to generalize beyond training data. Within the DeLb-PINN architecture, DNNs are used as adaptive real-time feedforward approximations of the unknown terms $M(q_d)$, $V_m(q_d,\dot{q}_d)$, $G(q_d)$, and $F(\dot{q}_d)$ (see Figure 1). Compared to traditional DNN control techniques, the use of DeLb-PINN is motivated by the desire to integrate the known physics of (3) into the control architecture. The implementation of multiple DNNs within the DeLb-PINN structure involves various mathematical challenges due to the $M(q)$ and $V_m(q,\dot{q})$ terms being matrices multiplied by a vector. This, and the fact that the result in (2) does not apply for matrix approximations motivates the use of the vectorization operator which results in additional technical challenges when developing the weight adaptation laws for M and V_m.

Based on the universal function approximation property, and the unknown terms $M(q_d)$, $V_m(q_d,\dot{q}_d)$, $G(q_d)$, and $F(\dot{q}_d)$ can be modeled using the DNN formulation in (1) as[2]

$$\mathrm{vec}(M(q_d)) = \Phi_M(x_M, \theta_M^*) + \varepsilon_M(x_M), \tag{8}$$

$$\mathrm{vec}(V_m(q_d,\dot{q}_d)) = \Phi_{V_m}\left(x_{V_m}, \theta_{V_m}^*\right) + \varepsilon_{V_m}(x_{V_m}), \tag{9}$$

$$G(q_d) = \Phi_G(x_G, \theta_G^*) + \varepsilon_G(x_G), \tag{10}$$

$$F(\dot{q}_d) = \Phi_F(x_F, \theta_F^*) + \varepsilon_F(x_F), \tag{11}$$

[2]Motivated by the DCAL strategy in [16], the functions in (8)-(11) are expressed in terms of q_d and $\dot{q}_d$ which are continuous bounded trajectories, and hence, the functions lie on the space of continuous functions over $\mathcal{Q}$. Therefore, the universal function approximation property can be applied without restrictions on the initial conditions of the state of the system which facilitates a global result in the subsequent stability analysis.

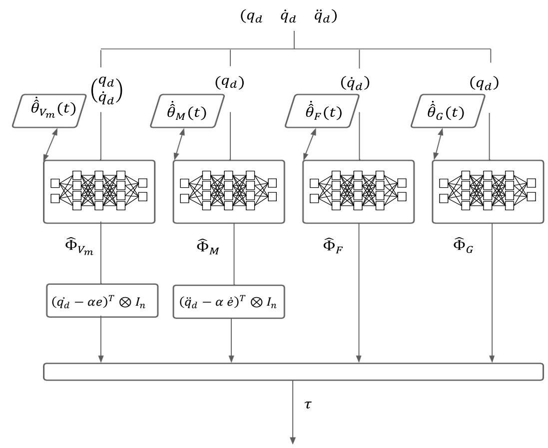

Figure 1. A diagrammatic illustration of the DeLb-PINN architecture which leverages the known structure of the dynamics in (3), and uses DNN-based adaptive estimates of $\mathrm{vec}(M)$, $\mathrm{vec}(V_m)$, G, and F given by $\widehat{\Phi}_M$, $\widehat{\Phi}_{V_m}$, $\widehat{\Phi}_G$, and $\widehat{\Phi}_F$, respectively.

respectively. The ideal DNN estimate is denoted $\Phi_i \in \mathbb{R}^{L_{k_i+1}}$ for $i \in \{M, V_m, F, G\}$ and the input to the DNNs are defined as $x_M \triangleq q_d$, $x_{V_m} \triangleq [q_d, \dot{q}_d]^\top$, $x_G \triangleq q_d$, and $x_F \triangleq \dot{q}_d$. The unknown function approximation errors are $\varepsilon_M : \mathbb{R}^n \to \mathbb{R}^{n^2}$, $\varepsilon_{V_m} : \mathbb{R}^{2n} \to \mathbb{R}^{n^2}$, $\varepsilon_G : \mathbb{R}^n \to \mathbb{R}^n$, and $\varepsilon_F : \mathbb{R}^n \to \mathbb{R}^n$ for $\Phi_M(x_M, \theta_M^*)$, $\Phi_{V_m}\left(x_{V_m}, \theta_{V_m}^*\right)$, $\Phi_G(x_G, \theta_G^*)$, and $\Phi_F(x_F, \theta_F^*)$, respectively. The ideal weights of the DNN, number of hidden layers, and number of nodes in the j^{th} layer are denoted by $\theta_i^* \triangleq \left[\mathrm{vec}(V_{0,i})^\top, ..., \mathrm{vec}(V_{k_i,i})^\top\right]^\top \in \mathbb{R}^{\sum_{j=0}^{k_i} L_{j,i} L_{j+1,i}}$, $k_i \in \mathbb{N}$, $L_{j,i} \in \mathbb{R}_{>0}$ for all $i \in \{M, V_m, G, F\}$ and $j \in \{0, ..., k_i\}$, respectively. The following assumption is made to facilitate the subsequent development.

