robotics

MDPI

Article

Model-Free Gradient-Based Adaptive Learning Controller for an Unmanned Flexible Wing Aircraft

Mohammed Abouheaf [1], Wail Gueaieb [1,*] and Frank Lewis [2]

[1] School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada; mohammed.abouheaf@uottawa.ca

[2] Department of Electrical Engineering, University of Texas at Arlington, Arlington, TX 76019, USA; lewis@uta.edu

* Correspondence: wail.gueaieb@uottawa.ca; Tel.: +1-613-562-5800

Received: 1 September 2018; Accepted: 20 October 2018; Published: 23 October 2018

Abstract: Classical gradient-based approximate dynamic programming approaches provide reliable and fast solution platforms for various optimal control problems. However, their dependence on accurate modeling approaches poses a major concern, where the efficiency of the proposed solutions are severely degraded in the case of uncertain dynamical environments. Herein, a novel online adaptive learning framework is introduced to solve action-dependent dual heuristic dynamic programming problems. The approach does not depend on the dynamical models of the considered systems. Instead, it employs optimization principles to produce model-free control strategies. A policy iteration process is employed to solve the underlying Hamilton–Jacobi–Bellman equation using means of adaptive critics, where a layer of separate actor-critic neural networks is employed along with gradient descent adaptation rules. A Riccati development is introduced and shown to be equivalent to solving the underlying Hamilton–Jacobi–Bellman equation. The proposed approach is applied on the challenging weight shift control problem of a flexible wing aircraft. The continuous nonlinear deformation in the aircraft's flexible wing leads to various aerodynamic variations at different trim speeds, which makes its auto-pilot control a complicated task. Series of numerical simulations were carried out to demonstrate the effectiveness of the suggested strategy.

Keywords: model-free control; flexible wing aircraft; reinforcement learning; optimal control

1. Introduction

Various Approximate Dynamic Programming (ADP) methods have been employed to solve the optimal control problems for single and multi-agent systems [1–6]. They are divided into different classes according to the way the temporal difference equations and the associated optimal strategies are evaluated. The ADP approaches that consider gradient-based forms provide fast converging approaches, but they require the complete knowledge of the dynamical model of the system under consideration [7]. The solution of the flexible wing control problem requires model-free approaches, since the aerodynamics of the flexible wing aircraft are highly nonlinear and they variate continuously [8–16]. This type of aircraft has large uncertainties embedded in their aerodynamic models. Herein, an online adaptive learning approach, based on a gradient structure, is employed to solve the challenging control problem of flexible wing aircrafts. This approach does not need any of the aerodynamic information of the aircraft. It is based on a model-free control strategy approximation.

Several ADP approaches have been adopted to solve the difficulties associated with the dynamic programming solutions which involve the curse of dimensionality in the state and action spaces [2–5,17,18]. They are employed in different applications such as machine learning, autonomous systems, multi-agent systems, consensus and synchronization, and decision making problems [19–21].

Typical optimal control methods tend to solve the underlying Hamilton–Jacobi–Bellman (HJB) equation of the dynamical system by applying the optimality principles [22,23]. An optimal control problem is usually formulated as an optimization problem with a cost function that identifies the optimization objectives and a mathematical process to find the respective optimal strategies [6,7,18,22–28]. To implement the optimal control solutions stemming from the ADP approaches, numerous solving frameworks are considered based on combinations of Reinforcement Learning (RL) and adaptive critics [1,5,18,25,27]. Reinforcement Learning approaches use various forms of temporal difference equations to solve the optimization problems associated with the dynamical systems [1,18]. This implies finding ways to penalize or reward the attempted control strategies to optimize a certain objective function. This is accomplished in a dynamic learning environment where the agent applies its acquired knowledge to update its experience about the merit of using the attempted policies. RL methods implement the temporal difference solutions using two main coupled steps. The first approximates the value of a given strategy, while the second approximates the optimal strategy itself. The sequence of these coupled steps can be implemented with either value or policy iteration method [18]. RL has also been proposed to solve problems with multi-agent structures and objectives [29] as well as cooperative control problems using dynamic graphical games [21,26,30]. Action Dependent Dual Heuristic Dynamic Programming (ADDHP) depends on the system's dynamic model [7,26,28]. Herein, the relation between the Hamiltonian and Bellman equation is used to solve for the governing costate expressions and hence a policy iteration process is proposed to find an optimal solution. Dual Heuristic Dynamic Programming (DHP) approaches for graphical games are developed in [21,26,30]. However, these approaches require in-advance knowledge of the system's dynamics and, in some cases of the multi-agent systems, they rely on complicated costate structures to include the neighbors influences.

Adaptive critics are typically implemented within reinforcement learning solutions using neural network approximations [18,27]. The actor approximates the optimal strategy, while the value of the assessed strategy is approximated by the critic [18]. Real-time optimal control solutions using adaptive critics are introduced in [3]. Adaptive critics provide prominent solution frameworks for the adaptive dynamic programming problems [31]. They are employed to produce expert paradigms that can undergo learning processes while solving the underlying optimization challenges. Moreover, they have been invoked to solve a wide spectrum of optimal control problems in continuous and discrete-time domains, where actor-critic schemes are evoked within an Integral Reinforcement Learning context [32,33]. An action-dependent solving value function is proposed to play some zero-sum games in [34], where one critic and two actors are adapted forward in time to solve the game. An online distributed actor-critic scheme is suggested to implement a Dual Heuristic Dynamic Programming solution for the dynamic graphical games in [7,24] without overlooking the neighbors' effects, which is a major concern in the classical DHP approaches. The solution provided by each agent is implemented by single actor-critic approximators. Another actor-critic development is applied to implement a partially-model-free adaptive control solution for a deterministic nonlinear system in [35]. A reduced solving value function approach employed an actor-critic scheme to solve the graphical games, where only partial knowledge about the system dynamics is necessary [26]. An actor-critic solution framework is adopted for an online policy iteration process with a weighted-derivative performance index form in [33]. A model-free optimal solution for graphical games is implemented using only one critic structure for each agent in [25]. The recent state-of-the-art adaptive critics implementations for numerous reinforcement learning solutions for the feedback control problems are surveyed in [36]. These involve the regulation and tracking problems for single- as well as multi-agent systems [36].

Flexible wing aircraft are usually modeled as two-mass systems (fuselage and wing). Both masses are coupled via different kinematic and dynamic constraints [8,13–15,37]. They involve the kinematic constraint at the connection point of the hang strap [38,39]. The keel tube works as a symmetric axis for this type of aircraft. The basic theoretical and experimental developments for the aerodynamic

modeling aspects of the flexible wing systems are introduced in [8,13–15,40,41]. Several wind tunnel experiments have been introduced for the hang glider in [14]. An approximate modeling approach of the flexible wing's aerodynamics led to equations of motion for the lateral and longitudinal directions with small perturbation models in [42]. The modeling process for the hang glider assumed a rigid wing modeling process, where the derivatives, due to the aerodynamics, were added at the last stage [11,12]. A comprehensive decoupled aerodynamic model for the hang glider is presented in [43]. A nine-degree-of-freedom aerodynamic model that employs a set of nonlinear state equations is developed in [38,39]. The control of the flexible wing aircraft follows a weight shift mechanism, where the lateral and longitudinal maneuvers or the roll/pitch control mechanism is achieved by changing the relative centers of gravity of the wing and the fuselage systems [9,10,13,14,37,44]. The geometry of the flexible wing's control arm influences the maximum allowed control moments [9]. The reduced center of gravity magnifies the static pitch stability [9]. Frequency response-based approaches are adopted to study the stability of flexible wing systems in [11,12]. The longitudinal stability of a fixed wing system can be used to understand that of the flexible wing vehicle provided some conditions are satisfied [37]. The lateral stability margins are shown to be larger compared to conventional fixed wing aircraft.

The contribution of this work is four-fold:

1. An online adaptive learning control approach is proposed to solve the challenging weight-shift control problem of flexible wing aircraft. The approach uses model-free control structures and gradient-based solving value functions. This serves as a model-free solution framework for the classical Action Dependent Dual Heuristic Dynamic Programming problems.
2. The work handles many concerns associated with implementing value and policy iteration solutions for ADDHP problems, which either necessitate partial knowledge about the system dynamics or involve difficulties in the evaluations of the associated solving value functions.
3. The relation between a modified form of Bellman equation and the Hamiltonian expression is developed to transfer the gradient-based solution framework from the Bellman optimality domain to an alternative domain that uses Hamilton–Jacobi–Bellman expressions. This duality allows for a straightforward solution setup for the considered ADDHP problem. This is supported by a Riccati development that is equivalent to solving the underlying Bellman optimality equation.
4. The proposed solution that is based on the combined-costate structure is implemented using a novel policy iteration approach. This is followed by an actor-critic implementation that is free of the computational expensive matrix inverse calculations.

The paper is organized as follows: Section 2 briefly explains the weight shift control mechanism of a flexible wing aircraft. Section 3 highlights the model-based solutions within the framework of optimal control theory along with the existing challenges. Section 4 discusses the duality between the Hamiltonian function and Bellman equation leading to the Hamilton–Jacobi–Bellman formulation, which is used to generalize the Action Dependent Dual Heuristic Dynamic Programming solution with a policy iteration process. Section 5 introduces the model-free gradient-based solution and the underlying Riccati development. Section 6 demonstrates the adaptive critics implementations for the proposed model-free gradient-based solution. Section 7 tests the validity of the introduced online adaptive learning control approach by applying it on two case studies. Finally, the paper is concluded with some concluding remarks in Section 8.

2. Control Mechanism of a Flexible Wing Aircraft

This section briefly introduces the idea of weight shift control along with a basic aerodynamic model of a flexible wing system. Herein, a flexible wing aircraft is modeled as a two-mass system (fuselage/pilot and wing) coupled through nonlinear kinematic constraints at the hang strap, as shown in Figure 1. The flexible wing is connected to the fuselage through a control bar. The aerodynamic forces are controlled via a weight shift mechanism, where the fuselage's center of gravity "floats" with respect to that of the wing [8–12,37,44]. Such a system is governed by complex aerodynamic

forces which makes it difficult to model to a satisfactory accuracy. Consequently, model-based control approaches may not be appropriate for the auto-pilot control of such systems.

