mathematics

MDPI

Article

Stability Analysis of Batch Offline Action-Dependent Heuristic Dynamic Programming Using Deep Neural Networks

Timotei Lala

Department of Automation and Applied Informatics, Politehnica University of Timisoara, 2, Bd. V. Parvan, 300223 Timisoara, Romania; timotei.lala@student.upt.ro; Tel.: +40-256-403040; Fax: +40-256-403214

Abstract: In this paper, the theoretical stability of batch offline action-dependent heuristic dynamic programming (BOADHDP) is analyzed for deep neural network (NN) approximators for both the action value function and controller which are iteratively improved using collected experiences from the environment. Our findings extend previous research on the stability of online adaptive ADHDP learning with single-hidden-layer NNs by addressing the case of deep neural networks with an arbitrary number of hidden layers, updated offline using batched gradient descend updates. Specifically, our work shows that the learning process of the action value function and controller under BOADHDP is uniformly ultimately bounded (UUB), contingent on certain conditions related to NN learning rates. The developed theory demonstrates an inverse relationship between the number of hidden layers and the learning rate magnitude. We present a practical implementation involving a twin rotor aerodynamical system to emphasize the impact difference between the usage of single-hidden-layer and multiple-hidden-layer NN architectures in BOADHDP learning settings. The validation case study shows that BOADHDP with multiple hidden layer NN architecture implementation obtains 0.0034 on the control benchmark, while the single-hidden-layer NN architectures obtain 0.0049, outperforming the former by 1.58% by using the same collected dataset and learning conditions. Also, BOADHDP is compared with online adaptive ADHDP, proving the superiority of the former over the latter, both in terms of controller performance and data efficiency.

Keywords: ADP; ADHDP; deep neural networks; batch learning; Lyapunov stability; uniformly ultimately bounded; gradient descent; Q-function; action value function

MSC: 68T05

Academic Editor: Xiaobing Feng

Received: 2 December 2024
Revised: 3 January 2025
Accepted: 6 January 2025
Published: 9 January 2025

Citation: Lala, T. Stability Analysis of Batch Offline Action-Dependent Heuristic Dynamic Programming Using Deep Neural Networks. *Mathematics* **2025**, *13*, 206. https://doi.org/10.3390/math13020206

1. Introduction

Adaptive dynamic programming (ADP) has emerged as a powerful methodology for tuning control systems in modern applications, where complexity, nonlinearity, and uncertainty are commonplace. Originating from Werbos' pioneering work [1], which was based on the seminal work on dynamic programming conducted by Bellman [2], ADP soon became a notable stream of research, with multiple ADP designs developed. Among the ADP designs, two distinct classes of solutions have emerged: heuristic dynamic programming (HDP) and dual heuristic programming (DHP) [3]. In the HDP framework, reinforcement learning is employed to determine the cost-to-go from the current state. The HDP convergence for the general nonlinear systems is presented in [4]. Conversely, in DHP, neural networks are used to learn the derivative of the cost function relative to the states, known as the costate vector [5]. The DHP convergence for linear systems was established

in [6]. For both of those two classes of algorithms, there exists the action-dependent (AD) adaptation [7]. ADP has also addressed the class of discrete-time control problems [8–13] and continuous time systems [14–16].

Apart from the theoretical contributions, ADP designs have been validated on a wide array of real applications. In [17], ADP is applied to a helicopter tracking and trimming control task. In [18], neural network controllers tuned with the ADHDP method are applied to an engine torque and exhaust air–fuel ratio control for an automotive engine. A practical implementation in the context of an electric water heater is presented in [19], where the collected sensor data were used to learn in a model-free manner the Q-function and the controller.

Convergence and stability proofs of the iterative processes involved in ADP-like techniques have also been developed. In [6], the adaptive critic method is described, where two networks approximate the controller and the Lagrangian multipliers associated with the optimal control, respectively. The convergence of the interleaved successive update of the two networks has been analyzed. In [20], an online generalized ADP is issued for a system with input constraints. Then, using a Lyapunov approach, a uniformly ultimate boundedness (UUB) stability is proved. The convergence of the value-iteration HDP is established for the nonlinear discrete-time systems in [4]. In paper [21], the authors derive the UUB stability for direct HDP algorithms, proving that the actor and critic weights remain bounded. The actor and critic were approximated by a multilayer perceptron (MLP) with three layers: input, hidden, and output. However, the updated weights were only the ones from the hidden and output layers, like in a linear basis function approach. To overcome the practical limitations imposed by linear basis-function-type approximators, such as scalability and overfitting, the authors from paper [22] extended the stability analysis from [21] to MLPs to update both the input-to-hidden-layer weights and the hidden-to-output weights.

Current research in the field of reinforcement learning (RL), which studies the class of stochastic systems and controllers, shows significant performances when using deep NNs for control applications for both discretized systems [23] and continuous control tasks [24]. The advantage of deep neural networks over single-layer networks lies in their increased approximation capacity, which is achieved through multiple hidden layers. These layers enable the composition of features at different abstraction levels, creating a robust hierarchical representation. This hierarchical structure allows deep networks to learn and model complex nonlinear relationships within data more effectively than shallow networks. Thus, using multilayer NNs in ADP applications can enhance learning convergence and the overall controller performance. Also, using batch learning methods, which update the NN weights using collected past experiences simultaneously, is more data efficient compared to the single-transition learning, where the weights are updated one transition at a time. This also breaks the temporal correlations, helping NNs better generalize across a system's state space. Typically, batch learning is combined with offline learning, where the weights are updated exclusively using a fixed dataset of transitions, without any adaptation during the controller's runtime. Methods such as those in [19,24–26] demonstrate the benefits of using batch learning through a technique known as experience replay. In contrast, ref. [27] highlights an approach where the entire dataset of collected transitions is used for learning, in an offline manner.

This paper makes two key contributions. First, we provide a novel theoretical stability of ADHDP when utilizing deep neural networks as function approximators for both the action value function and the controller and for when batch learning is issued on the entire dataset of collected transitions from the system. This stands as an improvement over the stability analyses performed in [21,22], which were based on single-hidden-layer NN

architectures updated online, with each transition collected during the system runtime. To this end, we prove that the batched offline ADHDP (BOADHDP) learning process is uniformly ultimately bounded (UUB) by using the Lyapunov stability approach. We show that the stability of the learning process is dependent on some conditions imposed on the NN learning rates and that these conditions also provide a relationship between the learning rate magnitudes and the number of hidden layers in the networks. Second, we issue a validation study on a twin rotor aerodynamical system (TRAS) to emphasize the superiority of employing multiple hidden layers in the NN approximators in the BOADHDP learning process. We also issue some comparison between BOADHDP and the online adaptive ADHDP algorithms from [21,22].

The rest of the paper is organized as follows. Section 2 describes the theoretical underpinnings of BOADHDP. Section 3 presents the multilayer neural network approximation of the action value function and controller. Section 4 provides the main theoretical results for the stability of BOADHDP. Section 5 illustrates the TRAS validation case study. Finally, the discussion and concluding remarks are presented in Section 6.

2. Problem Formulation

Let the discrete-time nonlinear system described by the state equation be

$$\boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, \boldsymbol{u}_k), \tag{1}$$

where $k \in \mathbb{N}$ denotes the time index, $\boldsymbol{x}_k = [x_{1,k}, \ldots, x_{n,k}]^T \in \Omega_X \subset \Re^n$ the system state, $\boldsymbol{u}_k = [u_{1,k}, \ldots, u_{m,k}]^T \in \Omega_U \subset \Re^m$ the control input, $F : \Omega_X \times \Omega_U \to \Omega_X$ the unknown continuously differentiable system function, and Ω_X and Ω_U the compact subsets of $\Re^n$ and $\Re^m$, respectively. The control input is generated by $\boldsymbol{u}_k = C(\boldsymbol{x}_k)$, with $C : \Omega_X \to \Omega_U$ a time-invariant, continuous state feedback controller function with respect to the state $\boldsymbol{x}$. For convention, vectors with $[x_{1,k}, \ldots, x_{n,k}]^T$ are column vectors, while the ones without the transposition are row vectors.

For the optimal control problem, the objective is to find the optimal controller that minimizes the infinite value function, defined as follows:

$$V(\boldsymbol{x}_k) = \sum_{i=k}^{\infty} r(\boldsymbol{x}_i,\ C(\boldsymbol{x}_i))\ = r(\boldsymbol{x}_k,\ C(\boldsymbol{x}_k)) + V(\boldsymbol{x}_{k+1}), \tag{2}$$

where function $r : \Omega_X \times \Omega_U \to \Re$, having $r(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq 0$, $r(0,0) = 0$, is known as the penalty function, defined as $r(\boldsymbol{x}_k, \boldsymbol{u}_k) = \Theta(\boldsymbol{x}_k) + C(\boldsymbol{x}_k)^T \boldsymbol{R} C(\boldsymbol{x}_k)$, where $\Theta : \Omega_X \to \Re$ is the penalty term describing the system's desired behavior as a positive semidefinite function, and $\boldsymbol{R} \in \Re^{m \times m}$ is a square positive definite command weighting matrix, as in [4]. The optimal value function [1] is defined as

$$V^*(\boldsymbol{x}_k) = \min_{C(\boldsymbol{x}_k)} \{ r(\boldsymbol{x}_k,\ C(\boldsymbol{x}_k)) + V^*(\boldsymbol{x}_{k+1}) \}. \tag{3}$$

The optimal controller is found by applying the *argmin*() operator to Equation (3), as

$$C^*(\boldsymbol{x}_k) = arg\min_{C(\boldsymbol{x}_k)} \{ r(\boldsymbol{x}_k,\ C(\boldsymbol{x}_k)) + V^*(\boldsymbol{x}_{k+1}) \}. \tag{4}$$

With the system function F unknown, one cannot apply the well-known ADP methods for the system (1) directly in order to arrive at (3) and (4). Therefore, the introduction of action value functions is mandatory to handle the model-free case.

The action value function proposed by [28] evaluates both the current state and the command. It is defined as

$$Q(\boldsymbol{x}_k, \boldsymbol{u}_k) = r(\boldsymbol{x}_k, \boldsymbol{u}_k) + V(\boldsymbol{x}_{k+1}). \tag{5}$$

Compared to the value function (2), the action value function represents the cost of issuing a command $\boldsymbol{u}_k$ in a state $\boldsymbol{x}_k$, plus the value function of the next state $\boldsymbol{x}_{k+1}$. Mainly, Equation (5) evaluates all possible actions $\boldsymbol{u}_k \in \Omega_U$ followed by the controller $C(\boldsymbol{x}_{k+1})$. Equation (5) can also be written, according to [28], as

$$Q(\boldsymbol{x}_k, \boldsymbol{u}_k) = r(\boldsymbol{x}_k, \boldsymbol{u}_k) + Q(\boldsymbol{x}_{k+1}, C(\boldsymbol{x}_{k+1})). \tag{6}$$

From [28], similarly to the value function (3), the optimal action value function is defined as

$$Q^*(\boldsymbol{x}_k) = \min_{C(\boldsymbol{x}_k)} \{r(\boldsymbol{x}_k, \boldsymbol{u}_k) + Q^*(\boldsymbol{x}_{k+1}, C(\boldsymbol{x}_{k+1}))\}, \tag{7}$$

and the optimal controller is represented by

$$C^*(\boldsymbol{x}_k) = arg\min_{C(\boldsymbol{x}_k)} \{r(\boldsymbol{x}_k, \boldsymbol{u}_k) + Q^*(\boldsymbol{x}_{k+1}, C(\boldsymbol{x}_{k+1}))\}. \tag{8}$$

ADHDP Algorithm

Arriving at the optimal action value function and controller requires an iterative procedure consisting of j steps, where the action value function and controller are continuously updated, according to [28]. Starting with an initial controller $C_0(\boldsymbol{x}_k)$ and an action value function, e.g., $Q_0(\boldsymbol{x}_k, \boldsymbol{u}_k) = 0$, the action value function evaluation is issued by

$$Q_1(\boldsymbol{x}_k, \boldsymbol{u}_k) = r(\boldsymbol{x}_k, \boldsymbol{u}_k) + Q_0(\boldsymbol{x}_{k+1}, C_0(\boldsymbol{x}_{k+1})). \tag{9}$$

Then, the controller is updated using

$$C_1(\boldsymbol{x}_k) = arg\min_{C(\boldsymbol{x}_k)} \{r(\boldsymbol{x}_k, \boldsymbol{u}_k) + Q_0(\boldsymbol{x}_{k+1}, C(\boldsymbol{x}_{k+1}))\}. \tag{10}$$

At the j^{th} iteration, the action value function update is

$$Q_{j+1}(\boldsymbol{x}_k, \boldsymbol{u}_k) = r(\boldsymbol{x}_k, \boldsymbol{u}_k) + Q_j\big(\boldsymbol{x}_{k+1}, C_j(\boldsymbol{x}_{k+1})\big), \tag{11}$$

while the controller update law is

$$C_{j+1}(\boldsymbol{x}_k) = arg\min_{C(\boldsymbol{x}_k)} \big\{r(\boldsymbol{x}_k, \boldsymbol{u}_k) + Q_j(\boldsymbol{x}_{k+1}, C(\boldsymbol{x}_{k+1}))\big\}. \tag{12}$$

The iteration scheme consisting of the repetitive application of Equations (11) and (12) runs as $j \to \infty$.

Remark 1. *A policy iteration algorithm requires an initially known stabilizing controller $C_0(\boldsymbol{x}_k)$, whereas, for value iteration schemes, this need is avoided.*

In the next section, the implementation of the controller and action value function update is issued using a neural network function approximation for $Q_j(\boldsymbol{x}_k, \boldsymbol{u}_k)$ and $C_j(\boldsymbol{x}_k)$.

3. Neural Network Implementation for BOADHDP

The recurrent ADP scheme described by Equations (11) and (12) is practically implemented using function approximators for the action value function and controller. To

this end, neural networks (NNs) are used, due to their universal function approximation property, which is able to handle multidimensional nonlinear systems (1). The tuning of the NN weights from each individual layer requires both input–output training data and the employment of the backpropagation mechanism, which can be best described as a gradient-based update rule.

The training data for the controller and action value function are collected from the controlled system (1) and take the form of transition tuples $(\boldsymbol{x}_k,\ \boldsymbol{u}_k, r(\boldsymbol{x}_k,\boldsymbol{u}_k),\ \boldsymbol{x}_{k+1})$ stored in a dataset $D_M = \{(\boldsymbol{x}_k,\ \boldsymbol{u}_k, r(\boldsymbol{x}_k,\boldsymbol{u}_k),\ \boldsymbol{x}_{k+1})\}$, with $k = 1 : M$. The main objective of the data collection phase is to uniformly sample the state space $\Omega_X \times \Omega_U$, sufficiently exploring the systems' dynamics.

The action value function and controller NN weight tuning algorithm, using a gradient descent, is described in Sections 3.1 and 3.2. The weight gradient update uses the entirety of the collected transitions from D_M, compared to the methods from [21,22] which use only one transition per gradient update. This method is called batch optimization, and its utilization is a common practice for the application of RL and ADP applied to complex nonlinear systems.

For the batch learning implementation, the action value function and controller update is made simultaneously for the entire dataset D_M. Therefore, let $\mathbf{X}_p = [\boldsymbol{x}_1,\ldots,\boldsymbol{x}_{M-1}]$, $\mathbf{X}_f = [\boldsymbol{x}_2,\ldots,\boldsymbol{x}_M]$ of size $n \times (M-1)$, and $\mathbf{Y} = [\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{M-1}]$ of size $m \times (M-1)$ be vectors that lump all states and commands collected in the dataset D_M. Also, let $\boldsymbol{\Xi} = \begin{bmatrix} \mathbf{X}_p \\ \mathbf{Y} \end{bmatrix}$ be the matrix of the concatenation of the states and command matrices converted into a matrix resembling the action value function input.