Assumption 1: There exists a known constant $\bar{\theta}_i \in \mathbb{R}_{>0}$ such that the unknown ideal weights can be bounded as $\|\theta_i^*\| \leq \bar{\theta}_i$ for all $i \in \{M, V_m, G, F\}$ [20, Assumption 1].

To ensure the output of the DNN is of appropriate dimension, the vectorization operator is applied to $M(q_d)$ and $V_m(q_d,\dot{q}_d)$. Applying the vectorization operator on $M(q_d)$ and $V_m(q_d,\dot{q}_d)$, using properties of the vectorization operator, and substituting in (8)-(11) into (7) yields

$$\begin{aligned} M(q)\dot{r} = {} & \tau(t) - \tau_d(t) - V_m(q,\dot{q})\, r + \widetilde{N} \\ & - \left((\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\right)\Phi_M(x_M, \theta_M^*) \\ & - \left((\dot{q}_d - \alpha e)^\top \otimes I_n\right)\Phi_{V_m}\left(x_{V_m}, \theta_{V_m}^*\right) \\ & - \Phi_G(x_G, \theta_G^*) - \Phi_F(x_F, \theta_F^*) - \varepsilon_G(x_G) \\ & - \left((\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\right)\varepsilon_M(x_M) - \varepsilon_F(x_F) \\ & - \left((\dot{q}_d - \alpha e)^\top \otimes I_n\right)\varepsilon_{V_m}(x_{V_m}), \end{aligned} \tag{12}$$

where the auxiliary function $\widetilde{N} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is defined as $\widetilde{N}(e, r, \dot{q}_d, \ddot{q}_d) \triangleq M_e(\ddot{q}_d - \alpha\dot{e}) + V_{me}(\dot{q}_d - \alpha e) +$

$G_e + F_e$. Since $\dot{q}_d$ and $\ddot{q}_d$ are bounded, $\widetilde{N}$ can be bounded as $\|\widetilde{N}\| \leq \rho(\|z\|)\|z\|$, where $z \triangleq \left[e^\top, r^\top\right]^\top$ and $\rho : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ denotes a known strictly increasing function based on the mean value theorem-based inequality in [21, Appendix A].

C. Control Design

The DeLb-PINN control design uses adaptive estimates of the involved DNNs while also utilizing knowledge of the model structure through the developed architecture in Figure 1. Based on the subsequent stability analysis, the control input is designed as

$$\begin{aligned}\tau(t) = &\left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)\widehat{\Phi}_{V_m} + \widehat{\Phi}_G + \widehat{\Phi}_F \\ &+ \left(\left(\ddot{q}_d - \alpha \dot{e}\right)^\top \otimes I_n\right)\widehat{\Phi}_M - k_1 r - e \\ &- \operatorname{sgn}(r)\left(\rho(\|z\|)\|z\| + k_2 + k_3\|(\dot{q}_d - \alpha e)^\top \otimes I_n\|\right)(x_F) \\ &- k_4 \operatorname{sgn}(r)\|(\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\|, \end{aligned} \tag{13}$$