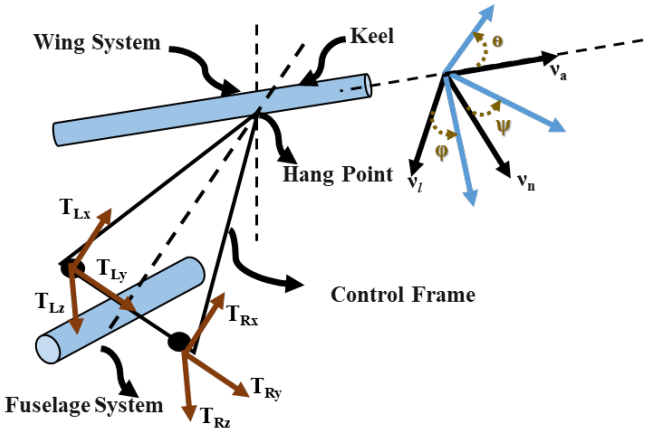

Figure 1. A flexible wing hang glider.

In this framework, the longitudinal and lateral motions are controlled through the force components applied on the control bar of the hang glider [38,39]. This development takes into account a nine-degree-of-freedom model that considers the kinematic interactions and the constraints between the fuselage and the wing at the hang point, as shown in Figure 1. The longitudinal and lateral dynamics are referred to the wing's frame, where the forces (nonlinear state equations) at the hang point are substituted for by some transformations in the wing's frame [39].

The decoupled longitudinal and lateral aerodynamic models satisfy the following assumptions [39]:

- The hang strap works as a kinematic constraint between the decoupled wing/fuselage systems.
- The fuselage system is assumed to be a rigid body connected to the wing system via a control triangle and a hang strap.
- The force components applied on the control bar are the input control signals.
- External forces, such as the aerodynamics and gravity, the associated moments, and the internal forces, are evaluated for both fuselage and wing systems.
- The fuselage's pitch–roll–yaw attitudes and pitch–roll–yaw attitude rates are referred to the wing's frame of motion through kinematic transformations.
- The complete aerodynamic model of the aircraft is reduced by substituting for the internal forces at the hang strap using the action/reaction laws.
- The pilot's frames of motion (i.e., longitudinal and lateral states) are referred to the respective wing's frames of motion.

The dynamics of the flexible wing aircraft are decoupled into longitudinal and lateral systems, such that [9,37,39,44].

$$\delta^{Lo/La}_{(k+1)} = A^{Lo/La}\,\delta^{Lo/La}_{k} + B^{Lo/La}\,u^{Lo/La}_{k}, \tag{1}$$

where $\delta^{Lo} = \begin{bmatrix} v_{aw} & v_{nw} & \dot{\theta}_w & \dot{\theta}_{fw} & \theta_{fw} & \theta_w \end{bmatrix}^T$ is the longitudinal state vector, $\delta^{La} = \begin{bmatrix} v_{lw} & \dot{\phi}_w & \dot{\psi}_w & \dot{\phi}_{fw} & \dot{\psi}_{fw} & \phi_{fw} & \psi_{fw} & \phi_w \end{bmatrix}^T$ is the lateral state vector, the force $T_{cq} = \frac{1}{2}\left(T_{Rq} + T_{Lq}\right)$ is the collective force in direction q, the force $T_{dq} = \frac{1}{2}\left(T_{Rq} - T_{Lq}\right)$ is the differential force in direction q, $u^{Lo} = \begin{bmatrix} T_{cx} & T_{cz} \end{bmatrix}^T$ represents the longitudinal control signals, and $u^{La} = \begin{bmatrix} T_{cy} & T_{dx} & T_{dz} \end{bmatrix}^T$ denotes the lateral control signals.

The modeling results of the flexible wing aircraft are based on the experimental and theoretical studies of [9], where the control mechanism employs force components on the control bar [39].

3. Optimal Control Problem

This section explains the main challenges associated with the optimal solution of the control problem using Action Dependent Dual Heuristic Dynamic Programming approaches. It should justify the need for a model-free gradient-based optimal control solution.

3.1. Bellman Equation Formulation

Consider a flexible wing hang glider characterized by the following discrete-time state space equation:

$$\delta_{k+1} = A\,\delta_k + B\,u_k, \tag{2}$$

where $\delta_k \in R^n$ is a vector of the (longitudinal/lateral) states and $u_k \in R^m$ is a vector of the (longitudinal/lateral) force components applied on the control bar, A and B are the (longitudinal/lateral) state space matrices, and k is the time index.

A quadratic convex performance index is introduced to assess the quality of the taken control actions, such that

$$J = \sum_{k=0}^{\infty} F(\delta_k, u_k), \tag{3}$$

where F is a convex utility function given by

$$F(\delta_k, u_k) = \frac{1}{2}\left(\delta_k^T\,Q\,\delta_k + u_k^T\,R\,u_k\right), \tag{4}$$

where $Q \geq 0 \in R^{n\times n}$ and $R > 0 \in R^{m\times m}$ are symmetric time-invariant positive semi-definite and positive definite weighting matrices, respectively.

The structure in Equation (3) is used to suggest a solution form. First, the solving value function $V(\delta_k, u_k)$ is assumed to depend on the the state δ_k and the control strategy u_k so that

$$V(\delta_k, u_k) = \sum_{i=k}^{\infty} F(\delta_i, u_i). \tag{5}$$

This yields a temporal difference (Bellman) equation defined by

$$V(\delta_k, u_k) = \frac{1}{2}\left(\delta_k^T\,Q\,\delta_k + u_k^T\,R\,u_k\right) + V(\delta_{k+1}, u_{k+1}). \tag{6}$$

The value function in Equation (5) is assumed to have the following form

$$V(\delta_k, u_k) = \frac{1}{2}\,[\delta_k^{\mathrm{T}}\;\; u_k^{\mathrm{T}}]\;\; S \;\; \begin{bmatrix} \delta_k \\ u_k \end{bmatrix}, \tag{7}$$

where $S = \begin{bmatrix} S_{\delta\delta} & S_{\delta u} \\ S_{u\delta} & S_{uu} \end{bmatrix}$.

3.2. Model-Based Policy Formulation

Herein, a model-based optimal control strategy and the associated costate equation are derived by applying the Bellman's optimality principles to Bellman equation (Equation (6)). Below, a model-free policy solution is introduced. To evaluate the optimal control strategy, the optimality principles are applied to $V(\dots)$.

$$\begin{aligned} \text{argmin}_{u_k} V(\delta_k, u_k) &= \frac{\partial V(\delta_k, u_k)}{\partial u_k} = 0 \\ \Rightarrow u^o &= -R^{-1}B^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ \frac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_{k+1}} V(\delta_{k+1}, u_{k+1}), \end{aligned} \tag{8}$$

where $\nabla_{\delta u_{k+1}} V(\delta_{k+1}, u_{k+1}) = S \cdot [\delta_{k+1}^T \; u_{k+1}^T]^T$. Applying this model-based optimal policy in Equation (6) yields the following Bellman's optimality equation:

$$V^o(\delta_k, u_k^o) = \frac{1}{2}\left(\delta_k^{\mathrm{T}} Q \delta_k + u_k^{o\mathrm{T}} R\, u_k^o\right) + V^o(\delta_{(k+1)}, u_{(k+1)}^o). \tag{9}$$

The gradient-based solution requires the knowledge of the costate equation associated with the system in Equation (2). The costate equation is evaluated as follows

$$\nabla_{\delta_k} V(\delta_k, u_k) = Q\delta_k + A^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ \frac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_{k+1}} V(\delta_{k+1}, u_{k+1}), \tag{10}$$

The main concern about this gradient-based development is that both the optimal strategy in Equation (8) and the associated costate Equation (10) depend on the dynamical model of the system (i.e., A and B). The following development shows how it is possible to avoid this shortcoming using dynamical information in deciding on the optimal control strategies.

3.3. Model-Free Policy Formulation

In the sequel, a model-free policy structure is introduced along with the optimal control solution algorithms. Applying the Bellman's optimality principles [22] yields the optimal control strategy u_k^o so that

$$\text{argmin}_{u_k} V(\delta_k, u_k) = \text{argmin}_{u_k} \frac{1}{2} [\delta_k^{\mathrm{T}} \; u_k^{\mathrm{T}}] \; S \begin{bmatrix} \delta_k \\ u_k \end{bmatrix}.$$

Note that the optimality principle is applied to the left-hand-side of Equation (6). This yields the following model-free control policy

$$u_k^o = K \cdot \delta_k, \tag{11}$$

where the control gain K is given by $K = -S_{u_k u_k}^{-1} \cdot S_{u_k \delta_k}$. Substituting Equation (11) into Equation (6) yields a dual (equivalent) Bellman's optimality equation (Equation (9)). The Bellman optimality equation (Equation (9)) will be used to propose different Action Dependent Dual Heuristic Dynamic solutions for the optimal control problem in hand, as shown below.

To propose gradient-based solutions, the gradient of the Bellman equation (Equation (6)) with respect to the state δ_k is calculated.

$$\nabla_{\delta_k} V(\delta_k, u_k) = Q\,\delta_k + A^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ K \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_{k+1}} V(\delta_{k+1}, u_{k+1}), \tag{12}$$

where $\nabla_{\delta_k} V(\delta_k, u_k) = \partial V(\delta_k, u_k)/\partial \delta_k$ and $\nabla_{\delta u_k} V(\delta_k, u_k) = S \cdot [\delta_k^T \; u_k^T]^T, \forall k$.

The optimal strategy (Equation (11)) and the costate (Equation (12)) are used to propose different gradient-based solution forms. These are generalizations of the ADDHP solution, where a slight modification on the approximation of the control policy is introduced. In the sequel, solutions based on value iteration and policy iteration processes are presented.

Remark 1. *Although Algorithm 1 and 2 use model-free policy structures (Equations (14) and (16)), the gradient expressions (Equations (13) and (15)) depend on the system's drift dynamics (matrix A), which is a real challenge*

for systems with uncertain or unknown dynamics. Moreover, it is difficult to evaluate the matrix S, and so V, in Equation (15) using the policy iteration process. As such, a new approach is required to benefit the gradient-based solution form without the need for a system's dynamic model. To do that, a dual development using the Hamiltonian framework is needed.

