IET Journals
The Institution of Engineering and Technology

Online Model-Free Reinforcement Learning for the Automatic Control of a Flexible Wing Aircraft

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Mohammed Abouheaf[1,2*], *Wail Gueaieb*[1], *and Frank Lewis*[3]

[1] *School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario, Canada*
[2] *Electrical Engineering, College of Energy Engineering, Aswan University, Aswan, Egypt*
[3] *Electrical Engineering, University of Texas at Arlington, Arlington, Texas, United States*
* *E-mail: mohammed.abouheaf@uottawa.ca*

Abstract: The control problem of the flexible wing aircraft is challenging due to the prevailing and high nonlinear deformations in the flexible wing system. This urged for new control mechanisms that are robust to the real-time variations in the wing's aerodynamics. An online control mechanism based on a value iteration reinforcement learning process is developed for flexible wing aerial structures. It employs a model-free control policy framework and a guaranteed convergent adaptive learning architecture to solve the system's Bellman optimality equation. A Riccati equation is derived and shown to be equivalent to solving the underlying Bellman equation. The online reinforcement learning solution is implemented using means of an adaptive-critic mechanism. The controller is proven to be asymptotically stable in the Lyapunov sense. It is assessed through computer simulations and its superior performance is demonstrated on two scenarios under different operating conditions.

Keywords: Flexible Wing Aircraft, Optimal Control, Reinforcement Learning, Value Iteration, Adaptive Critics.

1 Introduction

Unmanned flexible wing aircraft have gained an increasing attention due to their low operating cost, long endurance, and relatively short runway. Recent research investigations tackled several aspects of flexible wing systems, including modeling approaches and control mechanisms. Unlike fixed wing aircraft, the high nonlinear deformation in the flexible wing makes it very challenging to model such type of aircraft and to design autonomous controllers for them [1–9]. Classical control approaches depend on accurate system models, and so may not be suitable for these unmanned aerial vehicles (UAVs). In this paper, an innovative model-free adaptive learning controller is developed for the automatic real-time control of flexible wing aircraft. This work brings together ideas form the machine learning, optimal control, and adaptive critics.

A significant body of fundamental research has already been conducted on flexible wing aircraft [2–7, 9, 10]. The early studies of hang gliders involved aeroelastic and aerodynamic modeling approaches, which resulted in approximate aerodynamic equations of motion [11]. A mobile test rig was developed in [2] to test the aerodynamic characteristics of a set of hang glider wings. In addition, a fundamental database of wind tunnel tests for hang gliders is provided in [3]. Powton [12] studied the effects of the aerodynamic nonlinearities in terms of the wing's camber and twist variations. A study of the aerodynamic models for flexible wing aircraft along with a practical measurement setup was conducted in [1]. Furthermore, a number of researchers attempted to decouple the aircraft's motion along the longitudinal and lateral planes. For example, Kroo [11] developed longitudinal and lateral small perturbation equations of motion in [11]. A longitudinal-lateral aerodynamic model is usually derived by assuming a rigid wing while the aerodynamic derivatives are added at the end of the modeling process [13, 14]. Another of such models was developed by Spottiswoode [15].

The dynamics of a flexible wing system are usually modeled by considering two interacting bodies (the wing and the pilot/fuselage frame) under geometric and kinematic constraints [16]. A nine-degree-of-freedom (DOF) flight dynamical model has been recently proposed by Ochi [8] with a pilot-handling feedback control mechanism. The work followed a rigorous analytical process which led to a set of nonlinear state equations.

One of the most commonly methods of controlling a flexible wing aircraft is to use a weight shift mechanism [2–4]. In this case, the motion of the aircraft is governed by the shift in the relative centers of gravity of the wing and the fuselage (as opposed to the fuselage weight shift itself) [1]. It is shown that the geometry of the control arms affects the the maximum control moments in a flexible wing system [1] and that the static pitching stability can be increased by lowering the center of gravity. A mathematical model is employed in [1] to determine the stability of a flexible wing vehicle within a speed envelope. It is shown that the lateral stability margins are larger compared to those of a typical airplane. The initial investigations about the longitudinal stability of the hang glider were conducted in [6]. The classical longitudinal stability is shown to be effective in understanding the robustness of the hang glider provided that some conditions are satisfied [7]. The effect of the wing's aerodynamic nonlinearities on the pitching moment is studied in [12]. The longitudinal stability and control properties of the flexible wing UAV are investigated in [17]. More stability analyses were conducted in [13, 14] but in the frequency domain this time. A regulation system combined of two control structures for planar vertical takeoff and landing aircraft is proposed in [18]. The vertical variable is regulated using saturated-feedback linearization mechanism, while the angular and horizontal variables are controlled using combined sliding mode-PD control mechanism. Regulator schemes are proposed for magnetic generator and hexarotor mechanical systems where the stability is analyzed using Lyapunov mathematical framework [19]. Another controller based on a Lyapunov method and robust feedback linearization scheme is proposed for nonlinear processes in [20].

Artificial Intelligence (AI) has become a vivid area to solve various optimal control problems for single and multi-agent systems. Dynamic programming is typically applied to solve optimization problems in the framework of optimal control theory [21, 22]. However, dynamic programming methods generally suffer from the curse of dimensionality in both the state and action spaces. Hence, Approximate Dynamic Approaches (ADP) are developed to

alleviate these difficulties by using Reinforcement Learning (RL) approaches and means of adaptive critics to provide temporal difference solutions for the optimal control problems [21–25]. The ADP approaches can be classified into four main categories: (i) Heuristic Dynamic Programming; (ii) Dual Heuristic Dynamic Programming; (iii) Action Dependent Heuristic Dynamic Programming; and (iv) Action Dependent Dual Heuristic Dynamic Programming. These categories depend on the form of the solving value function and the accompanying choice of the optimal control actions. Different versions of approximate dynamic programming solutions are proposed to solve multi-agent control problems in [26–28]. These include a class of innovative implementations for the respective costate equations of the underlying control problems. Classical mechanics are used to derive the basic optimality principles to solve for optimal control problems in [29–31]. The minimum cost-to-go function is shown to be equivalent to the solution of the underlying Hamilton-Jacobi-Bellman equation (HJB) [29–31]. The optimal control theory provides the platform to derive the optimality conditions to solve various types of control problems using the ADP approaches [21, 29]. These conditions involve expressions for the optimal value functions and the respective optimal control policies, and hence result in the underlying Bellman optimality, Hamilton-Jacobi-Bellman (HJB), and costate equations.

Reinforcement Learning (RL) approaches optimize the performance of the dynamical systems by minimizing the respective infinite horizon cumulative sum of a cost function [25, 32]. They are developed using two-step processes referred to as either value iteration or policy iteration, depending on the strategy followed to solve the underlying problem [23–25]. The main differences between the two are in the structures of the evaluated value functions, which in turn affects the update of the optimal policies. This is equivalent to solving the Hamilton-Jacobi-Bellman equation for the considered dynamical system. An online policy iteration approach is used to solve dynamic graphical games in [26]. Some RL solutions are implemented using means of the adaptive critics in [25]. The implementations were carried out using separate actor-critic neural network structures, where the actor approximates the optimal policy, while the critic approximates the optimal value function [33]. The analytical methodology for the critic learning process was presented in [33]. The actor-critic learning mechanism is implemented forward-in-time and is accomplished via a dynamic learning environment, where the quality of the taken actions are evaluated using a utility function [25]. The adaptive critics are used to implement some optimal control problem solutions in [22, 34, 35]. More RL approaches are used to solve a number of multi-agent optimization problems in game theoretic frameworks in [36–39].

This work contributes with an innovative computational and mathematical framework to design model-free control schemes for a class of nonlinear processes with unknown dynamical models like flexible wing systems. The proposed control scheme has the ability to provide model-free optimal control decisions while remaining robust to the unmodeled uncertainties and disturbances. The adaptive learning scheme is able to solve the underlying model-free Bellman optimality equation (temporal difference equation) of the dynamical system in real-time. The dynamic learning system is realized using value iteration process with two supporting separate neural network structures, where gradient descent approach is used to tune the neural networks weights. A Riccati development is introduced to understand the duality between the model-based control solution and its equivalent model-free temporal difference solution.

The paper is organized as follows: Section 2 briefly introduces the weight shift control mechanism of a flexible wing aircraft and defines its state space representation. Section 3 discusses the optimal control formulation of the problem along with its Heuristic Dynamic Programming solution. In Section 4, we lay out the mathematical foundation for the model-free optimal control development, leading to the online adaptive learning algorithm along with its convergence proof. The duality between the developed model-free Bellman optimality equation and the associated Riccati formulation is explained in Section 5. The proposed online adaptive control scheme is based on an adaptive-critic technique that is implemented through two artificial neural networks. Section 6 details the design of these neural networks. Section 7 presents different simulation scenarios to demonstrate the salient features of the proposed controller. Finally, the manuscript is concluded with a few concluding remarks in Section 8.

2 Flexible Wing Aircraft Model

The basic principles of operation of a flexible wing aircraft are briefly explained here. The framework adopted in this manuscript is based on decoupling the flight dynamics of a flexible wing system into longitudinal and lateral frames of motion. This is based on the findings in [1, 7]. The resulting dynamical models are used later on to validate the performance of the adaptive learning control algorithm.

The flexible wing aircraft is characterized by the mechanical coupling of two bodies: a wing and a fuselage which is suspended below the wing through a pivot (hang strap). A schematic diagram of a flexible wing system is shown in Figure 1. The coupling between the two structures imposes a set of dynamical and kinematic constraints on the system. The flexible wing aircraft employs an interesting pitch-roll control mechanism by pitching and rolling the control bar. These angles are the only two signals used to control the hang glider. They are denoted by α (roll) and β (pitch) in Figure 1. Such pitch and roll movements on the control bar lead to a relative shift between the fuselage and the wing's centers of gravity. This creates aerodynamic moments which act on the aircraft's center of gravity. The inherent flexibility of the wing results in high nonlinear aerodynamics. The dynamics of the aircraft are referred to the wing's frames of motion (longitudinal and lateral). The approximate flight dynamics are modeled at trim speeds in [1, 7]. The classical control approaches require a precise mathematical model of the system, which is difficult to obtain for systems with this magnitude of complexity.

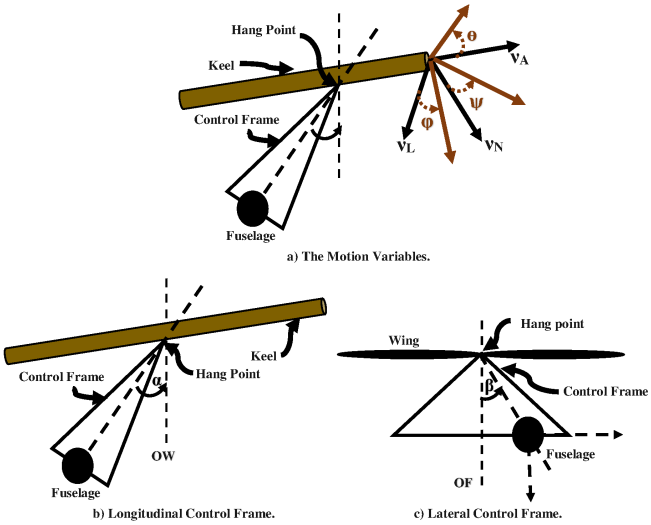

Fig. 1: Schematic Diagram of the Flexible Wing System.