Stating $\widetilde{Q}_j(\boldsymbol{x}_k,\ \boldsymbol{u}_k, \boldsymbol{W}_Q)$ and $\widetilde{C}_j(\boldsymbol{x}_k, \boldsymbol{W}_C)$ as the action value function and controller functions, respectively, approximated by NNs, and with $\hat{\boldsymbol{W}}_Q$ and $\hat{\boldsymbol{W}}_C$ representing the entirety of the action value function and controller weights, respectively, the gradient descend update is next detailed.

3.1. Action Value Function NN Approximation

The action value function NN has the scope of approximating (11). Having as inputs the state $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$, the state action function NN is described as

$$Q_j(\mathbf{X}_p,\ \mathbf{Y}, \boldsymbol{W}_Q) = Q\left(\mathbf{X}_p,\ \mathbf{Y}, \boldsymbol{W}_Q^j\right) = \boldsymbol{z}_Q^{L_Q} = \boldsymbol{W}_{Q,j}^{L_Q}\boldsymbol{\kappa}_Q^{L_Q-1}, \tag{13}$$

where

$$\boldsymbol{z}_Q^{l_Q} = \boldsymbol{W}_{Q,j}^{l_Q}\boldsymbol{\kappa}_Q^{l_Q-1}, \text{for } l_Q = 1,\ldots,L_Q,$$
$$\boldsymbol{\kappa}_Q^{l_Q} = \phi\left(\boldsymbol{z}_Q^{l_Q}\right),$$

and L_Q is the total number of layers, $\boldsymbol{W}_{Q,j}^{l} \in \mathfrak{R}^{h_Q^{l_Q} \times h_Q^{l_Q-1}}$ is the ideal hidden-layer weight matrix from the iteration j and layer l_Q, and $h_Q^{l_Q}$ is the number of neurons from layer l_Q. The size of $Q\left(\mathbf{X}_p,\ \mathbf{Y}, \boldsymbol{W}_Q^j\right)$ is $1 \times M$. Here, $\phi(\cdot) = \tanh(\cdot)$ represents the activation function and can take any form, such as $\tanh(\cdot)$, $\mathrm{ReLu}(\cdot)$, $\mathrm{sigmoid}(\cdot)$, and so on. The vector $\boldsymbol{\kappa}^l$ is the l_Q layer activation output. For the first layer, we have $\boldsymbol{\kappa}_Q^0 = \boldsymbol{\Xi}$.

Generally, weights ${\boldsymbol{W}_{Q,j}^{l_Q}}^T$, for $l_Q = 0,\ldots,\ L_Q$ are generally unknown due to the existing approximation errors in the weight update backpropagation rule. Hence, working with the real action value function $Q_j(\mathbf{X}_p,\ \mathbf{Y}, \boldsymbol{W}_Q)$ is not realistic, but only with some

approximations of it. Noting with $\hat{\boldsymbol{W}}_{Q,j}$ the entirety of the action value function weights, the output of the approximate action value function NN has the form of

$$\hat{Q}_j(\mathbf{X}_p,\ \mathbf{Y}, \hat{\boldsymbol{W}}_Q) = \hat{Q}\left(\mathbf{X}_p,\ \mathbf{Y}, \hat{\boldsymbol{W}}_Q^j\right) = \hat{\boldsymbol{z}}_Q^{L_Q} = \hat{\boldsymbol{W}}_{Q,j}^{L_Q} \hat{\boldsymbol{\kappa}}_Q^{L_Q-1}, \tag{14}$$

where

$$\boldsymbol{z}_Q^{l_Q} = \boldsymbol{W}_{Q,j}^{l_Q} \boldsymbol{\kappa}_Q^{l_Q-1}, \text{ for } l_Q = 1, \ldots, L_Q,$$
$$\boldsymbol{\kappa}_Q^{l_Q} = \phi\left(\boldsymbol{z}_Q^{l_Q}\right),$$

and where $\hat{\boldsymbol{W}}_{Q,j}^{l} \in \mathfrak{R}^{h_Q^{l_Q} \times h_Q^{l_Q-1}}$ represents an estimation of the ideal weights for $l_Q = 0, \ldots, L_Q$. To update the action value function NN weights, an internal gradient update loop is issued for the $i_Q = 0, \ldots, \mathrm{I}_Q$ steps, having the weights initialized with $\hat{\boldsymbol{W}}_{Q,j,0} = \hat{\boldsymbol{W}}_{Q,j}$. At each iteration i, the following optimization problem needs to be solved,

$$\hat{\boldsymbol{W}}_{Q,j,i_Q+1} = \underset{\hat{\boldsymbol{W}}}{\operatorname{argmin}} \frac{1}{M} \boldsymbol{E}_{Q,j,i_Q}, \tag{15}$$

where

$$\boldsymbol{E}_{Q,j,i_Q} = \boldsymbol{e}_{Q,j,i_Q} \boldsymbol{e}_{Q,j,i_Q}{}^T \tag{16}$$

and

$$\boldsymbol{e}_{Q,j,i_Q} = \left(\hat{Q}(\mathbf{X}_p,\ \mathbf{Y}, \hat{\boldsymbol{W}}) - \boldsymbol{\eta}_{Q,j,i_Q}\right)^T, \tag{17}$$

having $\boldsymbol{\eta}_{Q,j,i_Q} = r(\mathbf{X}_p, \mathbf{Y}) - \gamma \hat{Q}\left(\mathbf{X}_f, \widetilde{C}\left(\mathbf{X}_f, \hat{\boldsymbol{W}}_{C,j}\right), \hat{\boldsymbol{W}}_{Q,j,i_Q}\right)$.

Here, $\boldsymbol{e}_{Q,j,i_Q}$ represents the prediction error in the form of a TD error. The state action function weights are updated by the rule

$$\hat{\boldsymbol{W}}_{Q,j,\,i_Q+1}^{l_Q} = \hat{\boldsymbol{W}}_{Q,j,\,i_Q}^{l_Q} - \alpha_Q \frac{\partial \boldsymbol{E}_{Q,j,i_Q}}{\partial \hat{\boldsymbol{W}}_{Q,j,\,i_Q}^{l_Q}}, \tag{18}$$

where $\alpha_Q > 0$ is the action value function NN learning rate and

$$\frac{\partial \boldsymbol{E}_{Q,j,i_Q}}{\partial \hat{\boldsymbol{W}}_{Q,j,\,i_Q}^{l_Q}} = \frac{\partial \boldsymbol{E}_{Q,j,i_Q}}{\partial \hat{\boldsymbol{z}}_Q^{l_Q}} \frac{\partial \hat{\boldsymbol{z}}_Q^{l_Q}}{\partial \hat{\boldsymbol{W}}_{Q,j,\,i_Q}^{l_Q}} = \frac{\partial \boldsymbol{E}_{Q,j,i_Q}}{\partial \hat{\boldsymbol{z}}_Q^{l_Q}} \hat{\boldsymbol{\kappa}}_Q^{l_Q-1^T} \tag{19}$$

$$\frac{\partial \boldsymbol{E}_{Q,j,i_Q}}{\partial \hat{\boldsymbol{z}}_Q^{l_Q}} = \frac{\partial \boldsymbol{E}_{Q,j,i_Q}}{\partial \hat{\boldsymbol{z}}_Q^{l_Q+1}} \frac{\partial \hat{\boldsymbol{z}}_Q^{l_Q+1}}{\partial \boldsymbol{\kappa}_Q^{l_Q}} \frac{\partial \boldsymbol{\kappa}_Q^{l_Q}}{\partial \hat{\boldsymbol{z}}_Q^{l_Q}} = \hat{\boldsymbol{W}}_{Q,j,\,i_Q}^{l_Q+1^T} \frac{\partial \boldsymbol{E}_{Q,j,i_Q}}{\partial \hat{\boldsymbol{z}}_Q^{l_Q+1}} \bigodot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right) \tag{20}$$

The sign $\bigodot$ corresponds to the Hadamard product. Then, the weights $\hat{\boldsymbol{W}}_{Q,j+1}$ of $j+1$ are actualized as $\hat{\boldsymbol{W}}_{Q,j+1} = \hat{\boldsymbol{W}}_{Q,j,\mathrm{I}_Q}$.

3.2. Controller NN Approximation

The controller NN has the scope of approximating $C(\boldsymbol{x}_k)$. Noting with $\boldsymbol{W}_{C,j}$ the entirety of the controller weights, and having as input the state $\mathbf{X}_p$, the output is computed as

$$C_j(\mathbf{X}_p, \boldsymbol{W}_C) = C(\mathbf{X}_p, \boldsymbol{W}_{C,j}) = \boldsymbol{z}_C{}^{L_C} = \boldsymbol{W}_{C,j}^{L_C} \boldsymbol{\kappa}_C{}^{L_C-1}, \tag{21}$$

where

$$\boldsymbol{z}_C{}^{l_C} = \boldsymbol{W}_{C,j}^{l_C} \boldsymbol{\kappa}_C{}^{l_C-1}, \text{ for } l_C = 1, \ldots, L_C$$

$$\boldsymbol{\kappa}_C{}^{l_C} = \phi\left(\boldsymbol{z}_C{}^{l_C}\right),$$

and where $\boldsymbol{W}_{C,j}^{l_C} \in \Re^{h_Q^{l_C} \times h_Q^{l_C-1}}$ represents an estimation of the iteration j of the ideal weights for the $l_C = 0, \ldots, L_C$ layers. Noting with $\hat{\boldsymbol{W}}_{C,j}$ the estimation of the ideal weights, the output of the controller NN is

$$\hat{C}_j(\mathbf{X}_p, \hat{\boldsymbol{W}}_C) = \hat{C}(\mathbf{X}_p, \hat{\boldsymbol{W}}_{C,j}) = \hat{\boldsymbol{z}}_C^{L_C} = \hat{\boldsymbol{W}}_{C,j}^{L_C} \hat{\boldsymbol{\kappa}}_C^{L_C-1} \tag{22}$$

with

$$\hat{\boldsymbol{z}}_C^{l_C} = \hat{\boldsymbol{W}}_{C,j}^{l_C} \hat{\boldsymbol{\kappa}}_C^{l_C-1}, \text{ for } l_C = 1, \ldots, L_C$$

$$\hat{\boldsymbol{\kappa}}_C^{l_C} = \phi\left(\hat{\boldsymbol{z}}_C^{l_C}\right)$$

and where $\hat{\boldsymbol{W}}_{C,j}^{l_C}$ represents an estimation of the real weights. To update the controller weights, one needs to issue an internal gradient update loop for the $i_C = 0, \ldots, \mathrm{I}_C$ steps, having the weights initialized with $\hat{\boldsymbol{W}}_{C,j,0} = \hat{\boldsymbol{W}}_{C,j}$. At each iteration i_C, the following optimization problem needs to be minimized for the entirety of the collected dataset, as follows,

$$\hat{\boldsymbol{W}}_{C,j,i_C} = \underset{\hat{W}}{\operatorname{argmin}} \frac{1}{M} \boldsymbol{E}_{C,j,i_C} \tag{23}$$

where

$$\boldsymbol{E}_{C,j,i_C} = \boldsymbol{e}_{C,j,i_C} \boldsymbol{e}_{C,j,i_C}{}^T \tag{24}$$

and

$$\boldsymbol{e}_{C,j,i_C} = \hat{Q}(\mathbf{X}_p,\ \hat{C}(\mathbf{X}_p, \hat{\boldsymbol{W}}), \hat{\boldsymbol{W}}_{Q,j+1}) \tag{25}$$

where $\alpha_C > 0$ represents the controller NN learning rate.

The update of each individual weights is

$$\hat{\boldsymbol{W}}_{C,j,\ i_C+1}^{l_C} = \hat{\boldsymbol{W}}_{C,j,\ i_C}^{l_C} - \alpha_C \frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{W}}_{C,j,\ i_C}^{l_C}} \tag{26}$$

where

$$\frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{W}}_{C,j,\ i_C}^{l_C}} = \frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{z}}_C^{l_C}} \frac{\partial \hat{\boldsymbol{z}}_C^{l_C}}{\partial \hat{\boldsymbol{W}}_{Q,j,\ i_C}^{l_C}} = \frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{z}}_C^{l_C}} \hat{\boldsymbol{\kappa}}_C^{l_C-1^T} \tag{27}$$

$$\frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{z}}_C^{l_C}} = \frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{z}}_C^{l_C+1}} \frac{\partial \hat{\boldsymbol{z}}_C^{l_C+1}}{\partial \boldsymbol{\kappa}^{l_C}} \frac{\partial \boldsymbol{\kappa}^{l_C}}{\partial \hat{\boldsymbol{z}}_C^{l_C}} = \hat{\boldsymbol{W}}_{C,j,\ i_C}^{l_C+1^T} \frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{z}}_C^{l_C+1}} \bigodot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right). \tag{28}$$

To issue the update (28), it is necessary to compute the gradient of the action value function with respect to the controller output. This is computed as

$$\frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{z}}_C^{L_C}} = \frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{C}(\mathbf{X}_p, \hat{\boldsymbol{W}}_{C,j,i_C})} = \frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{z}}_Q^1} \frac{\partial \hat{\boldsymbol{z}}_Q^1}{\partial \hat{\boldsymbol{\kappa}}_Q^0} \frac{\partial \hat{\boldsymbol{\kappa}}_Q^0}{\partial \hat{C}(\mathbf{X}_p, \hat{\boldsymbol{W}}_{C,j,i_C})} = \boldsymbol{\Psi}^{\mathrm{T}} \left(\hat{\boldsymbol{W}}_{C,j,i}^{1^T} \frac{\partial \boldsymbol{E}_{C,j,i_C}}{\partial \hat{\boldsymbol{z}}_Q^1} \right) = \boldsymbol{\Omega}_{j,i} \tag{29}$$

where $\boldsymbol{\Psi} = \begin{bmatrix} \mathbf{0}_{n \times m} \\ \boldsymbol{I}_m \end{bmatrix}$ and $\boldsymbol{I}_m$ are the identity matrix, of dimensions $m \times m$, and $\mathbf{0}_{n \times m}$, a $n \times m$ matrix of zeros. Then, the weights $\hat{\boldsymbol{W}}_{C,j+1}$ of $j+1$ are actualized as $\hat{\boldsymbol{W}}_{C,j+1} = \hat{\boldsymbol{W}}_{C,j,\mathrm{I}_C}$.

3.3. Batch Offline ADHDP with Multiple-Hidden-Layer NN Algorithm

Next, the BOADHDP algorithm using multiple-hidden-layer NN function approximators is detailed. The algorithm consists of consecutive steps where the action value and controller NNs are updated.

1. Initialize α_Q, α_C, $\mathrm{I_Q}$, $\mathrm{I_C}$, Δ_Q. Initialize the NN architectures for $\hat{Q}_j(\mathbf{X}_p, \mathbf{Y}, \hat{\boldsymbol{W}}_Q)$ and $\hat{C}_j(\mathbf{X}_p, \hat{\boldsymbol{W}}_C)$ by setting L_Q, L_C, and their respective weights. Let $j = 0$ and $i_Q = i_C = 0$.
2. Collect M transitions from system (1) and construct the database D_M.
3. At iteration j, set $i_Q = 0$ and $\hat{\boldsymbol{W}}_{Q,j,i_Q} = \hat{\boldsymbol{W}}_{Q,j}$. Then, update the weights from all L_Q layers using (18) for $i_Q = \overline{0, \mathrm{I_Q}}$. Finally, set $\hat{\boldsymbol{W}}_{Q,j+1} = \hat{\boldsymbol{W}}_{Q,j,\mathrm{I_Q}}$.
4. Set $i_C = 0$ and $\hat{\boldsymbol{W}}_{C,j,i_C} = \hat{\boldsymbol{W}}_{C,j}$. Then, update the weights from all L_C layers using (26) for $i_C = \overline{0, \mathrm{I_C}}$. Finally, set $\hat{\boldsymbol{W}}_{C,j+1} = \hat{\boldsymbol{W}}_{C,j,\mathrm{I_C}}$.
5. If the condition $\left\| \hat{\boldsymbol{W}}_{Q,j} - \hat{\boldsymbol{W}}_{Q,j-1} \right\| < \Delta_Q$ is not met, update $j = j + 1$ and go to Step 3. Else, stop the iterative algorithm.