where $k_1, k_2, k_3, k_4 \in \mathbb{R}_{>0}$ are user-defined control gains, and the adaptive estimate of each DNN is denoted as $\widehat{\Phi}_i \triangleq \Phi_i(x_{d,i}, \hat{\theta}_i)$ for all $i \in \{M, V_m, G, F\}$, where the estimated weights and biases are denoted by $\hat{\theta}_i = \left[\operatorname{vec}(\widehat{V}_{0,i})^\top, ..., \operatorname{vec}(\widehat{V}_{k,i})^\top\right]^\top \in \mathbb{R}^{L_{j,i} \times L_{j+1,i}}$, for all $(j,i) \in \{0, ..., k_i\} \times \{M, V_m, G, F\}$, where $\widehat{V}_{j,i} \in \mathbb{R}^{L_{j,i} \times L_{j+1,i}}$ is the subsequently designed estimated matrix of weights and biases in the j^{th} hidden layer for all $j \in \{0, ..., k_i\}$ and $i \in \{M, V_m, G, F\}$. The following shorthand notations are introduced for brevity in the subsequent analysis $\Phi_i^* \triangleq \Phi_i(x_{d,i}, \theta_i^*)$, $\widetilde{\Phi}_i \triangleq \Phi_i^* - \widehat{\Phi}_i$, and $\widetilde{\Phi}_i \triangleq \Phi_i^* - \widehat{\Phi}_i$ for all $i \in \{M, V_m, G, F\}$. Substituting the control input in (13) into (12) yields the closed-loop error system

$$\begin{aligned}M(q)\dot{r} = &-\left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)\widetilde{\Phi}_{V_m} - \widetilde{\Phi}_G - \widetilde{\Phi}_F \\ &- \left(\left(\ddot{q}_d - \alpha \dot{e}\right)^\top \otimes I_n\right)\widetilde{\Phi}_M - k_1 r - e \\ &- \operatorname{sgn}(r)(\rho(\|z\|)\|z\| + k_2 + k_3\|(\dot{q}_d - \alpha e)^\top \otimes I_n\|) \\ &- k_4 \operatorname{sgn}(r)\|(\ddot{q}_d - \alpha \dot{e})^\top \otimes I_n\| - \tau_d(t) \\ &- V_m(q, \dot{q})r + \widetilde{N} - \left(\left(\ddot{q}_d - \alpha \dot{e}\right)^\top \otimes I_n\right)\varepsilon_M(x_M) \\ &- \varepsilon_G(x_G) - \varepsilon_F(x_F) - \left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)\varepsilon_{V_m}(x_{V_m}). \end{aligned} \tag{14}$$

IV. DNN Weight Adaptation Laws

The implementation of real-time Lyapunov-based adaptation laws allows the DeLb-PINN architecture to continuously adapt in real-time. Based on the subsequent analysis, the weight adaptation laws are designed as

$$\dot{\hat{\theta}}_M = -\operatorname{proj}\left(\Gamma_M \widehat{\Phi}_M'^\top \left(\left(\ddot{q}_d - \alpha \dot{e}\right)^\top \otimes I_n\right)^\top r\right), \tag{15}$$

$$\dot{\hat{\theta}}_{V_m} = -\operatorname{proj}\left(\Gamma_{V_m} \widehat{\Phi}_{V_m}'^\top \left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)^\top r\right), \tag{16}$$

$$\dot{\hat{\theta}}_F = -\operatorname{proj}\left(\Gamma_F \widehat{\Phi}_F'^\top r\right), \tag{17}$$

$$\dot{\hat{\theta}}_G = -\operatorname{proj}\left(\Gamma_G \widehat{\Phi}_G'^\top r\right), \tag{18}$$

where $\Gamma_i \in \mathbb{R}^{\sum_{j=0}^{k_i} L_{j,i} L_{j+1,i} \times \sum_{j=0}^{k_i} L_{j,i} L_{j+1,i}}$ is a positive-definite adaptation gain matrix for $i \in \{M, V_m, G, F\}$. The operator $\operatorname{proj}(\cdot)$ denotes the projection operator defined in [22, Appendix E, Eq. E.4], which is used to ensure that $\hat{\theta}_i(t) \in \mathcal{B}_i \triangleq \left\{\theta_i \in \mathbb{R}^{\Sigma_{j=0}^{k_i} L_{j,i} L_{j+1,i}} : \|\theta_i\| \leq \bar{\theta}_i\right\}$, for all $t \in \mathbb{R}_{\geq 0}$ and $i \in \{M, V_m, G, F\}$. The Jacobian of the DNN $\widehat{\Phi}_i'$ can be represented as $\widehat{\Phi}_i' \triangleq \left[\widehat{\Phi}_{0,i}', ..., \widehat{\Phi}_{j,i}'\right]$, where the shorthand notation $\widehat{\Phi}_{j,i}'$ is defined as $\widehat{\Phi}_{j,i}' \triangleq \frac{\partial \Phi_{j,i}(x_i, \hat{\theta}_i)}{\partial \hat{\theta}_i}$, for all $j \in \{0, ..., k_i\}$ and $i \in \{M, V_m, G, F\}$. Using (2), the chain rule, and the properties of the vectorization operator the terms $\widehat{\Phi}_{0,i}'$ and $\widehat{\Phi}_{j,i}'$ can be expressed as