The following discrete-time linear model is developed for a flexible wing aircraft based on the nonlinear state space representation and the semi-experimental study conducted in [1]:

$$\boldsymbol{Z}_{k+1} = \boldsymbol{A}\boldsymbol{Z}_k + \boldsymbol{B}\boldsymbol{u}_k \tag{1}$$

where the states along the longitudinal and lateral frames are $\boldsymbol{Z}^{Lon} = [\nu_A \ \nu_N \ \dot{\theta} \ \theta]^T$ and $\boldsymbol{Z}^{Lat} = [\nu_L \ \dot{\phi} \ \dot{\psi} \ \phi \ \psi]^T$, respectively. The states θ, ϕ, and ψ, denote the glider's pitch, roll, and yaw attitude angles. The dot on top of a signal symbolizes its time derivative. The variables ν_A, ν_N, and ν_L represent the axial, normal, and lateral linear velocities. The control signals of the longitudinal and lateral subsystems are $\boldsymbol{u}^{Lon} = \alpha$ and $\boldsymbol{u}^{Lat} = \beta$, respectively.

It is worth pointing out that this model does not account for the interaction forces between the fuselage and the wing. Also, it

models the fuselage's reaction control moment as a spring-damper system. More information about the assumptions and limitations of the model can be found in [1]. It is also important to articulate the fact that the above model is only used for simulating the hang glider's dynamical behavior under the control strategy to be detailed later. The latter, however, is model independent.

3 Heuristic Dynamic Programming Solution

This section, shows the Heuristic Dynamic Programming solution for the underlying optimal control problem of the flexible wing aircraft. This solution is implemented online using a value iteration process, where the solution depends partially on the dynamics of the flexible wing system. The following development is used to frame the model-free adaptive learning controller.

Consider a system described by the following discrete-time state space mode:

$$\boldsymbol{Z}_{(k+1)} = \boldsymbol{A}\,\boldsymbol{Z}_k + \boldsymbol{B}\,\boldsymbol{u}_k, \tag{2}$$

where k is the time index, $\boldsymbol{Z} \in \mathbb{R}^n$ is a vector of the states, $\boldsymbol{u} \in \mathbb{R}^m$ is a vector of the control input signals, and $\boldsymbol{A}$ and $\boldsymbol{B}$ are the system and input matrices, respectively. A performance measure index $J = \sum_{k=0}^{\infty} U(\boldsymbol{Z}_k, \boldsymbol{u}_k)$ is employed to measure the quality of the taken control policies using a quadratic convex cost function $U(\boldsymbol{Z}_k, \boldsymbol{u}_k) = \frac{1}{2}\left(\boldsymbol{Z}_k^{\mathrm{T}}\,\boldsymbol{Q}\,\boldsymbol{Z}_k + \boldsymbol{u}_k^{\mathrm{T}}\,\boldsymbol{R}\,\boldsymbol{u}_k\right)$, and where $\boldsymbol{R} > 0 \in \mathbb{R}^{m\times m}$ and $\boldsymbol{Q} \geq 0 \in \mathbb{R}^{n\times n}$ are positive and semi-positive definite matrices that are symmetric. Let $S(\boldsymbol{Z}_k)$ be a value function that is quadratic in the states $\boldsymbol{Z}_\ell$, such that

$$S(\boldsymbol{Z}_k) = \sum_{r=k}^{\infty} U(\boldsymbol{Z}_r, \boldsymbol{u}_r) \tag{3}$$

It is possible to define Bellman equation as follows:

$$S(\boldsymbol{Z}_k) \;=\; \frac{1}{2}(\boldsymbol{Z}_k^{\mathrm{T}}\,\boldsymbol{Q}\,\boldsymbol{Z}_k + \boldsymbol{u}_k^{\mathrm{T}}\,\boldsymbol{R}\,\boldsymbol{u}_k) + S(\boldsymbol{Z}_{(k+1)}). \tag{4}$$

The optimal policies are evaluated using the Bellman optimality principles, which result in the stationarity condition (the optimal control policy u^o) [26, 28, 29]. This is achieved by applying the gradient of (4) with respect to the control signal $\boldsymbol{u}_k$, $S^o(\boldsymbol{Z}_k) = \mathrm{argmin}_{\boldsymbol{u}_k} S(\boldsymbol{Z}_k)$, which yields

$$\boldsymbol{u}_k^o = -\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\,\nabla S^o(\boldsymbol{Z}_{(k+1)}), \tag{5}$$

where $\nabla S^o(\boldsymbol{Z}_k) = \partial S^o(\boldsymbol{Z}_k)\,/\,\partial \boldsymbol{Z}_k$. Plugging the optimal policy (5) in (4) leads to the Bellman optimality equation (defining the optimal value function $S^o(\dots)$),

$$S^o(\boldsymbol{Z}_k) = \frac{1}{2}\left(\boldsymbol{Z}_k^{\mathrm{T}}\,\boldsymbol{Q}\,\boldsymbol{Z}_k + \boldsymbol{u}_k^{o\mathrm{T}}\,\boldsymbol{R}\,\boldsymbol{u}_k^o\right) + S^o(\boldsymbol{Z}_{(k+1)}). \tag{6}$$

The solution of the Bellman optimality equation (6) can be implemented either offline or online. A value iteration process is employed here to solve the underlying Heuristic Dynamic Programming Problem as follows:

Algorithm 1: HDP-Value Iteration Solution

1.Initialize $S^0(\boldsymbol{Z}_k)$ and $\boldsymbol{u}_k^0$.
2.Evaluate the value $S^{(\ell+1)}(\dots)$

$$S^{(\ell+1)}(\boldsymbol{Z}_k) = U^\ell(\boldsymbol{Z}_k, \boldsymbol{u}_k) + S^\ell(\boldsymbol{Z}_{(k+1)}), \tag{7}$$

where ℓ is the iteration index.
3.Update the control action

$$\boldsymbol{u}_k^{(\ell+1)} = -\boldsymbol{R}^{-1}\boldsymbol{B}^{\mathrm{T}}\,\nabla S^{(\ell+1)}(\boldsymbol{Z}_{(k+1)}). \tag{8}$$

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

The weighting matrix R is a positive definite matrix, which guarantees that (5) and consequently (8) have non-singular values. Later on, the optimal control policy will not be a function of R.

This algorithm represents an online solution for the optimal control problem. However, the optimal control action (8) uses partial knowledge about the system's dynamics (i.e., $\boldsymbol{u}^\ell$ depends on the input control matrix $\boldsymbol{B}$). In the following, we will introduce a control mechanism that is able to produce model-free policies.

4 Model-Free Control Mechanism

Herein, an online model-free control algorithm is proposed. First, a modified form of Bellman optimality equation (6) is obtained with a model-free policy structure. Then, a convergence analysis of the resultant adaptive learning controller is carried out.

4.1 Modified Bellman Equation

Consider the modified form of the value function (3), which is quadratic in the states $\boldsymbol{Z}_k$ and the control signals u_k. This yields the following modified Bellman equation:

$$S(\boldsymbol{Z}_k, \boldsymbol{u}_k) = U(\boldsymbol{Z}_k, \boldsymbol{u}_k) + S(\boldsymbol{Z}_{(k+1)}, \boldsymbol{u}_{(k+1)}). \tag{9}$$

Neglecting the high-order terms of Taylor's series expansion of the value function (9), $S(\dots)$ can be approximated by

$$S(\boldsymbol{Z}_k, \boldsymbol{u}_k) \approx \frac{1}{2}\begin{bmatrix}\boldsymbol{Z}_k^{\mathrm{T}} & \boldsymbol{u}_k^{\mathrm{T}}\end{bmatrix}\boldsymbol{M}\begin{bmatrix}\boldsymbol{Z}_k \\ \boldsymbol{u}_k\end{bmatrix}, \tag{10}$$

where $\boldsymbol{M} = \begin{bmatrix}\boldsymbol{M}_{\boldsymbol{ZZ}} & \boldsymbol{M}_{\boldsymbol{Zu}} \\ \boldsymbol{M}_{\boldsymbol{uZ}} & \boldsymbol{M}_{\boldsymbol{uu}}\end{bmatrix} > 0 \in \mathbb{R}^{(n+m)\times(n+m)}$.

Similarly, the Bellman's optimality principles are applied to the difference equation (9), or equivalently the value function (10), to obtain the optimal policy $\boldsymbol{u}_k^o$ and the optimal value $S^o(\dots)$.

$$\boldsymbol{u}_k^o = \mathrm{argmin}_{\boldsymbol{u}_k} S(\boldsymbol{Z}_k, \boldsymbol{u}_k) = -\boldsymbol{M}_{\boldsymbol{u}_k\boldsymbol{u}_k}^{-1}\cdot\boldsymbol{M}_{\boldsymbol{u}_k\boldsymbol{Z}_k}\cdot\boldsymbol{Z}_k. \tag{11}$$

Remark 1. *The optimal policy (11) relies on the successive approximations of value function $S(\dots)$ using Bellman temporal difference equation (9) (i.e., evaluating matrix M) as will be explained in the following online model-free value iteration process. Later on, the convergence analysis emphasizes the positive definiteness of the obtained solutions towards the optimal solution starting with initial positive definite value function (i.e., $M^0 > 0$). This guarantees that the optimal policies will not have any singular values since the block matrix $\boldsymbol{M}_{\boldsymbol{uu}}$ is positive definite at all evaluation steps.*

Substituting (11) in (9) yields a model-free of Bellman optimality equation,

$$S^o(\boldsymbol{Z}_k, \boldsymbol{u}_k) = \frac{1}{2}\left(\boldsymbol{Z}_k^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{Z}_k + \boldsymbol{u}_k^{o\mathrm{T}}\boldsymbol{R}\boldsymbol{u}_k^o\right) + S^o(\boldsymbol{Z}_{(k+1)}, \boldsymbol{u}_{(k+1)}^o). \tag{12}$$

Based on this solution, a model-free control algorithm is developed to solve for the optimal value function $S^o(\dots)$ and the optimal strategy $\boldsymbol{u}^o$, evaluated by (11), as listed below.

Algorithm 2: Value Iteration Control Algorithm

1.Initialize $S^0(\boldsymbol{Z}_k, \boldsymbol{u}_k^0)$ and $\boldsymbol{u}_k^0$ with admissible values.
2.Update the value function $S^{(\ell+1)}(\dots)$

$$S^{(\ell+1)}(\boldsymbol{Z}_k, \boldsymbol{u}_k) = U^\ell(\boldsymbol{Z}_k, \boldsymbol{u}_k) + S^\ell(\boldsymbol{Z}_{(k+1)}, \boldsymbol{u}_{(k+1)}), \tag{13}$$

where ℓ is the iteration index.

3.Update the policy

$$u_{(k+1)}^{(\ell+1)} = -\boldsymbol{M}_{\boldsymbol{uu}}^{-1(\ell+1)} \boldsymbol{M}_{\boldsymbol{uZ}}^{(\ell+1)} \boldsymbol{Z}_{(k+1)}^{\ell+1}. \quad (14)$$

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

4.2 Value Iteration Policy Convergence

For the controller to be useful it is important to have guaranteed convergence property. This can be proven by studying the evolution of the value functions $S^{\ell}(\dots)$ during the online learning process of Algorithm 2.

