Online Adaptive Incremental Reinforcement Learning Flight Control for a CS-25 Class Aircraft

Heyer, Stefan; Kroezen, Dave; van Kampen, Erik-jan

DOI
10.2514/6.2020-1844

Publication date
2020

Document Version
Final published version

Published in
AIAA Scitech 2020 Forum

AIAA SciTech Forum
6-10 January 2020, Orlando, FL
AIAA Scitech 2020 Forum

10.2514/6.2020-1844

Online Adaptive Incremental Reinforcement Learning Flight Control for a CS-25 Class Aircraft

S. Heyer *, D. Kroezen †, E. van Kampen ‡
Delft University of Technology, P.O. Box 5058, 2600GB Delft, The Netherlands

In recent years Adaptive Critic Designs (ACDs) have been applied to adaptive flight control of uncertain, nonlinear systems. However, these algorithms often rely on representative models as they require an offline training stage. Therefore, they have limited applicability to a system for which no accurate system model is available, nor readily identifiable. Inspired by recent work on Incremental Dual Heuristic Programming (IDHP), this paper derives and analyzes a Reinforcement Learning (RL) based framework for adaptive flight control of a CS-25 class fixed-wing aircraft. The proposed framework utilizes Artificial Neural Networks (ANNs) and includes an additional network structure to improve learning stability. The designed learning controller is implemented to control a high-fidelity, six-degree-of-freedom simulation of the Cessna 550 Citation II PH-LAB research aircraft. It is demonstrated that the proposed framework is able to learn a near-optimal control policy online without a priori knowledge of the system dynamics nor an offline training phase. Furthermore, it is able to generalize and operate the aircraft in not previously encountered flight regimes as well as identify and adapt to unforeseen changes to the aircraft's dynamics.

Nomenclature

$\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{a}$	=	state, reference state, and action vectors
r, g	=	reward and return
γ, τ, κ	=	discount, mixing, and forgetting factors
$f(\boldsymbol{s}, \boldsymbol{a}), r(\boldsymbol{s}^R, \boldsymbol{s})$	=	state-transition and reward functions
$\pi(\boldsymbol{s}, \boldsymbol{s}^R), \pi^*(\boldsymbol{s}, \boldsymbol{s}^R)$	=	policy and optimal policy
$\hat{\pi}(\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{w}^A)$	=	parametric approximation of the policy
$v(\boldsymbol{s}, \boldsymbol{s}^R)$	=	state-value function
$\lambda(\boldsymbol{s}, \boldsymbol{s}^R)$	=	state-value partial derivative w.r.t. the state vector
$\hat{\lambda}(\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{w}^C)$	=	parametric approximation of state-value partial derivative w.r.t. state vector
$\hat{\lambda}'(\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{w}^{C'})$	=	parametric approximation of state-value partial derivative w.r.t. state vector
$\boldsymbol{P}, \boldsymbol{Q}$	=	Boolean selection and symmetric weight matrices
$t, \Delta t, T$	=	time, sample time, and simulation time duration
$\boldsymbol{F}, \hat{\boldsymbol{F}}$	=	state matrix and state matrix estimate
$\boldsymbol{G}, \hat{\boldsymbol{G}}$	=	input matrix and input matrix estimate
$\boldsymbol{w}^A, \boldsymbol{w}^C, \boldsymbol{w}^{C'}$	=	actor, critic, and target critic parameter vectors
$\Delta\boldsymbol{s}, \Delta\boldsymbol{a}, \Delta\hat{\boldsymbol{s}}$	=	state and action vector increment and state vector increment prediction
$\boldsymbol{e}$	=	partial derivative of the Temporal Difference (TD) error w.r.t. the state vector
L^A, L^C	=	actor and critic losses
η^A, η^C	=	actor and critic learning rates
$\hat{\boldsymbol{\Theta}}, \boldsymbol{\Lambda}, \boldsymbol{X}$	=	Recursive Least Squares (RLS) parameter, covariance, and measurement matrices
$\boldsymbol{\epsilon}$	=	innovation vector
$\delta_e, \delta_a, \delta_r$	=	elevator, aileron, and rudder deflections
$p, q, r, \alpha, \beta, \phi, \theta$	=	roll rate, pitch rate, yaw rate, angle of attack, sideslip angle, roll angle, and pitch angle
$p^R, q^R, \beta^R, \phi^R, H^R, \gamma^R$	=	roll rate, pitch rate, sideslip angle, roll angle, altitude, and flight path angle reference values
V_{tas}, H, γ, n	=	true airspeed, altitude, flight path angle, and load factor

*M.Sc., Faculty of Aerospace Engineering, Control and Simulation Division, Delft University of Technology
†M.Sc., Faculty of Aerospace Engineering, Control and Simulation Division, Delft University of Technology
‡Assistant Professor, Faculty of Aerospace Engineering, Control and Simulation Division, Delft University of Technology, AIAA Member

I. Introduction

In recent years, the aerospace domain has experienced an unprecedented increase in interest in autonomous operations. Autonomous systems, especially when operated in complex urban environments, need to be able to adapt to sudden, unexpected changes in the environment [1] and to changes in their dynamics, also referred to as fault tolerance. Additionally to the requirement of online operation, model-independence is important as for many aerospace applications, no accurate system model is available, nor readily identifiable [2, 3].

Current flight control systems for passenger aircraft predominantly rely on classical control techniques, which make use of multiple linear controllers. Gain scheduling methods are then applied to switch between the numerous linear controllers, each designed for a specific operating point [4]. These gains are determined offline, in advance, by means of a model of the system. Their non-adaptive nature makes them unsuitable for autonomous systems.

Since the 1960s, adaptive control has been an active research field, developing control strategies that adapt to changing system conditions online. Many, such as Nonlinear Dynamic Inversion (NDI) [5, 6] and Backstepping (BS) [7], with the focus on dealing with system nonlinearities. Although successfully applied [8–12], these methods strongly rely on an accurate model of the system dynamics. Recently developed incremental control methods, such as Incremental Nonlinear Dynamic Inversion (INDI), Incremental Backstepping (IBS) and incremental adaptive sliding mode control have decreased the model-dependence, in exchange for the need of high sample rate measurements [13–18]. Furthermore, major steps have been made in INDI through the first stability and robustness analysis in the presence of external disturbances [19] and a first successful demonstration on an CS-25 certified aircraft [20].

Originally inspired by the idea of replicating biological learning mechanisms [21], Reinforcement Learning (RL) is a field of Machine Learning (ML) that is best characterized by learning from interaction [22]. Ever since RL has been studied and applied to the field of adaptive and optimal control. Traditional RL methods were formulated for discrete state and actions spaces, which were sufficiently small such that approximate value functions could be represented as tables. Continuous and high-dimensional spaces prevalent in control applications would lead to an exponential growth in computational complexity known as the curse of dimensionality [23]. The curse of dimensionality was mitigated with the introduction of function approximators, such as Artificial Neural Networks (ANNs), characterizing the field of Approximate Dynamic Programming (ADP) [24].

With Adaptive Critic Designs (ACDs), a class of ADP methods, several applications were successfully explored in the 2000s, including adaptive flight control for a missile system [25], business jet [26], helicopter [27] and military aircraft [28]. However, these methods often need an extra structure to approximate the system dynamics. Furthermore, when applied online, a preceding offline learning phase is required, mainly due to non-trivial identification of the system dynamics. The offline identification phase itself requires a representative simulation model. In [29–31] novel frameworks, named Incremental Heuristic Dynamic Programming (IHDP) and Incremental Dual Heuristic Programming (IDHP) have been proposed to improve online adaptability and most importantly, eliminate the current need of an offline learning phase, by identifying an incremental model of the system in real-time. However, these novel frameworks have yet to be applied to and validated on complex, high-dimensional aerospace models and real systems.

The main contribution of this paper is to present the design and analysis of a RL based adaptive flight controller for a CS-25 class fixed-wing research aircraft, that can learn a near-optimal control policy online without a priori knowledge of the system dynamics nor an offline training phase. In this work, a novel learning algorithm, based on IDHP, is proposed and applied to a six-degree-of-freedom, high-fidelity, nonlinear simulation model of a Cessna 550 Citation II PH-LAB research aircraft. Through simulation, it is shown that the designed controller, is able to learn to control the aircraft, without a priori system knowledge, during a short online training phase. However, learning instability inherent to IDHP can lead to failures, which are unacceptable during operation in real systems. Therefore the proposed algorithm utilizes a separate target critic network to improve learning stability during Temporal Difference (TD) backups. Furthermore, it is shown that the controller is able to generalize and operate the aircraft in not previously encountered flight regimes and identify and adapt to unforeseen changes to the aircraft's dynamics.

The remainder of this paper is structured as follows. In Section II the control problem is formulated and the proposed learning framework is derived. Section III introduces the high-fidelity simulation model and presents the implementation of the learning framework into the controller design. Subsequently, in Section IV, the controller is tested and evaluated in three distinct cases: (1) online training, (2) online operation and (3) online adaption. Lastly, this paper is concluded in Section V.

Fig. 1 CS-25 class Cessna 550 Citation II PH-LAB research aircraft operated by the Delft University of Technology.

II. Foundations

This section starts by formulating the flight control task as a RL problem. Subsequently, the proposed learning framework is introduced, followed by its update rules and training strategy.

A. Problem Formulation

In the framework of RL the state-transition and reward function are commonly defined as processes of the environment, whose mechanics are hidden from the agent [22]. This paper regards them as separate entities, where the reward function is not a hidden process of the environment, but a designed functional. The state-transition function $f(\boldsymbol{s}, \boldsymbol{a}) \in \mathbb{R}^{m\times 1}$ characterizes the discrete-time, deterministic, nonlinear plant as in Eq. (1) with the state vector $\boldsymbol{s} \in \mathbb{R}^{m\times 1}$, the action vector $\boldsymbol{a} \in \mathbb{R}^{n\times 1}$ and the assumption of synchronous, high-frequency sampling.