Algorithm 1 Value Iteration Gradient-based Solution

1. Initialize $\nabla_{\delta_k} V^0(\delta_k, u_k)$ and u_k^o.
2. Evaluate $\nabla_{\delta_k} V^{\ell+1}(..)$ using

$$\nabla_{\delta_k} V^{\ell+1}(\delta_k, u_k) = Q\delta_k^\ell + A^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ K^\ell \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_k} V^\ell(\delta_{k+1}, u_{k+1}), \tag{13}$$

   where ℓ is the iteration index.
3. Update the approximation of the optimal strategy using

$$u_k^{\ell+1} = -\left[S_{u_k u_k}^{-1} \cdot S_{u_k \delta_k}\right]^{\ell+1} \cdot \delta_k. \tag{14}$$

4. Halt on convergence of $\|S^{\ell+1}(..) - S^\ell(..)\|$.

Algorithm 2 Policy Iteration Gradient-based Solution

1. Initialize $\nabla_{\delta_k} V^0(\delta_k, u_k)$ and use admissible u_k^o.
2. Evaluate $\nabla_{\delta_k} V^\ell(..)$ using

$$\nabla_{\delta_k} V^\ell(\delta_k, u_k) = Q\delta_k^\ell + A^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ K^\ell \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_k} V^\ell(\delta_{k+1}, u_{k+1}). \tag{15}$$

3. Update the approximation of the optimal strategy using

$$u_k^{\ell+1} = -\left[S_{u_k u_k}^{-1} \cdot S_{u_k \delta_k}\right]^{\ell} \cdot \delta_k. \tag{16}$$

4. Halt on convergence of $\|S^{\ell+1}(..) - S^\ell(..)\|$.

4. Hamiltonian-Jacobi–Bellman Formulation

The following Hamilton–Jacobi and Hamilton–Jacobi–Bellman developments are necessary to propose the model-free ADDHP control solutions. They find the relation between the costate variable of the Hamiltonian function and the solving value function through Bellman equation via a Hamilton–Jacobi framework. Then, the Hamilton–Jacob-Bellman development is used to propose the model-free ADDHP solution.

4.1. The Hamiltonian Mechanics

Optimal control problems, in general, are solved using the Hamiltonian mechanics, where the necessary conditions of optimality are found by means of Lagrange dynamics [22]. The objective of the optimization problem is to chose a policy μ_k to minimize a cost function F such that $\operatorname{argmin}_{\mu_k} F(\delta_k, \mu_k)$, subject to the following constraints:

$$\begin{aligned} \mu_k &= \chi(\delta_k) = C\,\delta_k, \\ \delta_{(k+1)} &\equiv \varrho\,(\delta_k, \mu_k), \end{aligned} \tag{17}$$

where $\chi \in R^{m\times 1}$ and $\varrho \in R^{n\times 1}$ are some mapping functions, and $C \in R^{m\times n}$ is a row gain matrix.

The Hamiltonian expression for the problem is given by

$$H(\delta_k, \lambda_{(k+1)}, \mu_k) = \lambda_{(k+1)}^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ \mu_{k+1} \end{bmatrix} + F(\delta_k, \mu_k), \tag{18}$$

where $\lambda_k \in R^{(n+m)\times 1}$ is the Lagrange multiplier or the costate variable. Merging Equation (17) into Equation (18) leads to

$$H(\delta_k, \lambda_{(k+1)}, \mu_k) = \lambda_{(k+1)}^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix} \delta_{k+1} + F(\delta_k, \mu_k). \tag{19}$$

Remark 2. *Similar to the optimal policy in Equation (8) derived using Bellman equation (Equation (6)), an optimal model-based control strategy based on the Hamiltonian can be obtained so that*

$$\begin{aligned} &argmin_{\mu_k} H(\delta_k, \nabla_{\delta\mu_{k+1}} V(\delta_{k+1}, \mu_{k+1}), \mu_k) = \frac{\partial H(\dots)}{\partial \mu_k} = 0 \\ &\Rightarrow \mu^* = -R^{-1} B^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ \dfrac{\partial \mu_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} \nabla_{\delta\mu_{k+1}} V(\delta_{k+1}, \mu_{k+1}). \end{aligned} \tag{20}$$

The following Hamilton–Jacobi theorem finds the relation between the costate variable λ_k and the value function $V(\delta_k, u_k), \forall k$.

Theorem 1. *Let the Hamiltonian function be given by Equation (18) and the value function $V(\delta_k, \mu_k)$ be defined by Equation (6). Then, this value function satisfies the following Hamilton–Jacobi equation:*

$$V(\delta_{k+1}, \mu_{k+1}) - V(\delta_k, \mu_k) + H(\delta_k, \nabla_{\delta\mu_{k+1}} V(\delta_{k+1}, \mu_{k+1}), \mu_k) - \nabla_{\delta\mu_{k+1}} V(\delta_{k+1}, \mu_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ \mu_{k+1} \end{bmatrix} = 0, \tag{21}$$

where $\lambda_{k+1} = \nabla_{\delta\mu_{k+1}} V(\delta_{k+1}, \mu_{k+1})$.

Proof. The augmented value function $V(\delta_k, \mu_k)$ is

$$V(\delta_k, \mu_k) = \sum_{l=k}^{\infty} \left\{ F(\delta_k, \mu_k) + \lambda_{(l+1)}^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix} \left[\varrho(\delta_\ell, \mu_\ell) - \delta_{(l+1)} \right] \right\}. \tag{22}$$

The Hamiltonian in Equation (18) is rearranged such that

$$H(\delta_\ell, \lambda_{(\ell+1)}, \mu_\ell) = \lambda_{(\ell+1)}^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix} \delta_{\ell+1} + F(\delta_\ell, \mu_\ell).$$

Using this expression into the augmented function in Equation (22) yields

$$V(\delta_{k+1}, \mu_{k+1}) - V(\delta_k, \mu_k) + H(\delta_k, \lambda_{(k+1)}, \mu_k) - \lambda_{(k+1)}^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix} \delta_{(k+1)} = 0. \tag{23}$$

Finding the gradient of Equation (23) with respect to $\delta_{(k+1)}$ yields

$$\nabla_{\delta_{k+1}} V(\delta_{k+1}, \mu_{k+1}) + \left(\frac{\partial \lambda_{k+1}}{\partial \delta_{k+1}}\right)^{\mathrm{T}} \left(\frac{\partial H(\delta_k, \lambda_{k+1}, \mu_k)}{\partial \lambda_{k+1}}\right) - \left(\left(\frac{\partial \lambda_{k+1}}{\partial \delta_{k+1}}\right)^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix} \delta_{k+1} + \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix}^{\mathrm{T}} \lambda_{k+1}\right) = 0.$$

This equation can be rearranged such that

$$\nabla_{\delta_{k+1}} V(\delta_{k+1}, \mu_{k+1}) - \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix}^{\mathrm{T}} \lambda_{k+1} + \left(\frac{\partial \lambda_{k+1}}{\partial \delta_{k+1}}\right)^{\mathrm{T}} \left(\frac{\partial H(\delta_k, \lambda_{k+1}, \mu_k)}{\partial \lambda_{k+1}} - \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix} \delta_{k+1}\right) = 0.$$

$$\frac{\partial H(\delta_k, \lambda_{k+1}, \mu_k)}{\partial \lambda_{k+1}} = \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix} \delta_{k+1} \Rightarrow \nabla_{\delta_{k+1}} V(\delta_{k+1}, \mu_{k+1}) = \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix}^{\mathrm{T}} \lambda_{k+1}.$$

This expression is equivalent to

$$[I_{n\times n}\ C^{\mathrm{T}}] S \begin{bmatrix} \delta_{k+1} \\ \mu_{k+1} \end{bmatrix} = \begin{bmatrix} I_{n\times n} \\ C \end{bmatrix}^{\mathrm{T}} \lambda_{k+1},$$

which yields

$$\lambda_{k+1} = S \begin{bmatrix} \delta_{k+1} \\ \mu_{k+1} \end{bmatrix}. \tag{24}$$

Then, the costate variable λ_{k+1} can be written in terms of the gradient of the value function $\nabla_{\delta u_{k+1}} V(\dots)$, such that

$$\lambda_{k+1} = \nabla_{\delta \mu_{k+1}} V(\delta_{k+1}, \mu_{k+1}) = \begin{bmatrix} \frac{\partial V(\delta_{k+1}, \mu_{k+1})}{\partial \delta_{k+1}} \\ \frac{\partial V(\delta_{k+1}, \mu_{k+1})}{\partial \mu_{k+1}} \end{bmatrix}.$$

Therefore, the value function $V(\dots)$ satisfies the HJ equation (Equation (21)). □

4.2. Hamiltonian–Bellman Solutions Duality

The following results show the conditions at which the Hamiltonian and Bellman-based solutions are dual.

Theorem 2. *(a) Let $\hat{V}(\dots)$ satisfy the following Hamilton–Jacobi–Bellman equation*

$$H\left(\delta_k, \nabla_{\delta u_{k+1}} \hat{V}(\delta_{k+1}, u^o_{k+1}), u^o_k\right) = 0 \tag{25}$$

$$\Rightarrow \nabla_{\delta u_{k+1}} \hat{V}(\delta_{k+1}, u^o_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u^o_{k+1} \end{bmatrix} + F(\delta_k, u^o_k) = 0, \tag{26}$$

with $\hat{V}(0) = 0$, where

$$u^o = -R^{-1} B^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ \frac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_{k+1}} \hat{V}(\delta_{k+1}, u_{k+1}). \tag{27}$$

Then, $\hat{V}(\dots)$ satisfies the Bellman optimality (Equation (9)).

(b) Let (A, B) be reachable. If $V^*(\dots)$ satisfies Equation (9), then it satisfies the Hamilton–Jacobi–Bellman Equation (25).

Proof. (a) The value function $\hat{V}(\delta_k)$ with the optimal policy u^o (Equation (27)) satisfies the HJB equation (Equation (25)). Then, Theorem 1 yields

$$\hat{V}(\delta_{k+1}, u^o_{k+1}) - \hat{V}(\delta_k, u^o_k) = -\nabla_{\delta u^o_{k+1}} \hat{V}(\delta_{k+1}, u^o_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u^o_{k+1} \end{bmatrix}.$$

Therefore, $\hat{V}(\dots)$ satisfies Equation (9).