Theorem 1. *Consider the dynamic system (2) with the adaptive control law of Algorithm 2. Then,*

1.*The equilibrium point* $(\boldsymbol{Z}_k, \boldsymbol{u}_k) = (\boldsymbol{0}, \boldsymbol{0})$ *is asymptotically stable.*
2.*The sequence of the value function* $S^{\ell}(\dots)$*, for* $\ell = 0, 1, \dots,$ *is monotonically ascending and converges to the optimal solution of (12).*
3.*The sequence of policies (14) is stabilizing.*

Proof:

1.The solving value function $S(\dots)$, which takes the form (10), is a Lyapunov candidate function. This function is continuously differentiable with respect to $\boldsymbol{Z}$ and $\boldsymbol{u}$, where $S(\boldsymbol{Z}_k = \boldsymbol{0}, \boldsymbol{u}_k = \boldsymbol{0}) = 0$, $S(\boldsymbol{Z}_k, \boldsymbol{u}_k) = \sum_{r=k}^{\infty} U(\boldsymbol{Z}_r, \boldsymbol{u}_r) > 0, \forall (\boldsymbol{Z}_k, \boldsymbol{u}_k) \neq (\boldsymbol{0}, \boldsymbol{0})$.
The policy (14) is stationary with the following value

$$S(\boldsymbol{Z}_k) = \boldsymbol{Z}_k^T \left(\boldsymbol{M}_{\boldsymbol{Z}_k \boldsymbol{Z}_k} - \boldsymbol{M}_{\boldsymbol{Z}_k \boldsymbol{u}_k} \boldsymbol{M}_{\boldsymbol{u}_k \boldsymbol{u}_k}^{-1} \boldsymbol{M}_{\boldsymbol{u}_k \boldsymbol{Z}_k} \right) \boldsymbol{Z}_k > 0.$$

Schur complement of the symmetric matrix $\boldsymbol{M}_{\boldsymbol{u}_k \boldsymbol{u}_k}$ is positive definite such that $\left(\boldsymbol{M}_{\boldsymbol{Z}_k \boldsymbol{Z}_k} - \boldsymbol{M}_{\boldsymbol{Z}_k \boldsymbol{u}_k} \boldsymbol{M}_{\boldsymbol{u}_k \boldsymbol{u}_k}^{-1} \boldsymbol{M}_{\boldsymbol{u}_k \boldsymbol{Z}_k} \right) > 0$. This means that, block matrices $\boldsymbol{M}_{\boldsymbol{Z}_k \boldsymbol{Z}_k}$ and $\boldsymbol{M}_{\boldsymbol{u}_k \boldsymbol{u}_k}$ are also positive definite. Hence, the hessians of $S(\boldsymbol{Z}_k, \boldsymbol{u}_k)$ with respect to $\boldsymbol{Z}_k$ and $\boldsymbol{u}_k$ are given by $\dfrac{\partial^2 S(\boldsymbol{Z}_k, \boldsymbol{u}_u k)}{\partial \boldsymbol{Z}_k^2} = \boldsymbol{M}_{\boldsymbol{Z}_k \boldsymbol{Z}_k} > 0$ and $\dfrac{\partial^2 S(\boldsymbol{Z}_k, \boldsymbol{u}_k)}{\partial \boldsymbol{u}_k^2} = \boldsymbol{M}_{\boldsymbol{u}_k \boldsymbol{u}_k} > 0$ respectively. Therefore, $(\boldsymbol{0}, \boldsymbol{0})$ is an equilibrium point for the convex function $S(\dots)$. Since the following holds

$$S(\boldsymbol{Z}_{(k+1)}, \boldsymbol{u}_{(k+1)}) - S(\boldsymbol{Z}_k, \boldsymbol{u}_k) = -U(\boldsymbol{Z}_k, \boldsymbol{u}_k) < 0, \forall (\boldsymbol{Z}_k, \boldsymbol{u}_k) \neq (\boldsymbol{0}, \boldsymbol{0}). \quad (15)$$

Therefore, $S(..)$ is a Lyapunov function and the equilibrium point $(\boldsymbol{0}, \boldsymbol{0})$ is asymptotically stable.
2.Let the initial value function $S^0 \geq 0$ be upper bounded above by a constant C. Applying an arbitrary policy $\boldsymbol{\mu}$ yields

$$0 \leq S^0(\boldsymbol{Z}_k, \tilde{\boldsymbol{u}}) \leq S^0(\boldsymbol{Z}_k, \boldsymbol{\mu}_k) \leq C, \quad (16)$$

where $\tilde{\boldsymbol{u}}$ is a policy satisfying (14).
Using a stabilizing policy $\boldsymbol{u}$ and the iterative process (13) we get

$$S^{\ell+1}(\boldsymbol{Z}_k, \boldsymbol{u}) - S^{\ell}(\boldsymbol{Z}_k, \boldsymbol{u}) = S^{\ell}(\boldsymbol{Z}_{(k+1)}, \boldsymbol{u}) + \frac{1}{2}\left(\boldsymbol{Z}_k^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{Z}_k + \boldsymbol{u}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u} \right) - S^{\ell-1}(\boldsymbol{Z}_{(k+1)}, \boldsymbol{u}) - \frac{1}{2}\left(\boldsymbol{Z}_k^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{Z}_k + \boldsymbol{u}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u} \right).$$

This equality can be extended into the following arrangement:

$$\begin{aligned} & S^{\ell+1}(\boldsymbol{Z}_k, \boldsymbol{u}) - S^{\ell}(\boldsymbol{Z}_k, \boldsymbol{u}) \\ &= S^{\ell}(\boldsymbol{Z}_{(k+1)}, \boldsymbol{u}) - S^{\ell-1}(\boldsymbol{Z}_{(k+1)}, \boldsymbol{u}) \\ &= S^{(\ell-1)}(\boldsymbol{Z}_{(k+2)}, \boldsymbol{u}) - S^{\ell-2}(\boldsymbol{Z}_{(k+2)}, \boldsymbol{u}) \\ &= S^{(\ell-2)}(\boldsymbol{Z}_{(k+3)}, \boldsymbol{u}) - S^{\ell-3}(\boldsymbol{Z}_{(k+3)}, \boldsymbol{u}) \\ &= \vdots \\ &= S^{1}(\boldsymbol{Z}_{(k+\ell)}, \boldsymbol{u}) - S^{0}(\boldsymbol{Z}_{(k+\ell)}, \boldsymbol{u}). \end{aligned}$$

Rearranging this equality in a finite summation form yields

$$\begin{aligned} & S^{\ell+1}(\boldsymbol{Z}_k, \boldsymbol{u}) \\ &= S^{1}(\boldsymbol{Z}_{(k+\ell)}, \boldsymbol{u}) - S^{0}(\boldsymbol{Z}_{(k+\ell)}, \boldsymbol{u}) + S^{\ell}(\boldsymbol{Z}_k, \boldsymbol{u}) \\ &= S^{1}(\boldsymbol{Z}_{(k+\ell)}, \boldsymbol{u}) - S^{0}(\boldsymbol{Z}_{(k+\ell)}, \boldsymbol{u}) \\ &+ S^{1}(\boldsymbol{Z}_{(k+\ell-1)}, \boldsymbol{u}) - S^{0}(\boldsymbol{Z}_{(k+\ell-1)}, \boldsymbol{u}) + S^{\ell-1}(\boldsymbol{Z}_k, \boldsymbol{u}) \\ &= S^{1}(\boldsymbol{Z}_{(k+\ell)}, \boldsymbol{u}) - S^{0}(\boldsymbol{Z}_{(k+\ell)}, \boldsymbol{u}) \\ &+ S^{1}(\boldsymbol{Z}_{(k+\ell-1)}, \boldsymbol{u}) - S^{0}(\boldsymbol{Z}_{(k+\ell-1)}, \boldsymbol{u}) \\ &+ S^{1}(\boldsymbol{Z}_{(k+\ell-2)}, \boldsymbol{u}) - S^{0}(\boldsymbol{Z}_{(k+\ell-2)}, \boldsymbol{u}) + S^{\ell-2}(\boldsymbol{Z}_k, \boldsymbol{u}) \\ &= \sum_{n=0}^{\ell} S^{1}(\boldsymbol{Z}_{(k+n)}, \boldsymbol{u}) - \sum_{n=0}^{\ell} S^{0}(\boldsymbol{Z}_{(k+n)}, \boldsymbol{u}) + S^{0}(\boldsymbol{Z}_k, \boldsymbol{u}) \\ &= \sum_{n=0}^{\ell} S^{1}(\boldsymbol{Z}_{(k+n)}, \boldsymbol{u}) - \sum_{n=1}^{\ell} S^{0}(\boldsymbol{Z}_{(k+n)}, \boldsymbol{u}). \end{aligned}$$

Using the utility function U in this summation yields

$$\begin{aligned} S^{\ell+1}(\boldsymbol{Z}_k, \boldsymbol{u}) &= S^{0}(\boldsymbol{Z}_{(k+\ell+1)}, \boldsymbol{u}) \\ &+ \sum_{n=0}^{\ell} \frac{1}{2}\left(\boldsymbol{Z}_{(k+n)}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{Z}_{(k+n)} + \boldsymbol{u}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u} \right). \\ &\leq S^{0}(\boldsymbol{Z}_{(k+\ell+1)}, \boldsymbol{u}) \\ &+ \sum_{n=0}^{\infty} \frac{1}{2}\left(\boldsymbol{Z}_{(k+n)}^{\mathrm{T}} \boldsymbol{Q} \boldsymbol{Z}_{(k+n)} + \boldsymbol{u}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u} \right). \quad (17) \end{aligned}$$

Using the stability result of part 1, we deduce that the summation $\sum_{n=0}^{\infty} U(\boldsymbol{Z}_{(k+n)}, \boldsymbol{u})$ is upper bounded by a constant $\bar{C}$. Then, (16) and (17) lead to, $0 \leq S^{\ell+1}(\boldsymbol{Z}_k, \boldsymbol{u}) \leq C + \bar{C}, \forall k, \ell$. Therefore,

$$0 \leq \dots \leq S^0 \leq S^1 \leq \dots \leq S^{\ell} \dots \leq S^* = \lim_{\ell \to \infty} S^{\ell}. \quad (18)$$

Hence, the sequence S^{ℓ}, $\ell = 0, 1, \dots$, converges to an optimal value function S^* which is the optimal solution of the modified model-free Bellman optimality equation (12).
3.Inequality (18) and the sequence of policies (14) lead to

$$0 \leq S^0(\boldsymbol{Z}_k, \boldsymbol{u}^0) \leq S^1(\boldsymbol{Z}_k, \boldsymbol{u}^1) \leq \dots \leq S^{\ell}(\boldsymbol{Z}_k, \boldsymbol{u}^{\ell}) \dots \leq S^*(\boldsymbol{Z}_k, \boldsymbol{u}^*). \quad (19)$$

Thus,

$$0 \leq S^{\ell}(\boldsymbol{Z}_k, \boldsymbol{u}_k^o) \leq S^{\ell}(\boldsymbol{Z}_k, \boldsymbol{u}_k), \forall \ell. \quad (20)$$

for some arbitrary stabilizing policy $\boldsymbol{u}$ and optimal policy $\boldsymbol{u}^o$.
Inequalities (19) and (20) guarantee that the sequence of policies $\boldsymbol{u}_k^{\ell}$, $\ell = 0, 1, \dots$, is stabilizing and convergent to the optimal policy $\boldsymbol{u}^* = -\boldsymbol{M}_{\boldsymbol{uu}}^{-1*} \cdot \boldsymbol{M}_{\boldsymbol{uZ}}^{*} \cdot \boldsymbol{Z}_k$, where $S^*(\boldsymbol{Z}_k, \boldsymbol{u}_k) =$

$\frac{1}{2}[\boldsymbol{Z}_k^{\mathrm{T}}\ \boldsymbol{u}_k^{\mathrm{T}}]M^*\begin{bmatrix}\boldsymbol{Z}_k\\ \boldsymbol{u}_k\end{bmatrix}$ is the optimal value function and $\boldsymbol{M}^* = \begin{bmatrix}\boldsymbol{M}^*_{\boldsymbol{ZZ}} & \boldsymbol{M}^*_{\boldsymbol{Zu}}\\ \boldsymbol{M}^*_{\boldsymbol{uZ}} & \boldsymbol{M}^*_{\boldsymbol{uu}}\end{bmatrix}$ is the optimal gain.