The flight control problem is a variable set-point tracking task with the reference state vector $\boldsymbol{s}^R \in \mathbb{R}^{p\times 1}$. Therefore, the goal is to learn a deterministic policy $\pi(\boldsymbol{s}, \boldsymbol{s}^R) \in \mathbb{R}^{n\times 1}$ that maximizes the scalar return g, defined by Eq. (2) and Eq. (3), respectively. The return, defined by the state-value function $v(\boldsymbol{s}, \boldsymbol{s}^R)$, represents a discounted sum of future scalar rewards r, where the scalar discount factor $\gamma \in [0, 1]$ is a property of the agent. Equation (4) is the reward function. It is defined as the negative, weighted, squared state tracking error, with the Boolean selection matrix $\boldsymbol{P} \in \mathbb{R}^{p\times m}$ and the symmetric weight matrix $\boldsymbol{Q} \in \mathbb{R}^{p\times p}$. Furthermore, the reward function is differentiable as required for the update operations of Dual Heuristic Programming (DHP) frameworks. Its partial derivative with respect to the state vector $\frac{\partial r}{\partial \boldsymbol{s}} \in \mathbb{R}^{1\times m}$ is defined in Eq. (5).

$$\boldsymbol{s}_{t+1} = f(\boldsymbol{s}_t, \boldsymbol{a}_t) \tag{1}$$

$$\boldsymbol{a}_t = \pi(\boldsymbol{s}_t, \boldsymbol{s}_t^R) \tag{2}$$

$$g_t = v(\boldsymbol{s}_t, \boldsymbol{s}_t^R) = \sum_{k=0}^{\infty} \gamma^k r_{t+1+k} \tag{3}$$

$$r_{t+1} = r(\boldsymbol{s}_t^R, \boldsymbol{s}_{t+1}) = -\left[\boldsymbol{P}\boldsymbol{s}_{t+1} - \boldsymbol{s}_t^R\right]^T \boldsymbol{Q} \left[\boldsymbol{P}\boldsymbol{s}_{t+1} - \boldsymbol{s}_t^R\right] \tag{4}$$

$$\frac{\partial r_{t+1}}{\partial \boldsymbol{s}_{t+1}} = \frac{\partial r(\boldsymbol{s}_t^R, \boldsymbol{s}_{t+1})}{\partial \boldsymbol{s}_{t+1}} = -2\left[\boldsymbol{P}\boldsymbol{s}_{t+1} - \boldsymbol{s}_t^R\right]^T \boldsymbol{Q}\boldsymbol{P} \tag{5}$$

B. Learning Framework

A schematic of the feed-forward signal flow of the proposed learning framework is presented in Fig. 2. The schematic includes both processes and parametric structures. The proposed learning framework is derived from the IDHP framework as presented in [30], with its three parametric structures: the actor $\hat{\pi}(\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{w}^A) \in \mathbb{R}^{n\times 1}$, the critic $\hat{\lambda}(\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{w}^C) \in \mathbb{R}^{1\times m}$, and the incremental model of the plant. Whereas the actor approximates the control policy, the critic approximates the partial derivative of the state-value function with respect to the state, with the parameter vectors $\boldsymbol{w}^A$ and $\boldsymbol{w}^C$, respectively.

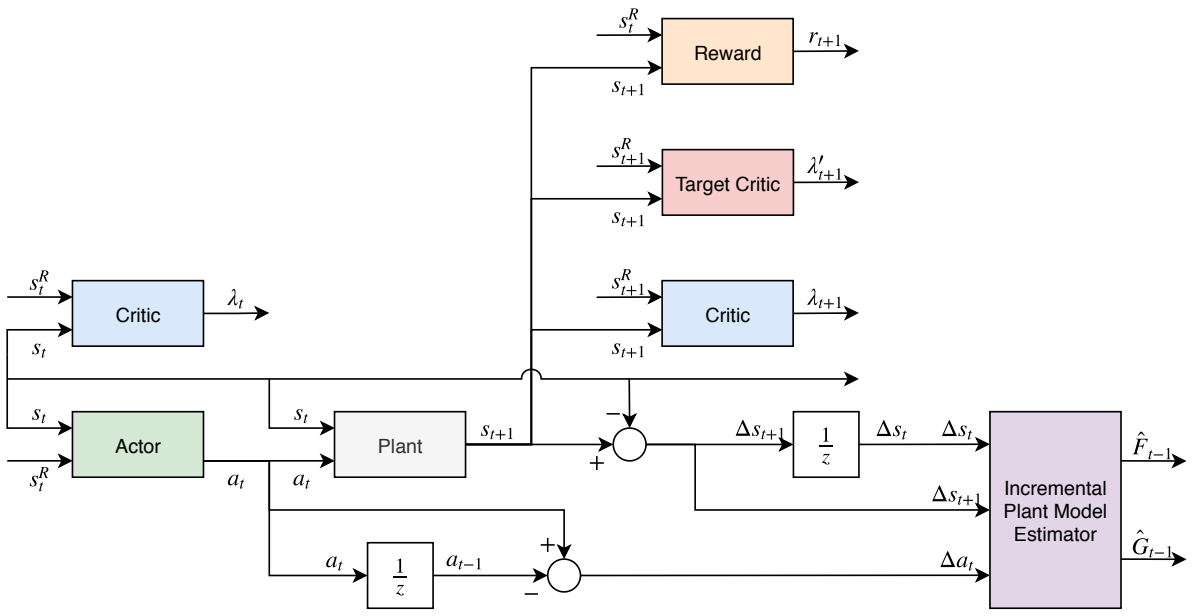

Fig. 2 Schematic of the feed-forward signal flow of the learning framework, where the different parametric structures and processes are illustrated by distinct colors.

1. Target Network

The proposed learning framework is an on-policy approach and therefore does not appertain to the deadly triad*. Nevertheless, any form of learning instability is undesirable during online operation. To improve learning stability a separate target critic $\hat{\lambda}'(\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{w}^{C'}) \in \mathbb{R}^{1\times m}$, inspired by [32, 33], is proposed for the TD backups. The proposed approach slows down learning as the propagation of the state-value derivatives is delayed, but this is outweighed by the improved learning stability. Analogous to the actor and critic, the target critic has a parameter vector $\boldsymbol{w}^{C'}$.

2. Incremental Model

By means of a first-order Taylor series expansion of the discrete-time, nonlinear plant at the operating point $[\boldsymbol{s}_0, \boldsymbol{a}_0]$ a linear approximation of the system is established as in Eq. (6), with the partial derivatives of the state-transition function, $\boldsymbol{F}(\boldsymbol{s}_0, \boldsymbol{a}_0) = \frac{\partial f(\boldsymbol{s}_0, \boldsymbol{a}_0)}{\partial \boldsymbol{s}_0} \in \mathbb{R}^{m\times m}$ and $\boldsymbol{G}(\boldsymbol{s}_0, \boldsymbol{a}_0) = \frac{\partial f(\boldsymbol{s}_0, \boldsymbol{a}_0)}{\partial \boldsymbol{a}_0} \in \mathbb{R}^{m\times n}$, also referred to as the state matrix and input matrix, respectively. By choosing the operating point $[\boldsymbol{x}_0, \boldsymbol{a}_0] = [\boldsymbol{x}_{t-1}, \boldsymbol{a}_{t-1}]$ and rearranging Eq. (6), the incremental form of the discrete-time, linear approximation of the system is obtained, as in Eq. (7), with $\Delta\boldsymbol{s}_{t+1} = \boldsymbol{s}_{t+1} - \boldsymbol{s}_t$, $\Delta\boldsymbol{a}_{t+1} = \boldsymbol{a}_{t+1} - \boldsymbol{a}_t$, $\boldsymbol{F}_{t-1} = \boldsymbol{F}(\boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1})$, and $\boldsymbol{G}_{t-1} = \boldsymbol{G}(\boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1})$.

$$\boldsymbol{s}_{t+1} \approx f(\boldsymbol{s}_0, \boldsymbol{a}_0) + \boldsymbol{F}(\boldsymbol{s}_0, \boldsymbol{a}_0)[\boldsymbol{s}_t - \boldsymbol{s}_0] + \boldsymbol{G}(\boldsymbol{s}_0, \boldsymbol{a}_0)[\boldsymbol{a}_t - \boldsymbol{a}_0] \tag{6}$$

$$\Delta\boldsymbol{s}_{t+1} \approx \boldsymbol{F}_{t-1}\Delta\boldsymbol{s}_t + \boldsymbol{G}_{t-1}\Delta\boldsymbol{a}_t \tag{7}$$

Assuming a high sampling frequency and a slow-varying system, the incremental model as in Eq. (7), provides a linear, time-varying approximation of the nonlinear system [15, 34]. An online system identification algorithm is utilized to generate estimates of the time-varying state and input matrix, $\hat{\boldsymbol{F}}_{t-1} \approx \boldsymbol{F}_{t-1}$ and $\hat{\boldsymbol{G}}_{t-1} \approx \boldsymbol{G}_{t-1}$

*The deadly triad refers to the instability and divergent behavior that methods which combine off-policy learning, bootstrapping, and function approximation, exhibit [22].

3. Network Topology

In this paper single-hidden-layer, fully-connected Multilayer Perceptrons (MLPs) ANNs are chosen as parametric structures for the actor, critic, and target critic, as they: (1) easily manage dimensional large input and output spaces (2) support batch or incremental learning methods (3) are differentiable (4) can approximate any nonlinear function on a compact space arbitrarily well (5) support flexible design (6) are widely applied in intelligent flight control applications [26–28, 30, 35–41]. The hyperbolic tangent activation function is utilized and the hidden layers consist of 10 neurons.