(b) The Hamiltonian with the value function $\hat{V}(\dots)$, arbitrary policy u_k, and optimal policy u_k^o, evaluated using the optimal value function $\hat{V}(\dots)$ yields

$$H\Big(\delta_k, \nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u_{k+1}), u_k\Big) = H\Big(\delta_k, \nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u^o_{k+1}), u^o_k\Big) + \frac{1}{2}(u_k - u^o_k)^{\mathrm{T}} R (u_k - u^o_k)$$
$$+\nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u_{k+1} \end{bmatrix} - \nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u^o_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u^o_{k+1} \end{bmatrix}.$$

$H\Big(\delta_k, \nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u^o_k), u^o_{k+1}\Big) = 0$. Then,

$$H\Big(\delta_k, \nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u_{k+1}), u_k\Big) = \frac{1}{2}(u_k - u^o_k)^{\mathrm{T}} R (u_k - u^o_k) + \nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u_{k+1} \end{bmatrix}$$
$$-\nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u^o_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u^o_{k+1} \end{bmatrix}. \tag{28}$$

Bellman equation (Equation (6)) can be rearranged such that

$$V(\delta_k, u_k) = \tfrac{1}{2}\left(\delta_k^T\, Q\, \delta_k + u_k^T\, R\, u_k\right) + V(\delta_{k+1}, u_{k+1}) + \nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u_{k+1} \end{bmatrix}$$
$$-\nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u_{k+1} \end{bmatrix}. \tag{29}$$

Equations (28) and (29), and the results from Theorem 1, yield

$$V(\delta_k, u_k) = V(\delta_{k+1}, u_{k+1}) + \frac{1}{2}(u_k - u^o_k)^{\mathrm{T}} R (u_k - u^o_k) - \nabla_{\delta u_{k+1}}\hat{V}(\delta_{k+1}, u^o_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u^o_{k+1} \end{bmatrix}.$$

Applying the optimality principles (i.e., taking the derivative of $V(\dots)$ with respect to u_k) leads to the optimal value function $V^*(\dots)$ and the respective optimal policy u_k^*.

$$\frac{\partial V(\delta_k, u_k)}{\partial u_k} = 0 \;\Rightarrow\; u_k^* - u_k^o = -\,R^{-1}\,B^{\mathrm{T}} \times$$
$$\left(\left[\begin{bmatrix} I_{n\times n} \\ \dfrac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_{k+1}} V^*(\delta_{k+1}, u_{k+1}) \right|_{u=u^*} - \left[\begin{bmatrix} I_{n\times n} \\ \dfrac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_{k+1}} \hat{V}(\delta_{k+1}, u_{k+1}) \right|_{u=u^o} \right). \tag{30}$$

The Hessian of the Hamiltonian (as a function of δ_k, $\hat{V}(\dots)$ and u_k) and the Hessian of the Bellman equation (as a function of δ_k, $V^*(\dots)$ and u_k) are given by

$$\frac{\partial^2 H(\dots)}{\partial\, u_k^2} = \frac{\partial^2 \hat{V}(\dots)}{\partial\, u_k^2} = R + B^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ \dfrac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} \hat{S} \begin{bmatrix} I_{n\times n} \\ \dfrac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix} B,$$
$$\frac{\partial^2 V^*(\dots)}{\partial\, u_k^2} = R + B^{\mathrm{T}} \begin{bmatrix} I_{n\times n} \\ \dfrac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} S^* \begin{bmatrix} I_{n\times n} \\ \dfrac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix} B,$$

where $\hat{S} > 0$ and $S^* > 0$ are the positive-definite solution matrices associated with $\hat{V}(\dots)$ and $V^*(\dots)$, respectively.

Thus, $\partial^2 H(\dots)/\partial\, u_k^2 > 0$ and $\partial^2 \hat{V}(\dots)/\partial\, u_k^2 > 0$. Therefore, the optimal policies u_k^* and u_k^o are unique and $u_k^* = u_k^o$. Consequently, according to Equation (30), $V^*(\dots)$ satisfies the HJB

equation (Equation (25)) (i.e., $V^*(\dots) = \hat{V}(\dots)$) if the system is reachable. This can be explained by incorporating the difference between the costate equations evaluated by the Hamiltonian function and Bellman Equation in Equation (30) so that

$$\nabla_{\delta_k} V^*(\delta_k, u_k) - \nabla_{\delta_k} \hat{V}(\delta_k, u_k) = A^{\mathrm{T}} \left(\left| \begin{bmatrix} I_{n\times n} \\ \frac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_{k+1}} V^*(\delta_{k+1}, u_{k+1}) \right|_{u=u^*} - \left| \begin{bmatrix} I_{n\times n} \\ \frac{\partial u_{k+1}}{\partial \delta_{k+1}} \end{bmatrix}^{\mathrm{T}} \nabla_{\delta u_{k+1}} \hat{V}(\delta_{k+1}, u_{k+1}) \right|_{u=u^o} \right).$$

These results conclude the duality between the Hamiltonian function and Bellman equation for the Action Dependent Dual Heuristic Dynamic Programming solutions. □

5. The Adaptive Learning Solution and Riccati Development

This section introduces the online gradient-based model-free adaptive learning solution which uses the previous HJB development. Then, a Riccati development for the underlying optimal control problem is introduced (it is equivalent to solving the underlying Bellman's optimality (Equation (9)) or the HJB equation (Equation (25)).

5.1. Model-Free Gradient-Based Solution

The results of Theorem 2 are used to develop a gradient-based algorithm which generalizes the ADDHP solution for the optimal control problem using a model-free policy structure. This adaptive learning solution is based on an online policy iteration process. The duality between the Hamilton–Jacobi–Bellman (HJB) equation (Equation (25)) and Bellman optimality (Equation (9)) is leveraged to propose a gradient-based approach that leads to a model-free control strategy. Algorithm 3 is as follows:

Algorithm 3 Online Policy Iteration Process

1. Initialize the costate $\nabla V^0(\delta_k)$ and the policy u_k^o.
2. Evaluate $\nabla_{\delta u} V^\ell(..)$

$$\nabla_{\delta u_{k+1}} V^\ell(\delta_{k+1}, u_{k+1}^\ell)^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ u_{k+1}^\ell \end{bmatrix} = -F(\delta_k, u_k^\ell). \tag{31}$$

3. Update the approximation of the optimal strategy,

$$u_k^{\ell+1} = -\left[S_{u_k u_k}^{-1} \cdot S_{u_k \delta_k} \right]^\ell \cdot \delta_k. \tag{32}$$

4. Terminate on convergence of $\| \nabla V^{\ell+1}(..) - \nabla V^\ell(..) \|$.

5.2. Riccati Development

The following result shows the equivalent Riccati development of the underlying optimal control solution.

Theorem 3. *Let the solution of Equation (9), or equivalently Equation (25), be given by* $V(\delta_k, u_k) = \frac{1}{2}[\delta_k^T \; u_k^T] \Psi \begin{bmatrix} \delta_k \\ u_k \end{bmatrix}$ *and the optimal strategy follows Equation (8). Then, there is a Riccati solution that is given by*

$$\Psi^{r+1} = \begin{bmatrix} Q + A^{\mathrm{c}\mathrm{T}} \Psi^r A^c & A^{\mathrm{T}} \Psi^r B^c \\ B^{\mathrm{c}\mathrm{T}} \Psi^r A^c & R + B^{\mathrm{c}\mathrm{T}} \Psi^r B^c \end{bmatrix}. \tag{33}$$

Note that the parameters of Equation (33) are defined in the proof below.

Proof. The optimal policy in Equation (8) can be written as $u_k = -\hat{\Psi}\,\delta_k$, where $\Psi = \begin{bmatrix} \Psi_{\delta\delta} & \Psi_{\delta u} \\ \Psi_{u\delta} & \Psi_{uu} \end{bmatrix}$ and $\hat{\Psi} = \Psi_{uu}^{-1}\,\Psi_{u\delta}$.

Therefore, the value function $V_{(k+1)}$ can be expressed as

$$V\left(\delta_{(k+1)}, u_{(k+1)}\right) = \frac{1}{2}\delta_{(k+1)}^{\mathrm{T}}\tilde{\Psi}\ \delta_{(k+1)}, \tag{34}$$

where $\tilde{\Psi} = \Psi_{\delta\delta} - \Psi_{\delta u}\Psi_{uu}^{-1}\,\Psi_{u\delta}$.

Substituting the policy in Equation (8) and the value function (Equation (34)) into Equation (2), yields

$$\delta_{(k+1)} = A\delta_k - B\,R^{-1}\,B^{\mathrm{T}}\tilde{\Psi}(A\,\delta_k\ +\ B\,u_k).$$

Then,

$$\delta_{(k+1)} = A^c\,\delta_k + B^c\,u_k, \tag{35}$$

where $A^c = A - B\,R^{-1}\,B^{\mathrm{T}}\tilde{\Psi}A$ and $B^c = -B\,R^{-1}\,B^{\mathrm{T}}\tilde{\Psi}B$.

Substituting Equations (35) and (34) into Bellman equation (Equation (9)) leads to

$$V\left(\delta_k, u_k\right) = \frac{1}{2}\left(\delta_k^{\mathrm{T}}\,Q\,\delta_k + u_k^{\mathrm{T}}\,R\,u_k\right) + \frac{1}{2}\delta_k^{\mathrm{T}}A^{c\mathrm{T}}\tilde{\Psi}A^c\delta_k + \frac{1}{2}u_k^{\mathrm{T}}B^{c\mathrm{T}}\tilde{\Psi}B^c u_k + \frac{1}{2}u_k^{\mathrm{T}}B^{c\mathrm{T}}\tilde{\Psi}A^c\delta_k + \frac{1}{2}\delta_k^{\mathrm{T}}A^{c\mathrm{T}}\tilde{\Psi}B^c u_k.$$

Therefore,

$$[\delta_k^T\ u_k^T]\Psi\begin{bmatrix}\delta_k \\ u_k\end{bmatrix} = [\delta_k^T\ u_k^T]\begin{bmatrix} Q + A^{c\mathrm{T}}\tilde{\Psi}A^c & A^{\mathrm{T}}\tilde{\Psi}B^c \\ B^{c\mathrm{T}}\tilde{\Psi}A^c & R\ + B^{c\mathrm{T}}\tilde{\Psi}B^c \end{bmatrix}\begin{bmatrix}\delta_k \\ u_k\end{bmatrix}.$$

Then,

$$\Psi = \begin{bmatrix} Q + A^{c\mathrm{T}}\tilde{\Psi}A^c & A^{\mathrm{T}}\tilde{\Psi}B^c \\ B^{c\mathrm{T}}\tilde{\Psi}A^c & R\ + B^{c\mathrm{T}}\tilde{\Psi}B^c \end{bmatrix}.$$

This equation yields the Riccati form of Equation (33). □

6. Adaptive Critics Implementations

This section shows the neural network approximation for the online policy iteration solution proposed by Algorithm 3. This implementation represents the optimal value approximation separately form the policy approximation. However, they are both coupled through the Bellman equation or the Hamiltonian function.

6.1. Actor-Critic Neural Networks Implementation

Herein, a simple layer of actor and critic neural network structures is considered. The actor is used to approximate the optimal strategy of Equation (32) while the critic approximates the optimal value in Equation (31). The learning environment involves selecting the values that minimize a cost function along with the associated approximation of the optimal strategies resulting from the feedback and the assessment of the taken strategies. This is done online in real-time where the system dynamics are not required. The weights are adapted through a gradient descent approach.