4. UUB Convergence

In this section, the convergence of the NN weights to a fixed point is examined. By using a Lyapunov function, the stability of the weight evolution to the fixed point is proved to be UUB under some specific conditions.

4.1. Lyapunov Approach Description

Each iteration j of the BOADHDP algorithm consists of a total cumulated number of $\mathrm{I} = \mathrm{I_Q} + \mathrm{I_C}$ gradient steps for both action value function and controller. Let a new iteration index be defined as $i = 1 : j * \mathrm{I}$, namely $i \in \left[1, \ldots, \mathrm{I_Q}, \mathrm{I_Q} + 1, \ldots, \mathrm{I}, \mathrm{I} + 1, \ldots, \mathrm{I} + \mathrm{I_Q}, \mathrm{I} + \mathrm{I_Q} + 1, \ldots, 2\mathrm{I}, \ldots\right]$, which represents a fine-grained iteration over both gradient action value function and controller. During $i \in \left[j\mathrm{I}, j\mathrm{I} + \mathrm{I_Q}\right]$, only the action value function neural network weights $\hat{\boldsymbol{W}}_{Q,j,i}$ are updated using (18), while $\hat{\boldsymbol{W}}_{C,j,i}$ remains unchanged. On the other side, for $i \in \left[j\mathrm{I} + \mathrm{I_Q}, j\mathrm{I} + \mathrm{I_Q} + \mathrm{I_C}\right]$, only the controller weights $\hat{\boldsymbol{W}}_{C,j,i}$ are updated using (26), while the action value function weights $\hat{\boldsymbol{W}}_{Q,j,i}$ remain unchanged. To simplify the notation, we substitute $\hat{\boldsymbol{W}}_{Q,j,i}$ and $\hat{\boldsymbol{W}}_{C,j,i}$ with $\hat{\boldsymbol{W}}_{Q,i}$ and $\hat{\boldsymbol{W}}_{C,i}$, respectively.

Let $\boldsymbol{W}_Q^*$ and $\boldsymbol{W}_C^*$ represent the optimal weights of the action value function NN and the controller NN, and let the weight estimation errors between the approximation of the real weights and the optimal ones be $\overline{\boldsymbol{W}}_{Q,i} = \hat{\boldsymbol{W}}_{Q,i} - \boldsymbol{W}_Q^*$, $\overline{\boldsymbol{W}}_{C,i} = \hat{\boldsymbol{W}}_{C,i} - \boldsymbol{W}_C^*$.

Therefore, the difference between the estimated weights and the optimal ones at each layer of both the action value function and the controller NN at each iteration i is, according to (18) and (26),

$$\overline{\boldsymbol{W}}_{Q,i+1}^{l_Q} = \hat{\boldsymbol{W}}_{Q,i+1}^{l_Q} - \boldsymbol{W}_Q^{l_Q,*} = \hat{\boldsymbol{W}}_{Q,i}^{l_Q} - \alpha_Q \frac{\partial \boldsymbol{E}_{Q,i}}{\partial \hat{\boldsymbol{W}}_{Q,i}^{l_Q}} - \boldsymbol{W}_Q^{l_Q,*} = \overline{\boldsymbol{W}}_{Q,i}^{l_Q} - \alpha_Q \frac{\partial \boldsymbol{E}_{Q,i}}{\partial \hat{\boldsymbol{W}}_{Q,i}^{l_Q}} \tag{30}$$

$$\overline{\boldsymbol{W}}_{C,i+1}^{l_C} = \hat{\boldsymbol{W}}_{C,i+1}^{l_C} - \boldsymbol{W}_C^{l_C,*} = \hat{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \frac{\partial \boldsymbol{E}_{C,i}}{\partial \hat{\boldsymbol{W}}_{C,i}^{l_C}} - \boldsymbol{W}_C^{l_C,*} = \overline{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \frac{\partial \boldsymbol{E}_{C,i}}{\partial \hat{\boldsymbol{W}}_{C,i}^{l_C}}. \tag{31}$$

Then, based on (14), (18), (22), and (29), define the following dynamical system with the nonlinear difference equation system, where P represents a nonlinear function,

$$\left\{ \begin{bmatrix} \overline{\boldsymbol{W}}_{Q,i+1} \\ \overline{\boldsymbol{W}}_{C,i+1} \end{bmatrix} = \begin{bmatrix} \overline{\boldsymbol{W}}_{Q,i} \\ \overline{\boldsymbol{W}}_{C,i} \end{bmatrix} - P \begin{pmatrix} \hat{\boldsymbol{W}}_{Q,i}, \hat{\boldsymbol{W}}_{Q,i-1}, \phi\left(\hat{\boldsymbol{W}}_{Q,i}^1 \Xi\right), \phi\left(\hat{\boldsymbol{W}}_{Q,i-1}^1 \Xi\right), \ldots, \phi\left(\hat{\boldsymbol{W}}_{Q,i}^{L_Q} \hat{\boldsymbol{\kappa}}_Q^{L_Q - 1}\right), \phi\left(\hat{\boldsymbol{W}}_{Q,i-1}^{L_Q} \hat{\boldsymbol{\kappa}}_Q^{L_Q - 1}\right) \\ \hat{\boldsymbol{W}}_{C,i}, \hat{\boldsymbol{W}}_{C,i}, \phi\left(\hat{\boldsymbol{W}}_{C,i}^1 \mathbf{X}_p\right), \phi\left(\hat{\boldsymbol{W}}_{C,i-1}^1 \mathbf{X}_p\right), \ldots, \phi\left(\hat{\boldsymbol{W}}_{C,i}^{L_C} \hat{\boldsymbol{\kappa}}_C^{L_C - 1}\right), \phi\left(\hat{\boldsymbol{W}}_{C,i-1}^{L_C} \hat{\boldsymbol{\kappa}}_C^{L_C - 1}\right) \end{pmatrix}. \tag{32}$$

Definition 1. *The equilibrium point of a system (32) is said to be uniformly ultimately bounded (UUB) with bound $\chi > 0$ if, for any $\psi > 0$ and $i_0 > 0$, there exists a positive number $N = N(\psi, \chi)$ independent of i_0, such that $\left\| \begin{matrix} \overline{\boldsymbol{W}}_{Q,i} \\ \overline{\boldsymbol{W}}_{C,i} \end{matrix} \right\| \leq \chi$ for all $i \geq N + i_0$ when $\left\| \begin{matrix} \overline{\boldsymbol{W}}_{Q,i_0} \\ \overline{\boldsymbol{W}}_{C,i_0} \end{matrix} \right\| \leq \psi$.*

4.2. Preliminary Results

In the following, the UUB property of the system (32) is demonstrated for the update rules (18) and (26), both of which make the weights of the two approximating NNs enter a region with the center in the optimal weights $\boldsymbol{W}_Q^*$ and $\boldsymbol{W}_C^*$. Some fundamental assumptions are next introduced.

Assumption 1. *The optimal NN weights for the action value function and the controller and the activation function $\phi(\cdot)$ are bounded by positive constants, i.e., $\left\| W_Q^* \right\| \leq W_{Q,max}^*$, $\left\| W_C^* \right\| \leq W_{C,max}^*$, $\|\phi(\cdot)\| \leq \phi_{max}$.*

Lemma 1. *Under Assumption 1, it is implied that the first difference of $\Gamma_{Q,i}^{L_Q} = \frac{1}{\alpha_Q} tr\left\{ \overline{\boldsymbol{W}}_{Q,i}^{L_Q\,T} \overline{\boldsymbol{W}}_{Q,i}^{L_Q} \right\}$ is given by*

$$\Delta\Gamma_{Q,i}^{L_Q} = -tr\left\{ 2\overline{\boldsymbol{W}}_{Q,i}^{L_Q} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right) \boldsymbol{e}_{Q,i}{}^T \right\} + \alpha_Q tr\left\{ \boldsymbol{e}_{Q,i} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right)^T \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right) \boldsymbol{e}_{Q,i}{}^T \right\}. \tag{33}$$

Proof. Let $\Delta\Gamma_{Q,i}^{L_Q}$ be described as

$$\Delta\Gamma_{Q,i}^{L_Q} = \frac{1}{\alpha_Q} tr\left\{ \overline{\boldsymbol{W}}_{Q,i+1}^{L_Q\,T} \overline{\boldsymbol{W}}_{Q,i+1}^{L_Q} - \overline{\boldsymbol{W}}_{Q,i}^{L_Q\,T} \overline{\boldsymbol{W}}_{Q,i}^{L_Q} \right\}. \tag{34}$$

Using (19), (20), and (30), we get

$$\overline{\boldsymbol{W}}_{Q,i+1}^{L_Q} = \overline{\boldsymbol{W}}_{Q,i}^{L_Q} - \alpha_Q \frac{\partial \boldsymbol{E}_{Q,i}}{\partial \hat{\boldsymbol{W}}_{Q,\,i}^{L_Q}} = \overline{\boldsymbol{W}}_{Q,i}^{L_Q} - \alpha_Q \frac{\partial \boldsymbol{E}_{Q,i}}{\partial \boldsymbol{e}_{Q,i}} \frac{\partial \boldsymbol{e}_{Q,i}}{\partial \hat{\boldsymbol{z}}^L} \frac{\partial \hat{\boldsymbol{z}}^L}{\partial \hat{\boldsymbol{W}}_{Q,\,i}^{L_Q}} = \overline{\boldsymbol{W}}_{Q,i}^{L_Q} - \alpha_Q \boldsymbol{e}_{Q,i} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right)^T \tag{35}$$

Based on this, we have

$$\begin{aligned} \Delta\Gamma_{Q,i}^{L_Q} &= \tfrac{1}{\alpha_Q} tr\left\{ \left(\overline{\boldsymbol{W}}_{Q,i}^{L_Q} - \alpha_Q \boldsymbol{e}_{Q,i} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right)^T \right)^T \left(\overline{\boldsymbol{W}}_{Q,i}^{L_Q} - \alpha_Q \boldsymbol{e}_{Q,i} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right)^T \right) - \overline{\boldsymbol{W}}_{Q,i}^{L_Q\,T} \overline{\boldsymbol{W}}_{Q,i}^{L_Q} \right\} \\ &= \tfrac{1}{\alpha_Q} tr\left\{ \left(\overline{\boldsymbol{W}}_{Q,i}^{L_Q\,T} - \alpha_Q \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right) \boldsymbol{e}_{Q,i}{}^T \right) \left(\overline{\boldsymbol{W}}_{Q,i}^{L_Q} - \alpha_Q \boldsymbol{e}_{Q,i} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right)^T \right) - \overline{\boldsymbol{W}}_{Q,i}^{L_Q\,T} \overline{\boldsymbol{W}}_{Q,i}^{L_Q} \right\} \\ &= \tfrac{1}{\alpha_Q} tr\left\{ -2\alpha_Q \overline{\boldsymbol{W}}_{Q,i}^{L_Q} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right) \boldsymbol{e}_{Q,i}{}^T + \alpha_Q^2 \boldsymbol{e}_{Q,i} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right)^T \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right) \boldsymbol{e}_{Q,i}{}^T \right\} \\ &= -tr\left\{ 2\overline{\boldsymbol{W}}_{Q,i}^{L_Q} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right) \boldsymbol{e}_{Q,i}{}^T \right\} + \alpha_Q tr\left\{ \boldsymbol{e}_{Q,i} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right)^T \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1} \right) \boldsymbol{e}_{Q,i}{}^T \right\}. \end{aligned} \tag{36}$$

□

Lemma 2. *Under Assumption 1, it is implied that the first difference of $\Gamma_{Q,i}^{l_Q} = \frac{1}{\alpha_Q} tr\left\{ \overline{\boldsymbol{W}}_{Q,i}^{l_Q\,T} \overline{\boldsymbol{W}}_{Q,i}^{l_Q} \right\}$, for $l_Q = \overline{1 : L_Q - 1}$ is given by*

$$\begin{aligned} &-tr\left\{ 2\overline{\boldsymbol{W}}_{Q,i}^{l_Q\,T} \left(\overline{\boldsymbol{W}}_{Q,i}^{l_Q+1\,T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q} \right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1} \right) \right\} + \\ &\quad tr\left\{ \alpha_Q \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1} \right)^T \left(\boldsymbol{\Phi}_i^{l_Q+1\,T} \hat{\boldsymbol{W}}_{Q,i}^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q} \right)^T \right) \left(\overline{\boldsymbol{W}}_{Q,i}^{l_Q+1\,T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q} \right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1} \right) \right\}. \end{aligned} \tag{37}$$

Proof. For any $\Delta\Gamma_{Q,i}^{l_Q}$, with $l_Q = \overline{1:L_Q-1}$, we have

$$\Delta\Gamma_{Q,i}^{l_Q} = \frac{1}{\alpha_Q} tr\left\{ \overline{\boldsymbol{W}}_{Q,i+1}^{l_Q}{}^T \overline{\boldsymbol{W}}_{Q,i+1}^{l_Q} - \overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T \overline{\boldsymbol{W}}_{Q,i}^{l_Q} \right\}. \tag{38}$$

Based on (19), (20), and (30), we get

$$\overline{\boldsymbol{W}}_{Q,i+1}^{l_Q} = \overline{\boldsymbol{W}}_{Q,i}^{l_Q} - \alpha_Q \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \bigodot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right), \tag{39}$$

with $\boldsymbol{\Phi}_i^{l_Q+1} = \frac{\partial \boldsymbol{E}_{Q,i}}{\partial \hat{\boldsymbol{z}}_Q^{l_Q+1}}$. Based on (38) and (39), one gets

$$\begin{aligned}
&\Delta\Gamma_{Q,i}^{l_Q} \\
&= \tfrac{1}{\alpha_Q} tr\left\{ \left(\overline{\boldsymbol{W}}_{Q,i}^{l_Q} - \alpha_Q \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right) \right)^T \left(\overline{\boldsymbol{W}}_{Q,i}^{l_Q} \right. \right. \\
&\left. \left. -\alpha_Q \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right) \right) - \overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T \overline{\boldsymbol{W}}_{Q,i}^{l_Q} \right\} \\
&= \tfrac{1}{\alpha_Q} tr\left\{ \left(\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T - \alpha_Q \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T \left(\boldsymbol{\Phi}_i^{l_Q+1^T} \hat{\boldsymbol{W}}_{Q,i}^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T \right) \right) \left(\overline{\boldsymbol{W}}_{Q,i}^{l_Q} \right. \right. \\
&\left. \left. -\alpha_Q \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right) \right) - \overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T \overline{\boldsymbol{W}}_{Q,i}^{l_Q} \right\} \\
&= \tfrac{1}{\alpha_Q} tr\left\{ -2\alpha_Q \overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right) \right. \\
&\left. +\alpha_Q^2 \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T \left(\boldsymbol{\Phi}_i^{l_Q+1^T} \hat{\boldsymbol{W}}_{Q,i}^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T \right) \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right) \right\} \\
&= -tr\left\{ 2\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right) \right\} \\
&+tr\left\{ \alpha_Q \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T \left(\boldsymbol{\Phi}_i^{l_Q+1^T} \hat{\boldsymbol{W}}_{Q,i}^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T \right) \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right) \right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right) \right\}.
\end{aligned} \tag{40}$$

□

Lemma 3. *Under Assumption 1, it is implied that the first difference of* $\Gamma_{C,i}^{L_C} = \frac{1}{\alpha_C} tr\left\{ \overline{\boldsymbol{W}}_{C,i}^{L_C}{}^T \overline{\boldsymbol{W}}_{C,i}^{L_C} \right\}$ *is given by*

$$-tr\left\{ 2\overline{\boldsymbol{W}}_{C,i}^{l_C} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right) \boldsymbol{\Omega_i}^T \right\} + \alpha_C tr\left\{ \boldsymbol{\Omega_i} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right)^T \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right) \boldsymbol{\Omega_i}^T \right\}. \tag{41}$$