In [26, 27] an additional, offline-trained, trim network is employed to provide a global mapping between the nominal control positions and the system's operating condition. The actor in the framework proposed in this paper is able to learn a notion of a local flight-condition-dependent tracking policy, eliminating the need for a pretrained trim network. Consequently, the input of the actor, critic, and target critic networks include both the state vector and reference state tracking error $\begin{bmatrix} \boldsymbol{s} & \boldsymbol{Ps} - \boldsymbol{s}^R \end{bmatrix}^T \in \mathbb{R}^{(m+p)\times 1}$.

Whereas the input layers of the actor, critic, and target critic have the same structure, their output layers are different. The actor's output layer utilizes a hyperbolic tangent function, which is scaled according to the individual saturation limits of the control surfaces of the PH-LAB research aircraft as presented in Section III. In the case of the elevator for which the saturation limits are asymmetric, the limit with the largest absolute magnitude is utilized as scaling constant. The critic and target critic have a linear output layer. The topology of the neural networks is illustrated in Fig. 3.

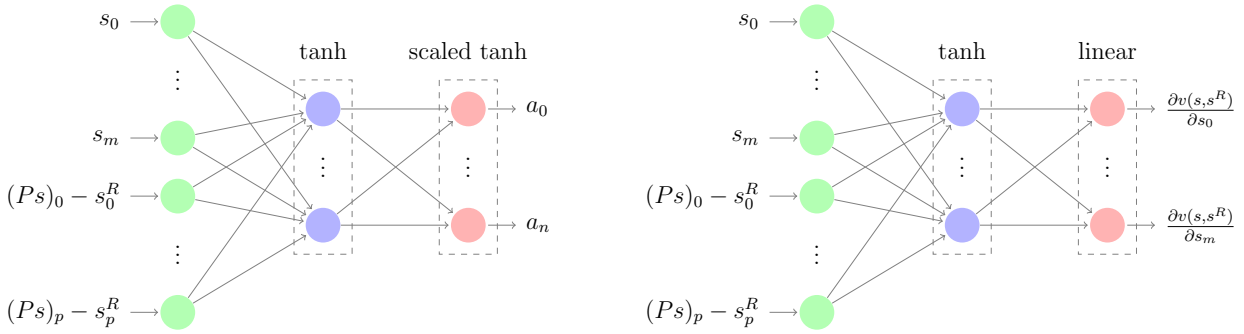

Fig. 3 Neural network topology used for actor (left) and network topology used for critic and target critic (right), with input, hidden, and output layers colored in green, blue, red, respectively. Layers can have either a (scaled) hyperbolic tangent or a linear activation function. The hidden layers have 10 neurons.

C. Update Rules

The parametric structures are updated at each time step as new information becomes available through an interaction with the plant. The leaning process is governed by the update rules, as presented in this section.

1. Critic

The critic is updated through a bootstrapping, TD backup operation. Equation (8) defines the mean squared error loss of the critic L^C to be minimized, with the error $\boldsymbol{e} \in \mathbb{R}^{1\times m}$. The error $\boldsymbol{e}$ is defined as the partial derivative of the TD error with respect to the state vector, as in Eq. (9).

$$L_t^C = \frac{1}{2}\boldsymbol{e}_t\boldsymbol{e}_t^T \tag{8}$$

$$\begin{aligned} \boldsymbol{e}_t &= -\frac{\partial\left[r(\boldsymbol{s}_t^R, \boldsymbol{s}_{t+1}) + \gamma v(\boldsymbol{s}_{t+1}, \boldsymbol{s}_{t+1}^R) - v(\boldsymbol{s}_t, \boldsymbol{s}_t^R)\right]}{\partial \boldsymbol{s}_t} \\ &= -\left[\frac{\partial r(\boldsymbol{s}_t^R, \boldsymbol{s}_{t+1})}{\partial \boldsymbol{s}_{t+1}} + \gamma\hat{\lambda}'(\boldsymbol{s}_{t+1}, \boldsymbol{s}_{t+1}^R, \boldsymbol{w}_t^{C'})\right]\frac{\partial \boldsymbol{s}_{t+1}}{\partial \boldsymbol{s}_t} + \hat{\lambda}(\boldsymbol{s}_t, \boldsymbol{s}_t^R, \boldsymbol{w}_t^C) \end{aligned} \tag{9}$$

In Eq. (9) the target $\left[\frac{\partial r(\boldsymbol{s}_t^R, \boldsymbol{s}_{t+1})}{\partial \boldsymbol{s}_{t+1}} + \gamma\hat{\lambda}'(\boldsymbol{s}_{t+1}, \boldsymbol{s}_{t+1}^R, \boldsymbol{w}_t^{C'})\right]\frac{\partial \boldsymbol{s}_{t+1}}{\partial \boldsymbol{s}_t}$ is computed by means of an evaluation of the reward

process, a forward pass through the target critic, a backward pass through the actor, and the current state and input matrix estimates of the incremental model. The latter becomes evident through the expansion of the term $\frac{\partial \boldsymbol{s}_{t+1}}{\partial \boldsymbol{s}_t}$ as presented in Eq. (10). As ANNs are inherently differentiable, the differentiability requirement of the actor in Eq. (10) is met.

$$\begin{aligned}\frac{\partial \boldsymbol{s}_{t+1}}{\partial \boldsymbol{s}_t} &= \frac{\partial f(\boldsymbol{s}_t, \boldsymbol{a}_t)}{\partial \boldsymbol{s}_t} + \frac{\partial f(\boldsymbol{s}_t, \boldsymbol{a}_t)}{\partial \boldsymbol{a}_t}\frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{s}_t} \\ &= \hat{\boldsymbol{F}}_{t-1} + \hat{\boldsymbol{G}}_{t-1}\frac{\partial \hat{\pi}(\boldsymbol{s}_t, \boldsymbol{s}_t^R, \boldsymbol{w}_t^A)}{\partial \boldsymbol{s}_t}\end{aligned} \tag{10}$$

Equation (11) defines the gradient of the critic's loss with respect to its parameter vector. [†] The critic is updated by stepping in opposite direction to the gradient of the loss with learning rate η^C, as defined in Eq. (12).

$$\frac{\partial L_t^C}{\partial \boldsymbol{w}_t^C} = \frac{\partial L_t^C}{\partial \hat{\lambda}(\boldsymbol{s}_t, \boldsymbol{s}_t^R, \boldsymbol{w}_t^C)}\frac{\partial \hat{\lambda}(\boldsymbol{s}_t, \boldsymbol{s}_t^R, \boldsymbol{w}_t^C)}{\partial \boldsymbol{w}_t^C} = \boldsymbol{e}_t\frac{\partial \hat{\lambda}(\boldsymbol{s}_t, \boldsymbol{s}_t^R, \boldsymbol{w}_t^C)}{\partial \boldsymbol{w}_t^C} \tag{11}$$

$$\boldsymbol{w}_{t+1}^C = \boldsymbol{w}_t^C - \eta_t^C\frac{\partial L_t^C}{\partial \boldsymbol{w}_t^C} \tag{12}$$

2. Target Critic

For the target critic, soft target updates are utilized as first introduced in [33]. The target critic is initialized as a copy of the critic and is subsequently updated according to a weighted sum as defined by Eq. (13). With the scalar mixing factor $\tau \ll 1$. The use of the target critic improves the learning stability as the target state-value derivatives are constrained to change slowly.

$$\boldsymbol{w}_{t+1}^{C'} = \tau \boldsymbol{w}_{t+1}^C + [1-\tau]\boldsymbol{w}_t^{C'} \tag{13}$$

3. Actor

The actor is updated towards an optimal policy $\pi^*(\boldsymbol{s}, \boldsymbol{s}^R) \in \mathbb{R}^{n\times 1}$, as defined in Eq. (14), which maximizes the state-value function $v(\boldsymbol{s}, \boldsymbol{s}^R)$. Consequently, the loss of the actor L^A and its partial derivative with respect to its parameter vector are defined by Eq. (15) and Eq. (16), respectively. Accordingly, the update procedure of the actor involves, the reward process, the critic, and the input matrix estimate of the incremental model. Equation (17) defines the gradient step with learning rate η^A.

$$\pi^*(\boldsymbol{s}_t, \boldsymbol{s}_t^R) = \arg\max_{a_t} v(\boldsymbol{s}_t \boldsymbol{s}_t^R) \tag{14}$$

$$L_t^A = -v(\boldsymbol{s}_t, \boldsymbol{s}_t^R) = -\left[r(\boldsymbol{s}_t^R, \boldsymbol{s}_{t+1}) + \gamma v(\boldsymbol{s}_{t+1}, \boldsymbol{s}_{t+1}^R)\right] \tag{15}$$

$$\begin{aligned}\frac{\partial L_t^A}{\partial \boldsymbol{w}_t^A} &= -\frac{\partial\left[r(\boldsymbol{s}_t^R, \boldsymbol{s}_{t+1}) + \gamma v(\boldsymbol{s}_{t+1}, \boldsymbol{s}_{t+1}^R)\right]}{\partial \boldsymbol{w}_t^A} \\ &= -\left[\frac{\partial r(\boldsymbol{s}_t^R, \boldsymbol{s}_{t+1})}{\partial \boldsymbol{s}_{t+1}} + \gamma\frac{\partial v(\boldsymbol{s}_{t+1}, \boldsymbol{s}_{t+1}^R)}{\partial \boldsymbol{s}_{t+1}}\right]\frac{\partial \boldsymbol{s}_{t+1}}{\partial \boldsymbol{a}_t}\frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{w}_t^A} \\ &= -\left[\frac{\partial r(\boldsymbol{s}_t^R, \boldsymbol{s}_{t+1})}{\partial \boldsymbol{s}_{t+1}} + \gamma\hat{\lambda}(\boldsymbol{s}_{t+1}, \boldsymbol{s}_{t+1}^R, \boldsymbol{w}_{t+1}^C)\right]\hat{\boldsymbol{G}}_{t-1}\frac{\partial \hat{\pi}(\boldsymbol{s}_t, \boldsymbol{s}_t^R, \boldsymbol{w}_t^A)}{\partial \boldsymbol{w}_t^A}\end{aligned} \tag{16}$$

$$\boldsymbol{w}_{t+1}^A = \boldsymbol{w}_t^A - \eta_t^A\frac{\partial L_t^A}{\partial \boldsymbol{w}_t^A} \tag{17}$$