The value function $V(\delta_k, u_k)$ is approximated by the following quadratic form:

$$\hat{V}(.|W_c) = \frac{1}{2}[\delta_k^T\ \ u_k^T]\ W_c^{\mathrm{T}}\ \begin{bmatrix}\delta_k \\ u_k\end{bmatrix},$$

where $W_c^T = \begin{bmatrix} W_{c\delta\delta}^T & W_{c\delta u}^T \\ W_{cu\delta}^T & W_{cuu}^T \end{bmatrix} \in R^{(n+m)\times(n+m)}$ is a critic weight matrix.

Consequently, the approximation of $\nabla_{\delta u_k} \hat{V}(\dots)$ follows

$$\nabla_{\delta u_k} \hat{V}(.|W_c) = W_c^{\mathrm{T}} \begin{bmatrix} \delta_k \\ u_k \end{bmatrix}.$$

Similarly, the optimal strategy of Equation (32) is approximated by $\hat{u} = W_a^{\mathrm{T}} \ \delta_k$, where $W_a^T \in R^{m\times n}$ is the actor's weight matrix.

To proceed with the policy iteration solution of Algorithm 3, the matrix W_c needs to be transformed to a vector form, such that

$$W_c^{\mathrm{T}} \begin{bmatrix} \delta_k \\ u_k \end{bmatrix} = \bar{W}_c^T \bar{\gamma}_k,$$

where $\bar{W}_c^T \in R^{1\times(n+m)(n+m+1)/2}$ and $\bar{\gamma}_k \in R^{(n+m)(n+m+1)/2\times 1}$ are the vector transformations of the matrix W_c (upper triangle entries) and its respective combination vector evaluated using the entries from $\begin{bmatrix} \delta_k \\ u_k \end{bmatrix}$.

This can be used to formulate Equation (31), such as

$$\bar{W}_c^T \bar{\gamma}_{(k+1)} + \frac{1}{2}(\delta_k^{\mathrm{T}} Q \delta_k + \hat{u}_k^{\mathrm{T}} R \hat{u}_k) = 0.$$

Therefore, the target value of the critic approximation of $-\nabla_{\delta\hat{u}_{k+1}} \hat{V}(\delta_{k+1}, \hat{u}_{k+1})^{\mathrm{T}} \begin{bmatrix} \delta_{k+1} \\ \hat{u}_{k+1} \end{bmatrix}$ is expressed as

$$T^{critic} = \frac{1}{2}(\delta_k^{\mathrm{T}} Q \delta_k + \hat{u}_k^{\mathrm{T}} R \hat{u}_k). \tag{36}$$

Similarly, the target value of the actor approximation, or the optimal strategy, is defined by

$$T^{actor} = -\left[{W_{cu_ku_k}^T}^{-1} \cdot {W^T}_{cu_k\delta_k}\right] \delta_k. \tag{37}$$

The error in the critic approximation is

$$\varepsilon^{critic} = \zeta\left(-\bar{W}_c^T \bar{\gamma}_{k+1} - T^{critic}\right), \tag{38}$$

where $\zeta(\dots)$ is a stacking factor that stores $(n+m)\times(n+m+1)/2$ consecutive values of its argument.

In a similar fashion, the error in the actor's approximation may be written as

$$\varepsilon^{actor} = W_a^{\mathrm{T}} \ \delta_k - T^{actor}. \tag{39}$$

A gradient decent tuning approach is used to tune the actor and the critic weights as follows:

$$W_a^{(\ell+1)\mathrm{T}} = W_a^{\ell\mathrm{T}} - \eta_a \, \varepsilon^{actor} \delta_k^{\ell\mathrm{T}}, \tag{40}$$

$$\bar{W}_c^{(\ell+1)\mathrm{T}} = \bar{W}_c^{\ell\mathrm{T}} - \eta_c \varepsilon^{critic} \zeta \bar{\gamma}_{(k+1)}^{\ell}, \tag{41}$$

where η_a and η_c are the learning rates for the actor and critic weights, respectively.

The following Algorithm 4 shows the online implementation of Algorithm 3 using the actor-critic neural network approximations.

Algorithm 4 Online Model-Free Actor-Critic Neural Network Solution

1. Initialize the neural network weights W_a^0 and W_c^0.
2. Start outside loop (ℓ iterations)
   (a) Initialize the states δ_0^0.
   Start inner loop (q iterations)
   - Transfer the outer critic weights $W_c^q = W_c^\ell$.
   - Evaluate $\delta_{(k+1)}^q$ and $\hat{u}^q$ using Equations (2) and (37).

   End inner loop when $q = (n+m) \times (n+m+1)/2$.
   (b) Evaluate the critic weights using Equation (41).
   (c) Update the actor weights using Equation (40).
3. Terminate on convergence of $\left\| W_c^{\ell+1}(\dots) - W_c^\ell(\dots) \right\|$.

7. Simulation Results

A flexible wing hang glider is used to validate the developed model-free online adaptive learning approach [9]. The continuously varying dynamics of the flexible wing system poses a challenging control problem. This means that the controller operates in a highly uncertain dynamical environment.

The simulation results highlight the stability properties achieved by the controller in addition to monitoring its robustness against the disturbances and the dynamics' uncertainties. Two simulation scenarios are considered: Case I shows the controller's performance in nominal conditions (i.e., at a certain trim speed), while Case II tests the robustness of the developed controller by comparing its performance to the classical Riccati control approach under various disturbances.

7.1. Simulation Parameters

The longitudinal and lateral state space matrices of the flexible wing system, A^{Lo}, B^{Lo}, A^{La}, and B^{La}, at a given trim speed, are used to generate the online measurements [9].

The actor and critic learning parameters are set to $\eta_a = \eta_c = 0.001$. The weight matrices for the longitudinal (R^{Lo}, Q^{Lo}) and lateral (R^{La}, Q^{La}) directions are taken as

$$
\begin{aligned}
R^{Lo} &= 10^{-4} \times \begin{bmatrix} 0.1000 & 0.4000 \end{bmatrix}, \\
Q^{Lo} &= \begin{bmatrix} 0.0100 & 0.0400 & 0.1013 & 0.1013 & 0.4053 & 0.4053 \end{bmatrix}, \\
R^{La} &= 10^{-4} \times \begin{bmatrix} 0.0250 & 0.1000 & 0.4000 \end{bmatrix}, \\
Q^{La} &= \begin{bmatrix} 0.0400 & 0.1013 & 0.1013 & 0.1013 & 0.1013 & 0.4053 & 0.4053 & 0.4053 \end{bmatrix}.
\end{aligned}
$$

The eigenvalue structures of the simulated case studies are given the following graphical notations. The open-loop eigenvalues are denoted by □. The $*$ refer to the closed-loop eigenvalues during the learning process. The eigenvalues resulting from the model-free approach are symbolized by $\times$. The eigenvalues evaluated by the Riccati solution are shown as $\circ$.

7.2. Simulation Case I

This case shows the simulation outcome when the adaptive learning algorithm is applied to control the decoupled longitudinal and lateral dynamical systems in real-time. The open- and closed-loop poles are tabulated in Table 1. The online controller was able to asymptotically stabilize the longitudinal and lateral open-loop systems. The dominant modes are damped much faster than the open-loop system, as shown by the eigenvalue structures in Figure 2. This is further emphasized by Figure 3,

The state space matrices for the longitudinal decoupled dynamics A^{Lo} and B^{Lo} are

$$A^{Lo} = \begin{bmatrix} 0.9906 & 0.0272 & -0.0982 & 0.0006 & 0.1400 & -0.0927 \\ -0.0065 & 0.9828 & 0.0737 & 0.0002 & 0.0504 & -0.0312 \\ 0.0018 & 0.0084 & 0.9501 & 0.0000 & 0.0018 & -0.0002 \\ 0.0057 & -0.0024 & 0.0990 & 0.9990 & -0.1735 & -0.0002 \\ 0.0000 & -0.0000 & 0.0005 & 0.0100 & 0.9991 & -0.0000 \\ 0.0000 & 0.0000 & 0.0097 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix},$$

$$B^{Lo} = \begin{bmatrix} -0.0004 & 0.0002 \\ -0.0002 & 0.0001 \\ -0.0005 & 0.0002 \\ 0.0011 & -0.0005 \\ 0.0000 & -0.0000 \\ -0.0000 & 0.0000 \end{bmatrix}.$$

The state space matrices for the lateral decoupled dynamics A^{La} and B^{La} are

$$A^{La} = \begin{bmatrix} 0.9923 & 0.0437 & -0.0939 & -0.0012 & 0.0000 & -0.2393 & -0.0806 & 0.0924 \\ -0.0150 & 0.7661 & 0.0816 & 0.0000 & -0.0000 & 0.0019 & 0.0003 & -0.0007 \\ 0.0009 & -0.0009 & 0.9949 & 0.0000 & 0.0000 & 0.0011 & 0.0004 & 0.0000 \\ 0.0078 & 0.2202 & -0.0748 & 0.9985 & 0.0000 & -0.2791 & -0.0937 & 0.0004 \\ -0.0060 & -0.0796 & 0.0328 & 0.0000 & 1.0000 & 0.0027 & -0.0004 & -0.0003 \\ 0.0001 & 0.0013 & -0.0005 & 0.0100 & -0.0037 & 0.9986 & -0.0005 & 0.0000 \\ -0.0000 & -0.0004 & 0.0002 & 0.0000 & 0.0106 & 0.0000 & 1.0000 & -0.0000 \\ -0.0001 & 0.0088 & 0.0038 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix},$$

$$B^{La} = \begin{bmatrix} -0.0008 & -0.0000 & 0.0002 \\ 0.0002 & -0.0000 & -0.0000 \\ -0.0000 & 0.0000 & -0.0000 \\ -0.0010 & -0.0000 & 0.0002 \\ 0.0006 & -0.0003 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 \\ 0.0000 & -0.0000 & -0.0000 \\ 0.0000 & -0.0000 & -0.0000 \end{bmatrix}.$$

where the dynamics and the input force control signals are shown to be stable. The adaption process of the actor and the critic neural network weights for both motion frames are shown in Figure 4. The plots demonstrate the converging behavior of the controller. Note that the weights appear in groups as they are too close to show individually. The actor weights are updated after $(n+m) \times (n+m+1)/2$ steps.