$$\begin{aligned}\widehat{\Phi}_{0,i}' &\triangleq \left(\overset{\curvearrowleft}{\prod_{l=1}^{k_i}} \widehat{V}_{l,i}^\top \hat{\phi}_{l,i}'\right)\left(I_{L_{1,i}} \otimes x_{a,i}^\top\right), \\ \widehat{\Phi}_{j,i}' &\triangleq \left(\overset{\curvearrowleft}{\prod_{l=j+1,i}^{k_i}} \widehat{V}_{l,i}^\top \hat{\phi}_{l,i}'\right)\left(I_{L_{j+1,i}} \otimes \hat{\phi}_{j,i}^\top\right), \end{aligned} \tag{19}$$

for all $j \in \{1, ..., k_i\}$ and $i \in \{M, V_m, G, F\}$, respectively, where the shorthand notation $\hat{\phi}_{j,i} \triangleq \phi_{j,i}(\Phi_{j-1}(x_i, \hat{\theta}_i))$ and $\hat{\phi}_{j,i}' \triangleq \phi_{j,i}'(\Phi_{j-1}(x_i, \hat{\theta}_i))$ denotes DNN activation function at the j^{th} layer and its Jacobian, respectively and the gradient of the activation function can be represented by $\phi_{j,i}' : \mathbb{R}^{L_{j,i}} \to \mathbb{R}^{L_{j,i} \times L_{j,i}}$ and is defined as $\phi_{j,i}'(y) \triangleq \frac{\partial}{\partial \varrho}\phi_{j,i}(\varrho)|_{\varrho = y}$, for all $y \in \mathbb{R}^{L_{j,i}}$, $i \in \{M, V_m, G, F\}$, and $j \in \{1, ..., k_i\}$.

V. Stability Analysis

Theorem 1. *For the dynamical system in (3), the controller in (13) and the adaptation laws developed in (15)-(18) ensure global asymptotic tracking in the sense that $\|e(t)\| \to 0$ and $\|r(t)\| \to 0$ as $t \to \infty$, provided the following gain conditions are satisfied*

$$\begin{aligned} k_2 &> \bar{d} + \bar{\Delta}_G + \bar{\Delta}_F + \bar{\varepsilon}_G + \bar{\varepsilon}_F, \\ k_3 &> \bar{\Delta}_{V_m} + \bar{\varepsilon}_{V_m}, \\ k_4 &> \bar{\Delta}_M + \bar{\varepsilon}_M. \end{aligned} \tag{20}$$

Proof: Let the states $w : \mathbb{R}_{\geq 0} \to \mathbb{R}^\Psi$ be defined as $w \triangleq \left[r^\top, e^\top, \tilde{\theta}_{V_m}^\top, \tilde{\theta}_G^\top, \tilde{\theta}_F^\top, \tilde{\theta}_M^\top\right]^\top$, where $\Psi \triangleq 2n + \Sigma_{i \in \mathcal{I}} \Sigma_{j=0}^{k_i} L_{j,i} \times L_{j+1,i}$ and $\mathcal{I} \triangleq \{M, V_m, F, G\}$. Consider the candidate Lyapunov function $\mathcal{V}_L : \mathbb{R}^\Psi \to \mathbb{R}_{\geq 0}$ defined as

$$\begin{aligned}\mathcal{V}_L(w) \triangleq &\frac{1}{2} r^\top M(q) r + \frac{1}{2} e^\top e + \frac{1}{2}\tilde{\theta}_{V_m}^\top \Gamma_{V_m}^{-1} \tilde{\theta}_{V_m} \\ &+ \frac{1}{2}\tilde{\theta}_G^\top \Gamma_G^{-1} \tilde{\theta}_G + \frac{1}{2}\tilde{\theta}_F^\top \Gamma_F^{-1} \tilde{\theta}_F + \frac{1}{2}\tilde{\theta}_M^\top \Gamma_M^{-1} \tilde{\theta}_M. \end{aligned} \tag{21}$$