[†]While the target term is also dependent on $\boldsymbol{w}_t^C$ through the update rules of the target critic, this is generally neglected [33, 42]

4. Incremental Model

The estimates of the state and input matrices are represented by the parameter matrix $\hat{\boldsymbol{\Theta}} \in \mathbb{R}^{(m+n)\times m}$, as presented in Eq. (18). The parameter matrix is accompanied by a covariance matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{(m+n)\times(m+n)}$, which expresses a measure of confidence of the estimates. The update process of the Recursive Least Squares (RLS) estimator starts with a prediction of state increments $\Delta\hat{\boldsymbol{s}}$ based on the latest measurements $\boldsymbol{X} \in \mathbb{R}^{(m+n)\times 1}$ and the current parameter estimates $\hat{\boldsymbol{\Theta}}$, as defined in Eq. (19) and Eq. (20). Subsequently, the prediction error $\boldsymbol{\epsilon} \in \mathbb{R}^{1\times m}$, named innovation, is computed, as defined in Eq. (21). Conclusively, the parameter estimates and covariance matrix are updated based on Eq. (22) and Eq. (23), with the scalar forgetting factor $\kappa \in [0, 1]$, which exponentially weights older measurements.

$$\hat{\boldsymbol{\Theta}}_{t-1} = \begin{bmatrix} \hat{\boldsymbol{F}}_{t-1}^T \\ \hat{\boldsymbol{G}}_{t-1}^T \end{bmatrix} \tag{18}$$

$$\boldsymbol{X}_t = \begin{bmatrix} \Delta\boldsymbol{s}_t \\ \Delta\boldsymbol{a}_t \end{bmatrix} \tag{19}$$

$$\Delta\hat{\boldsymbol{s}}_{t+1}^T = \boldsymbol{X}_t^T \hat{\boldsymbol{\Theta}}_{t-1} \tag{20}$$

$$\boldsymbol{\epsilon}_t = \Delta\boldsymbol{s}_{t+1}^T - \Delta\hat{\boldsymbol{s}}_{t+1}^T \tag{21}$$

$$\hat{\boldsymbol{\Theta}}_t = \hat{\boldsymbol{\Theta}}_{t-1} + \frac{\boldsymbol{\Lambda}_{t-1}\boldsymbol{X}_t}{\kappa + \boldsymbol{X}_t^T\boldsymbol{\Lambda}_{t-1}\boldsymbol{X}_t}\boldsymbol{\epsilon}_t \tag{22}$$

$$\boldsymbol{\Lambda}_t = \frac{1}{\kappa}\left[\boldsymbol{\Lambda}_{t-1} - \frac{\boldsymbol{\Lambda}_{t-1}\boldsymbol{X}_t\boldsymbol{X}_t^T\boldsymbol{\Lambda}_{t-1}}{\kappa + \boldsymbol{X}_t^T\boldsymbol{\Lambda}_{t-1}\boldsymbol{X}_t}\right] \tag{23}$$

D. Training Strategy

Algorithm 1 outlines the training strategy of the agent. The proposed strategy targets minimal model-dependency and computational complexity. In [24, 26] the agent's update operations are (partially) conducted on state predictions computed with a model of the plant. Although this approach enables the use of additional optimization schemes at each time step, it has limited applicability to time-critical, online operations, such as the PH-LAB research aircraft. Furthermore, these methods strongly rely on both the model's prediction performance and its capability to estimate the state-transition derivatives.

The method proposed in this paper utilizes current state measurements for the update operations instead. This reduces the model-dependency to only the state-transition derivatives, making the controller's performance less vulnerable to imperfect models. In addition, the computational complexity of the learning algorithm is improved, as it requires less (re)evaluations of the actor, critic, and target critic.

III. Controller Design

This section starts by introducing the simulation model of the PH-LAB research aircraft utilized in this paper. Subsequently, the flight controller design is presented, followed by the hyperparameters.

A. High-Fidelity Simulation Model

Operated by the Faculty of Aerospace Engineering of the Delft University of Technology, the Cessna 550 Citation II PH-LAB depicted in Fig. 1 is a multipurpose research platform. In this paper, a high-fidelity, nonlinear, six-degrees-of-freedom model of a Cessna 500 Citation I, developed with the Delft University Aircraft Simulation Model and Analysis Tool (DASMAT), is utilized. Despite the differences of the aircraft in fuselage size, engine power, and wing size, the model is still representative of the PH-LAB research aircraft [43]. Additionally, the simulation model includes engine dynamics, actuator dynamics, and sensor models. The sensor models are not used, as clean measurements are assumed in this paper. The actuator dynamics are modeled with a first order actuator model and control surface deflection saturation. The limits are listed in Table 1. Furthermore, the aircraft's yaw damper is disabled to provide the agent with full control authority over the control surfaces. The engine's thrust setting is controlled by an internal

Algorithm 1 Learning Framework

Require:

simulation parameters $\Delta t, T$

agent parameters $\gamma, \eta^A, \eta^C, \tau, \kappa$

differentiable deterministic policy parameterization $\hat{\pi}(\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{w}^A)$

differentiable state-value function derivative parameterization $\hat{\lambda}(\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{w}^C)$

differentiable target state-value function derivative parameterization $\hat{\lambda}'(\boldsymbol{s}, \boldsymbol{s}^R, \boldsymbol{w}^{C'})$

reward function derivative $\frac{\partial r(\boldsymbol{s}^R, \boldsymbol{s})}{\partial \boldsymbol{s}}$

Initialize:

$\boldsymbol{w}_0^A, \boldsymbol{w}_0^C, \hat{\boldsymbol{\Theta}}_0, \boldsymbol{\Lambda}_0, \boldsymbol{s}_0$

$\boldsymbol{w}_0^{C'} \leftarrow \boldsymbol{w}_0^C$

Compute:

for $i = 0$ **to** $int\left(\frac{T}{\Delta t}\right) - 1$ **do**

  get $\boldsymbol{s}_i^R$

  if $i = 1$ **then**

    $\boldsymbol{w}_i^A \leftarrow \boldsymbol{w}_{i-1}^A$

    $\boldsymbol{w}_i^C \leftarrow \boldsymbol{w}_{i-1}^C$

    $\boldsymbol{w}_i^{C'} \leftarrow \boldsymbol{w}_{i-1}^{C'}$

  end if

  if $i > 1$ **then**

    $\begin{bmatrix} \hat{\boldsymbol{F}}_{i-2}^T \\ \hat{\boldsymbol{G}}_{i-2}^T \end{bmatrix} \leftarrow \hat{\boldsymbol{\Theta}}_{i-2}$

    $\frac{\partial \boldsymbol{s}_i}{\partial \boldsymbol{s}_{i-1}} \leftarrow \hat{\boldsymbol{F}}_{i-2} + \hat{\boldsymbol{G}}_{i-2} \frac{\partial \hat{\pi}(\boldsymbol{s}_{i-1}, \boldsymbol{s}_{i-1}^R, \boldsymbol{w}_{i-1}^A)}{\partial \boldsymbol{s}_{i-1}}$

    $\frac{\partial r_i}{\partial \boldsymbol{s}_i} \leftarrow \frac{\partial r(\boldsymbol{s}_{i-1}^R, \boldsymbol{s}_i)}{\partial \boldsymbol{s}_i}$

    $\hat{\boldsymbol{\lambda}}_{i-1} \leftarrow \hat{\lambda}(\boldsymbol{s}_{i-1}, \boldsymbol{s}_{i-1}^R, \boldsymbol{w}_{i-1}^C)$

    $\hat{\boldsymbol{\lambda}}_i \leftarrow \hat{\lambda}(\boldsymbol{s}_i, \boldsymbol{s}_i^R, \boldsymbol{w}_{i-1}^C)$

    $\hat{\boldsymbol{\lambda}}_i' \leftarrow \hat{\lambda}'(\boldsymbol{s}_i, \boldsymbol{s}_i^R, \boldsymbol{w}_{i-1}^{C'})$

    $\Delta \boldsymbol{w}_i^A \leftarrow \eta^A \left[\frac{\partial r_i}{\partial \boldsymbol{s}_i} + \gamma \hat{\boldsymbol{\lambda}}_i\right] \hat{\boldsymbol{G}}_{i-2} \frac{\partial \hat{\pi}(\boldsymbol{s}_{i-1}, \boldsymbol{s}_{i-1}^R, \boldsymbol{w}_{i-1}^A)}{\partial \boldsymbol{w}_{i-1}^A}$

    $\Delta \boldsymbol{w}_i^C \leftarrow -\eta^C \left[-\left[\frac{\partial r_i}{\partial \boldsymbol{s}_i} + \gamma \hat{\boldsymbol{\lambda}}_i'\right] \frac{\partial \boldsymbol{s}_i}{\partial \boldsymbol{s}_{i-1}} + \hat{\boldsymbol{\lambda}}_{i-1}\right] \frac{\partial \hat{\boldsymbol{\lambda}}_{i-1}}{\partial \boldsymbol{w}_{i-1}^C}$

    $\boldsymbol{w}_i^A \leftarrow \boldsymbol{w}_{i-1}^A + \Delta \boldsymbol{w}_i^A$

    $\boldsymbol{w}_i^C \leftarrow \boldsymbol{w}_{i-1}^C + \Delta \boldsymbol{w}_i^C$