Proof. Let $\Delta\Gamma_{C,i}^{L_C}$ be described as

$$\Delta\Gamma_{C,i}^{L_C} = \frac{1}{\alpha_C} tr\left\{ \overline{\boldsymbol{W}}_{C,i+1}^{L_C}{}^T \overline{\boldsymbol{W}}_{C,i+1}^{L_C} - \overline{\boldsymbol{W}}_{C,i}^{L_C}{}^T \overline{\boldsymbol{W}}_{C,i}^{L_C} \right\}. \tag{42}$$

Based on (27), (28), and (31), let

$$\overline{\boldsymbol{W}}_{C,i+1}^{l_C} = \overline{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \frac{\partial \boldsymbol{E}_{C,i}}{\partial \hat{\boldsymbol{W}}_{C,i}^{l_C}} = \overline{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \boldsymbol{\Omega_i} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right)^T. \tag{43}$$

Therefore,

$$\begin{aligned}\Delta\Gamma_{C,i}^{L_C} &= \tfrac{1}{\alpha_C} tr\left\{ \left(\overline{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \boldsymbol{\Omega_i} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right)^T \right)^T \left(\overline{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \boldsymbol{\Omega_i} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right)^T \right) - \overline{\boldsymbol{W}}_{C,i}^{L_C\,T} \overline{\boldsymbol{W}}_{C,i}^{L_C} \right\} = \\ &\tfrac{1}{\alpha_C} tr\left\{ \left(\overline{\boldsymbol{W}}_{C,i}^{l_C\,T} - \alpha_C \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right) \boldsymbol{\Omega_i}^T \right) \left(\overline{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \boldsymbol{\Omega_i} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right)^T \right) - \overline{\boldsymbol{W}}_{C,i}^{L_C\,T} \overline{\boldsymbol{W}}_{C,i}^{L_C} \right\} = \\ &\tfrac{1}{\alpha_C} tr\left\{ -2\alpha_C \overline{\boldsymbol{W}}_{C,i}^{l_C} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right) \boldsymbol{\Omega_i}^T + \alpha_C^2 \boldsymbol{\Omega_i} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right)^T \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right) \boldsymbol{\Omega_i}^T \right\} = \\ &-tr\left\{ 2\overline{\boldsymbol{W}}_{C,i}^{l_C} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right) \boldsymbol{\Omega_i}^T \right\} + \alpha_C tr\left\{ \boldsymbol{\Omega_i} \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right)^T \phi\left(\hat{\boldsymbol{z}}_C^{L_C-1}\right) \boldsymbol{\Omega_i}^T \right\}. \end{aligned} \tag{44}$$

□

Lemma 4. *Under Assumption 1, it is implied that the first difference of* $\Gamma_{C,i}^{l_C} = \frac{1}{\alpha_C} tr\left\{ \overline{\boldsymbol{W}}_{C,i}^{l_C\,T} \overline{\boldsymbol{W}}_{C,i}^{l_C} \right\}$, *for* $l_C = \overline{1 : L_C - 1}$, *is given by*

$$\begin{aligned} &-tr\left\{ 2\overline{\boldsymbol{W}}_{C,i}^{l_C\,T} \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right) \right\} \\ &+\alpha_C tr\left\{ \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right)^T \left(\boldsymbol{\chi}_i^{l_C+1\,T} \hat{\boldsymbol{W}}_{C,i}^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right)^T \right) \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right) \right\} \end{aligned} \tag{45}$$

Proof. For any $\Delta\Gamma_{C,i}^{l_C}$, with $l_C = \overline{1 : L_C - 1}$, we have

$$\Delta\Gamma_{C,i}^{l_C} = \frac{1}{\alpha_C} tr\left\{ \overline{\boldsymbol{W}}_{C,i+1}^{l_C\,T} \overline{\boldsymbol{W}}_{C,i+1}^{l_C} - \overline{\boldsymbol{W}}_{C,i}^{l_C\,T} \overline{\boldsymbol{W}}_{C,i}^{l_C} \right\} \tag{46}$$

Based on (27), (28), and (31), we get

$$\overline{\boldsymbol{W}}_{C,i+1}^{l_C} = \overline{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \bigodot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right), \tag{47}$$

with $\boldsymbol{\chi}_i^{l_C+1} = \frac{\partial \boldsymbol{E}_{C,i}}{\partial \hat{\boldsymbol{z}}_C^{l_C+1}}$. Based on (46) and (47), one gets

$$\begin{aligned} &\Delta\Gamma_{C,i}^{l_C} \\ &= \tfrac{1}{\alpha_C} tr\left\{ \left(\overline{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right) \right)^T \left(\overline{\boldsymbol{W}}_{C,i}^{l_C} \right. \right. \\ &\left. \left. -\alpha_C \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right) \right) - \overline{\boldsymbol{W}}_{C,i}^{l_C\,T} \overline{\boldsymbol{W}}_{C,i}^{l_C} \right\} \\ &= \tfrac{1}{\alpha_C} tr\left\{ \left(\overline{\boldsymbol{W}}_{C,i}^{l_C\,T} - \alpha_C \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right)^T \left(\boldsymbol{\chi}_i^{l_C+1\,T} \hat{\boldsymbol{W}}_{C,i}^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right)^T \right) \right) \left(\overline{\boldsymbol{W}}_{C,i}^{l_C} \right. \right. \\ &\left. \left. -\alpha_C \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right) \right) - \overline{\boldsymbol{W}}_{C,i}^{l_C\,T} \overline{\boldsymbol{W}}_{C,i}^{l_C} \right\} \\ &= \tfrac{1}{\alpha_C} tr\left\{ -2\alpha_C \overline{\boldsymbol{W}}_{C,i}^{l_C\,T} \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right) \right. \\ &\left. +\alpha_C^2 \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right)^T \left(\boldsymbol{\chi}_i^{l_C+1\,T} \hat{\boldsymbol{W}}_{C,i}^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right)^T \right) \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right) \right\} \\ &= -tr\left\{ 2\overline{\boldsymbol{W}}_{C,i}^{l_C\,T} \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right) \right\} \\ &+\alpha_C tr\left\{ \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right)^T \left(\boldsymbol{\chi}_i^{l_C+1\,T} \hat{\boldsymbol{W}}_{C,i}^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right)^T \right) \left(\hat{\boldsymbol{W}}_{C,i}^{l_C+1\,T} \boldsymbol{\chi}_i^{l_C+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_C^{l_C}\right) \right) \phi\left(\hat{\boldsymbol{z}}_C^{l_C-1}\right) \right\}. \end{aligned} \tag{48}$$

□

4.3. Main Stability Analysis

This section provides the main stability theory for the error estimation of system (32).

Theorem 1. *Running BOADHDP algorithm from Section 3.3, which iteratively updates* $\hat{\boldsymbol{W}}_{Q,i}$ *and* $\hat{\boldsymbol{W}}_{C,i}$ *using (18) and (26), the action value function and controller weights converge to their optimal weights* $\boldsymbol{W}_Q^*$ *and* $\boldsymbol{W}_C^*$*, respectively, such that* $\overline{\boldsymbol{W}}_{Q,i} \to 0$ *and* $\overline{\boldsymbol{W}}_{C,i} \to 0$ *if*

$$\alpha_Q < \frac{2\left(\overline{\boldsymbol{W}}_{Q,\,max}{}^2 \phi_{Q,max}{}^2 + \sum_{l_q=1}^{L_Q-1} \overline{\boldsymbol{W}}_{Q,\,max} \phi_{Q,max}^2 \hat{\boldsymbol{W}}_{Q,\,max} \prod_{l=l_Q}^{L_Q-1} \hat{\boldsymbol{W}}_{Q,\,max} \dot{\phi}_{Q,max}\right)}{\overline{\boldsymbol{W}}_{Q,\,max}{}^2 \phi_{Q,max}{}^4 + \sum_{l_q=1}^{L_Q-1} \hat{\boldsymbol{W}}_{Q,\,max}^2 \phi_{Q,max}{}^4 \prod_{l=l_Q}^{L_Q-1} \hat{\boldsymbol{W}}_{Q,\,max}^2 \dot{\phi}_{Q,max}{}^2} = \alpha_{Q,max} \tag{49}$$

$$\alpha_C < \frac{2\left(\overline{\boldsymbol{W}}_{C,\,max} \boldsymbol{\Omega}_{max} \phi_{max} + \sum_{l_C=1}^{L_C-1} \overline{\boldsymbol{W}}_{C,\,max} \phi_{max} \boldsymbol{\Omega}_{max} \prod_{l=l_C}^{L_C-1} \hat{\boldsymbol{W}}_{C,\,max} \dot{\phi}_{C,max}\right)}{\boldsymbol{\Omega}_{max}{}^2 \phi_{max}{}^2 + \sum_{l_C=1}^{L_C-1} \phi_{max}{}^2 \boldsymbol{\Omega}_{max}{}^2 \prod_{l=l_C}^{L_C-1} \hat{\boldsymbol{W}}_{C,\,max}^2 \dot{\phi}_{C,max}{}^2} = \alpha_{C,max}. \tag{50}$$

Proof. According to (18) and (26), we have, for each layer of action value function and controller NN,

$$\overline{\boldsymbol{W}}_{Q,i+1}^{l_Q} = \hat{\boldsymbol{W}}_{Q,i+1}^{l_Q} - \boldsymbol{W}_Q^{l_Q,*} = \hat{\boldsymbol{W}}_{Q,i}^{l_Q} - \alpha_Q \frac{\partial \boldsymbol{E}_{Q,i}}{\partial \hat{\boldsymbol{W}}_{Q,i}^{l_Q}} - \boldsymbol{W}_Q^{l_Q,*} = \overline{\boldsymbol{W}}_{Q,i}^{l_Q} - \alpha_Q \frac{\partial \boldsymbol{E}_{Q,i}}{\partial \hat{\boldsymbol{W}}_{Q,i}^{l_Q}}, \tag{51}$$

$$\overline{\boldsymbol{W}}_{C,i+1}^{l_C} = \hat{\boldsymbol{W}}_{C,i+1}^{l_C} - \boldsymbol{W}_C^{l_C,*} = \hat{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \frac{\partial \boldsymbol{E}_{C,i}}{\partial \hat{\boldsymbol{W}}_{C,i}^{l_C}} - \boldsymbol{W}_C^{l_C,*} = \overline{\boldsymbol{W}}_{C,i}^{l_C} - \alpha_C \frac{\partial \boldsymbol{E}_{C,i}}{\partial \hat{\boldsymbol{W}}_{C,i}^{l_C}}. \tag{52}$$

Let the Lyapunov function candidate be defined for each weight matrix according to each action value function and controller NN layer l_Q and l_C be described as

$$\Gamma_Q = \Gamma_{Q_i}^1 + \ldots + \Gamma_{Q,i}^{L_Q} = \frac{1}{\alpha_Q} tr\left\{\overline{\boldsymbol{W}}_{Q,i}^1{}^T \overline{\boldsymbol{W}}_{Q,i}^1\right\} + \ldots + \frac{1}{\alpha_Q} tr\left\{\overline{\boldsymbol{W}}_{Q,i}^{L_Q}{}^T \overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\} \tag{53}$$

$$\Gamma_C = \Gamma_{C,\mathrm{i}}^1 + \ldots + \Gamma_{C,\mathrm{i}}^{L_C} = \frac{1}{\alpha_C} tr\left\{\overline{\boldsymbol{W}}_{C,i}^1{}^T \overline{\boldsymbol{W}}_{C,i}^1\right\} + \ldots + \frac{1}{\alpha_C} tr\left\{\overline{\boldsymbol{W}}_{C,i}^{L_C}{}^T \overline{\boldsymbol{W}}_{C,i}^{L_C}\right\} \tag{54}$$

The joint action value function and controller Lyapunov function is

$$\Gamma = \Gamma_Q + \Gamma_C \tag{55}$$

Let the difference of the Lyapunov candidates be

$$\Delta\Gamma_Q = \Delta\Gamma_{Q,i}^1 + \ldots + \Delta\Gamma_{Q,i}^{L_Q} \tag{56}$$

$$\Delta\Gamma_C = \Delta\Gamma_{C,\mathrm{i}}^1 + \ldots + \Delta\Gamma_{C,\mathrm{i}}^{L_C}, \tag{57}$$

and the joint Lyapunov differences be $\Delta\Gamma = \Delta\Gamma_Q + \Delta\Gamma_C$.

Next, the proof is divided in two parts: one proving that $\Delta\Gamma_Q < 0$, if inequality (49) is respected, and one proving that $\Delta\Gamma_C < 0$, if inequality (50) is respected.

(a) Let $\Delta\Gamma_{Q,i}^{L_Q} = -tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{L_Q} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right) \boldsymbol{e}_{Q,i}{}^T\right\} + \alpha_Q tr\left\{\boldsymbol{e}_{Q,i} \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right) \boldsymbol{e}_{Q,i}{}^T\right\}$, according to Lemma 1, and $\Delta\Gamma_{Q,i}^{l_Q} = -tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1}{}^T \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\} + tr\left\{\alpha_Q \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T \left(\boldsymbol{\Phi}_i^{l_Q+1}{}^T \hat{\boldsymbol{W}}_{Q,i}^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T\right) \left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1}{}^T \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right) \phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\}$, for all layers $l_Q = \overline{1 : L_Q - 1}$, based on Lemma 2.

The sum $\Delta\Gamma_Q = \Delta\Gamma_{Q,i}^1 + \ldots + \Delta\Gamma_{Q,i}^{L_Q}$, $\forall l_Q = \overline{1 : L_Q - 1}$, is lower than 0 if

$$
\begin{aligned}
&\Delta\Gamma_Q \\
&= -tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\} + \alpha_Q tr\left\{\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\} + \dots \\
&-tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\} \\
&+tr\left\{\alpha_Q\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T\left(\boldsymbol{\Phi}_i^{l_Q+1^T}\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T\right)\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\} \\
&< 0 \\
&\Longleftrightarrow \alpha_Q tr\left\{\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\} + \dots \\
&+\alpha_Q tr\left\{\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T\left(\boldsymbol{\Phi}_i^{l_Q+1^T}\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T\right)\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\} \\
&< tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\} + \dots + tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\} \\
&\Longleftrightarrow \alpha_Q\left(tr\left\{\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\} + \dots\right. \\
&\left.+tr\left\{\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T\left(\boldsymbol{\Phi}_i^{l_Q+1^T}\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T\right)\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\}\right) \\
&< tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\} + \dots \\
&+tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\}
\end{aligned}
\tag{58}
$$

For the terms corresponding to layer L_Q from (58), we have

$$
tr\left\{\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\} \le \left\|\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|^2 \tag{59}
$$

Also, $tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\}$ is written as

$$
\begin{aligned}
&tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\} \\
&= tr\left\{\left(\overline{\boldsymbol{W}}_{Q,i}^{L_Q}{}^T + \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right)\left(\overline{\boldsymbol{W}}_{Q,i}^{L_Q} + \boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right)\right\} \\
&-tr\left\{\overline{\boldsymbol{W}}_{Q,i}^{L_Q}{}^T\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\} - tr\left\{\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\} \\
&\le \left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}{}^T + \phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\boldsymbol{e}_{Q,i}{}^T\right\|^2 - \left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\|^2 \\
&-\left\|\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|^2 \\
&\le \left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\|^2 + 2\left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\|\left\|\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\| \\
&+\left\|\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|^2 - \left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\|^2 - \left\|\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|^2 \\
&= 2\left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\|\left\|\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|.
\end{aligned}
\tag{60}
$$

Based on the TD error definition (17), we can write $\boldsymbol{e}_{Q,i} = \overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right) - \boldsymbol{\eta}_{Q,i}$. Then, (59) is described as

$$
\begin{aligned}
\left\|\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|^2 &= \left\|\left(\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right) - \boldsymbol{\eta}_{Q,i}\right)\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|^2 \\
&= \left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T - \boldsymbol{\eta}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|^2 \\
&\le \left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|^2.
\end{aligned}
\tag{61}
$$

Also, (60) is described as

$$\begin{aligned} 2\left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\|\left\|\boldsymbol{e}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\| &= 2\left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\|\left\|\left(\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)-\boldsymbol{\eta}_{Q,i}\right)\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\| = \\ 2\left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\|\left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T-\boldsymbol{\eta}_{Q,i}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\| &\leq \\ 2\left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\right\|\left\|\overline{\boldsymbol{W}}_{Q,i}^{L_Q}\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\phi\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)^T\right\|. \end{aligned} \tag{62}$$