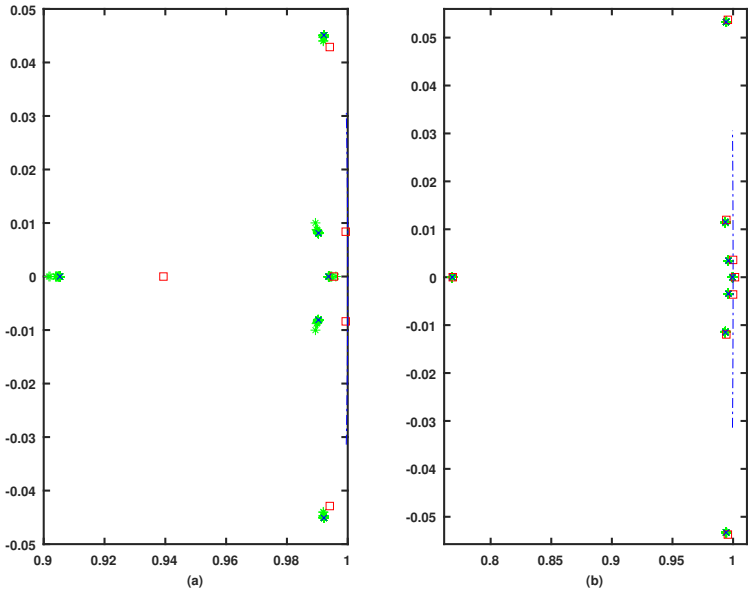

Figure 2. Case I. The eigenvalue structures during the learning process: (**a**) the longitudinal system; and (**b**) the lateral system.

Table 1. Open and closed-loop eigenvalues (Case I).

Longitudinal Dynamics	
Open-Loop System	$0.9394, 0.9950\, e^{\pm 0.0431}$ $0.9954, 0.9994\, e^{\pm 0.0084}$
Closed-Loop system	$0.9053, 0.9904\, e^{\pm 0.0082}$ $0.9932\, e^{\pm 0.0454}, 0.9937$
Lateral Dynamics	
Open-Loop System	$0.7687, 0.9945\, e^{\pm 0.0120}$ $0.9971\, e^{\pm 0.0539}$ $1.0000\, e^{\pm 0.0036}, 1.0016$
Closed-Loop system	$0.7684, 0.9937\, e^{\pm 0.0116}$ $0.9957\, e^{\pm 0.0535}$ $0.9961\, e^{\pm 0.0035}, 0.9995$

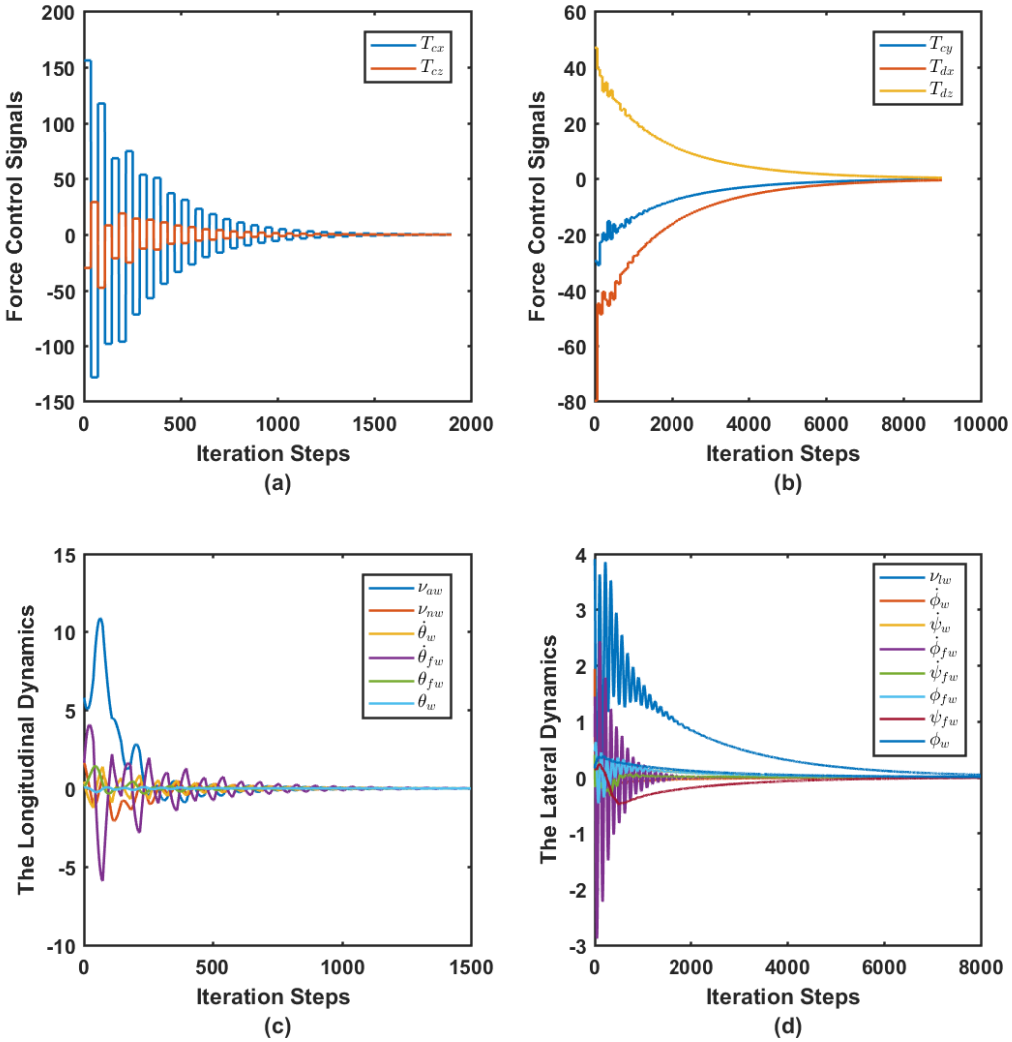

Figure 3. Case I. The longitudinal and lateral force control signals and the dynamics: (**a**) the longitudinal force control signals; (**b**) the lateral force control signals; (**c**) the longitudinal dynamics; and (**d**) the lateral dynamics.

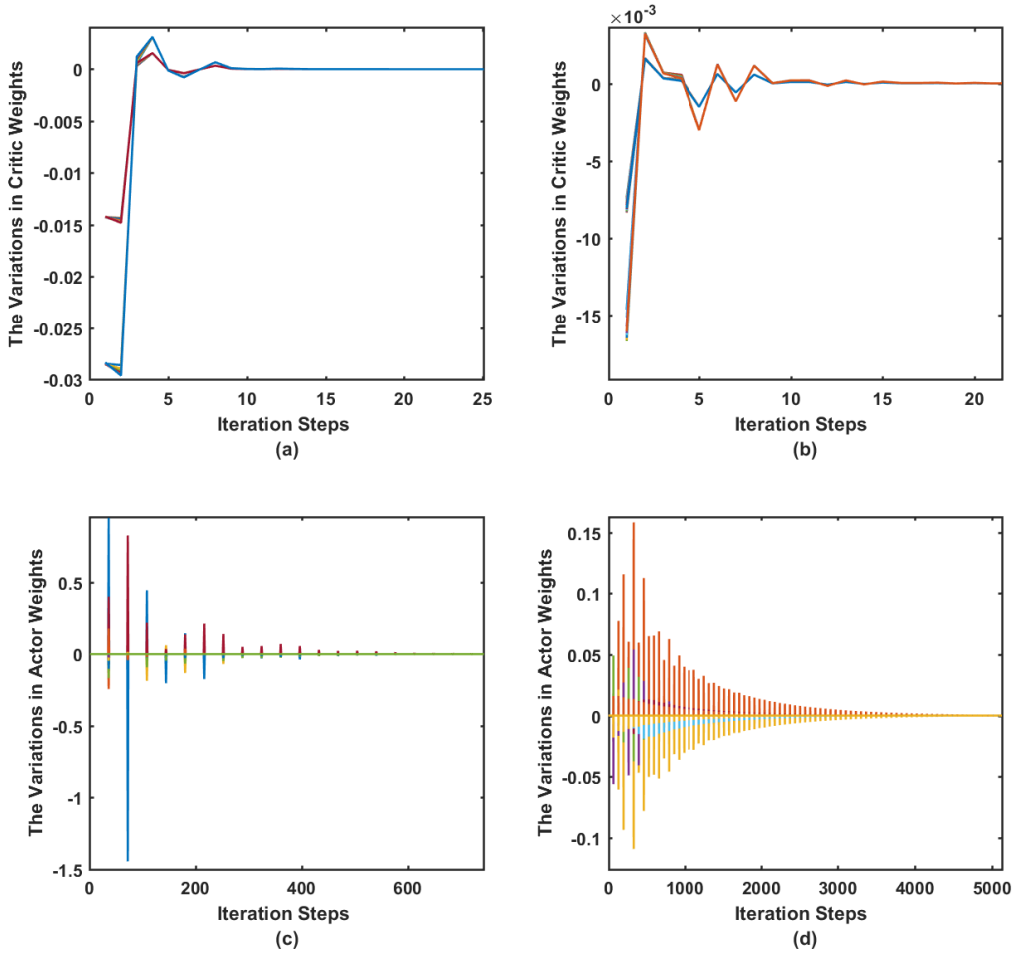

Figure 4. Case I. The variations in the neural networks' weights: (**a**) the variations in the actor's weights for the longitudinal case; (**b**) the variations in the actor's weights for the lateral case; (**c**) the variations in the critic's weights for the longitudinal case; and (**d**) the variations in the critic's weights for the lateral case.

7.3. Simulation Case II

This case tests the robustness of the online reinforcement learning algorithm against the uncertainties in the dynamic environment of the flexible wing system (i.e., the matrices $A^{Lo/La}$ and $B^{Lo/La}$) on top of the disturbances in the longitudinal and lateral states $\delta^{Lo/La}$. The dynamic uncertainties and disturbances in the states are sampled from a normal Gaussian distribution with amplitudes of up to $\pm 50\%$ and $\pm 20\%$ of the nominal values, respectively. This scenario combines the Riccati classical control technique and the developed online adaptive learning approach such that

$$\delta_{k+1} = (A + \tilde{A}_k)(\delta_k + \tilde{\delta}_k) + Bu_k^{Ric} + (\tilde{B} - B)u_k^{RL}.$$

where $\tilde{A}_k$ and $\tilde{B}_k$ are the real-time uncertainties in the drift dynamics A and the control input matrices B. The terms u_k^{Ric} and u_k^{RL} are the control input signals calculated by the Riccati and the online reinforcement learning approaches, respectively.