    $\boldsymbol{w}_i^{C'} \leftarrow \tau \boldsymbol{w}_i^C + [1 - \tau] \boldsymbol{w}_{i-1}^{C'}$

    $\Delta \boldsymbol{s}_{i-1} \leftarrow \boldsymbol{s}_{i-1} - \boldsymbol{s}_{i-2}$

    $\Delta \boldsymbol{a}_{i-1} \leftarrow \boldsymbol{a}_{i-1} - \boldsymbol{a}_{i-2}$

    $\Delta \boldsymbol{s}_i \leftarrow \boldsymbol{s}_i - \boldsymbol{s}_{i-1}$

    $\boldsymbol{X}_{i-1} \leftarrow \begin{bmatrix} \Delta \boldsymbol{s}_{i-1} \\ \Delta \boldsymbol{a}_{i-1} \end{bmatrix}$

    $\Delta \hat{\boldsymbol{s}}_i^T \leftarrow \boldsymbol{X}_{i-1}^T \hat{\boldsymbol{\Theta}}_{i-2}$

    $\boldsymbol{\epsilon}_{i-1} \leftarrow \Delta \boldsymbol{s}_i^T - \Delta \hat{\boldsymbol{s}}_i^T$

    $\hat{\boldsymbol{\Theta}}_{i-1} \leftarrow \hat{\boldsymbol{\Theta}}_{i-2} + \frac{\boldsymbol{\Lambda}_{i-2} \boldsymbol{X}_{i-1}}{\kappa + \boldsymbol{X}_{i-1}^T \boldsymbol{\Lambda}_{i-2} \boldsymbol{X}_{i-1}} \boldsymbol{\epsilon}_{i-1}$

    $\boldsymbol{\Lambda}_{i-1} \leftarrow \frac{1}{\kappa} \left[\boldsymbol{\Lambda}_{i-2} - \frac{\boldsymbol{\Lambda}_{i-2} \boldsymbol{X}_{i-1} \boldsymbol{X}_{i-1}^T \boldsymbol{\Lambda}_{i-2}}{\kappa + \boldsymbol{X}_{i-1}^T \boldsymbol{\Lambda}_{i-2} \boldsymbol{X}_{i-1}}\right]$

  end if

  $\boldsymbol{a}_i \leftarrow \hat{\pi}(\boldsymbol{s}_i, \boldsymbol{s}_i^R, \boldsymbol{w}_i^A)$

  get $\boldsymbol{s}_{i+1}$ by taking action $\boldsymbol{a}_i$

end for

airspeed controller. The simulation model is run with a sampling frequency of 50 Hz. It contains and accepts a large number of states and control inputs, respectively. A subset of it spans the agent-plant interface as defined by Eq. (24).

Table 1 Aerodynamic control surface saturation limits utilized in the actuator dynamics model of the Cessna 500 Citation I simulation model.

Control Surface	Saturation Limits
δ_e	$[-20.05, 14.90]$ deg
δ_a	$[-37.24, 37.24]$ deg
δ_r	$[-21.77, 21.77]$ deg

$$\boldsymbol{s} = \begin{bmatrix} p & q & r & V_{tas} & \alpha & \beta & \phi & \theta & H \end{bmatrix}^T \quad \boldsymbol{a} = \begin{bmatrix} \delta_e & \delta_a & \delta_r \end{bmatrix}^T \tag{24}$$

B. Flight Controller

The adaptive learning framework is applied to angular rate control of the pitch and roll rate, augmented with an outer control loop, as illustrated in Fig. 4. Rate control exhibits the lowest learning complexity due to the direct dynamic relation between the angular rates and control surfaces. The outer control loop consists of conventional PID controllers and provides a higher-level control interface, which enables reference tracking of an altitude and roll angle profile. Under the assumption of a symmetric aircraft, a decoupled controller design is employed utilizing separate longitudinal and lateral learning controllers, with the state, reference state, and actions vectors as in Eq. (25) and Eq. (26). The resulting selection and weight matrices are presented in Eq. (27) and Eq. (28). For large roll angles, there is limited control of the flight path angle through the pitch rate. On the other hand, using separate controllers with each their own learning framework instance reduces the complexity of the problem to be learned and simplifies the online incremental model identification. As separate critics are used, there is no need to specify relative weights between the longitudinal and lateral tracking errors.

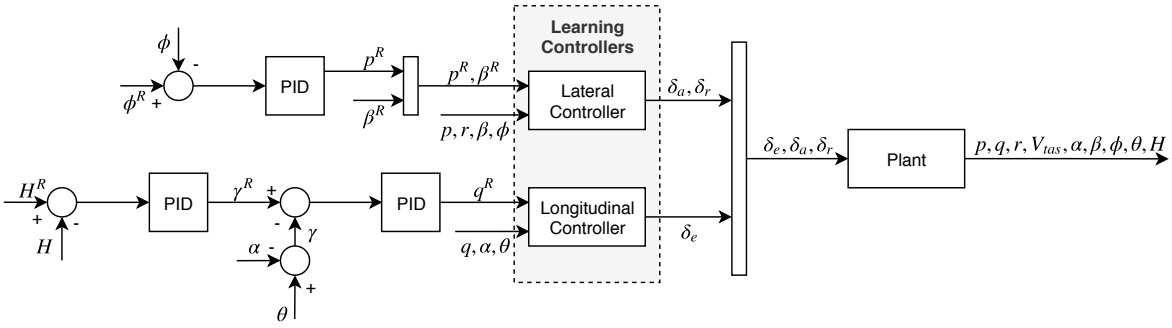

Fig. 4 Schematic of flight controller with individual longitudinal and lateral learning controllers.

$$\boldsymbol{s}^{lon} = \begin{bmatrix} q & \alpha & \theta \end{bmatrix}^T \quad \boldsymbol{s}^{R^{lon}} = \begin{bmatrix} q^R \end{bmatrix}^T \quad \boldsymbol{a}^{lon} = \begin{bmatrix} \delta_e \end{bmatrix}^T \tag{25}$$

$$\boldsymbol{s}^{lat} = \begin{bmatrix} p & r & \beta & \phi \end{bmatrix}^T \quad \boldsymbol{s}^{R^{lat}} = \begin{bmatrix} p^R & \beta^R \end{bmatrix}^T \quad \boldsymbol{a}^{lat} = \begin{bmatrix} \delta_a & \delta r \end{bmatrix}^T \tag{26}$$

$$\boldsymbol{P}^{lon} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad \boldsymbol{Q}^{R^{lon}} = \begin{bmatrix} 1 \end{bmatrix} \tag{27}$$

$$\boldsymbol{P}^{lat} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \boldsymbol{Q}^{R^{lat}} = \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix} \tag{28}$$

As can be observed from Eq. (25) the airspeed is not provided to the longitudinal controller. The exclusion of the airspeed is motivated by the fact that the online identification of the airspeed related parameters of the incremental model is nontrivial due to time scale separation and relatively small local variations in airspeed. The latter is exacerbated by the internal airspeed controller. Additional to a reference roll rate, the lateral learning controller receives a zero degree reference for the sideslip angle, such that it can learn to minimize it, with the goal of flying coordinated turns.

C. Hyperparameters

The hyperparameters utilized in the remainder of this paper are summarized in Table 2. These parameters are utilized for both the longitudinal and lateral controllers and are determined empirically by experimentation and common guidelines proposed in [24]. In Section IV a distinction is made between an online training, online operation, and online adaption phase. As the name suggests, the training phase focuses on the learning process of the controllers. During the operation phase, the agent is employed to fly representative maneuvers. In the adaption phase, changes are applied to the plant and the agent during flight. In all phases, the agent is constantly learning through interaction. During the training phase, a faster learning process is preferred over a lower chance of converging to a local minimum. Accordingly, the learning rates during the training phase are higher, than in the operation phase. The utilization of the target critic depends on the experiment being conducted in Section IV. For the training phase the RLS incremental model estimator, the actor, and the critic are initialized according to the values as listed in Table 2. During the operation phase, the learning framework is initialized at its pretrained state.

The forgetting factor κ of the RLS estimator is set to one. This is done to ensure consistency of the estimator. During periods of poor or no excitation (which is the case for the majority of time during operation), the covariance matrix increases exponentially when $\kappa < 1$, also referred to as estimator windup. Once, the system is excited again, abrupt changes in the estimate occur despite no changes in the actual system. In addition to poor excitation, non-uniformly distributed information over all parameters and time-scale separation in the variation of parameters also lead to estimator windup. Many approaches to mitigate these issues have been proposed. In [44–46] non-uniformly distributed information is dealt with by selective amplification of the covariance matrix, called directional forgetting. Similarly, [47, 48] propose vector-type or selective forgetting, were individual parameter dependent forgetting factors are used, to deal with the effects of time-scale separation. The design of an advanced online parameter identification algorithm for the PH-LAB is out of the scope of this research.

Setting the forgetting factor to one has the disadvantage that the estimator does not forget older data. Consequently, the estimator becomes less adaptive over time. In this paper, this problem is mitigated by resetting the covariance matrix to its initial value when a large change in the system is detected through the estimator's innovation term, as elaborated in Section IV.

Table 2 Hyperparameters of learning framework instances of both lateral and longitudinal controllers.

Parameter	Value
η^A, η^C, τ	training phase $5, 10, 0.01$ or $5, 10, 1$ operation phase $1, 2, 0.01$
$\boldsymbol{w}_0^A, \boldsymbol{w}_0^C$	$\mathcal{N}_{trunc}(\mu = 0, \sigma = 0.05)$
$\hat{\boldsymbol{F}}_0, \hat{\boldsymbol{G}}_0, \Lambda_0$	$I, \mathbf{0}, I \cdot 10^8$
κ	1.0
γ	0.8

IV. Results and Discussion

In this section, the results of the conducted experiments are presented and discussed. Three distinct cases are evaluated. First, the framework's capability to train online without a priori knowledge of the system dynamics is evaluated. In addition, the online identification of the incremental model, as well as, the effect of the target critic on the learning stability are discussed. Subsequently, experiments are conducted to demonstrate that the agent is able to operate the aircraft during representative maneuvers at a variety of flight conditions. Last but not least, the framework's capability of dealing with unforeseen changes, such as changes in the aircraft's dynamics, is analyzed.