□

5 Riccati Control Solution

The next development lays out the mathematical foundation for the Riccati solution for the underlying optimal control problem. This solution is equivalent to solving "recursively" the modified Bellman optimality equation (12) using the model-free policies (11).

Theorem 2. *Let the value* $S(\boldsymbol{Z}_k, \boldsymbol{u}_k) = \frac{1}{2}[\boldsymbol{Z}_k^T\ \boldsymbol{u}_k^T]\cdot\boldsymbol{\Psi}\cdot\begin{bmatrix}\boldsymbol{Z}_k\\ \boldsymbol{u}_k\end{bmatrix}$ *be the value iteration function used to solve the Bellman optimality condition (12) following policies (11), where* $\boldsymbol{\Psi}$ *is a Riccati gain matrix of the following form:*

$$\boldsymbol{\Psi} = \begin{bmatrix}\boldsymbol{\Psi}_{\boldsymbol{ZZ}} & \boldsymbol{\Psi}_{\boldsymbol{Zu}}\\ \boldsymbol{\Psi}_{\boldsymbol{uZ}} & \boldsymbol{\Psi}_{\boldsymbol{uu}}\end{bmatrix}$$

Also, let $\hat{\boldsymbol{\Psi}} = \boldsymbol{\Psi}_{\boldsymbol{uu}}^{-1}\boldsymbol{\Psi}_{\boldsymbol{uZ}}$ *and* $\tilde{\boldsymbol{\Psi}} = \boldsymbol{\Psi}_{\boldsymbol{ZZ}} - \boldsymbol{\Psi}_{\boldsymbol{uu}}^{-1}\boldsymbol{\Psi}_{\boldsymbol{uZ}}$*. Then, the Riccati solution can be written in the following recursive expression:*

$$\boldsymbol{\Psi}^{\ell+1} = \begin{bmatrix}\boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}^{\ell}\boldsymbol{A} + \hat{\boldsymbol{\Psi}}^{\ell\mathrm{T}}\boldsymbol{R}\hat{\boldsymbol{\Psi}}^{\ell} + \boldsymbol{Q} & \boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}^{\ell}\boldsymbol{B}\\ \boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}^{\ell}\boldsymbol{A} & \boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}^{\ell}\boldsymbol{B}\end{bmatrix}. \quad (21)$$

Proof: Using the above definition of the value function $S(\dots)$, the optimal policy may be expressed as

$$\boldsymbol{u}_k^o = -\boldsymbol{\Psi}_{\boldsymbol{uu}}^{-1}\boldsymbol{\Psi}_{\boldsymbol{uZ}}\boldsymbol{Z}_k = -\hat{\boldsymbol{\Psi}}\boldsymbol{Z}_k. \quad (22)$$

The value of function $S(\dots)$ at time-step k is

$$S\big(\boldsymbol{Z}_{(k+1)}, \boldsymbol{u}_{(k+1)}\big) = \frac{1}{2}\boldsymbol{Z}_{(k+1)}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\boldsymbol{Z}_{(k+1)}. \quad (23)$$

Using (2), (22), and (23) in (12), yields

$$\begin{aligned}S(\boldsymbol{Z}_k, \boldsymbol{u}_k) = {}& \frac{1}{2}\Big(\boldsymbol{Z}_k^{\mathrm{T}}\,\boldsymbol{Q}\,\boldsymbol{Z}_k + \boldsymbol{Z}_k^{\mathrm{T}}\hat{\boldsymbol{\Psi}}^{\mathrm{T}}\,\boldsymbol{R}\,\hat{\boldsymbol{\Psi}}\,\boldsymbol{Z}_k\Big)\\ & + \frac{1}{2}\Big(\boldsymbol{Z}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\boldsymbol{A}\boldsymbol{Z}_k + \boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\boldsymbol{B}\boldsymbol{u}_k\\ & + \boldsymbol{u}_k^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\boldsymbol{A}\boldsymbol{Z}_k + \boldsymbol{Z}_k^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\boldsymbol{B}\boldsymbol{u}_k\Big). \quad (24)\end{aligned}$$

This equation can be rearranged as

$$[\boldsymbol{Z}_k^T\ \boldsymbol{u}_k^T]\boldsymbol{\Psi}\begin{bmatrix}\boldsymbol{Z}_k\\ \boldsymbol{u}_k\end{bmatrix} = [\boldsymbol{Z}_k^T\ \boldsymbol{u}_k^T]\begin{bmatrix}\boldsymbol{Q} + \hat{\boldsymbol{\Psi}}^{\mathrm{T}}\boldsymbol{R}\hat{\boldsymbol{\Psi}} + \boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\boldsymbol{A} & \boldsymbol{A}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\boldsymbol{B}\\ \boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\boldsymbol{A} & \boldsymbol{B}^{\mathrm{T}}\tilde{\boldsymbol{\Psi}}\boldsymbol{B}\end{bmatrix}\begin{bmatrix}\boldsymbol{Z}_k\\ \boldsymbol{u}_k\end{bmatrix}.$$

Therefore, a Riccati equation can be formulated as

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

This solution form is equivalent to solving the underlying modified Bellman equation (12) using the model-free policies (11), which results in the recursive solution form (21). □

6 Actor-Critic Design

In order to apply Algorithm 2 to solve the model-free optimality equation (12) using the model-free policies (11), it is necessary to approximate the value function $\hat{S}^o\big(\boldsymbol{Z}_k, \boldsymbol{u}_k^o\big)$ and the optimal policy $\boldsymbol{u}_k^o$. This is accomplished through an actor-critic neural network structure. The critic approximates the value function while the actor approximates the optimal policy. Each of the two structures is implemented as a single-layer feedforward neural network where the weights are tuned in real-time using a gradient descent approach.

Since the model-free optimal control policy is a function of $\boldsymbol{Z}_k$, a policy may be approximated by the actor's output as $\hat{\boldsymbol{u}}_k = \boldsymbol{W}_a^{\mathrm{T}}\boldsymbol{Z}_k$, where the matrix $\boldsymbol{W}_a^{\mathrm{T}} \in \mathbb{R}^{n\times m}$ represents the weights of the actor neural network. The value function is quadratic in the states and the model-free control policy. Then, a similar critic structure can be proposed, such that $\hat{S}(\boldsymbol{Z}_k, \hat{\boldsymbol{u}}_k) = \frac{1}{2}\boldsymbol{\chi}_k^{\mathrm{T}}\boldsymbol{W}_c^{\mathrm{T}}\boldsymbol{\chi}_k$, where the matrix $\boldsymbol{W}_c^{\mathrm{T}} \in \mathbb{R}^{(n+m)\times(n+m)}$ represents the weights of the critic's neural network and $\boldsymbol{\chi}_k^{\mathrm{T}} = [\boldsymbol{Z}_k^{\mathrm{T}}\ \hat{\boldsymbol{u}}_k^{\mathrm{T}}]$.

The actor's approximation error is $\delta_{\boldsymbol{W}_a} = \hat{\boldsymbol{u}}_k - \boldsymbol{u}_k^{target}$, where the target value of the mode-free optimal policy is

$$\boldsymbol{u}_k^{target} = -\Big[(\boldsymbol{W}_{c\hat{\boldsymbol{u}}_k\hat{\boldsymbol{u}}_k})^{-1}\,\boldsymbol{W}_{c\hat{\boldsymbol{u}}_k\boldsymbol{Z}_k}^{T}\Big]\cdot\boldsymbol{Z}_k. \quad (25)$$

Following a gradient descent approach to tune the weights leads to the following weight adaptation law:

$$\boldsymbol{W}_a^{(\ell+1)\mathrm{T}} = \boldsymbol{W}_a^{(\ell)\mathrm{T}} - \eta_{W_a}\Big(\hat{\boldsymbol{u}}_k^{(\ell)} - \boldsymbol{u}_k^{(\ell)desired}\Big)\boldsymbol{Z}_k^{\mathrm{T}},$$

where $0 < \eta_{W_a} < 1$ is an actor's learning rate.

Similarly, the critic's approximation error is $\delta_{W_c} = \hat{S}_k - S_k^{target}$, where the target value function is

$$S_k^{target} = \frac{1}{2}\Big(\boldsymbol{Z}_k^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{Z}_k + \hat{\boldsymbol{u}}_k^{\mathrm{T}}\boldsymbol{R}\hat{\boldsymbol{u}}_k\Big) + \hat{S}(\boldsymbol{Z}_{(k+1)}, \hat{\boldsymbol{u}}_{(k+1)}).$$

Differentiating with respect to the approximation weights and applying the chain rule yields the following adaptive law:

$$\boldsymbol{W}_c^{(\ell+1)\mathrm{T}} = \boldsymbol{W}_c^{(\ell)\mathrm{T}} - \eta_{W_c}\Big(\hat{S}_k^{(\ell)} - S_k^{(\ell)desired}\Big)\boldsymbol{\chi}_k\boldsymbol{\chi}_k^{\mathrm{T}}, \quad (26)$$

with $0 < \eta_{W_c} < 1$ being a critic's learning rate.

The following algorithm summarizes the online neural network implementation of the adaptive actor-critic technique.

Algorithm 3: Neural Network Implementation of the Actor-Critic Technique

1.Initialize the actor and critic weights, $\boldsymbol{W}_a^0$ and $\boldsymbol{W}_c^0$.
2.Calculate the approximate model-free policy $\hat{\boldsymbol{u}}_k^{\ell}$ using (24).
3.Apply (25) to update the actor weights.
4.Calculate the measurements $\hat{\boldsymbol{Z}}_{(k+1)}^{\ell}$.
5.Apply (26) to update the critic weights.
6.Terminate upon convergence of $\|\hat{S}^{\ell+1}(\dots) - \hat{S}^{\ell}(\dots)\|$.

7 Results and Discussion

To validate the model-free adaptive controller, it is applied to a flexible wing aircraft model. Two case studies are simulated. They are designed to assess the convergence of the learning process and the controller's stability and robustness in the absence of a dynamic model and in the face of various disturbance levels.

The flexible wing airplane model experimented in [1] is used to generate the necessary measurements at a trim speed of $10.8\,\mathrm{m\,s^{-1}}$. This value is based on experimental data collected by Kilkenny in [5] to construct a flight envelope at different speeds to obtain valid aerodynamic data for a flexible wing system. Table 1 shows the configuration parameters of the Hiway Demon hang glider adopted in

Table 1 Hang Glider (Hiway Demon) Configuration Parameters

Parameter	Value
pilot mass	80 kg
wing mass	31 kg
wing area	16.26 m^2
wing span	10 m
reference chord length	1.626 m
hang point position	0.04 m
hang strap length	1.2 m
control frame position	0.06 m
control frame height	1.65 m
distance between the pilot's hands on the control bar	0.3 m

this manuscript. Since this work is concerned with the automatic control of such types of aerial vehicles, the pilot may be regarded as a cargo mass, for instance.