The candidate Lyapunov function in (21) satisfies the inequality $\underline{\beta}\left\|w\right\|^2 \leq \mathcal{V}_L\left(w\right) \leq \overline{\beta}\left\|w\right\|^2$, where $\underline{\beta}, \overline{\beta} \in \mathbb{R}_{\geq 0}$ are known constants. Taking the time-derivative of $\mathcal{V}_L(w)$, applying the chain rule for differential inclusions in [23, Thm 2.2] and applying (14) and (15)-(18) yields

$$\begin{aligned}\dot{\mathcal{V}}_L \overset{a.a.t.}{\in} & r^\top \Big(-\left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right) \widetilde{\Phi}_{V_m} \\ & - \widetilde{\Phi}_G - \widetilde{\Phi}_F - \left(\left(\ddot{q}_d - \alpha \dot{e}\right)^\top \otimes I_n\right) \widetilde{\Phi}_M \\ & - k_1 r - e - K\left[\text{sgn}\right](r)\left(k_2 + k_4 \|(\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\|\right) \\ & - K\left[\text{sgn}\right](r)\left(k_3 \|\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\| + \rho(\|z\|)\|z\|\right) \\ & - \tau_d\left(t\right) - V_m\left(q,\dot{q}\right) r + \tilde{N} - \left(\left(\ddot{q}_d - \alpha\dot{e}\right)^\top \otimes I_n\right) \varepsilon_M\left(x_M\right) \\ & - \varepsilon_G\left(x_G\right) - \varepsilon_F\left(x_F\right) - \left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right) \varepsilon_{V_m}\left(x_{V_m}\right)\Big) \\ & + \frac{1}{2} r^\top \dot{M} r + e^\top (r - \alpha e) + \text{K[proj]}\left(\Gamma_G \widehat{\Phi}_G'^\top r\right)^\top \Gamma_G^{-1}\tilde{\theta}_G \\ & + \text{K[proj]}\left(\Gamma_F \widehat{\Phi}_F'^\top r\right)^\top \Gamma_F^{-1}\tilde{\theta}_F \\ & + \text{K[proj]}\left(\Gamma_M \widehat{\Phi}_M'^\top \left(\left(\ddot{q}_d - \alpha\dot{e}\right)^\top \otimes I_n\right)^\top r\right)^\top \Gamma_M^{-1}\tilde{\theta}_M \\ & + \text{K[proj]}\left(\Gamma_{V_m} \widehat{\Phi}_{V_m}'^\top \left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)^\top r\right)^\top \Gamma_{V_m}^{-1}\tilde{\theta}_{V_m}.\end{aligned} \tag{22}$$

To facilitate the calculation of $\widetilde{\Phi}_i$, a first-order Taylor series approximation-based error model is given by [20, Eq. 22]

$$\widetilde{\Phi}_i = \widehat{\Phi}_i'\tilde{\theta}_i + \mathcal{O}^2(\|\tilde{\theta}_i\|), \tag{23}$$

for $i \in \{M, V_m, G, F\}$, where $\mathcal{O}^2\left(\|\tilde{\theta}_i\|\right)$ denotes the higher order terms which can be bounded by $\bar{\Delta}_i \in \mathbb{R}_{>0}$ such that $\|\mathcal{O}^2\left(\|\tilde{\theta}_i\|\right)\| \leq \bar{\Delta}_i$ for $i \in \{M, V_m, G, F\}$. Substituting (23) into (22), applying Property 2, combining like terms, and bounding yields