The eigenvalue structures in Table 2 reveal that the disturbed open-loop systems are unstable. However, the combined approach was able to asymptotically stabilize them. Furthermore, it is able to provide faster longitudinal and lateral dominant modes compared to those obtained by the Riccati solution. This is further emphasized by the dynamics and the force control signals shown in Figure 5. The comparison between the eigenvalue structures obtained in Tables 1 and 2 reveals that the combined approach resulted in faster response compared to the Riccati solution and the original response of the

longitudinal and the lateral systems. The convergence behavior of the actor-critic weights is shown in Figure 6, where the adaptation of the weights takes longer this time due to the higher complexity of the problem in hand. The eigenvalues evolution during the learning process is shown in Figure 7. The eigenvalues eventually converge to a stable region.

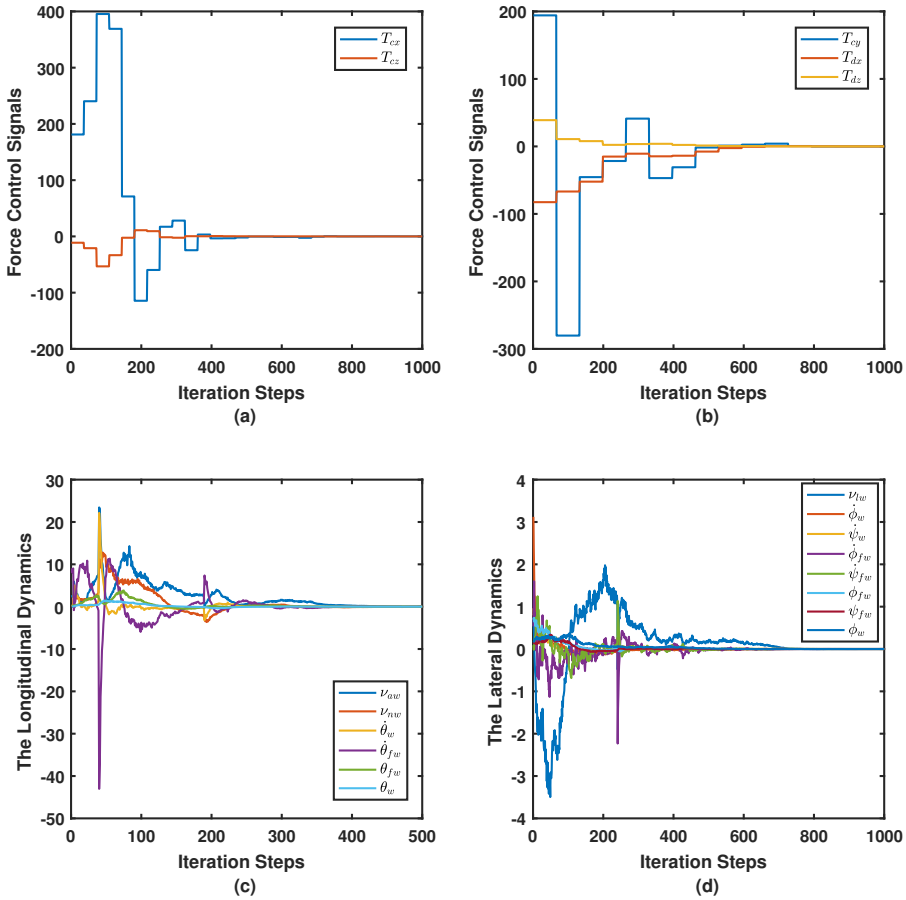

Figure 5. Case II. The longitudinal and lateral force control signals and the dynamics: (**a**) the longitudinal force control signals; (**b**) the lateral force control signals; (**c**) the longitudinal dynamics; and (**d**) the lateral dynamics.

Table 2. Open and closed-loop eigenvalues (Case II).

Longitudinal Dynamics	
Open-Loop System (Disturbed System)	0.9387, 0.9938, 0.9940 $e^{\pm 0.0404}$ 1.0002 $e^{\pm 0.0080}$
Closed-Loop System (Riccati Solution)	0.8289, 0.9551, 0.9818 $e^{\pm 0.0308}$ 0.9953 $e^{\pm 0.0092}$
Closed-Loop System (Model-Free Solution)	0.8271, 0.9533, 0.9824 $e^{\pm 0.0298}$ 0.9932 $e^{\pm 0.0104}$
Lateral Dynamics	
Open-Loop System (Disturbed System)	0.7346, 0.9972 $e^{\pm 0.0578}$ 0.9948 $e^{\pm 0.0121}$, 0.9996 $e^{\pm 0.0038}$, 1.0021
Open-Loop System (Riccati Solution)	0.6685, 0.7963, 0.9778 $e^{\pm 0.0254}$ 0.9944 $e^{\pm 0.0126}$ 0.9750, 0.9945
Closed-Loop system (Model-Free Solution)	0.6622, 0.7956, 0.9492 0.9799 $e^{\pm 0.0156}$, 0.9942 $e^{\pm 0.0102}$ 0.9942

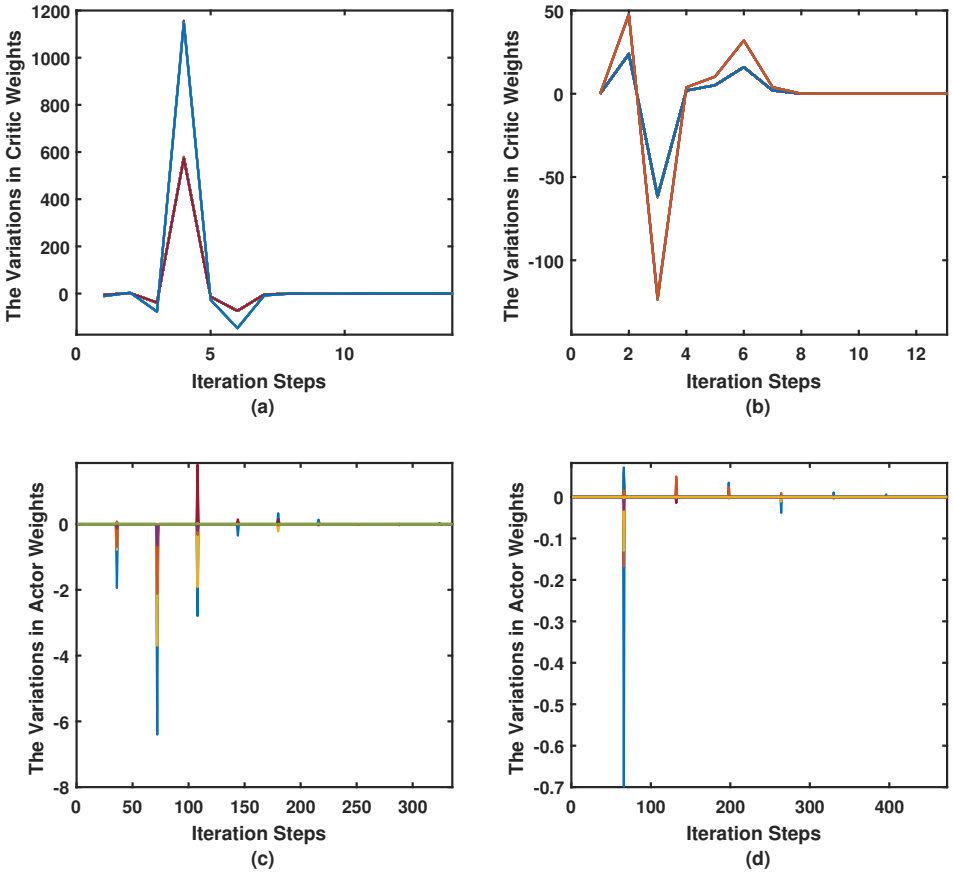

Figure 6. Case II. The variations in the neural networks' weights: (**a**) the variations in the actor's weights for the longitudinal case; (**b**) the variations in the actor's weights for the lateral case; (**c**) the variations in the critic's weights for the longitudinal case; and (**d**) the variations in the critic's weights for the lateral case.

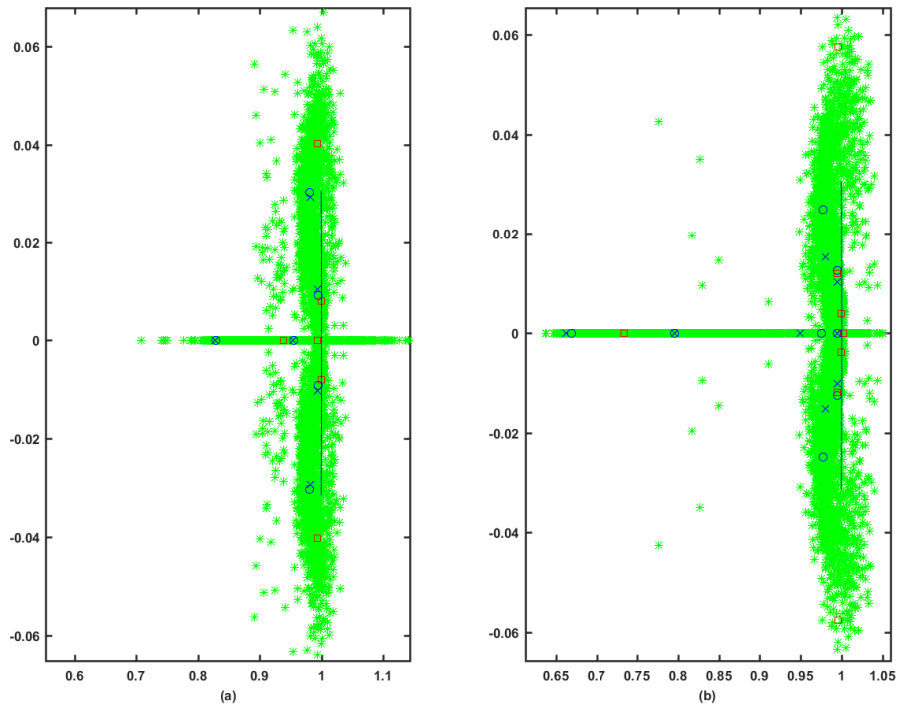

Figure 7. Case II. The eigenvalue structures during the learning process: (**a**) the longitudinal system; and (**b**) the lateral system.

8. Conclusions

A novel online policy iteration process is developed to generalize model-free gradient based solutions for optimal control problems. The approach is considered a sub-class of the classical action dependent dual heuristic dynamic programming. The mathematical layout showed the duality between the Hamilton–Jacobi–Bellman formulation and the underlying model-free Bellman's optimality setup. Unlike traditional costate-based solutions, the suggested method does not depend on the system's dynamics. A Riccati solution is developed and is shown to be equivalent to solving the Bellman's optimality equation. Artificial neural network-based approximations are employed to provide a real-time implementation of the policy iteration solution. This is accomplished using separate neural network structures to approximate the optimal strategy and the associated gradient of the solving value function. The performance of the proposed control scheme is demonstrated on a flexible wing aircraft. The simulation scenarios proved the effectiveness of the proposed controller under a wide range of uncertainties and disturbances in the system dynamics.