The discrete-time state space matrices of the decoupled longitudinal and lateral planes are given by

$$\boldsymbol{A}^{Lon} = \begin{bmatrix} 0.9982 & 0.0065 & 0.0012 & -0.0971 \\ -0.0139 & 0.9774 & 0.1055 & 0.0136 \\ 0.0027 & -0.0043 & 0.9858 & -0.0002 \\ 0.0000 & -0.0000 & 0.0099 & 1.0000 \end{bmatrix}$$

$$\boldsymbol{A}^{Lat} = \begin{bmatrix} 0.9977 & -0.0028 & -0.1069 & 0.0971 & -0.0131 \\ -0.0131 & 0.8092 & 0.0677 & -0.0007 & 0.0001 \\ 0.0026 & 0.0332 & 0.9802 & 0.0001 & -0.0000 \\ -0.0001 & 0.0090 & 0.0004 & 1.0000 & 0.0000 \\ 0.0000 & 0.0002 & 0.0099 & 0.0000 & 1.0000 \end{bmatrix}$$

$$\boldsymbol{B}^{Lon} = \begin{bmatrix} 0.0000 \\ 0.0040 \\ 0.0741 \\ 0.0004 \end{bmatrix}, \quad \boldsymbol{B}^{Lat} = \begin{bmatrix} -0.0003 \\ 0.0327 \\ 0.0049 \\ 0.0002 \\ 0.0000 \end{bmatrix}$$

with initial conditions $\boldsymbol{Z}_0^{Lon} = \begin{bmatrix} 28 & -1.0 & -0.6 & 1.0 \end{bmatrix}^T$ and $\boldsymbol{Z}_0^{Lat} = \begin{bmatrix} 10 & 0.9 & 0.9 & 1.0 & -0.5 \end{bmatrix}^T$. For a more realistic setup the control signals $\boldsymbol{u}^{Lon} = \alpha$ and $\boldsymbol{u}^{Lat} = \beta$ are bound to $\pm\pi/3$ rad.

The weighting matrices are set to

$$\boldsymbol{Q}^{Lon} = \begin{bmatrix} 0.0006 & 0.0400 & 1.0000 & 1.0000 \end{bmatrix}$$
$$\boldsymbol{Q}^{Lat} = \begin{bmatrix} 0.0006 & 0.2500 & 0.2500 & 1.0000 & 1.0000 \end{bmatrix}$$
$$R^{Lon} = 0.9803\,, \; R^{Lat} = 0.9803$$

The actor weights are initialized to

$$\boldsymbol{W}_a^{Lon} = \begin{bmatrix} 0.0317 & 0.0014 & -2.4171 & -3.0740 \end{bmatrix}^T$$
$$\boldsymbol{W}_a^{Lat} = \begin{bmatrix} 0.0404 & -0.6407 & -2.2064 & -1.8473 & -1.8872 \end{bmatrix}^T.$$

The critic weights are initialized to positive definite matrices, $\boldsymbol{W}_c^{Lon} = 10\, I_{5\times5}$, $\boldsymbol{W}_c^{Lat} = 10\, I_{6\times6}$. The actor and critic learning rates are fixed to $\eta_{W_a} = \eta_{W_c} = 0.001$. The simulations are performed in Matlab 2017 on a server with 16 virtual CPUs and 48 GB of memory. The sampling period is taken as 0.01 s.

7.1 Case Study 1

In the first case study, the aforementioned longitudinal and lateral state space models are used to simulate the hang glider. The system's open- and closed-loop poles are listed in Table 2. As can be noticed, both the longitudinal and lateral open-loop dynamics are unstable. Integrating the proposed controller in a feedback loop asymptotically stabilized the system by shifting the poles to a stable region.

The optimal control gains for longitudinal and lateral directions are given as follows:

$$\boldsymbol{W}_a^{Lon} = \begin{bmatrix} 0.5229 & -0.9582 & -2.6512 & -2.5554 \end{bmatrix}^T,$$
$$\boldsymbol{W}_a^{Lat} = \begin{bmatrix} 0.0120 & -0.9219 & -2.4250 & -0.9458 & -1.1173 \end{bmatrix}^T.$$

The evolution of the closed-loop poles during the learning process is depicted in Figure 2. The figure shows the open-loop poles with a o; the final closed-loop poles, resulting from the model-free control, with a $\times$; and the closed-loop poles during the learning process with $\bullet$. The pole search space covered both stable and unstable regions before eventually converging to asymptotically stable poles once the actor and critic weights are settled to their final values. The evolution of the adaptive weights of the actor and critic neural networks is demonstrated in Figure 3. After an initial fluctuation period, all the weights converged to their steady-state values, as proven in Theorem 1. This is also illustrated in Figure 4, which demonstrates the dynamics and control signals of the longitudinal and lateral subsystems. It took about (15 s) and (20 s) for the longitudinal and lateral states to decay to zero respectively. It is worth noticing that the controller started with an aggressive approach in an attempt to stabilize the system as fast as possible. This is clear from the control signals which reached their saturation values of $\pm\pi/3$ rad three times in the first (5 s) and (10 s) for the longitudinal and lateral systems respectively.

Table 2 Open- and closed-loop poles of the decoupled systems

The longitudinal system:	
Open-loop poles	$0.9801\, e^{\pm 0.0219}$ $1.0009\, e^{\pm 0.0116}$
Closed-loop poles	$0.8771,\ 0.8884$ $0.9975\, e^{\pm 0.0123}$
The lateral system:	
Open-loop poles	$0.7978,\ 0.9949$ $0.9973\, e^{\pm 0.0088}$ 1.0000
Closed-loop poles	$0.7825,\ 0.9727$ 0.9960 $0.9969\, e^{\pm 0.0082}$

7.2 Case Study 2

In this scenario, the robustness of the proposed approach is tested against unmodeled aerodynamics and uncertainties. Since, the aerodynamic models of the flexible wing aircraft are unknown, artificial nonlinearities are simulated by intentionally adding a degree of randomness (for each entry of A, B, and Z_k) around the nominal dynamical parameters (i.e., the trim flight condition). The different time-dependent white-noise disturbances and uncertainties are computed at every time step (i.e., evaluation step) from a normal distribution $\mathcal{N}(0,\,1)$. The uncertainty levels in the aerodynamics and the states are scaled up to $\pm50\%$ and $\pm20\%$ of their respective nominal values. The noise in the states is very useful in a real-world scenario as it reflects the typical noise in the feedback data measurements introduced by the sensors.

This simulation scenario employs a combined control scheme, where the overall optimal control signal is divided into parts; First, the Riccati approach is allowed to find the optimal control signal using the nominal aerodynamic parameters (i.e., nominal undisturbed aerodynamics at the trim speed). Second, the learning system is integrated to support the optimal online control decision for uncertainties. The Riccati solution assumes a perfect knowledge of the system's model. Hence, in this case, the adaptive model-free controller can be seen a regulatory controller that compensates for the

irregularities that are missed by the Riccati control signal. Hence, the resulting systems will have following form

$$\boldsymbol{Z}_{k+1} = (\boldsymbol{A} + \boldsymbol{\Delta A}_k)(\boldsymbol{Z}_k + \boldsymbol{\Delta Z}_k) + \left(\boldsymbol{B}\, \boldsymbol{u}_k^R + \Delta \boldsymbol{B}_k\, \boldsymbol{u}_k^{MF}\right),$$

where $\boldsymbol{\Delta Z}_k$, $\boldsymbol{\Delta A}_k$, and $\Delta \boldsymbol{B}_k$ are the induced randomnesses in each entry of the states $\boldsymbol{Z}_k$, system matrix $\boldsymbol{A}$, and control input matrix $\boldsymbol{B}$ respectively. u_k^R is the control input evaluated by the Riccati control solution, while u_k^{MF} is the control signal generated by the model-free controller.

Later on, the control law obtained by the model-free learning process (i.e., u_k^{MF}) is compared with other situations as if it is applied solely to the system (i.e., without the supporting Riccati regulation part), which simply means coming back to the first scenario, where the control signal is decided completely by the learning process.

The discrete-time state space matrices of the decoupled longitudinal and lateral frames at last evaluation instance k are given by

$$\boldsymbol{A}_k^{Lon} = \begin{bmatrix} 0.9983 & 0.0055 & 0.0014 & -0.0907 \\ -0.0145 & 0.9871 & 0.1127 & 0.0108 \\ 0.0019 & -0.0034 & 0.9864 & -0.0001 \\ 0 & 0 & 0.0091 & 1.0000 \end{bmatrix},$$

$$\boldsymbol{A}_k^{Lat} = \begin{bmatrix} 0.9971 & -0.0028 & -0.0987 & 0.1106 & -0.0126 \\ -0.0085 & 0.8307 & 0.0734 & -0.0005 & 0.0001 \\ 0.0013 & 0.0338 & 0.9860 & 0.0001 & 0 \\ 0 & 0.0076 & 0.0003 & 1.0000 & 0 \\ 0 & 0.0002 & 0.0092 & 0 & 1.0000 \end{bmatrix},$$

$$\boldsymbol{B}_k^{Lon} = \begin{bmatrix} 0 \\ 0.0031 \\ 0.0544 \\ 0.0003 \end{bmatrix}, \quad \boldsymbol{B}_k^{Lat} = \begin{bmatrix} -0.0002 \\ 0.0427 \\ 0.0030 \\ 0.0002 \\ 0 \end{bmatrix}.$$

The final optimal control gains for the longitudinal and lateral directions, recorded by the learning process at last evaluation instance k, are given as follows:

$$\boldsymbol{W}_a^{Lon} = \begin{bmatrix} 0.1961 & -0.0554 & -2.4252 & -3.0799 \end{bmatrix}^T,$$
$$\boldsymbol{W}_a^{Lat} = \begin{bmatrix} -0.0038 & -0.6563 & -2.3076 & -1.9291 & -2.0037 \end{bmatrix}^T.$$

In this scenario, the different eigenvalues which are evaluated using the disturbed system ($A + \Delta A_k$ and $B + \Delta B_k$) at instance k are represented graphically as follows; The open-loop poles of the disturbed system are given the o marks, while the notations o refer to the final closed-loop poles of the disturbed system employing Riccati control gains calculated using ($A + \Delta A_k$ and $B + \Delta B_k$). The closed-loop poles (represented by $\times$) are the result of employing the combined model-free approach and the Ricatti control solution using nominal system (A and B). The notations $\times$ indicate the closed-loop poles using the model-free control gains only. The notations • and • represent the spectrum of the closed-loop poles using the learning process and Riccati approach separately. Finally, the symbols • denote the spectrum of the closed-loop poles of the combined learning-Riccati process.

As listed in Tables 3 & 4, the open-loop poles of the disturbed longitudinal and lateral subsystems at last evaluation step k are unstable. Tables 3 & 4 and Figures 5 & 6 show the closed-loop poles of the longitudinal and lateral dynamics. Figure 6 highlights the ability of the online stand-alone learning process to explore more stable areas, which resulted in faster dominant modes compared to those achieved using Riccati or combined Riccati-learning processes.