$$\begin{aligned}\dot{\mathcal{V}}_L \overset{a.a.t.}{\leq} & r^\top \Big(-\left(\left(\ddot{q}_d - \alpha\dot{e}\right)^\top \otimes I_n\right)\left(\widehat{\Phi}_M'\tilde{\theta}_M + \mathcal{O}^2(||\tilde{\theta}_M||)\right) \\ & - \left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)\left(\widehat{\Phi}_{V_m}'\tilde{\theta}_{V_m} + \mathcal{O}^2(||\tilde{\theta}_{V_m}||)\right) \\ & - \left(\widehat{\Phi}_G'\tilde{\theta}_G + \mathcal{O}^2(||\tilde{\theta}_G||)\right) - \left(\widehat{\Phi}_F'\tilde{\theta}_F + \mathcal{O}^2(||\tilde{\theta}_F||)\right) \\ & - k_2 K\left[\text{sgn}\right](r) - k_3 K\left[\text{sgn}\right](r)\|(\dot{q}_d - \alpha e)^\top \otimes I_n\| \\ & - k_4 K\left[\text{sgn}\right](r)\|(\ddot{q}_d - \alpha\dot{e})^\top \otimes I_n\| - \tau_d\left(t\right) - k_1 r \\ & - \left(\left(\ddot{q}_d - \alpha\dot{e}\right)^\top \otimes I_n\right)\varepsilon_M\left(x_M\right) - \varepsilon_G\left(x_G\right) \\ & - \varepsilon_F\left(x_F\right) - \left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)\varepsilon_{V_m}\left(x_{V_m}\right)\Big) - e^\top \alpha e \\ & + \text{K[proj]}\left(\Gamma_G \widehat{\Phi}_G'^\top r\right)^\top \Gamma_G^{-1}\tilde{\theta}_G + \text{K[proj]}\left(\Gamma_F \widehat{\Phi}_F'^\top r\right)^\top \Gamma_F^{-1}\tilde{\theta}_F \\ & + \text{K[proj]}\left(\Gamma_M \widehat{\Phi}_M'^\top \left(\left(\ddot{q}_d - \alpha\dot{e}\right)^\top \otimes I_n\right)^\top r\right)^\top \Gamma_M^{-1}\tilde{\theta}_M \\ & + \text{K[proj]}\left(\Gamma_{V_m} \widehat{\Phi}_{V_m}'^\top \left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)^\top r\right)^\top \Gamma_{V_m}^{-1}\tilde{\theta}_{V_m}.\end{aligned} \tag{24}$$

Using [22, Lemma E.1.IV] and the fact that K[proj]$(\cdot)$ is the set of convex combinations of proj$(\cdot)$ and $(\cdot)$, the projection operator terms can be bounded as

$$\text{K[proj]}\left(\Gamma_G \widehat{\Phi}_G'^\top r\right)^\top \Gamma_G^{-1}\tilde{\theta}_G \leq r^\top \widehat{\Phi}_G'\tilde{\theta}_G, \tag{25}$$

$$\text{K[proj]}\left(\Gamma_F \widehat{\Phi}_F'^\top r\right)^\top \Gamma_F^{-1}\tilde{\theta}_F \leq r^\top \widehat{\Phi}_F'\tilde{\theta}_F, \tag{26}$$

$$\begin{aligned}&\text{K[proj]}\left(\Gamma_M \widehat{\Phi}_M'^\top \left(\left(\ddot{q}_d - \alpha\dot{e}\right)^\top \otimes I_n\right)^\top r\right)^\top \Gamma_M^{-1}\tilde{\theta}_M \\ &\qquad \leq r^T \left(\left(\ddot{q}_d - \alpha\dot{e}\right)^\top \otimes I_n\right)\widehat{\Phi}_M'\tilde{\theta}_M,\end{aligned} \tag{27}$$

$$\begin{aligned}&\text{K[proj]}\left(\Gamma_{V_m} \widehat{\Phi}_{V_m}'^\top \left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)^\top r\right)^\top \Gamma_{V_m}^{-1}\tilde{\theta}_{V_m} \\ &\qquad \leq r^T \left(\left(\dot{q}_d - \alpha e\right)^\top \otimes I_n\right)\widehat{\Phi}_{V_m}'\tilde{\theta}_{V_m}.\end{aligned} \tag{28}$$

Using (25)-(28), the fact that $r^\top$K[sgn](r) = $\|r\|_1$, and provided the gain conditions in (20) are satisfied, (24) can be bounded as

$$\dot{\mathcal{V}}_L(w) \overset{a.a.t.}{\leq} -k_1\|r\|^2 - \alpha\|e\|^2. \tag{29}$$