For the terms corresponding to all layers $l_Q = \overline{1:L_Q-1}$ from (58), we have

$$\begin{aligned} tr\left\{\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T\left(\boldsymbol{\Phi}_i^{l_Q+1^T}\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T\right)\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\} \leq \\ \left\|\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\|^2\left\|\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\|^2. \end{aligned} \tag{63}$$

Also, the term $tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\}$ is described as

$$\begin{aligned} &tr\left\{2\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\} \\ &= tr\left\{\left(\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T+\alpha_Q\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T\left(\boldsymbol{\Phi}_i^{l_Q+1^T}\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T\right)\right)\left(\overline{\boldsymbol{W}}_{Q,i}^{l_Q}+\alpha_Q\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right)\right\} \\ &-tr\left\{\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T\overline{\boldsymbol{W}}_{Q,i}^{l_Q}\right\}-tr\left\{\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T\left(\boldsymbol{\Phi}_i^{l_Q+1^T}\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T\right)\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right)\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\} \\ &\leq\left\|\overline{\boldsymbol{W}}_{Q,i}^{l_Q}{}^T+\alpha_Q\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)^T\left(\boldsymbol{\Phi}_i^{l_Q+1^T}\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)^T\right)\right\|^2-\left\|\overline{\boldsymbol{W}}_{Q,i}^{l_Q}\right\|^2 \\ &-\left\|\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\|^2\left\|\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\|^2 \\ &\leq\left\|\overline{\boldsymbol{W}}_{Q,i}^{l_Q}\right\|^2+2\left\|\overline{\boldsymbol{W}}_{Q,i}^{l_Q}\right\|\left\|\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\|\left\|\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\| \\ &+\left\|\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\|^2\left\|\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\|^2-\left\|\overline{\boldsymbol{W}}_{Q,i}^{l_Q}\right\|^2 \\ &-\left\|\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\|^2\left\|\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\|^2 \\ &=2\left\|\overline{\boldsymbol{W}}_{Q,i}^{l_Q}\right\|\left\|\phi\left(\hat{\boldsymbol{z}}_Q^{l_Q-1}\right)\right\|\left\|\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\|. \end{aligned} \tag{64}$$

With $\boldsymbol{\Phi}_i^{l_Q+1}=\hat{\boldsymbol{W}}_{Q,i}^{l_Q+2^T}\boldsymbol{\Phi}_i^{l_Q+2}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q+1}\right)$, based on (20), we get, for all NN layers from $l_Q+1, l_Q+2, \ldots, L_Q$,

$$\begin{aligned} \left\|\hat{\boldsymbol{W}}_{Q,\,i}^{l_Q+1^T}\boldsymbol{\Phi}_i^{l_Q+1}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\| &= \left\|\hat{\boldsymbol{W}}_{Q,\,i}^{l_Q+1^T}\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+2^T}\boldsymbol{\Phi}_i^{l_Q+2}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q+1}\right)\right)\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\| \\ &= \left\|\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+2^T}\ldots\left(\hat{\boldsymbol{W}}_{Q,i}^{L_Q{}^T}\boldsymbol{\Phi}_i^{L_Q}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\right)\ldots\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q+1}\right)\right)\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\| \\ &= \left\|\hat{\boldsymbol{W}}_{Q,i}^{l_Q+1^T}\left(\hat{\boldsymbol{W}}_{Q,i}^{l_Q+2^T}\ldots\left(\hat{\boldsymbol{W}}_{Q,i}^{L_Q{}^T}\boldsymbol{e}_{Q,i}\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{L_Q-1}\right)\right)\ldots\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q+1}\right)\right)\odot\dot{\phi}\left(\hat{\boldsymbol{z}}_Q^{l_Q}\right)\right\|. \end{aligned} \tag{65}$$

Based on the normed Hadamard product property $\|A \odot B\| \leq \|A\| \cdot \|B\|$, with A and B being matrices of the same size, (65) is described as

$$\begin{aligned}
&\left\|\hat{W}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{z}_Q^{l_Q}\right)\right\| \\
&\leq \left\|\hat{W}_{Q,\,i}^{l_Q+1}\right\| \cdot \left\|\hat{W}_{Q,i}^{l_Q+2^T} \ldots \left(\hat{W}_{Q,i}^{L_Q{}^T} e_{Q,i} \odot \dot{\phi}\left(\hat{z}_Q^{L_Q-1}\right)\right) \ldots \odot \dot{\phi}\left(\hat{z}_Q^{l_Q+1}\right)\right\| \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q}\right)\right\| \\
&\leq \left\|\hat{W}_{Q,i}^{l_Q+1}\right\| \cdot \left\|\hat{W}_{Q,i}^{l_Q+2}\right\| \cdot \ldots \cdot \left\|\hat{W}_{Q,i}^{L_Q}\right\| \cdot \|e_{Q,i}\| \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{L_Q-1}\right)\right\| \cdot \ldots \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q+1}\right)\right\| \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q}\right)\right\| \\
&= \left\|\hat{W}_{Q,i}^{l_Q+1}\right\| \cdot \left\|\hat{W}_{Q,i}^{l_Q+2}\right\| \cdot \ldots \cdot \left\|\hat{W}_{Q,i}^{L_Q}\right\| \cdot \left\|\hat{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right) - \eta_{Q,i}\right\| \\
&\cdot \left\|\dot{\phi}\left(\hat{z}_Q^{L_Q-1}\right)\right\| \cdot \ldots \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q+1}\right)\right\| \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q}\right)\right\| \\
&\leq \left(\left\|\hat{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right)\right\| - \|\eta_{Q,i}\|\right) \cdot \left\|\hat{W}_{Q,i}^{l_Q+1}\right\| \cdot \left\|\hat{W}_{Q,i}^{l_Q+2}\right\| \cdot \ldots \cdot \left\|\hat{W}_{Q,i}^{L_Q}\right\| \\
&\cdot \left\|\dot{\phi}\left(\hat{z}_Q^{L_Q-1}\right)\right\| \cdot \ldots \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q+1}\right)\right\| \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q}\right)\right\| \\
&\leq \left\|\hat{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right)\right\| \cdot \left\|\hat{W}_{Q,i}^{l_Q+1}\right\| \cdot \left\|\hat{W}_{Q,i}^{l_Q+2}\right\| \cdot \ldots \cdot \left\|\hat{W}_{Q,i}^{L_Q}\right\| \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{L_Q-1}\right)\right\| \\
&\cdot \ldots \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q+1}\right)\right\| \cdot \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q}\right)\right\| \\
&= \left\|\hat{W}_{Q,i}^{L_Q}\right\| \left\|\phi\left(\hat{z}_Q^{L_Q-1}\right)\right\| \prod_{l=l_Q}^{L_Q-1} \left\|\hat{W}_{Q,i}^{l+1}\right\| \left\|\dot{\phi}\left(\hat{z}_Q^{l}\right)\right\|.
\end{aligned} \tag{66}$$

Therefore, based on (61)–(64), the inequality (58) is written as

$$\begin{aligned}
&\alpha_Q \left(\left\| \overline{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right) \phi\left(\hat{z}_Q^{L_Q-1}\right)^T \right\|^2 + \ldots + \left\|\phi\left(\hat{z}_Q^{l_Q-1}\right)\right\|^2 \left\| \hat{W}_{Q,i}^{l_Q+1^T} \boldsymbol{\Phi}_i^{l_Q+1} \odot \dot{\phi}\left(\hat{z}_Q^{l_Q}\right) \right\|^2 \right) \\
&< 2\left\|\overline{W}_{Q,i}^{L_Q}\right\| \left\| \overline{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right) \phi\left(\hat{z}_Q^{L_Q-1}\right)^T \right\| + \cdots + 2\left\|\overline{W}_{Q,i}^{l_Q}\right\| \left\|\phi\left(\hat{z}_Q^{l_Q-1}\right)\right\| \left\| \hat{W}_{Q,i}^{l_Q+1^T\}} bm \Phi_i^{l_Q+1} \odot \dot{\phi}\left(\hat{z}_Q^{l_Q}\right) \right\| \\
&\iff \alpha_Q \left(\left\| \overline{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right) \phi\left(\hat{z}_Q^{L_Q-1}\right)^T \right\|^2 + \ldots + \left\|\phi\left(\hat{z}_Q^{l_Q-1}\right)\right\|^2 \left\| \hat{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right) \right\|^2 \prod_{l=l_Q}^{2L_Q-1} \left\| \hat{W}_{Q,i}^{l+1} \right\|^2 \left\| \dot{\phi}\left(\hat{z}_Q^{l}\right) \right\|^2 \right) \\
&< 2\left\|\overline{W}_{Q,i}^{L_Q}\right\| \left\| \overline{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right) \phi\left(\hat{z}_Q^{L_Q-1}\right)^T \right\| + \cdots + 2\left\|\overline{W}_{Q,i}^{l_Q}\right\| \left\|\phi\left(\hat{z}_Q^{l_Q-1}\right)\right\| \left\| \hat{W}_{Q,i}^{L_Q} \right\| \left\| \phi\left(\hat{z}_Q^{L_Q-1}\right) \right\| \prod_{l=l_Q}^{L_Q-1} \left\| \hat{W}_{Q,i}^{l+1} \right\| \left\| \dot{\phi}\left(\hat{z}_Q^{l}\right) \right\| \\
&\iff \alpha_Q \left(\left\| \overline{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right) \phi\left(\hat{z}_Q^{L_Q-1}\right)^T \right\|^2 + \sum_{l_q=1}^{L_Q-1} \left\|\phi\left(\hat{z}_Q^{l_Q-1}\right)\right\|^2 \left\| \hat{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right) \right\|^2 \prod_{l=l_Q}^{2L_Q-1} \left\| \hat{W}_{Q,i}^{l+1} \right\|^2 \left\| \dot{\phi}\left(\hat{z}_Q^{l}\right) \right\|^2 \right) \\
&< 2\left\|\overline{W}_{Q,i}^{L_Q}\right\| \left\| \overline{W}_{Q,i}^{L_Q} \phi\left(\hat{z}_Q^{L_Q-1}\right) \phi\left(\hat{z}_Q^{L_Q-1}\right)^T \right\| + 2\sum_{l_q=1}^{L_Q-1} \left\|\overline{W}_{Q,i}^{l_Q}\right\| \left\|\phi\left(\hat{z}_Q^{l_Q-1}\right)\right\| \left\| \hat{W}_{Q,i}^{L_Q} \right\| \left\| \phi\left(\hat{z}_Q^{L_Q-1}\right) \right\| \prod_{l=l_Q}^{L_Q-1} \left\| \hat{W}_{Q,i}^{l+1} \right\| \left\| \dot{\phi}\left(\hat{z}_Q^{l}\right) \right\|.
\end{aligned} \tag{67}$$

Let the following norm bounds be defined as follows:

$$\left\|\overline{W}_{Q,i}^{l_Q}\right\| \leq \overline{W}_{Q,\,max}, \ \left\|\hat{W}_{Q,i}^{l_Q}\right\| \leq \hat{W}_{Q,\,max}, \ \left\|\dot{\phi}\left(\hat{z}_Q^{l_Q}\right)\right\| \leq \dot{\phi}_{Q,max}, \text{ for all } l_Q = \overline{1:L_Q}.$$

Then, based on Assumption 1, the inequality (67) can be written as

$$\begin{aligned}
&\alpha_Q \left(\overline{W}_{Q,\,max}{}^2 \phi_{Q,max}{}^4 + \textstyle\sum_{l_q=1}^{L_Q-1} \hat{W}_{Q,\,max}^2 \phi_{Q,max}{}^4 \prod_{l=l_Q}^{L_Q-1} \hat{W}_{Q,\,max}^2 \dot{\phi}_{Q,max}{}^2 \right) < \\
&2\overline{W}_{Q,\,max}{}^2 \phi_{Q,max}{}^2 + 2\textstyle\sum_{l_q=1}^{L_Q-1} \overline{W}_{Q,\,max} \phi_{Q,max}^2 \hat{W}_{Q,\,max} \prod_{l=l_Q}^{L_Q-1} \hat{W}_{Q,\,max} \dot{\phi}_{Q,max}.
\end{aligned} \tag{68}$$

To guarantee that (68) is negative, the learning rate needs to be selected as follows:

$$\alpha_Q < \frac{2\left(\overline{W}_{Q,\,max}{}^2 \phi_{Q,max}{}^2 + \sum_{l_q=1}^{L_Q-1} \overline{W}_{Q,\,max} \phi_{Q,max}^2 \hat{W}_{Q,\,max} \prod_{l=l_Q}^{L_Q-1} \hat{W}_{Q,\,max} \dot{\phi}_{Q,max} \right)}{\overline{W}_{Q,\,max}{}^2 \phi_{Q,max}{}^4 + \sum_{l_q=1}^{L_Q-1} \hat{W}_{Q,\,max}^2 \phi_{Q,max}{}^4 \prod_{l=l_Q}^{L_Q-1} \hat{W}_{Q,\,max}^2 \dot{\phi}_{Q,max}{}^2} = \alpha_{Q,max}. \tag{69}$$

(b) Let $\Delta\Gamma_{C,i}^{L_C} = -tr\left\{2\overline{W}_{C,i}^{l_C}\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\} + \alpha_C tr\left\{\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\}$, according to Lemma 3, and $\Delta\Gamma_{C,i}^{l_C} = -tr\left\{2\overline{W}_{C,i}^{l_C}{}^T\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\} + \alpha_C tr\left\{\phi\left(\hat{z}_C^{l_C-1}\right)^T\left(\chi_i^{l_C+1^T}\hat{W}_{C,i}^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)^T\right)\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}$ for all layers $l_C = \overline{1:L_C-1}$, based on Lemma 4.