It is worth noticing how the learning instances (poles) in Figure 5 are denser compared to Figure 2. This is due to the longer time taken by the actor and critic weights to converge to their steady-state values, as can be witnessed from Figure 7. The final values of the closed-loop poles for the longitudinal and lateral systems listed in Tables 3 & 4 and hence shown in Figure 5 emphasized the superiority of the proposed learning approach even when it is working

Table 3 Open- and closed-loop poles of the longitudinal disturbed system at the last evaluation step k

The longitudinal system:	
Open-loop poles of the disturbed system $(\boldsymbol{A} + \Delta \boldsymbol{A}_k^{Lon})$ & $(\boldsymbol{B} + \Delta \boldsymbol{B}_k^{Lon})$	$0.9859\, e^{\pm 0.0200}$ $1.0002\, e^{\pm 0.0102}$
Closed-loop poles using Riccati approach; The control gains were calculated using system: $(\boldsymbol{A}$ & $\boldsymbol{B})$	$0.9634\, e^{\pm 0.0090}$ $0.9964\, e^{\pm 0.0089}$
Closed-loop poles using Riccati approach; The control gains were calculated using system: $(\boldsymbol{A} + \Delta \boldsymbol{A}_k^{Lon})$ & $(\boldsymbol{B} + \Delta \boldsymbol{B}_k^{Lon})$	$0.9623\, e^{\pm 0.0130}$ $0.9967\, e^{\pm 0.0093}$
Closed-loop poles using the combined Riccati-model-free approach	$0.9781\, e^{\pm 0.0178}$ $0.9964\, e^{\pm 0.0065}$
Closed-loop poles using the model-free approach	$0.8713,\ 0.9751$ $0.9963\, e^{\pm 0.0100}$

Table 4 Open- and closed-loop poles of the lateral disturbed system at the last evaluation step k

The lateral system:	
Open-loop poles of the disturbed system $(\boldsymbol{A} + \Delta \boldsymbol{A}_k^{Lat})$ & $(\boldsymbol{B} + \Delta \boldsymbol{B}_k^{Lat})$	$0.8167,\ 0.9947,\ 1$ $1.0012\, e^{\pm 0.0037}$
Closed-loop poles using Riccati approach; The control gains were calculated using system: $(\boldsymbol{A}$ & $\boldsymbol{B})$	0.8165 $0.9948\, e^{\pm 0.0046}$ $0.9976\, e^{\pm 0.0085}$
Closed-loop poles using Riccati approach; The control gains were calculated using system: $(\boldsymbol{A} + \Delta \boldsymbol{A}_k^{Lat})$ & $(\boldsymbol{B} + \Delta \boldsymbol{B}_k^{Lat})$	0.8156 $0.9914\, e^{\pm 0.0057}$ $0.9976\, e^{\pm 0.0055}$
Closed-loop poles using the combined Riccati-model-free approach	0.8152 $0.9947\, e^{\pm 0.0047}$ $0.9974\, e^{\pm 0.0082}$
Closed-loop poles using the model-free approach	$0.8105,\ 0.9824$ 0.9923 $0.9967\, e^{\pm 0.0070}$

in stand-alone mode. The dominant modes of the stand-alone learning controller are asymptotically stabilized and became much faster compared to all other situations. The spectrum of the closed-loop poles evolution calculated by stand-alone Riccati approach, combined Riccati-learning controller, and stand-alone learning control system, exhibited by Figure 6, assure that whenever the learning process exists, it widens the stability region and allows the system to take more quicker decisions, even in highly and continuously disturbed aerodynamic environment.

The intelligent adaptive controller stabilized the hang glider by again forcing the unstable poles to slide to an asymptotically stable region. Figure 8 demonstrates the asymptotic stability behavior of the decoupled dynamical systems. The longitudinal and lateral control input signals and their associated dynamics took almost (3 s) and (6 s) to practically vanish. Despite the relatively significant effort the controller had to apply to stabilize the lateral dynamics, the stabilization of the longitudinal dynamics were much easier to achieve. This is clear from the span of both control signals. This fast convergence and relative control comfort is due to the synergistic integration of the Riccati control signal, stemming from the nominal model, and the model-free control policy compensating for the imperfections in that model. Such behavior is tested by disabling the Riccati control component. The results are revealed in Figure 9. This time, it took the longitudinal and lateral dynamics more effort and a longer time to stabilize (about (10 s) and (15 s), respectively). It demonstrates that despite the time-variant noise in the system dynamics, the intelligent model-free controller is able to control the aircraft even without the help of a Riccati controller.

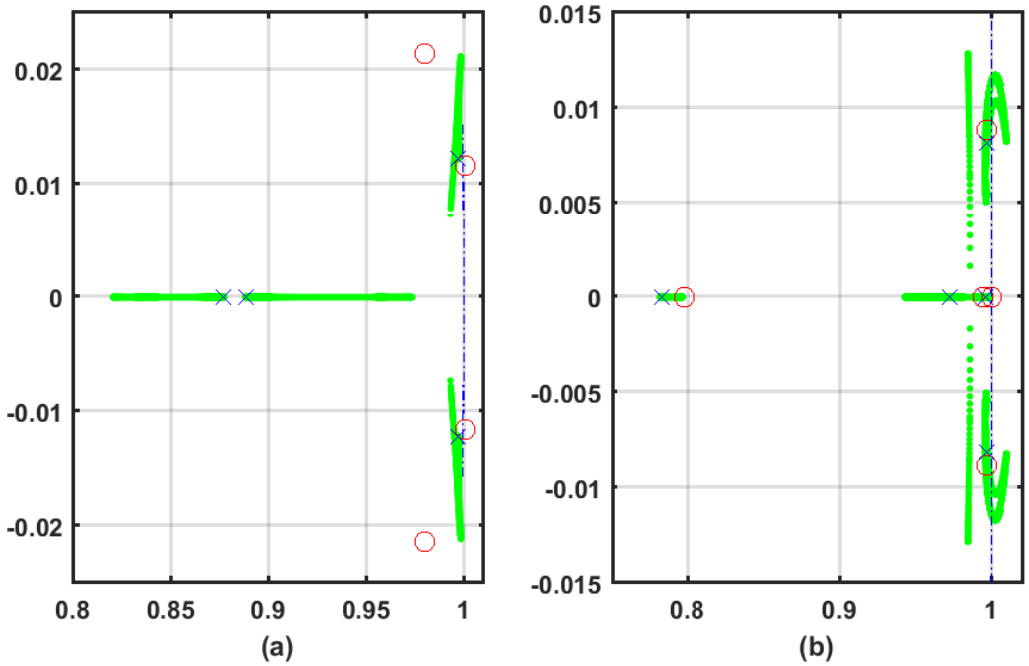

Fig. 2: Evolution of the closed-loop poles during learning: (a) longitudinal and (b) lateral dynamics.

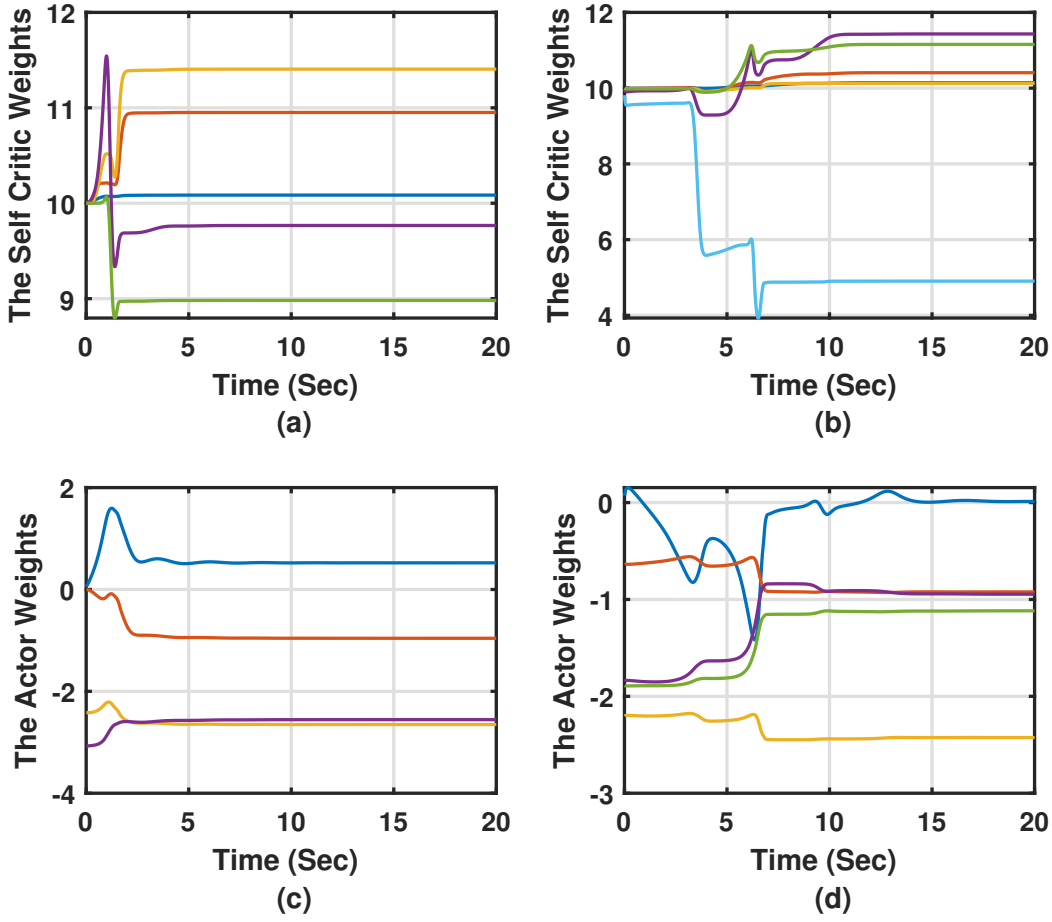

Fig. 3: Actor-critic weights: (a),(c) longitudinal and (b),(d) lateral dynamics.

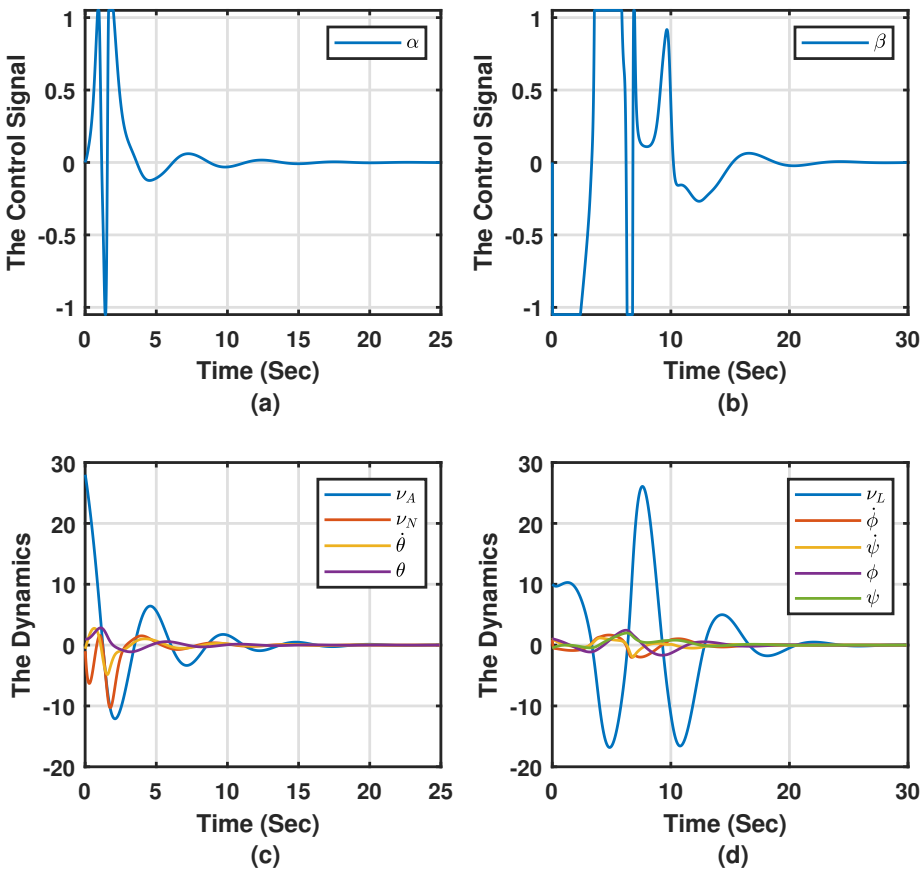

Fig. 4: The control signals and dynamics of (a),(c): longitudinal; (b),(d): lateral. All the signals are in their respective SI units.