A. Online Training Phase

In this section, the framework's capability of online learning without a priori knowledge of the plant is demonstrated. The longitudinal and lateral controllers are trained separately. During training of the longitudinal controller, the control surfaces associated with lateral motion are kept at their initial trim value and vice versa.

A sine function with an amplitude of 5 degrees per second and a frequency of 0.2 Hertz is utilized as pitch and roll rate reference. As elaborated in the previous section, the sideslip angle reference is zero. Different signals commonly used for (aircraft) system identification [49], such as frequency sweeps, doublets, 3211 doublets, and sinusoidal functions are good reference signal candidates. The proposed sinusoidal reference allows for a short training phase (under 60 seconds) with an acceptable load factor range, as depicted in Fig. 5. Persistent Excitation (PE) is essential to both state-space exploration in the learning process and dynamic excitation in the system identification process of the incremental model [22, 24, 49]. Exponentially decaying, sinusoidal excitation is applied to the elevator and ailerons to excite the system during the initial training phase. As the agent learns to track the dynamic reference signals, the excitation on the elevator and ailerons is reduced. Otherwise, the agent would learn to compensate for the applied excitation over time. Through the aircraft's dutch roll eigenmode, its yaw motion is also excited through the ailerons. Therefore no additional excitation is applied to the rudder.

As illustrated in Fig. 5 both the longitudinal and lateral controller is able to follow the reference signals after less than 30 seconds of training. The same observation is made from the actor and critic parameters as depicted in Fig. 6, as an overall convergence of the parameters can be observed. As a result of the motion in the longitudinal controller training phase, the airspeed oscillates. The agent perceives changes in airspeed as changes in the environment and constantly adapts to it. This can be observed in Fig. 6, where the actor and critic parameters of the longitudinal controller show minor oscillatory behavior.

From Fig. 5 it can be observed that it takes longer to learn to control the sideslip angle than the angular rates. This can be attributed to two main factors: (1) the control surfaces are more directly related to the angular rates than to the sideslip angle (2) the rudder is not directly excited. The consequence of the latter can also be observed in the online identification process of the incremental models.

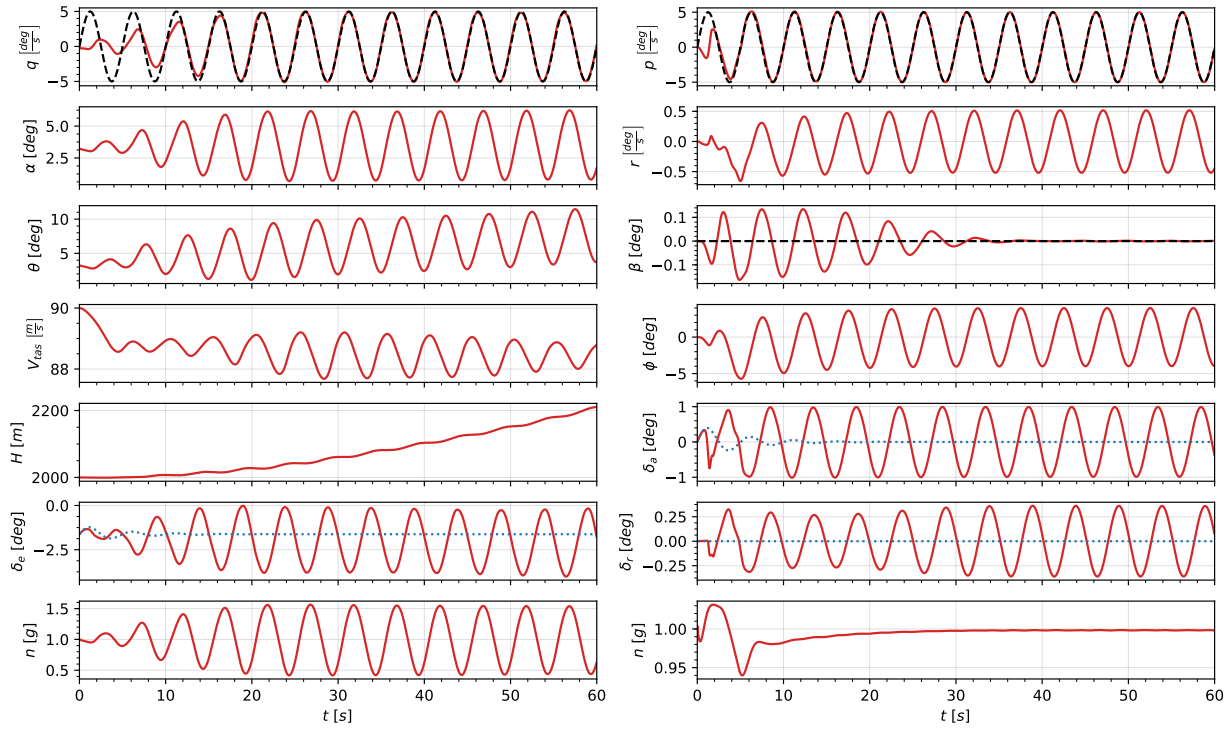

Fig. 5 Online training procedures of longitudinal (left) and lateral (right) controller, starting at trimmed operation condition $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$, with $(\tau, \eta^A, \eta^C) = (1, 5, 10)$. Reference and excitation signals are illustrated by black dashed and blue dotted lines, respectively.

1. Online Identification of Incremental Model

Figure 6 depicts the online identified state and input matrices for both the longitudinal and lateral incremental models. As elaborated in Section II, the state and input matrices play an important role in the update operations of both the actor and critic. For meaningful updates of the actor and critic, the online identified model parameters estimates should have the correct sign and to some extent correct relative magnitude. To allow for an evaluation of the identification performance, reference parameters values are provided by dotted lines. These reference values are derived from a linearization of the nonlinear plant at the initial trim condition at $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$. These constant values are not exact as: (1) they do not take in to account the actuator dynamics (2) nor the airspeed controller (3) do not take into account time-varying changes in the plant (4) the linearized model contains different states than the incremental model's regression matrix. Nonetheless, these reference values provide a good approximation to evaluate the performance of the online identified estimates.

It can be observed that both the parameters of the state and input matrices are identified in less than 10 seconds, providing the agent with local information about the plant, with the exception of one term. The term $\frac{\partial p}{\partial \delta_r}$ of the input matrix of the lateral incremental model, is initially incorrectly identified and slowly moves towards its correct final value. This behavior is partially a consequence of the fact that the rudder is not initially directly excited.

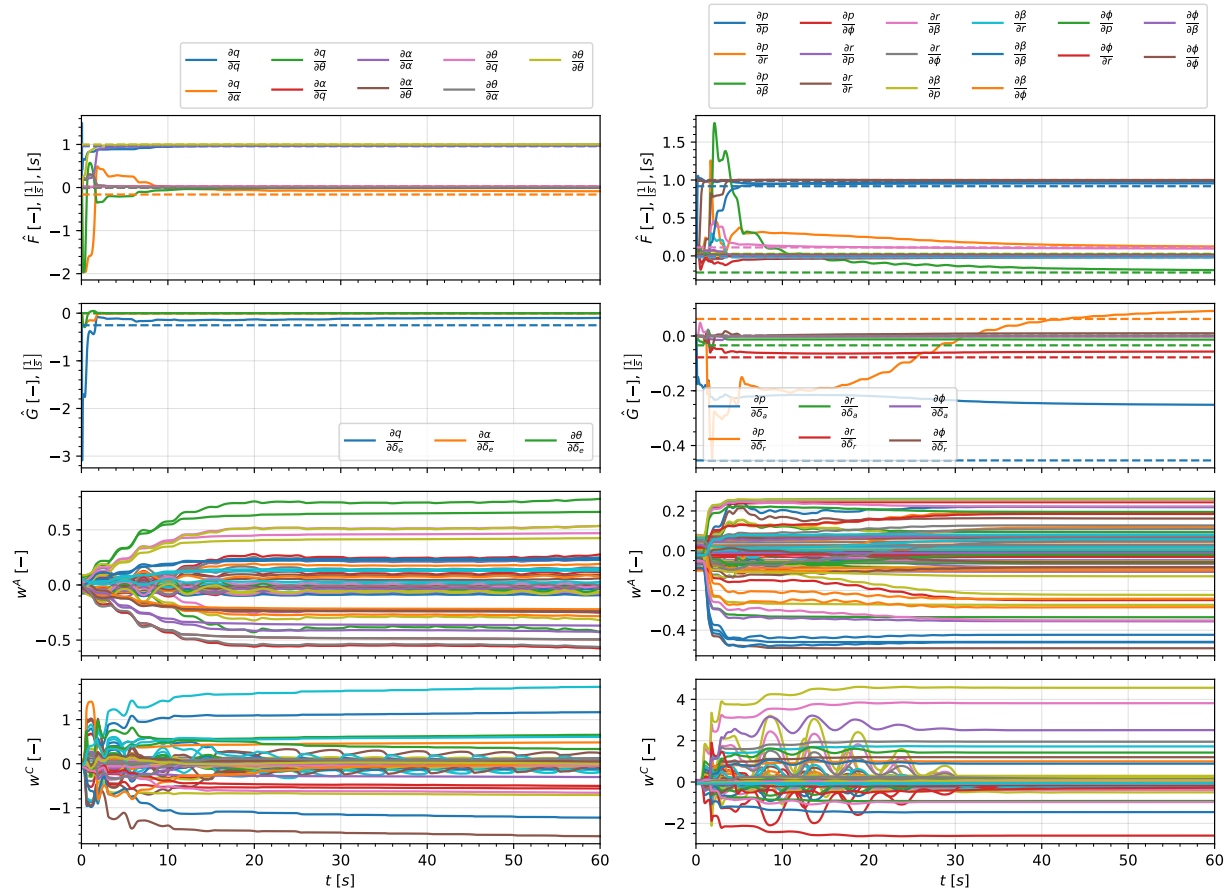

Fig. 6 State and input matrix estimates, and actor and critic parameters during online training of longitudinal (left) and lateral (right) controller, starting at trimmed operation condition $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$, with $(\tau, \eta^A, \eta^C) = (1, 5, 10)$ and $(\hat{\boldsymbol{F}}_0, \hat{\boldsymbol{G}}_0, \hat{\boldsymbol{\Theta}}_0) = (I, \boldsymbol{0}, I \cdot 10^8)$. Estimates derived from a plant linearization are represented by dotted lines.