The sum $\Delta\Gamma_C = \Delta\Gamma_{C,i}^1 + \ldots + \Delta\Gamma_{C,i}^{L_C}, \forall l_C = \overline{1:L_C-1}$, is lower than 0 if

$$
\begin{aligned}
&\Delta\Gamma_C\\
&= -tr\left\{2\overline{W}_{C,i}^{l_C}\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\} + \alpha_C tr\left\{\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\} + \ldots\\
&-tr\left\{2\overline{W}_{C,i}^{l_C}{}^T\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}\\
&+\alpha_C tr\left\{\phi\left(\hat{z}_C^{l_C-1}\right)^T\left(\chi_i^{l_C+1^T}\hat{W}_{C,i}^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)^T\right)\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\} < 0\\
&\Longleftrightarrow \alpha_C tr\left\{\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\} + \ldots\\
&+\alpha_C tr\left\{\phi\left(\hat{z}_C^{l_C-1}\right)^T\left(\chi_i^{l_C+1^T}\hat{W}_{C,i}^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)^T\right)\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}\\
&< tr\left\{2\overline{W}_{C,i}^{l_C}\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\} + \ldots + tr\left\{2\overline{W}_{C,i}^{l_C}{}^T\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}\\
&\Longleftrightarrow \alpha_C\left(tr\left\{\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\} +\right.\\
&\left.\ldots + tr\left\{\phi\left(\hat{z}_C^{l_C-1}\right)^T\left(\chi_i^{l_C+1^T}\hat{W}_{C,i}^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)^T\right)\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}\right)\\
&< tr\left\{2\overline{W}_{C,i}^{l_C}\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\} + \ldots + tr\left\{2\overline{W}_{C,i}^{l_C}{}^T\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}.
\end{aligned}
\tag{70}
$$

For the terms corresponding to layer L_C from (70), we have

$$
tr\left\{\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\} \le \left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|^2. \tag{71}
$$

Also, $tr\left\{2\overline{W}_{C,i}^{l_C}\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\}$ is described as

$$
\begin{aligned}
&tr\left\{2\overline{W}_{C,i}^{l_C}\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\}\\
&= tr\left\{\left(\overline{W}_{C,i}^{l_C}{}^T - \phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right)\left(\overline{W}_{C,i}^{l_C} - \boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right)\right\}\\
&-tr\left\{\overline{W}_{C,i}^{l_C}{}^T\overline{W}_{C,i}^{l_C}\right\} - \mathrm{tr}\left\{\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\phi\left(\hat{z}_C^{L_C-1}\right)\boldsymbol{\Omega_i}^T\right\}\\
&\le \left\|\overline{W}_{C,i}^{l_C} - \boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|^2 - \left\|\overline{W}_{C,i}^{l_C}\right\|^2 - \left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|^2\\
&\le \left\|\overline{W}_{C,i}^{l_C}\right\|^2 + 2\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\| + \left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|^2 - \left\|\overline{W}_{C,i}^{l_C}\right\|^2\\
&-\left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|^2 = 2\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|.
\end{aligned}
\tag{72}
$$

For the terms corresponding to all layers $l_C = \overline{1:L_C-1}$ from (70), we have

$$
\begin{aligned}
&tr\left\{\phi\left(\hat{z}_C^{l_C-1}\right)^T\left(\chi_i^{l_C+1^T}\hat{W}_{C,i}^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)^T\right)\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}\\
&\le \left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\|^2\\
&= \left\|\phi\left(\hat{z}_C^{l_C-1}\right)\right\|^2\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\right\|^2.
\end{aligned}
\tag{73}
$$

Also, the term $tr\left\{2\overline{W}_{C,i}^{l_C}{}^T\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}$ is described as

$$\begin{aligned}
&tr\left\{2\overline{W}_{C,i}^{l_C}{}^T\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}\\
&=tr\left\{\left(\overline{W}_{C,i}^{l_C}{}^T+\phi\left(\hat{z}_C^{l_C-1}\right)^T\left(\chi_i^{l_C+1^T}\hat{W}_{C,i}^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)^T\right)\right)\left(\overline{W}_{C,i}^{l_C}+\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right)\right\}\\
&-tr\left\{\overline{W}_{C,i}^{l_C}{}^T\overline{W}_{C,i}^{l_C}\right\}-tr\left\{\left(\hat{z}_C^{l_C-1}\right)^T\left(\chi_i^{l_C+1^T}\hat{W}_{C,i}^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)^T\right)\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\}\\
&\le\left\|\overline{W}_{C,i}^{l_C}{}^T+\phi\left(\hat{z}_C^{l_C-1}\right)^T\left(\chi_i^{l_C+1^T}\hat{W}_{C,i}^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)^T\right)\right\|^2-\left\|\overline{W}_{C,i}^{l_C}\right\|^2\\
&-\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\|^2\\
&\le\left\|\overline{W}_{C,i}^{l_C}\right\|^2+2\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\|\\
&+\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\|^2-\left\|\overline{W}_{C,i}^{l_C}\right\|^2\\
&-\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\phi\left(\hat{z}_C^{l_C-1}\right)\right\|^2\\
&=2\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\phi\left(\hat{z}_C^{l_C-1}\right)\right\|\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\right\|.
\end{aligned}\tag{74}$$

Having $\chi_i^{l_C+1}=\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C+1}\right)$, $\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\right\|$ can be written similarly to (65), as

$$\begin{aligned}
&\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\right\|\\
&=\left\|\hat{W}_{C,i}^{l_C+1^T}\left(\hat{W}_{C,i}^{l_C+2^T}\ldots\left(\hat{W}_{C,i}^{L_C}{}^T\boldsymbol{\Omega_i}\odot\dot{\phi}\left(\hat{z}_C^{L_C-1}\right)\right)\ldots\odot\dot{\phi}\left(\hat{z}_C^{l_C+1}\right)\right)\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right\|.
\end{aligned}\tag{75}$$

Based on the normed Hadamard product property, one gets

$$\begin{aligned}
&\left\|\hat{W}_{C,i}^{l_C+1^T}\left(\hat{W}_{C,i}^{l_C+2^T}\ldots\left(\hat{W}_{C,i}^{L_C}{}^T\boldsymbol{\Omega_i}\odot\dot{\phi}\left(\hat{z}_C^{L_C-1}\right)\right)\ldots\odot\dot{\phi}\left(\hat{z}_C^{l_C+1}\right)\right)\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right\|\\
&<\|\boldsymbol{\Omega_i}\|\cdot\left\|\hat{W}_{C,i}^{l_C+1}\right\|\cdot\left\|\hat{W}_{C,i}^{l_C+2}\right\|\cdot\ldots\cdot\left\|\hat{W}_{C,i}^{L_C}\right\|\cdot\left\|\dot{\phi}\left(\hat{z}_C^{L_C-1}\right)\right\|\cdot\ldots\cdot\left\|\dot{\phi}\left(\hat{z}_C^{l_C+1}\right)\right\|\\
&\cdot\left\|\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right\|=\|\boldsymbol{\Omega_i}\|\prod_{l=l_C}^{L_C-1}\left\|\hat{W}_{C,i}^{l+1}\right\|\left\|\dot{\phi}\left(\hat{z}_C^{l}\right)\right\|.
\end{aligned}\tag{76}$$

Therefore, based on (71)–(74), the inequality (70) is

$$\begin{aligned}
&\alpha_C\left(\left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|^2+\ldots+\left\|\phi\left(\hat{z}_C^{l_C-1}\right)\right\|^2\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\right\|^2\right)\\
&<2\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|+\ldots\\
&+2\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\phi\left(\hat{z}_C^{l_C-1}\right)\right\|\left\|\left(\hat{W}_{C,i}^{l_C+1^T}\chi_i^{l_C+1}\odot\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right)\right\|\\
&\Longleftrightarrow\alpha_C\left(\left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|^2+\ldots+\left\|\phi\left(\hat{z}_C^{l_C-1}\right)\right\|^2\|\boldsymbol{\Omega_i}\|^2\prod_{l=l_C}^{L_C-1}\left\|\hat{W}_{C,i}^{l+1}\right\|^2\left\|\dot{\phi}\left(\hat{z}_C^{l}\right)\right\|^2\right)\\
&<2\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|+\ldots\\
&+2\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\phi\left(\hat{z}_C^{l_C-1}\right)\right\|\|\boldsymbol{\Omega_i}\|\prod_{l=l_C}^{L_C-1}\left\|\hat{W}_{C,i}^{l+1}\right\|\left\|\dot{\phi}\left(\hat{z}_C^{l}\right)\right\|\\
&\Longleftrightarrow\alpha_C\left(\left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|^2+\sum_{l_C=1}^{L_C-1}\left\|\phi\left(\hat{z}_C^{l_C-1}\right)\right\|^2\|\boldsymbol{\Omega_i}\|^2\prod_{l=l_C}^{L_C-1}\left\|\hat{W}_{C,i}^{l_C+1}\right\|^2\left\|\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right\|^2\right)\\
&<2\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\boldsymbol{\Omega_i}\phi\left(\hat{z}_C^{L_C-1}\right)^T\right\|\\
&+2\sum_{l_C=1}^{L_C-1}\left\|\overline{W}_{C,i}^{l_C}\right\|\left\|\phi\left(\hat{z}_C^{l_C-1}\right)\right\|\|\boldsymbol{\Omega_i}\|\prod_{l=l_C}^{L_C-1}\left\|\hat{W}_{C,i}^{l+1}\right\|\left\|\dot{\phi}\left(\hat{z}_C^{l}\right)\right\|.
\end{aligned}\tag{77}$$

Let the following norm bounds be defined as follows:

$$\left\|\overline{W}_{C,i}^{l_C}\right\|\le\overline{W}_{C,\,max},\ \left\|\hat{W}_{C,i}^{l_C}\right\|\le\hat{W}_{C,\,max},\ \left\|\dot{\phi}\left(\hat{z}_C^{l_C}\right)\right\|\le\dot{\phi}_{C,max},\ \|\boldsymbol{\Omega_i}\|\le\boldsymbol{\Omega}_{max}\text{for all } l_C=\overline{1:L_C}.$$

Based on Assumption 1, the inequality (77) can be written as

$$\begin{aligned}
&\alpha_C\left(\boldsymbol{\Omega}_{max}{}^2\phi_{max}{}^2+\textstyle\sum_{l_C=1}^{L_C-1}\phi_{max}{}^2\boldsymbol{\Omega}_{max}{}^2\prod_{l=l_C}^{L_C-1}\hat{W}_{C,\,max}^2\dot{\phi}_{C,max}{}^2\right)\\
&<2\overline{W}_{C,\,max}\boldsymbol{\Omega}_{max}\phi_{max}+2\textstyle\sum_{l_C=1}^{L_C-1}\overline{W}_{C,\,max}\phi_{max}\boldsymbol{\Omega}_{max}\prod_{l=l_C}^{L_C-1}\hat{W}_{C,\,max}\dot{\phi}_{C,max}.
\end{aligned}\tag{78}$$

To guarantee that (78) is negative, the learning rate needs to be selected as follows:

$$\alpha_C<\frac{2\left(\overline{W}_{C,\,max}\boldsymbol{\Omega}_{max}\phi_{max}+\sum_{l_C=1}^{L_C-1}\overline{W}_{C,\,max}\phi_{max}\boldsymbol{\Omega}_{max}\prod_{l=l_C}^{L_C-1}\hat{W}_{C,\,max}\dot{\phi}_{C,max}\right)}{\boldsymbol{\Omega}_{max}{}^2\phi_{max}{}^2+\sum_{l_C=1}^{L_C-1}\phi_{max}{}^2\boldsymbol{\Omega}_{max}{}^2\prod_{l=l_C}^{L_C-1}\hat{W}_{C,\,max}^2\dot{\phi}_{C,max}{}^2}=\alpha_{C,max}.\tag{79}$$

In conclusion, by having the inequalities (69) and (79) respected, we get $\Delta\Gamma<0$. □

4.4. Results Interpretation

According to (69) and (79), as the number of hidden layers increases, the upper bounds for the learning rates α_Q and α_C decrease. This is due to the denominators in (69) and (79) being larger than their respective numerators, primarily because the denominators include squared terms. Therefore, the number of hidden layers in both neural networks is inversely proportional to the magnitude of their respective learning rates. For illustrative purposes, the action value function learning rate bound $\alpha_{Q,max}$ was plotted along the hidden layers $L_Q=1:15$ in Figure 1, based on (69). The norm bounds of the weights were selected as $\overline{W}_{Q,\,max}=\hat{W}_{Q,\,max}=2$ and $\phi_{Q,max}=\dot{\phi}_{Q,max}=1$ for the activation function $\phi(\cdot)=tanh(\cdot)$.

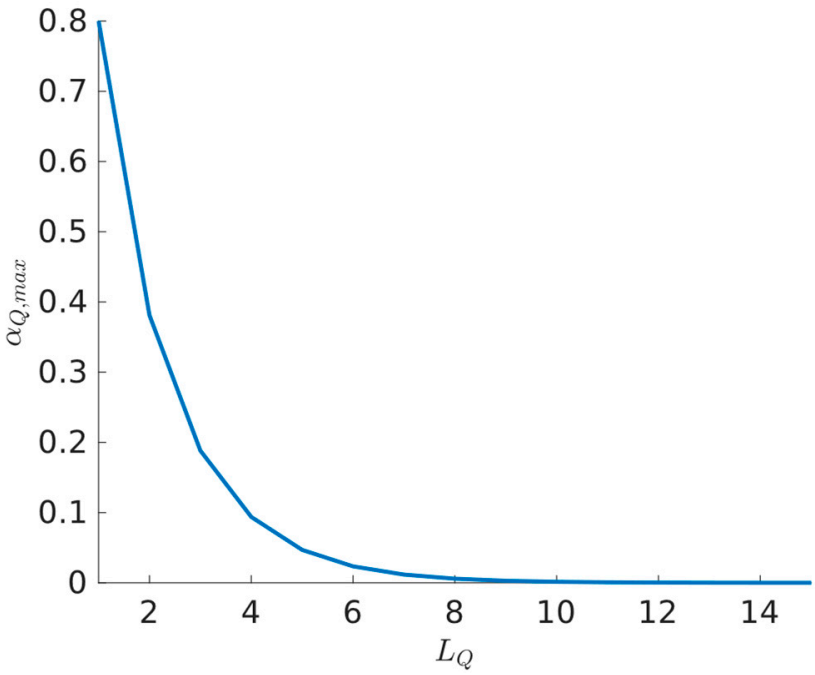

Figure 1. Relation between the number of NN layers and the bound of the learning rate $\alpha_{Q,max}$.

Remark 2. *This inversely proportional relationship between the number of NN hidden layers and the learning rate can be attributed to the gain in complexity of the NN optimization surface as the number of hidden layer increases. A high learning rate in such a scenario can lead to erratic updates in the intricate optimization surface, potentially causing the divergence of the learning process. While a smaller learning rate increases the risk of getting stuck in local minima, it is beneficial for a stable learning.*

5. Simulation Study

Next, the impact of employing multiple hidden layers in the NN approximators, batch learning, and offline computation in the ADHDP learning process, namely the BOADHDP algorithm from Section 3.3, was tested on an ORM tracking task on the TRAS system. First, the system is described along with the data collection settings for BOADHDP. This is followed by a comparison between the BOADHDP learning process using single-hidden-layer NNs and the one using two-hidden-layer NNs for approximating the action value function and the controller. Finally, the online adaptive ADHDHP algorithms from [21,22] are compared with BOADHDP, highlighting the advantages of the latter.

5.1. Data Collection Settings on TRAS System

The nonlinear system was characterized as a two-input and two-output system. The horizontal motion, or azimuth, operates as an integrator, whereas the vertical, or pitch, motion experiences different gravitational effects when moving upward versus downward. There was also an interconnection between these two channels. In Figure 2, a system setup

is shown. The model used was a simplified deterministic continuous-time state-space representation, consisting of two interconnected state-space subsystems:

$$\begin{cases} \dot{\omega}_h = \frac{(sat(U_h) - M_h(\omega_h))}{2.7} \cdot 10^{-5}, \\ K_h = (0.216F_h(\omega_h)\cos\alpha_v - 0.058\Omega_h + 0.0178sat(U_v)\cos\alpha_v), \\ \Omega_h = \frac{K_h}{(0.0238 \cdot \cos^2\alpha_v + 3 \cdot 10^{-3})}, \\ \dot{\alpha}_h = \Omega_h, \\ \dot{\omega}_v = \frac{(sat(U_v) - M_v(\omega_v))}{1.63} \cdot 10^{-4}, \\ \dot{\Omega}_v = \frac{1}{0.03} \begin{pmatrix} 0.2F_v(\omega_v) - 0.0127\Omega_v - 0.0935\sin\alpha_v \\ -9.28 \cdot 10^{-6}\Omega_v|\omega_v| + 4.17 \cdot 10^{-3}sat(U_h) - 0.05\cos\alpha_v \\ -0.021\Omega_h^2\sin\alpha_v\cos\alpha_v - 0.093\sin\alpha_v + 0.05 \end{pmatrix}, \\ \dot{\alpha}_v = \Omega_v, \end{cases} \tag{80}$$

where $sat()$ is the saturation function in the interval $[-1;\ 1]$. The horizontal azimuth control input was $U_h = u_1$ and the vertical pitch control was $U_v = u_2$. The system output was represented by the azimuth angle $\alpha_h \in [-\pi; \pi]$ and by the pitch angle $\alpha_v \in [-\pi/2;\ \pi/2]$. Nonlinear static characteristics were derived from experimental data through polynomial fitting as in [29]:

$$M_v(\omega) = 9.05 \times 10^{-12}\omega_v^3 + 2.76 \times 10^{-10}\,\omega_v^2 + 1.25 \times 10^{-4}\,\omega_v^1 + 1.66 \times 10^{-4}, \tag{81}$$

$$F_v(\omega) = -1.8 \times 10^{-18}\,\omega_v^5 - 7.8 \times 10^{-16}\,\omega_v^4 + 4.1 \times 10^{-11}\,\omega_v^3 + 2.7 \times 10^{-8}\,\omega^2 + 3.5 \times 10^{-4}\,\omega - 0.014, \tag{82}$$

$$M_h(\omega_h) = 5.95 \times 10^{-13}\,\omega_h^3 - 5.05 \times 10^{-10}\,\omega_h^2 + 1.02 \times 10^{-4}\,\omega_h^1 + 1.61 \times 10^{-3}\,\omega_h, \tag{83}$$

$$F_h(\omega_h) = -2.56 \times 10^{-20}\,\omega_h^5 + 4.09 \times 10^{-17}\,\omega_h^4 + 3.16 \times 10^{-12}\,\omega_h^3 - 7.34 \times 10^{-9}\,\omega_h^2 + 2.12 \times 10^{-5}\,\omega_h + 9.13 \times 10^{-3}. \tag{84}$$

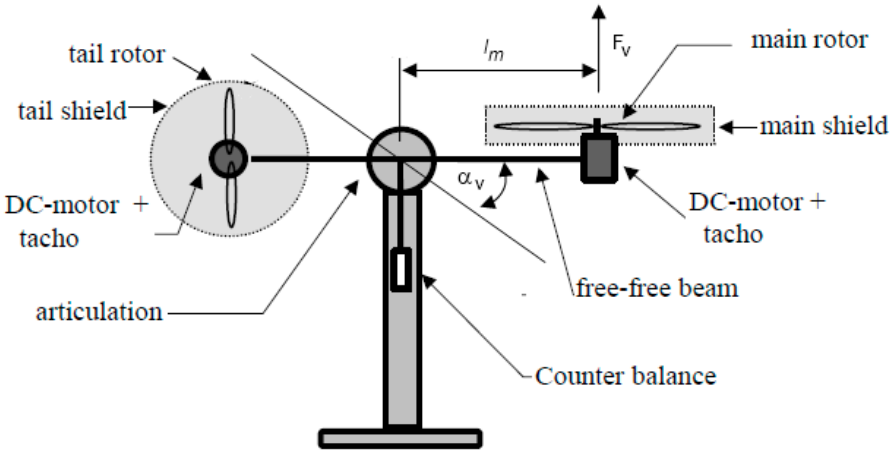

Figure 2. TRAS system setup [29].