Then, (29) can be further bounded as $\dot{\mathcal{V}}_L\left(w\right) \overset{a.a.t.}{\leq} -\min\left(k_1, \alpha\right)\|z\|^2$. Using (21) and the fact that $\dot{\mathcal{V}}_L(w) \leq 0$ implies $e, r, \tilde{\theta}_{V_m}, \tilde{\theta}_M, \tilde{\theta}_G, \tilde{\theta}_F \in \mathcal{L}_\infty$. Due to the use of the projection algorithm, $\hat{\theta}_{V_m}, \hat{\theta}_M, \hat{\theta}_G, \hat{\theta}_F \in \mathcal{L}_\infty$. The fact that $q_d, \dot{q}_d, e, r \in \mathcal{L}_\infty$ implies $q, \dot{q} \in \mathcal{L}_\infty$. Using (13) and the fact that $q_d, \dot{q}_d, \ddot{q}_d, e, r, \hat{\theta}_{V_m}, \hat{\theta}_M, \hat{\theta}_G, \hat{\theta}_F \in \mathcal{L}_\infty$ implies $\tau \in \mathcal{L}_\infty$. Therefore, the control input τ is bounded. Using the extension of LaSalle-Yoshizawa theorem for non-smooth systems in [24, Theorem 1], we can conclude that $\lim_{t\to\infty} \|e\left(t\right)\| = 0$ and $\lim_{t\to\infty} \|r\left(t\right)\| = 0$. ■

VI. Conclusion

In this paper, an adaptive DeLb-PINN architecture is developed for the control of general Euler-Lagrange systems. Specifically, this paper provides the first result on Lyapunov-derived adaptation laws for the weights of each layer of a DeLb-PINN adaptive controller. The mathematical challenges associated with developing Lyapunov-based adaptation laws for matrices were addressed using a vectorized representation of the matrix and then invoking properties of vectorization and Kronecker product operators to inform the adaptive update law design. Future work would include constraining the output of the DNNs further respect the positive definite property of the inertia matrix such as results in [5].

References

[1] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[2] 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.

[3] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," *arXiv*, 2020.
[4] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, "Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems," *Comput. Methods Appl. Mech. Eng.*, vol. 365, p. 113028, 2020.
[5] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," in *Int. Conf. Learn. Represent.*, 2019.
[6] 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.
[7] R. Wang, R. Walters, and R. Yu, "Incorporating symmetry into deep dynamics models for improved generalization," in *Int. Conf. Learn. Represent.*, 2021.
[8] M. Lutter, J. Silberbauer, J. Watson, and J. Peters, "Differentiable physics models for real-world offline model-based reinforcement learning," in *IEEE Int. Conf. Robot.*, pp. 4163–4170, 05 2021.
[9] 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.
[10] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," in *NeurIPS*, vol. 32, 2019.
[11] 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.
[12] 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.
[13] D. Le, M. Greene, W. Makumi, and W. E. Dixon, "Real-time modular deep neural network-based adaptive control of nonlinear systems," *IEEE Control Syst. Lett.*, vol. 6, pp. 476–481, 2022.
[14] O. S. Patil, D. M. Le, E. Griffis, and W. E. Dixon, "Deep residual neural network (ResNet)-based adaptive control: A Lyapunov-based approach," in *Proc. IEEE Conf. Decis. Control*, 2022.
[15] 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.
[16] N. Sadegh and R. Horowitz, "Stability and robustness analysis of a class of adaptive controllers for robotic manipulators," *Int J Rob Res*, vol. 9, no. 3, pp. 74–92, 1990.
[17] D. S. Bernstein, *Matrix mathematics*. Princeton university press, 2009.
[18] P. Kidger and T. Lyons, "Universal approximation with deep narrow networks," in *Conf. Learn. Theory*, pp. 2306–2327, 2020.
[19] 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.
[20] 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.
[21] M. de Queiroz, J. Hu, D. Dawson, T. Burg, and S. Donepudi, "Adaptive position/force control of robot manipulators without velocity measurements: Theory and experimentation," *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 27-B, no. 5, pp. 796–809, 1997.
[22] M. Krstic, I. Kanellakopoulos, and P. V. Kokotovic, *Nonlinear and Adaptive Control Design*. New York: John Wiley & Sons, 1995.
[23] D. Shevitz and B. Paden, "Lyapunov stability theory of nonsmooth systems," *IEEE Trans. Autom. Control*, vol. 39 no. 9, pp. 1910–1914, 1994.
[24] 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.