2. Online Training with Untrimmed Initialization

The online training phase as presented in the previous section is initiated at a trimmed condition. In practice, however, it is challenging to find an initial trimmed state without a priori knowledge of the plant. Therefore, the proposed framework should also be able to deal with untrimmed initial conditions. The effect of an untrimmed initialization on the training phase is therefore examined for the longitudinal controller by superimposing a random uniformly distributed elevator deflection offset around its trimmed state, in addition to the small random offset originating from the random initialization of the actor's parameters. Furthermore, the effects of the proposed utilization of a target critic on the learning speed and stability are examined, by comparing the results attained from experiments with and without a target critic. 500 runs are simulated for each case.

In Fig. 7 the area between the 25th and 75th percentile of the 500 runs are presented in red and blue, for a mixing factor of $\tau = 1.0$ and $\tau = 0.01$, respectively. In both cases the same learning rates are utilized $(\eta^A, \eta^C) = (5, 10)$. It can be observed that the interquartile range of the agent without target critic narrows within the first 22 seconds of training. In comparison, for the agent with a target critic, this is observed 20 seconds later. This implies that it takes longer for the majority of the runs to converge to an equivalent final policy. On the other hand, in the first 5 seconds of training the interquartile range of the agent with a target critic spans a smaller range of experienced pitch rate values than its counterpart. This implies that for the majority of the 500 runs, the initial training phase is less dynamic and erroneous, assisting the training stability. The latter is confirmed by the failure rates. The sample mean failure rate for 500 runs for the agent with and without target critic ($\tau = 0.01$ and $\tau = 1.0$) is 0% and 5.2%, respectively. Failure is defined as the occurrence of a float overflow in the learning process due to numerical instability.

Most of the time an unstable or diverging learning behavior leads to fast growth of the critic's parameters. As a consequence, the actor's loss gradients become very large leading to aggressive and large updates from which the agent cannot recover. The utilization of the proposed target critic stabilizes the critic's learning and as a consequence, no failures are encountered within 500 runs. As a comparison, without the proposed target critic, the training of the longitudinal controller has a failure rate of 5.2%. Safety of learning can be further improved with dedicated safe learning algorithms[50].

Consistent with findings in [33] the target critic slows down and stabilizes the learning process. Especially in an application in the PH-LAB aircraft, no failure is acceptable. Therefore, the augmentation of the current IDHP framework with the proposed target critic is vital to a successful implementation of the RL controller to the PH-LAB.

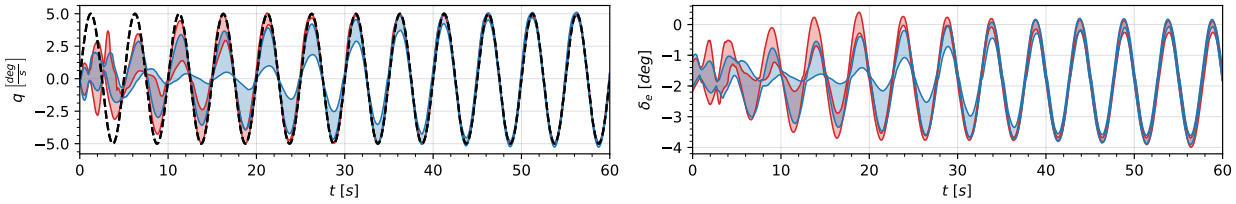

Fig. 7 Interquartile range of pitch rate and elevator deflection for 500 runs of the online training phase of the longitudinal controller with random untrimmed initialization around the operating point $(V_{tas}, H) = (90\,\frac{m}{s}, 2\,km)$. The reference signal is represented by a black, dashed line. The agent with and without a target critic are represented in blue and red, respectively.

B. Online Operation Phase

Subsequent to a successful online training phase of both the lateral and longitudinal controllers, these can be utilized to operate the aircraft. In this experiment, the goal is to demonstrate that the agent is able to operate the aircraft, successfully and reliably during representative maneuvers at a variety of flight conditions. For each of the four initial flight conditions as listed in Table 3, the agent follows an altitude and bank angle flight profile, which consists of a prolonged climb and a subsequent descent (with 5 $\frac{m}{s}$), combined with left and right turns (of 25 deg and 20 deg), as depicted in Fig. 8 and Fig. 9. For each flight condition 100 runs are simulated, where the plant is initialized at a trimmed state and the agent in its final state of the online training phase of the previous section.

Figure 8 illustrates all 100 flight maneuvers for flight condition FC0, starting at the same condition in which the previous online training phase was conducted. For all states and actions, as well as, the reference pitch and roll rate, the figure depicts the area between the minimum and maximum bounds of all 100 runs. The bounds emphasize that none of

the 100 runs experience a failure, as listed in Table 3. It can be observed that the agent flies the commanded height and bank angle profile, by following the reference pitch and roll rate, as commanded by the PID controllers. In addition, the agent succeeds to minimize the sideslip angle and conduct a well-coordinated turn, by actuating the rudder.

Although the online pretrained agent already succeeds in flying the aircraft, its performance continues to improve. From the min-max bound of the time history of the sideslip angle and rudder deflection, a minor oscillatory behavior is observed in the first two turns. As the agent carries out the first two turns it gains more experience and improves its performance. Consequently, the oscillatory behavior vanishes and min-max bound narrows for the two subsequent turns.

From the time history of the pitch rate and airspeed, it can be observed that the agent's pitch rate reference tracking performance temporarily degrades during quick changes in airspeed. Due to the separation between the airspeed controller and the longitudinal and lateral controllers, as elaborated in Section III, the learning controllers do not conceive a notion of the airspeed as a distinct state, but can only experience it as an external, temporary change in the perceived plant dynamics. Although the agent's performance temporary degrades, due to quick changes in airspeed, the agent still adapts to different airspeed regimes and their influence on the plant's dynamics.

This is demonstrated in Fig. 9, where the agent (pretrained at $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$) is utilized to operate at the flight condition FC3 with $(V_{tas}, H) = (140\ \frac{m}{s}, 5\ km)$. Despite the different flight condition, which the agent has not previously experienced, the agent is able to follow the flight profile in all 100 runs without failure, as depicted in by Fig. 9 and Table 3. Similarly, to the previous flight condition, the agent is still constantly adapting and improving upon new experiences as can be observed for example from the decrease in width and magnitude of the min-max bound of the time history of the sideslip angle.

Table 3 Description and sample failure rate of different flight conditions each simulated 100 times during the online operation phase. For all cases the agent is initialized at its final state of the online training phase conducted at $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$**.**

Flight Condition ID	$H_0\ m$	$V_{tas_0}\ \frac{m}{s}$	Failure Rate
FC0	2000	90	0%
FC1	2000	140	0%
FC2	5000	90	0%
FC3	5000	140	0%

C. Online Adaption

During operation, the controller needs to be able to adapt to uncertainties in the system, such as unexpected failures or time-varying components [8, 13, 51, 52]. Conventional ACDs that require an initial offline training phase, are unable to quickly adapt online to these changes [30]. In this section, the adaptability of the proposed framework is validated in two experiments. In the first step, a failure is introduced to the aileron to demonstrate the framework's capability to identify changes in the plant. Subsequently, another experiment is conducted where a disturbance is introduced directly in the actor's parameters. In both cases, the agent is initiated at its online pretrained state. The controller is commanded to fly two right rate-one turns while holding the initial altitude. The disturbances are induced at the start of the second turn at 110 seconds.

1. Adaptive Control in the Presence of Aileron Failure

To simulate the failure of one aileron, the aileron deflection as commanded by the actor is halved before it is passed as input to the plant. As a result, the perceived control effectiveness of the ailerons is halved and the controller has to double its aileron command to fly the turn. In this experiment, the covariance matrix of the lateral incremental model is reinitialized at $I \cdot 10^8$ once a fault is detected. Many methods for fault detection have been proposed in [3, 13, 51]. Here, a sudden increase of the innovation term, as presented in Eq. (21), is utilized to monitor for changes in the plant. As illustrated in Fig. 10 the controller is able to detect the change in the plant and complete the turn, despite the failure in the ailerons. From the estimates of the input matrix of the lateral incremental model, it can be observed that the online identification perceives the change in the plant's dynamics. For example the incremental parameter that corresponds to the $\frac{\partial p}{\partial \delta_a}$ term, changes from the initial value of $0.24\ \frac{1}{s}$ to $0.12\ \frac{1}{s}$.