The process was discretized by using a zero-order hold sampler on both inputs and outputs. With a sampling time of $T_s = 0.1$ s, the following discrete-time model was obtained,

$$\begin{cases} \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k), \\ \boldsymbol{y}_k = g(\boldsymbol{x}_k) = [\alpha_{k,h}, \alpha_{k,v}]^T, \end{cases} \tag{85}$$

where the system state was $\boldsymbol{x}_k = [\omega_{k,h}, \Omega_{k,h}, \alpha_{k,h}, \omega_{k,v}, \Omega_{k,v}, \alpha_{k,v}]^T \in \Re^6$ and the control input was $\boldsymbol{u}_k = [u_{k,h}, u_{k,v}]$, as in [29].

In the ORM tracking paradigm, the controlled system outputs track the output of the ORM model. In this application, the ORM was defined as in [29] and had the form of

$$\begin{cases} x^h_{k+1,m} = 0.9673 x^h_{k,m} + 0.0328 r_{k,h}, \\ x^v_{k+1,m} = 0.9673 x^v_{k,m} + 0.0328 r_{k,v}, \\ y_{k,m} = \left[y^h_{k,m}, y^v_{k,m}\right]^T = \left[x^h_{k,m}, x^v_{k,m}\right]^T, \end{cases} \tag{86}$$

where $r_{k,h}$ and $r_{k,v}$ are step input reference signals. Therefore, an extended state that comprises both the TRAS and the ORM states was defined as $\boldsymbol{x}^e_k = \left[\omega_{k,h}, \Omega_{k,h}, \alpha_{k,h}, \omega_{k,v}, \Omega_{k,v}, \alpha_{k,v}, x^h_{k,m}, x^v_{k,m}, r_{k,h}, r_{k,v}\right]^T \in \Re^{10}$.

For data collection, the linear diagonal controller

$$C(z, \boldsymbol{\theta}) = \begin{bmatrix} P_{11}(z)/\left(1 - z^{-1}\right) & 0 \\ 0 & P_{22}(z)/\left(1 - z^{-1}\right) \end{bmatrix},$$

$$P_{11}(z) = 2.9341 - 5.8689 z^{-1} + 3.9303 z^{-2} - 0.9173 z^{-3} - 0.0777 z^{-4},$$

$$P_{22}(z) = 0.6228 - 1.1540 z^{-1} + 0.5467 z^{-2} \tag{87}$$

was used in a closed loop with system (85), where the controller parameters were tuned using VRFT as in [29]. Having the closed loop stabilized, the successive step referenced input signals with amplitudes ranging in an interval of $r_{k,h} \in [-2; 2]$, and $r_{k,v} \in [-1.4; 1.1]$ were generated at 17 s and 25 s for the azimuth and pitch respectively. To guarantee a satisfactory exploration of the system's state-space domain, a random noise was added at each two timesteps. The random noise added on $C_{11}(Z)$ had an amplitude of $[-1.6; 1.6]$ and the one added on $C_{22}(Z)$ had an amplitude of $[-1.7; 1.7]$. A total of $M = 50,000$ transitions were collected, creating, therefore, the dataset $D_{50,000} = \left\{\left(\boldsymbol{x}^e_k,\ \boldsymbol{u}_k, r\left(\boldsymbol{x}^e_k, \boldsymbol{u}_k\right),\ \boldsymbol{x}^e_{k+1}\right)\right\}$, with $k = 1 : 50,000$. An excerpt of the data exploration is shown in Figure 3. Next, BOADHDP was issued for action value function and controller NN approximations for both the single-hidden-layer ($L_Q = 1$, $L_C = 1$) and the multilayer case ($L_Q = 2$, $L_C = 2$).

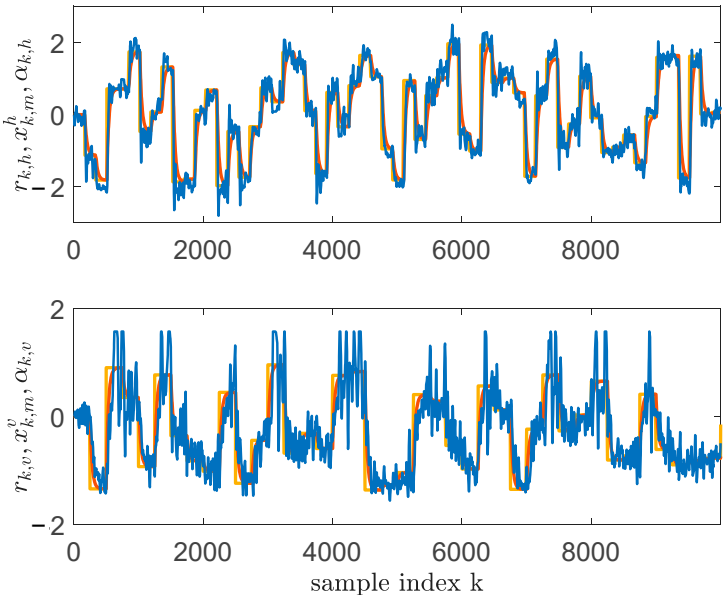

Figure 3. Data collection in relation to the TRAS system: $r_{k,h}$ and $r_{k,v}$ (yellow); $x^h_{k,m}$ and $x^v_{k,m}$ (red); $\alpha_{k,h}$ and $\alpha_{k,v}$ (blue).

5.2. Comparison of BOADHDP with Single-Layer and Multilayer NN Approximations

For the single-layer NNs, the form of the action value function was 12-50-1 and that of the controller was 10-10-2. The activation functions of the hidden layer were hyperbolic tangents and the ones of the output layer were linear. The weights were initialized using the Xavier initialization [29]. The internal gradient updates were $I_Q = 500$ and $I_C = 100$ and the learning rates were selected to be $\alpha_Q = 0.01$ and $\alpha_C = 0.001$. The penalty function took the form of $r(x_k^e, u_k) = \left(\alpha_{k,h} - x_{k,m}^h\right)^2 + \left(\alpha_{k,v} - x_{k,m}^v\right)^2$. The algorithm ran for a total number of 500 iterations. The performance of the NN controller was tested on a simulated scenario. In this scenario, the tracking capabilities were tested on a random reference signal generated from $[-1;1]$ for 2000 timesteps. Therefore, at each BOADHDP j^{th} iteration, the performance of the controller was measured by the function $J(x_k^e) = \left(\alpha_{k,h} - x_{k,m}^h\right)^2 + \left(\alpha_{k,v} - x_{k,m}^v\right)^2/2000$ on the simulated scenario, for $k = 1:2000$. The convergence of the action value function and the values of $J(x_k)$ is shown in Figure 4 in an orange color. This was computed by checking the norm between the weights from successive BOADHDP iterations, namely the norm $\left\|\hat{W}_Q^j - \hat{W}_Q^{j-1}\right\|_2^2$. The decreasing behavior of the successive weight norms from the first plot in Figure 4 proves the convergence of the action value function. The second plot presents the performance of the value function $J(x_k^e)$ under the simulated scenario for the controller obtained from each iteration j, namely $C_j(x_k^e, W_C)$. The tracking performance of the controller obtained at iteration $j = 500$ is shown in Figure 5. In this figure, the performance of the TRAS system (85) in a closed loop with the controller $C_{500}(x_k^e, W_C)$ is shown. The evolution in time of the output of the horizontal and the vertical axes is plotted in a blue color along with the reference signal (yellow) and reference model (orange), showing the tracking capacity of the $C_{500}(x_k^e, W_C)$ controller.

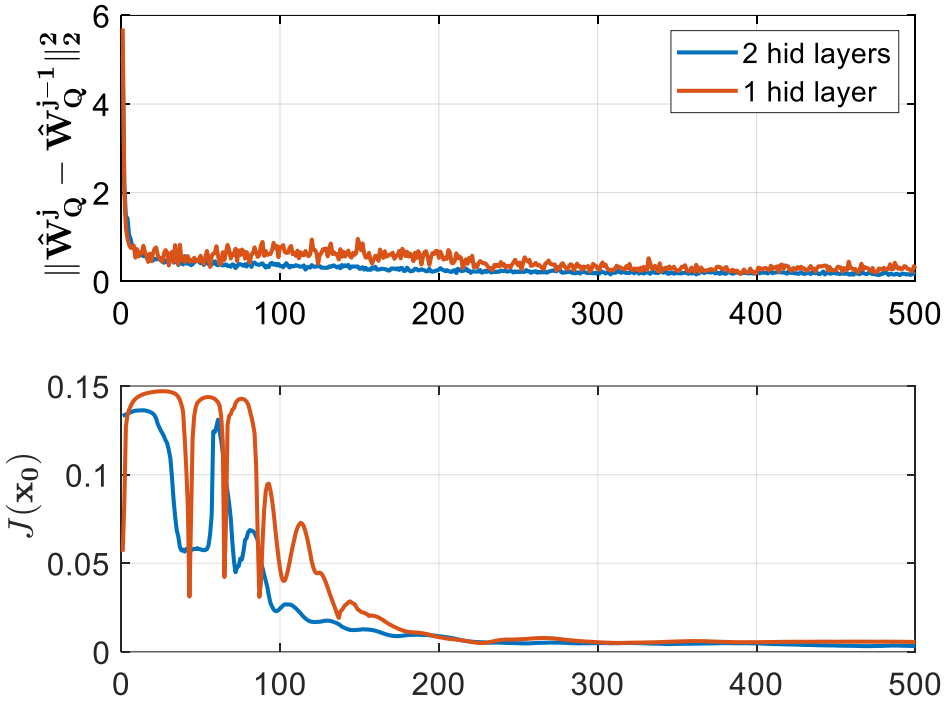

Figure 4. BOADHDP convergence in the TRAS system.

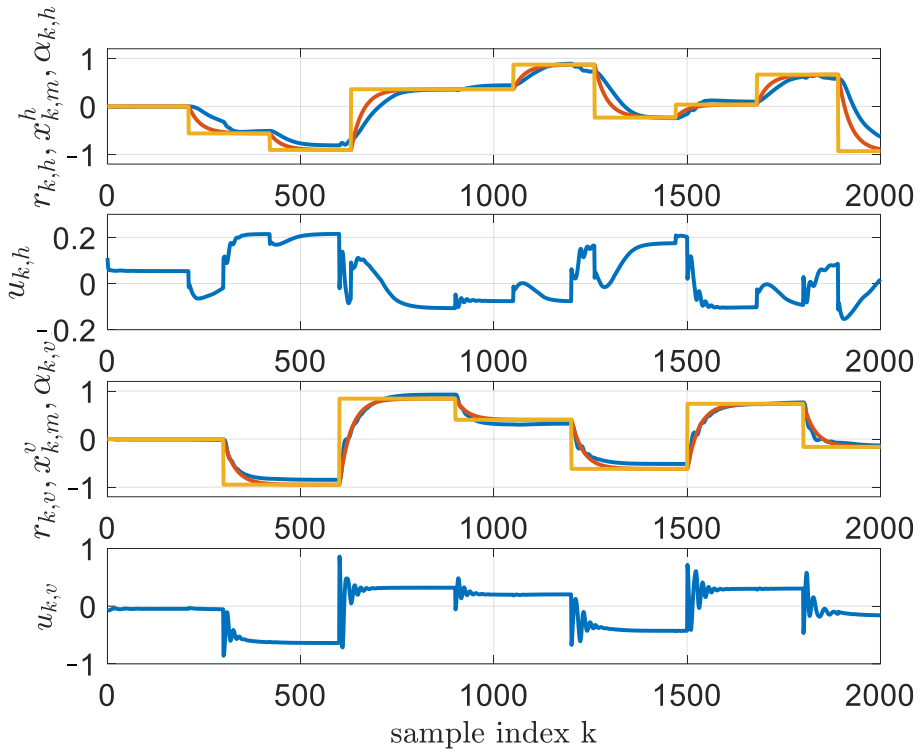

Figure 5. One-hidden-layer controller learned through BOADHDP, at iteration $j = 500$: $r_{k,h}$ and $r_{k,v}$ (yellow); $x^h_{k,m}$ and $x^v_{k,m}$ (red); $\alpha_{k,h}$ and $\alpha_{k,v}$ (blue). The commands $u_{k,h}$ and $u_{k,v}$ are for the horizontal and vertical axes (blue).

For the multilayer NN setup, the form of the action value function was 15-50-10-2 and that of the controller was 10-10-4-2. The activation functions of the two hidden layers were hyperbolic tangents and the ones from the output layer were linear. The weights were initialized using the Xavier initialization [29]. The internal gradient updates were $I_Q = 500$ and $I_C = 100$, and the learning rates took the values of $\alpha_Q = 0.01$ and $\alpha_C = 0.001$. The algorithm ran for a total number of 500 iterations. The convergence of the action value function and the values of $J(x_k)$ is shown in Figure 4 in a blue color. The tracking performance of the controller obtained at iteration $j = 500$ is shown in Figure 5.

From Figure 4, it can be seen that the convergence of the two-layer NN approximators for the action value function and the controller delivered more stable results. First, the norm of the action value function successive weight differences from the first plot was less noisy and provided a faster convergence in the two-layer case than the single-layer NN. Then, in the second plot, the function $J(x^e_k)$ converged faster to a lower value that correlated with a performant controller. Also, the values of $J(x^e_k)$ was 0.0049 for the single-layer NNs and 0.0031 for the two-layer implementation. The two-layer implementation outperformed the single-layer one by 1.58%. The difference in tracking performance can be seen in Figures 5 and 6, where the horizontal motion tracking improved in the case of the two-layer NN controller.