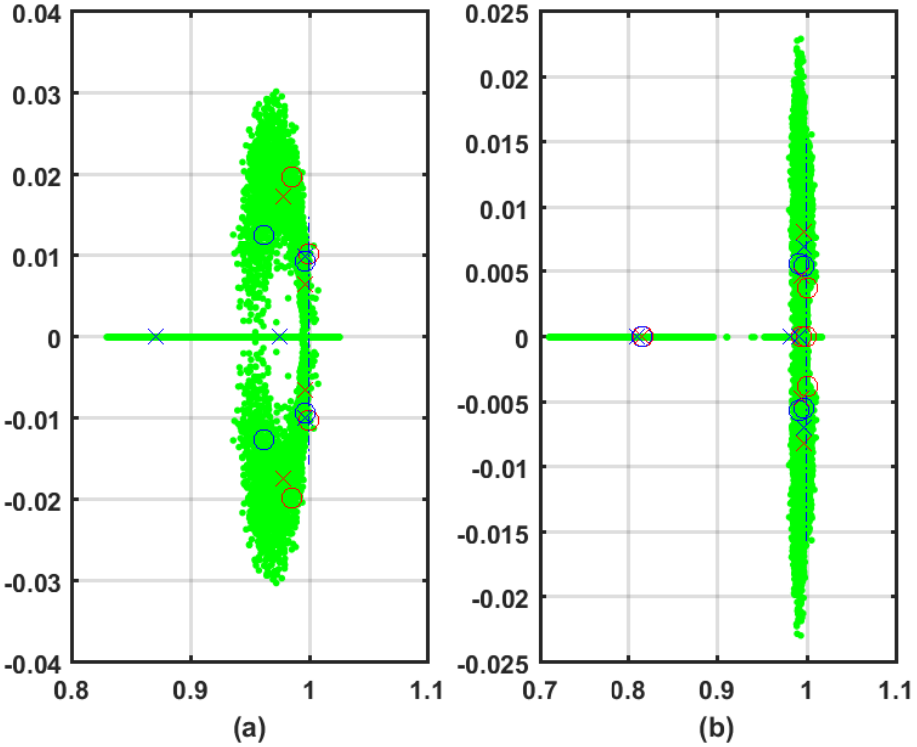

Fig. 5: Evolution of the closed-loop poles during learning of the disturbed system: (a) longitudinal and (b) lateral dynamics.

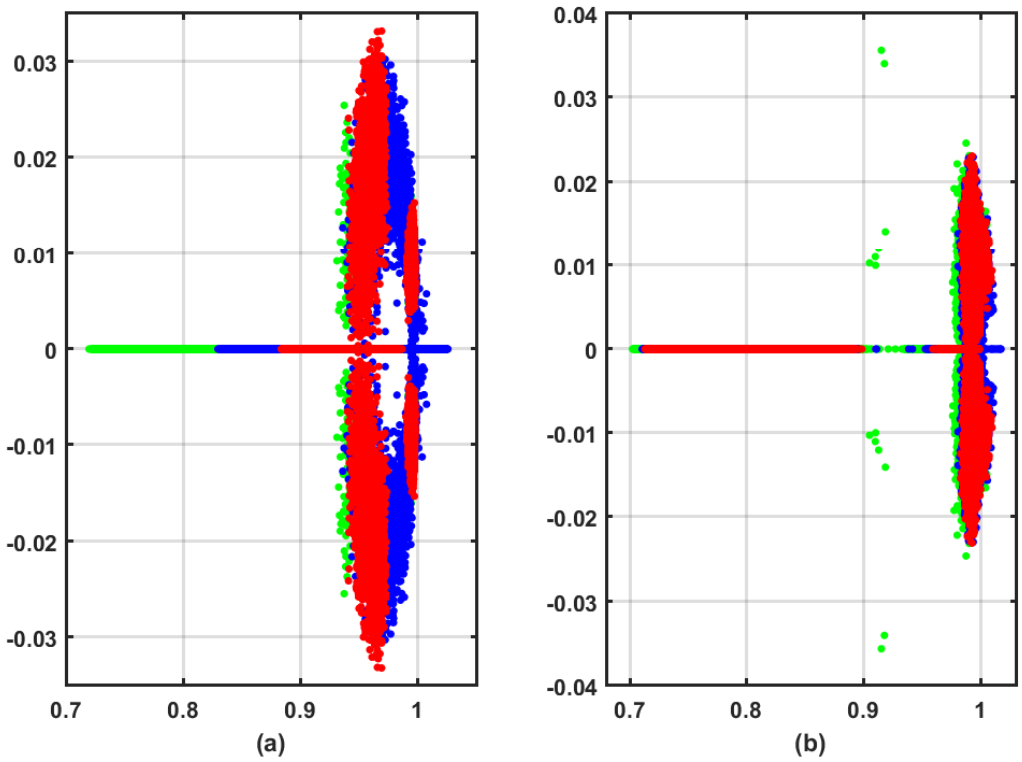

Fig. 6: Evolution of the closed-loop poles during learning of the disturbed system using the stand-alone adaptive learning scheme, Riccati approach, combined Riccati-learning mechanism: (a) longitudinal and (b) lateral dynamics.

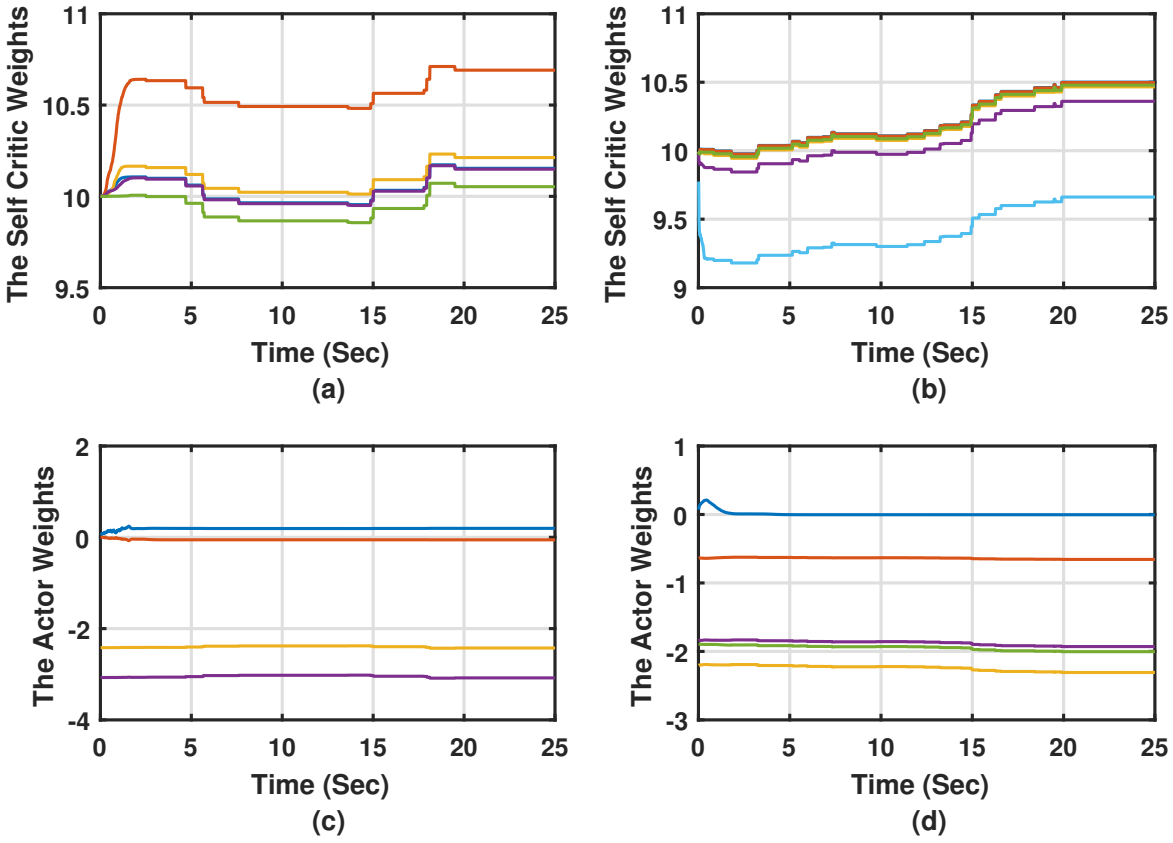

Fig. 7: Actor-critic weights for the disturbed system: (a),(c) longitudinal and (b),(d) lateral dynamics.

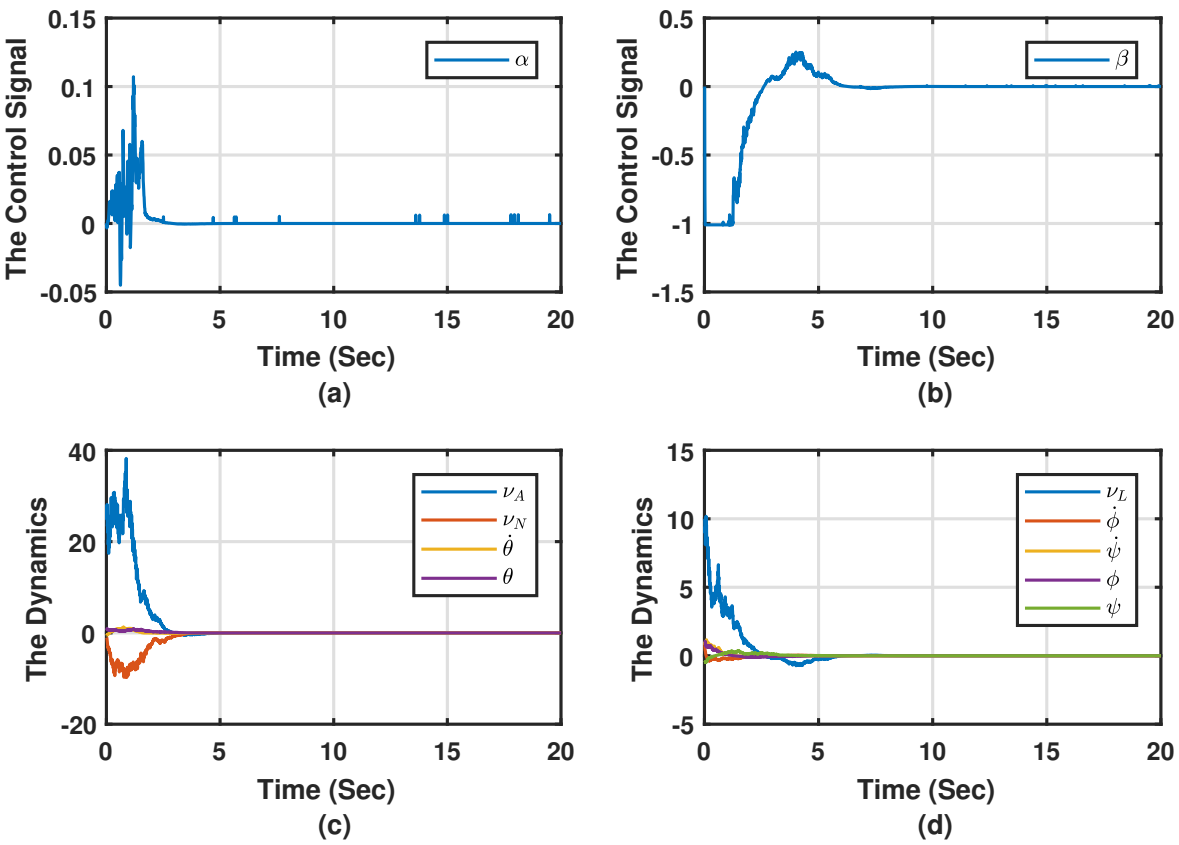

Fig. 8: The control signal and dynamics of the disturbed system. (a),(c): longitudinal; (b),(d): lateral. All the signals are in their respective SI units.