Author Contributions: This article is an outcome of the research primarily conducted by M.A. He conceived, designed, and performed the experiments. He also wrote most of the paper. W.G. supervised and directed the research. He also helped analyze the results and contributed in writing and editing the article. F.L. played an advisory role.

Funding: This research was partially funded by Ontario Centres of Excellence (OCE).

Conflicts of Interest: The authors declare no conflict of interest. The funding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

Variables	
ν_{aw}, ν_{lw}, ν_{nw}	Axial, lateral, and normal velocities in the wing's frame of motion.
θ_w, ϕ_w, ψ_w	Pitch, roll, and yaw angles in the wing's frame of motion.
$\dot{\theta}_w$, $\dot{\phi}_w$, $\dot{\psi}_w$	Pitch, roll, and yaw angle rates in the wing's frame of motion.
θ_{fw}, ϕ_{fw}, ψ_{fw}	Pitch, roll, and yaw angles of the fuselage relative to the wing's frame of motion.
$\dot{\theta}_{fw}$, $\dot{\phi}_{fw}$, $\dot{\psi}_{fw}$	Pitch, roll, and yaw angle rates of the fuselage relative to the wing's frame of motion.
$T_{R,L}$	Right and left internal forces on the control bar.
Subscripts	
$(\cdot)_{x,y,z}$	X, Y, and Z Cartesian components of $(\cdot)$, respectively.
Abbreviations	
ADP	Adaptive Dynamic Programming
ADDHP	Action Dependent Dual Heuristic Dynamic Programming
DHP	Dual Heuristic Dynamic Programming
HJB	Hamilton–Jacobi–Bellman
RL	Reinforcement Learning

References

1. Bertsekas, D.; Tsitsiklis, J. *Neuro-Dynamic Programming*, 1st ed.; Athena Scientific: Belmont, MA, USA, 1996.
2. Werbos, P. Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences. Ph.D. Thesis, Harvard University, Cambridge, MA, USA, 1974.
3. Werbos, P. Neural Networks for Control and System Identification. In Proceedings of the 28th Conference on Decision and Control, Tampa, FL, USA, 13–15 December 1989; pp. 260–265.
4. Miller, W.T.; Sutton, R.S.; Werbos, P.J. *Neural Networks for Control: A Menu of Designs for Reinforcement Learning Over Time*, 1st ed.; MIT Press: Cambridge, MA, USA, 1990; pp. 67–95.
5. Werbos, P. Approximate Dynamic Programming for Real-time Control and Neural Modeling. In *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*; Chapter 13; White, D.A., Sofge, D.A., Eds.; Van Nostrand Reinhold: New York, NY, USA, 1992.

6. Howard, R.A. *Dynamic Programming and Markov Processes*; Four Volumes; MIT Press: Cambridge, MA, USA, 1960.
7. Abouheaf, M.; Lewis, F. Dynamic Graphical Games: Online Adaptive Learning Solutions Using Approximate Dynamic Programming. In *Frontiers of Intelligent Control and Information Processing*; Chapter 1; Liu, D., Alippi, C., Zhao, D., Zhang, H., Eds.; World Scientific: Singapore, 2014; pp. 1–48.
8. Blake, D. Modelling The Aerodynamics, Stability and Control of The Hang Glider. Master's Thesis, Centre for Aeronautics—Cranfield University, Silsoe, UK, 1991.
9. Cook, M.; Spottiswoode, M. Modelling the Flight Dynamics of the Hang Glider. *Aeronaut. J.* **2005**, *109*, I–XX. [CrossRef]
10. Cook, M.V.; Kilkenny, E.A. An experimental investigation of the aerodynamics of the hang glider. In Proceedings of the an International Conference on Aerodynamics, London, UK, 15–18 October 1986.
11. De Matteis, G. Response of Hang Gliders to Control. *Aeronaut. J.* **1990**, *94*, 289–294. [CrossRef]
12. de Matteis, G. Dynamics of Hang Gliders. *J. Guid Control Dyn.* **1991**, *14*, 1145–1152. [CrossRef]
13. Kilkenny, E.A. *An Evaluation of a Mobile Aerodynamic Test Facility for Hang Glider Wings*; Technical Report 8330; College of Aeronautics, Cranfield Institute of Technology: Cranfield, UK, 1983.
14. Kilkenny, E. *Full Scale Wind Tunnel Tests on Hang Glider Pilots*; Technical Report; Cranfield Institute of Technology, College of Aeronautics, Department of Aerodynamics: Cranfield, UK, 1984.
15. Kilkenny, E.A. An Experimental Study of the Longitudinal Aerodynamic and Static Stability Characteristics of Hang Gliders. Ph.D. Thesis, Cranfield University, Silsoe, UK, 1986.
16. Vrancx, P.; Verbeeck, K.; Nowe, A. Decentralized Learning in Markov Games. *IEEE Trans. Syst. Man Cybern. Part B* **2008**, *38*, 976–981.
17. Webros, P.J. A Menu of Designs for Reinforcement Learning over Time. In *Neural Networks for Control*; Miller, W.T., III, Sutton, R.S., Werbos, P.J., Eds.; MIT Press: Cambridge, MA, USA, 1990; pp. 67–95.
18. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 1998.
19. Si, J.; Barto, A.; Powell, W.; Wunsch, D. *Handbook of Learning and Approximate Dynamic Programming*; The Institute of Electrical and Electronics Engineers, Inc.: New York, NY, USA, 2004.
20. Prokhorov, D.; Wunsch, D. Adaptive Critic Designs. *IEEE Trans. Neural Netw.* **1997**, *8*, 997–1007.
21. Abouheaf, M.; Lewis, F.; Vamvoudakis, K.; Haesaert, S.; Babuska, R. Multi-Agent Discrete-Time Graphical Games And Reinforcement Learning Solutions. *Automatica* **2014**, *50*, 3038–3053.
22. Lewis, F.; Vrabie, D.; Syrmos, V. *Optimal Control*, 3rd ed.; John Wiley: New York, NY, USA, 2012.
23. Bellman, R. *Dynamic Programming*; Princeton University Press: Princeton, NJ, USA, 1957.
24. Abouheaf, M.; Lewis, F. Approximate Dynamic Programming Solutions of Multi-Agent Graphical Games Using Actor-critic Network Structures. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, 4–9 August 2013; pp. 1–8.
25. Abouheaf, M.; Lewis, F.; Mahmoud, M.; Mikulski, D. Discrete-time Dynamic Graphical Games: Model-free Reinforcement Learning Solution. *Control Theory Technol.* **2015**, *13*, 55–69.
26. Abouheaf, M.; Gueaieb, W. Multi-Agent Reinforcement Learning Approach Based on Reduced Value Function Approximations. In Proceedings of the IEEE International Symposium on Robotics and Intelligent Sensors (IRIS), Ottawa, ON, Canada, 5–7 October 2017; pp. 111–116.
27. Widrow, B.; Gupta, N.K.; Maitra, S. Punish/reward: Learning with a Critic in Adaptive Threshold Systems. *IEEE Trans. Syst. Man Cybern.* **1973**, *SMC-3*, 455–465.
28. Webros, P.J. Neurocontrol and Supervised Learning: An Overview and Evaluation. 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. 65–89.
29. Busoniu, L.; Babuska, R.; Schutter, B.D. A Comprehensive Survey of Multi-Agent Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Part C* **2008**, *38*, 156–172.
30. Abouheaf, M.; Mahmoud, M. Policy Iteration and Coupled Riccati Solutions for Dynamic Graphical Games. *Int. J. Digit. Signals Smart Syst.* **2017**, *1*, 143–162.
31. Lewis, F.L.; Vrabie, D. Reinforcement learning and adaptive dynamic programming for feedback control. *IEEE Circuits Syst. Mag.* **2009**, *9*, 32–50.
32. Vrabie, D.; Lewis, F.; Pastravanu, O.; Abu-Khalaf, M. Adaptive Optimal Control for Continuous-Time Linear Systems Based on Policy Iteration. *Automatica* **2009**, *45*, 477–484.

33. Abouheaf, M.I.; Lewis, F.L.; Mahmoud, M.S. Differential graphical games: Policy iteration solutions and coupled Riccati formulation. In Proceedings of the 2014 European Control Conference (ECC), Strasbourg, France, 24–27 June 2014; pp. 1594–1599.
34. Asma Al-Tamimi, F.L.L.; Abu-Khalaf, M. Model-Free Q-Learning Designs for Linear Discrete-Time Zero-Sum Games with Application to H-infinity Control. *Automatica* **2007**, *43*, 473–481.
35. Bahare Kiumarsi, F.L.L. Actor–Critic-Based Optimal Tracking for Partially Unknown Nonlinear Discrete-Time Systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2105**, *26*, 140–151.
36. Kiumarsi, B.; Vamvoudakis, K.G.; Modares, H.; Lewis, F.L. Optimal and Autonomous Control Using Reinforcement Learning: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2042–2062.
37. Cook, M.V. The Theory of the Longitudinal Static Stability of the Hang-glider. *Aeronaut. J.* **1994**, *98*, 292–304. [CrossRef]
38. Ochi, Y. Modeling of the Longitudinal Dynamics of a Hang Glider. In Proceedings of the AIAA Modeling and Simulation Technologies Conference, American Institute of Aeronautics and Astronautics, Kissimmee, FL, USA, 5–9 January 2015; pp. 1591–1608.
39. Ochi, Y. Modeling of Flight Dynamics and Pilot's Handling of a Hang Glider. In Proceedings of the AIAA Modeling and Simulation Technologies Conference, American Institute of Aeronautics and Astronautics, Grapevine, TX, USA, 9–13 January 2017; pp. 1758–1776.
40. Sweeting, J. An Experimental Investigation of Hang Glider Stability. Master's Thesis, College of Aeronautics, Cranfield University, Silsoe, UK, 1981.
41. Cook, M. *Flight Dynamics Principles*; Butterworth-Heinemann: London, UK, 2012.
42. Kroo, I. *Aerodynamics, Aeroelasticity and Stability of Hang Gliders*; Stanford University: Stanford, CA, USA, 1983.
43. Spottiswoode, M. A Theoretical Study of the Lateral-directional Dynamics, Stability and Control of the Hang Glider. Master's Thesis, College of Aeronautics, Cranfield Institute of Technology, Cranfield, UK, 2001.
44. Cook, M.V. (Ed.) *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control*, 3rd ed.; Aerospace Engineering; Butterworth-Heinemann: Oxford, UK, 2013.