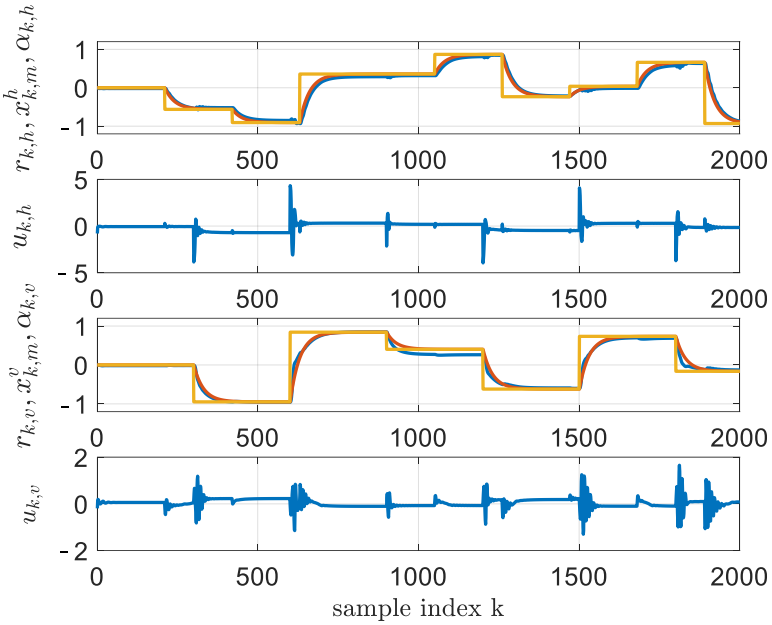

Figure 6. Two-hidden-layer controller learned through BOADHDP, at iteration $j = 500$: $r_{k,h}$ and $r_{k,v}$ (yellow); $x^h_{k,m}$ and $x^v_{k,m}$ (red); $\alpha_{k,h}$ and $\alpha_{k,v}$ (blue). The commands $u_{k,h}$ and $u_{k,v}$ are for the horizontal and vertical axes (blue).

5.3. Comparison Between BOADHDP and the Online Adaptive ADHDP

Next, the online adaptive ADHDP algorithms from [21,22] were applied to the TRAS system. The difference between ADHDP methods [] was that the former one only updates the weights from the hidden to the output layer, while the latter updates the entire NN weights.

For these algorithms, we used the same NN architectures as in the single-layer NN from BOADHDP, namely the form of the action value function NN was 12-50-1 and that of the controller was 10-10-2. The activation functions of the hidden layer were hyperbolic tangents and the ones of the output layer were linear. The weights were also initialized using the Xavier initialization [29]. The learning rates were selected to be $\alpha_Q = 0.01$ and $\alpha_C = 0.001$. The penalty function took the form of $r\left(\boldsymbol{x}^e_k, \boldsymbol{u}_k\right) = \left(\alpha_{k,h} - x^h_{k,m}\right)^2 + \left(\alpha_{k,v} - x^v_{k,m}\right)^2$.

Compared with BOADHDP, in these implementations, the adaptation of the NNs was made online, using only the transitions along with each time step of the simulated system. The algorithm ran for 200,000 time steps. Every 2000 steps, the controller weights were fixed and their performance was measured by the function $J\left(\boldsymbol{x}^e_k\right) = \left(\alpha_{k,h} - x^h_{k,m}\right)^2 + \left(\alpha_{k,v} - x^v_{k,m}\right)^2/2000$ under a simulated scenario, for $k = 1 : 2000$. The convergence of the action value function and of the controller performance of the simulated scenario can be seen in Figure 7 for the ADHDP algorithms from [21,22]. The tracking performance of the ADHDP algorithms from [21,22] on the TRAS system using the aforementioned learning settings is presented in Figure 8. The value of the $J\left(\boldsymbol{x}^e_k\right)$ was 0.0236 for the ADHDP algorithm from [21] and 0.0258 for the ADHDP algorithm from [22].

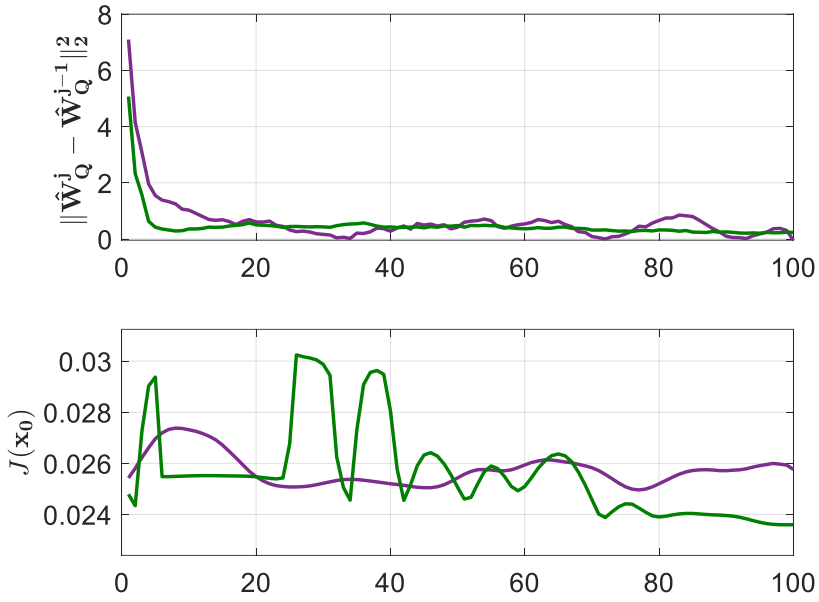

Figure 7. ADHDP convergence in relation to the TRAS system. ADHDP algorithm from [21] in purple and ADHDP algorithm from [22] in green.

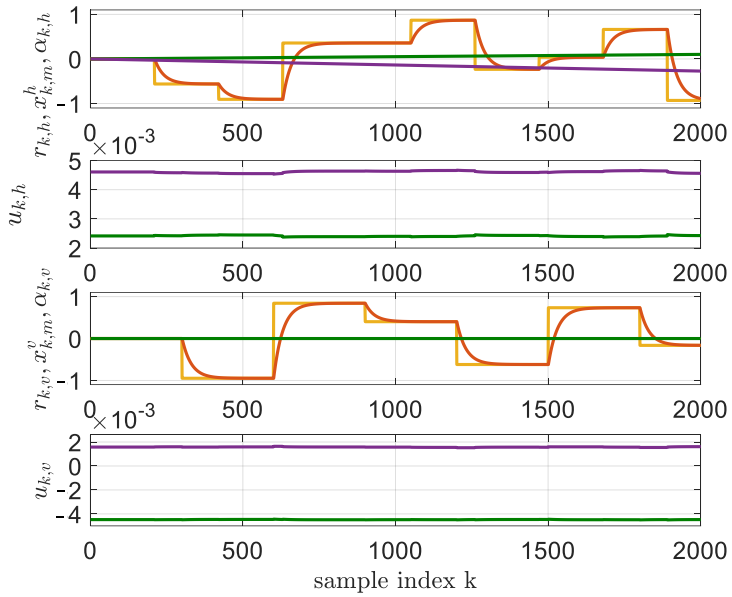

Figure 8. Tracking performance of the ADHDP algorithms from [21,22], at iteration $j = 150,000$: $r_{k,h}$ and $r_{k,v}$ (yellow); $x_{k,m}^{h}$ and $x_{k,m}^{v}$ (red); $\alpha_{k,h}$ and $\alpha_{k,v}$ (green—ADHDP algorithm from [21], purple—ADHDP algorithm from [22]). The commands $u_{k,h}$ and $u_{k,v}$ are for the horizontal and vertical axes (green—ADHDP algorithm from [21], purple—ADHDP algorithm from [22]).

The $J(\boldsymbol{x}_k^e)$ values of the BOADHDP and ADHDP algorithms from [21,22] are summarized in Table 1. Also, from Figures 5 and 8, it can be observed that the online adaptive ADHDP algorithms could not deliver the same performance as their batch and offline counterpart, BOADHDP. Furthermore, the ADHDP algorithms presented in [21,22] failed to enhance controller performance, even though they utilized four times as many collected transitions from the system. This difference in the performance of the BOADHDP algorithm stems, in part, from the batch nature of the learning process. By processing multiple collected transitions from the state action space at the same time during NN actualization, the gradient for the action value and controller NNs is averaged over all the transitions. In turn, this makes the NN update more stable. By issuing the gradient update in an offline manner, the same collected transitions are used at each iteration, making the convergence speed

faster. This stands in accordance with the observations from [28], where the authors proved the advantages of batch learning in comparison to the online adaptive single-transition learning from the classical ADHDP methods. Also, from this case study, it can be seen that the number of transitions required for learning was higher in the online adaptive case than in the batch offline case.

Table 1. Comparison between the BOADHDP (single- and multiple-hidden-layer NN approximations) and the ADHDP algorithms from [21,22].

Algorithm	$J(x_k^e)$
BOADHDP with NN approximation having a single hidden layer	0.0049
BOADHDP with NN approximation having two hidden layers	0.0031
ADHDP from [21]	0.0236
ADHDP from [22]	0.0258

6. Discussion and Conclusions

In this paper, we study the theoretical stability of BOADHDP with deep neural networks as function approximators for the action value function and the controller. To this end, we introduce a stability criterion for the iteratively updated action value function and controller NN. The theory uses the Lyapunov stability approach and shows that the weight estimation errors are UUB if some inequality constraints on the learning rate magnitudes are respected. This research extends the previous results from the literature, such as [21,22], both theoretically and practically.

- First, our Lyapunov stability is extended to address NN approximators for action value functions and controllers with multiple hidden layers. Although NNs with a single hidden layer are universal approximators, their usage for highly nonlinear applications is hindered by their generalization capabilities. In contrast, multilayer NNs can learn complex features effectively, reducing overfitting and generalization issues. The results outlined in Theorem 1 indicate also an indirect proportionality between the number of NN hidden layers and the magnitude of the learning rate, providing a practical heuristic approach for practical ADP applications of multilayer NNs.
- Second, our theoretical Lyapunov stability analysis addresses the usage of batch offline learning of the action value function and controller NNs. Although successful ADP applications have been reported using adaptive update methods, their practical use is often constrained by the significant number of iterations required for convergence. The adoption of batch learning has, thus, become standard practice, necessitating a theoretical Lyapunov stability coverage.
- Finally, from a practical point of view, we validate the advantage of using BOADHDP with multilayer NNs through a case study on a twin rotor aerodynamical system (TRAS). This study compares BOADHDP using neural networks with a single layer and two hidden layers as function approximators. The results show that the normed action value function weight convergence is smoother with two-hidden-layer networks, also leading to a controller with an enhanced performance on the control benchmark (0.0049 for the single-layer NNs and 0.0031 for the two-layer implementation, namely a 1.58% improvement). This demonstrates the superior capability of multilayer networks in managing complex, nonlinear control systems. Also, BOADHDP is compared with ADHDP algorithms from [21,22], with ADHDP algorithms from [21,22] obtaining 0.0236 and 0.0258, respectively, on the control benchmark, while also requiring four times more collected transitions from the TRAS system. This proves both the efficiency of the BOADHDP with respect to the number of collected transitions and

the performance of using batch offline learning methodologies, confirming the results from [28].

Our findings highlight the advantages of BOADHDP with deep neural networks in practical applications, underscoring the improved stability and performance in control tasks. Future research may explore extending this batched multilayer approach to adaptive learning scenarios. From a practical point of view, applications entailing deep neural networks and batch learning applications might benefit from this analysis.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the author on request.

Acknowledgments: I would like to thank Ioan Silea for reading this manuscript and for providing constructive feedback that improved the quality of our research.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Werbos, P.J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Science. Ph.D. Thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA, USA, 1974.
2. Bellman, R.E. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.
3. Werbos, P.J. Approximate dynamic programming for real time control and neural modeling. In *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*; White, D.A., Sofge, D.A., Eds.; Van Nostrand Reinhold: New York, NY, USA, 1992; pp. 493–525.
4. Al-Tamimi, A.; Lewis, F.L. Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof. *IEEE Trans. Syst. Man Cybern. B* **2008**, *38*, 943–949. [CrossRef] [PubMed]
5. Prokhorov, D.V.; Wunsch, D.C. Adaptive critic designs. *IEEE Trans. Neural Netw.* **1997**, *8*, 997–1007. [CrossRef] [PubMed]
6. Liu, X.; Balakrishnan, S.N. Convergence analysis of adaptive critic based optimal control. In Proceedings of the American Control Conference, Chicago, IL, USA, 28–30 June 2000.
7. White, D.A.; Sofge, D.A. *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*; Van Nostrand Reinhold: New York, NY, USA, 1992.
8. Padhi, R.; Unnikrishnan, N.; Wang, X.; Balakrishnan, S.N. A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems. *Neural Netw.* **2006**, *19*, 1648–1660. [CrossRef] [PubMed]
9. Balakrishnan, S.N.; Biega, V. Adaptive-critic-based neural networks for aircraft optimal control. *J. Guid. Control Dyn.* **1996**, *19*, 893–898. [CrossRef]
10. Dierks, T.; Jagannathan, S. Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1118–1129. [CrossRef] [PubMed]
11. Venayagamoorthy, G.K.; Harley, R.G.; Wunsch, D.C. Comparison of heuristic dynamic programming and dual heuristic programming adaptive critics for neurocontrol of a turbogenerator. *IEEE Trans. Neural Netw.* **2002**, *13*, 764–773. [CrossRef] [PubMed]
12. Ferrari, S.; Stengel, R.F. Online adaptive critic flight control. *J. Guid. Control Dyn.* **2004**, *27*, 777–786. [CrossRef]
13. Ding, J.; Jagannathan, S. An online nonlinear optimal controller synthesis for aircraft with model uncertainties. In Proceedings of the AIAA Guidance, Navigation and Control Conference, Toronto, ON, Canada, 2–5 August 2010.
14. Vrabie, D.; Pastravanu, O.; Lewis, F.; Abu-Khalaf, M. Adaptive optimal control for continuous-time linear systems based on policy iteration. *Automatica* **2009**, *45*, 477–484. [CrossRef]
15. Dierks, T.; Jagannathan, S. Optimal control of affine nonlinear continuous-time systems. In Proceedings of the American Control Conference, Baltimore, MA, USA, 30 June–2 July 2010.
16. Vamvoudakis, K.; Lewis, F. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica* **2010**, *46*, 878–888. [CrossRef]
17. Enns, R.; Si, J. Helicopter trimming and tracking control using direct neural dynamic programming. *IEEE Trans. Neural Netw.* **2003**, *14*, 929–939. [CrossRef]
18. Liu, D.; Javaherian, H.; Kovalenko, O.; Huang, T. Adaptive critic learning techniques for engine torque and air–fuel ratio control. *IEEE Trans. Syst. Man Cybern. B* **2008**, *38*, 988–993.

19. Ruelens, F.; Claessens, B.J.; Quaiyum, S.; De Schutter, B.; Babuška, R.; Belmans, R. Reinforcement learning applied to an electric water heater: From theory to practice. *IEEE Trans. Smart Grid* **2018**, *9*, 3792–3800. [CrossRef]
20. He, P.; Jagannathan, S. Reinforcement learning-based output feedback control of nonlinear systems with input constraints. *IEEE Trans. Syst. Man Cybern. B* **2005**, *35*, 150–154. [CrossRef] [PubMed]
21. Liu, F.; Sun, J.; Si, J.; Guo, W.; Mei, S. A boundness result for the direct heuristic dynamic programming. *Neural Netw.* **2012**, *32*, 229–235. [CrossRef] [PubMed]
22. Sokolov, Y.; Kozma, R.; Werbos, L.D.; Werbos, P.J. Complete stability analysis of a heuristic approximate dynamic programming control design. *Automatica* **2015**, *59*, 9–18. [CrossRef]
23. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
24. Lillicrap, T.; Hunt, J.; Pritzel, A.; Heess, N.; Erez, Y.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
25. Fujimoto, S.; Hoof, H.; Meger, D. Addressing function approximation error in actor-critic methods. In Proceedings of the Internation Conference on Machine Learning, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018.
26. Riedmiller, M. Neural Fitted Q Iteration–First Experiences with a Data Efficient Neural Reinforcement Learning Method. In Proceedings of the European Conference on Machine Learning, Porto, Portugal, 3–7 October 2005.
27. Radac, M.-B.; Lala, T. Learning output reference model tracking for higher-order nonlinear systems with unknown dynamics. *Algorithms* **2019**, *12*, 121. [CrossRef]
28. Watkins, C. Learning from Delayed Rewards. Ph.D. Thesis, Department of Computational Science, University of Cambridge, Cambridge, UK, 1989.
29. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010.