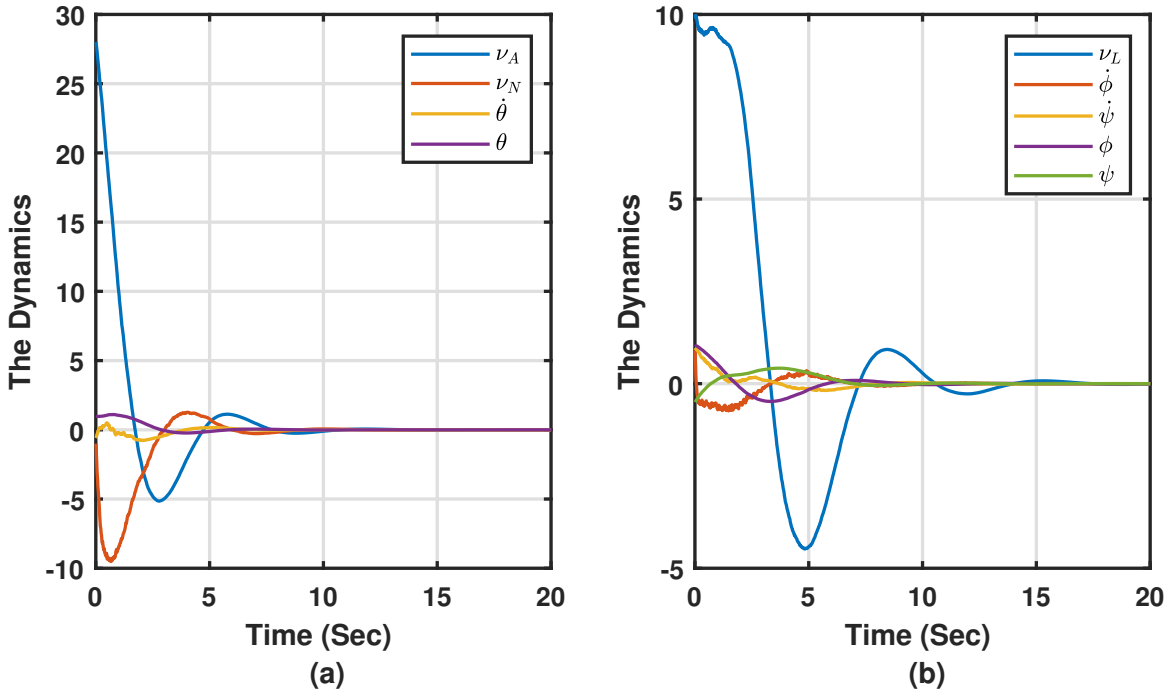

Fig. 9: The dynamics of the disturbed system (without Riccati control component). (a): longitudinal; (b): lateral. All the signals are in their respective SI units.

8 Conclusion

The automatic control of a flexible wing aircraft was addressed. It is challenging due to the lack of an accurate dynamical model faithfully describing the system dynamics. This work introduced an online adaptive learning controller to accomplish this task. It is based on machine learning and optimal control methods to generate model-free control decisions. Unlike traditional classical approaches, the proposed controller does not use or require any information about the system's dynamics. Instead, it finds optimal control approximations in real-time using few feedback signals that can be measured through sensors mounted aboard of the hang glider. The solution is implemented by employing a pair of actor-critic neural networks where the weights are adapted online through a gradient descent technique. The feedback loop is proven to be asymptotically stable while the reinforcement learning algorithm is guaranteed to converge. A Riccati solution framework is shown to be equivalent to solving the underlying model-free Bellman optimality equation. The validation results show the superior performance of the developed adaptive learning controller in terms of the stability and convergence even when the system is subject to significant time-variant disturbances.

9 References

1 Cook, M.V., Spottiswoode, M.: 'Modelling the flight dynamics of the hang glider', *The Aeronautical Journal*, 2005, **109**, (1102), pp. I–XX
2 Kilkenny, E.A. 'An evaluation of a mobile aerodynamic test facility for hang glider wings'. (College of Aeronautics, Cranfield Institute of Technology, 1983. 8330
3 Kilkenny, E. 'Full scale wind tunnel tests on hang glider pilots'. (Cranfield Institute of Technology, College of Aeronautics, Department of Aerodynamics, 1984.
4 Cook, M.V., Kilkenny, E.A. 'An experimental investigation of the aerodynamics of the hang glider'. In: Proceedings of the International Conference on Aerodynamics. (, 1986. pp. 1–10
5 Kilkenny, E.A. 'An experimental study of the longitudinal aerodynamic and static stability characteristics of hang gliders' [phdthesis]. Cranfield University, 1986
6 Blake, D. 'Modelling The Aerodynamics, Stability and Control of The Hang Glider' [mathesis]. College of Aeronautics, Cranfield Institute of Technology, 1991
7 Cook, M.V.: 'The theory of the longitudinal static stability of the hang-glider', *The Aeronautical Journal*, 1994, **98**, (978), pp. 292–304
8 Ochi, Y. 'Modeling of flight dynamics and pilot's handling of a hang glider'. In: AIAA Modeling and Simulation Technologies Conference. (American Institute of Aeronautics and Astronautics, 2017. pp. 1758–1776
9 Sweeting, J. 'An Experimental Investigation of Hang Glider Stability' [mathesis]. College of Aeronautics, Cranfield University, 1981
10 Cook, M.: 'Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control'. 3rd ed. Aerospace Engineering. (Butterworth-Heinemann, 2012)
11 Kroo, I. 'Aerodynamics, Aeroelasticity and Stability of Hang Gliders' [phdthesis]. Stanford University, 1983
12 Powton, J. 'A Theoretical Study of the Non-linear Aerodynamic Pitching Moment Characteristics of the Hang Glider and its Influence on Stability and Control' [mathesis]. College of Aeronautics, Cranfield Institute of Technology, 1995
13 De.Matteis, G.: 'Response of hang gliders to control', *The Aeronautical Journal*, 1990, **94**, (938), pp. 289–294
14 de Matteis, G.: 'Dynamics of hang gliders', *Journal of Guidance Control and Dynamics*, 1991, **14**, (6), pp. 1145–1152
15 Spottiswoode, M. 'A Theoretical Study of the Lateral-directional Dynamics, Stability and Control of the Hang Glider' [mathesis]. College of Aeronautics, Cranfield Institute of Technology, 2001
16 Ochi, Y. 'Modeling of the longitudinal dynamics of a hang glider'. In: AIAA Modeling and Simulation Technologies Conference. (American Institute of Aeronautics and Astronautics, 2015. pp. 1591–1608
17 Rollins, R. 'Study of Experimental Data to Assess the Longitudinal Stability and Control of the Hang Glider' [mathesis]. College of Aeronautics, Cranfield University, 2000
18 Aguilar.Ibañez, C.: 'Stabilization of the pvtol aircraft based on a sliding mode and a saturation function', *International Journal of Robust and Nonlinear Control*, 2017, **27**, (5), pp. 843–859
19 d. J..Rubio, J., Pieper, J., Meda-Campaña, J.A., Aguilar, A., Rangel, V.I., Gutierrez, G.J.: 'Modelling and regulation of two mechanical systems', *IET Science, Measurement Technology*, 2018, **12**, (5), pp. 657–665
20 Rubio, J.: 'Robust feedback linearization for nonlinear processes control', *ISA Transactions*, 2018, **74**, pp. 155 – 164
21 Howard, R.A.: 'Dynamic Programming and Markov Processes'. Four volumes. (Cambridge. MA: MIT Press, 1960)
22 Werbos, P. 13. In: White, D.A., Sorge, D.A., editors. 'Approximate dynamic programming for real-time control and neural modeling'. (New York: Van Nostrand Reinhold: Van Nostrand Reinhold, New York, 1992. pp. 493–525
23 Bertsekas, D.P., Tsitsiklis, J.N. 'Neuro-dynamic programming: An overview'. In: Proceedings of the IEEE Conference on Decision and Control. vol. 1. (, 1995. pp. 560–564
24 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, 1990)
25 Sutton, R.S., Barto, A.G.: 'Reinforcement Learning: An Introduction'. (Massachusetts USA: MIT Press, 1998)
26 Abouheaf, M., Gueaieb, W. 'Multi-agent reinforcement learning approach based on reduced value function approximations'. In: IEEE International Symposium on Robotics and Intelligent Sensors (IRIS). (, 2017. pp. 111–116
27 Abouheaf, M., Lewis, F. 'Approximate dynamic programming solutions of multi-agent graphical games using actor-critic network structures'. In: International Joint Conference on Neural Networks (IJCNN). (, 2013. pp. 1–8
28 Abouheaf, M., Lewis, F. Chapter 1. In: Liu, D., Alippi, C., Zhao, D., Zhang, H., editors. 'Dynamic graphical games: Online adaptive learning solutions using approximate dynamic programming'. (World Scientific, 2014. pp. 1–48
29 Lewis, F., Vrabie, D., Syrmos, V.: 'Optimal Control'. 3rd ed. (New York, USA: John Wiley, 2012)
30 Bellman, R.: 'Dynamic Programming'. (Princeton University Press, 1957)
31 Bryson, A.: 'Optimal control-1950 to 1985', *IEEE Control Systems*, 1996, **16**, (3), pp. 26–33
32 Sen, S., Weiss, G. 6. In: Weiss, G., editor. 'Learning in multi-agent systems'. (Cambridge, MA, USA: MIT Press, 1999. pp. 259–298
33 Widrow, B., Gupta, N.K., Maitra, S.: 'Punish/reward: Learning with a critic in adaptive threshold systems', *IEEE Transactions on Systems, Man, and Cybernetics*, 1973, **SMC-3**, (5), pp. 455–465
34 Werbos, P. 'Neural networks for control and system identification'. In: Proceedings of the 28th IEEE Conference on Decision and Control. vol. 1. (, 1989. pp. 260–265
35 Werbos, P. 'Beyond Regression: New Tools for Prediction and Analysis in the Behavior Sciences' [phdthesis]. Harvard University, 1974
36 Busoniu, L., Babuska, R., Schutter, B.D.: 'A comprehensive survey of multi-agent reinforcement learning', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2008, **38**, (2), pp. 156–172
37 Vrancx, P., Verbeeck, K., Nowe, A.: 'Decentralized learning in markov games', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2008, **38**, (4), pp. 976–981
38 Al.Tamimi, A., Lewis, F.L., Abu.Khalaf, M.: 'Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2008, **38**, (4), pp. 943–949
39 Vrabie, D., Pastravanu, O., Abu.Khalaf, M., Lewis, F.L.: 'Adaptive optimal control for continuous-time linear systems based on policy iteration', *Automatica*, 2009, **45**, (2), pp. 477–484