Consequently, the agent commands a larger aileron deflection, as illustrated in Fig. 10, to complete the right turn. In

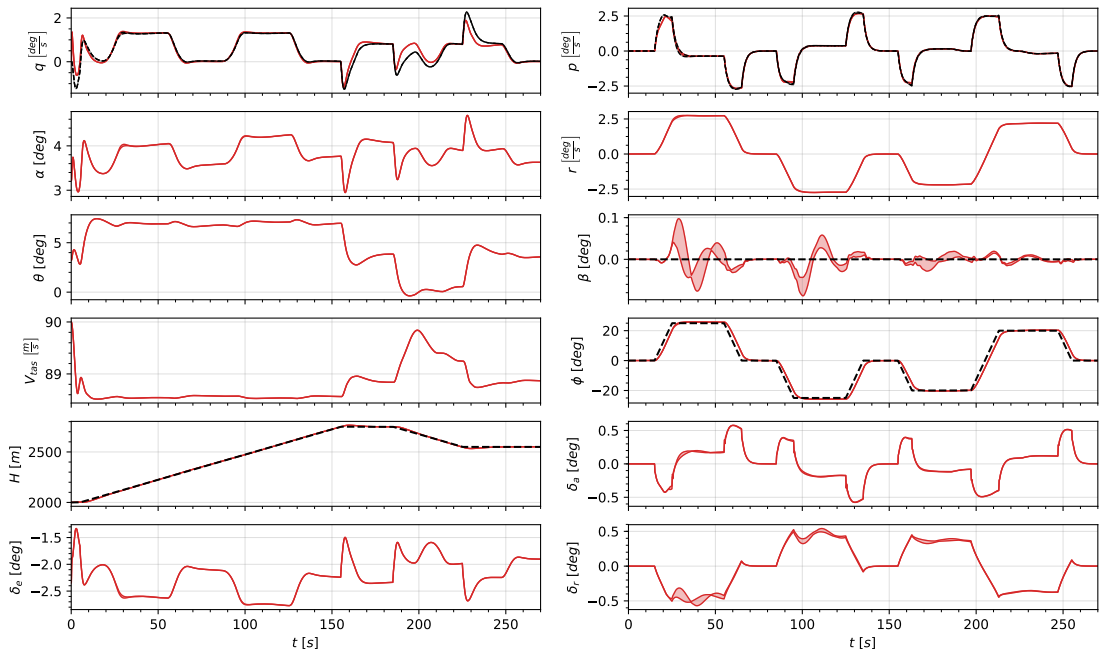

Fig. 8 Min-max bounds over 100 runs of the online operation phase, starting at trimmed operation condition FC0 $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$ and pretrained agent, with $(\tau, \eta^A, \eta^C) = (0.01, 1, 2)$. The agent was pretrained online at $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$. Reference signals are illustrated by black, dashed lines.

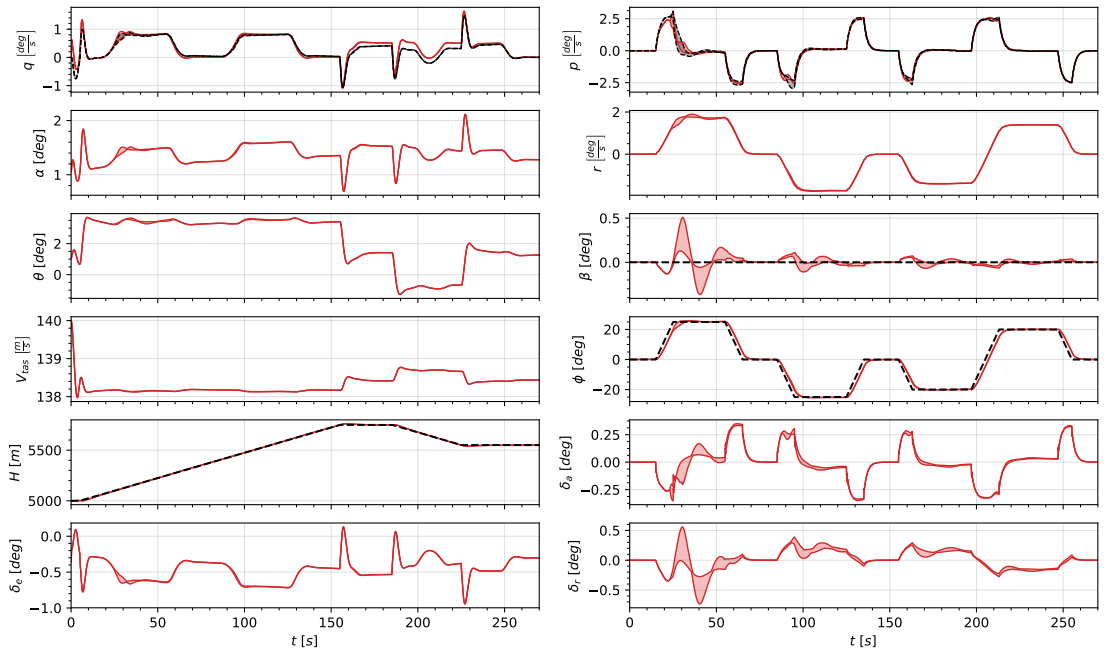

Fig. 9 Min-max bounds over 100 runs of the online operation phase, starting at trimmed operation condition FC3 $(V_{tas}, H) = (140\ \frac{m}{s}, 5\ km)$ and pretrained agent, with $(\tau, \eta^A, \eta^C) = (0.01, 1, 2)$. The agent was pretrained online at $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$. Reference signals are illustrated by black, dashed lines.

this case, the control adaption of the agent interacts with the outer loop PID controller. Therefore, in the next experiment, the adaptability of the agent is demonstrated by injecting a disturbance directly on the parameters of the actor.

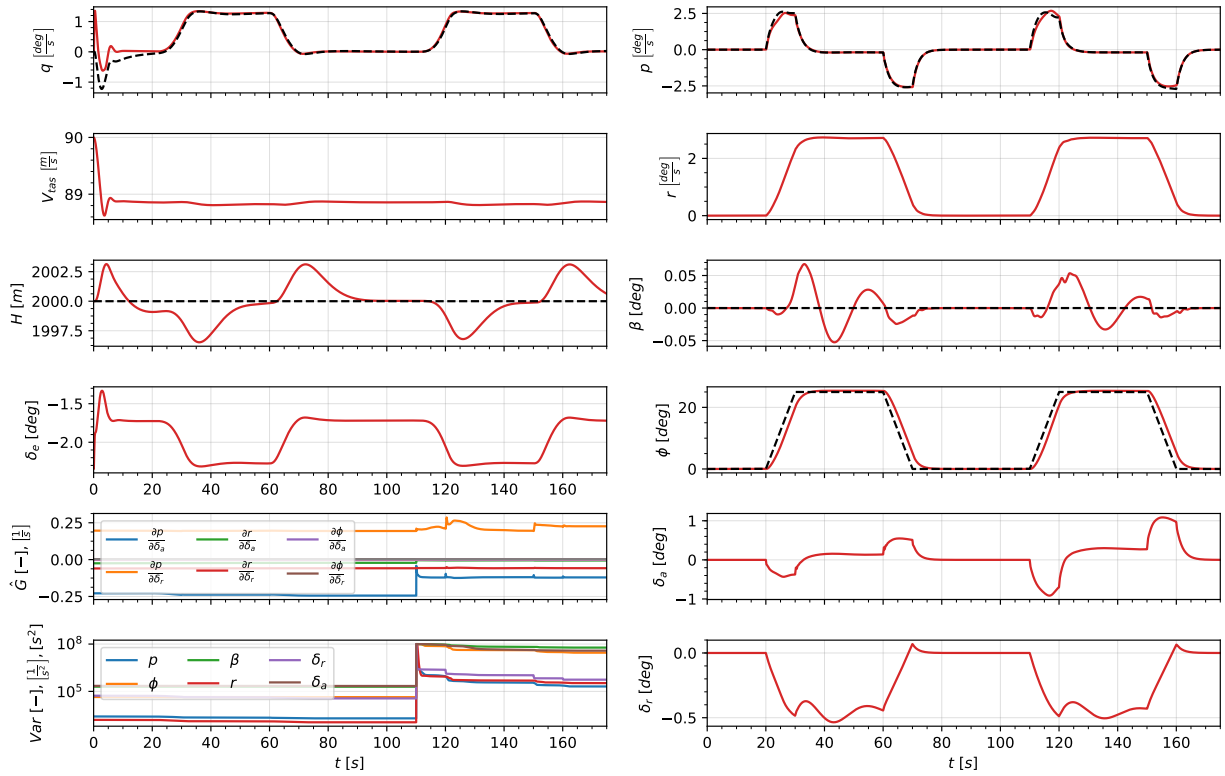

Fig. 10 Control adaption and online identification of the lateral incremental model during an aileron failure induced at 110 seconds. The plant is initiated at a trimmed state at $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$ and the longitudinal and lateral controller at their state after the initial online training phase. Reference signals are represented by black, dashed lines.

2. Adaptive Control in the Presence of Control Disturbance

Similarly to the previous experiment, the agent is commanded to fly two right rate-one turns, while maintaining its initial altitude. At 110 seconds the actor's parameters are disturbed by a zero-mean Gaussian noise with a standard deviation of 0.2. Consequently, the current policy is disturbed and the agent has to adapt to regain control over the aircraft and complete the flight maneuver.

As can be observed from Fig. 11, subsequent to the disturbance at 110 seconds the actor's parameters adapt towards a new near-optimal policy within 10 seconds. This adaption is also observed from the peak in the actor's loss gradients as depicted in the lower left box of Fig. 11. Furthermore, the actor's parameters converge to a different distribution, than its initial one. This implies that there are more than one near-optimal policies and that a near-optimal policy is not uniquely defined by one set of the actor's parameters.

From the time history of the sideslip angle and roll rate, it can be observed that the disturbance is detrimental to the agent's performance. More specifically, both the overall magnitude of the sideslip angle as well as roll rate tracking error increases during the initiation of the second right turn. Nonetheless, the agent still manages to follow the reference roll angle profile. As the agent exits the second turn at 150 seconds it has already recovered and improved upon its original performance.

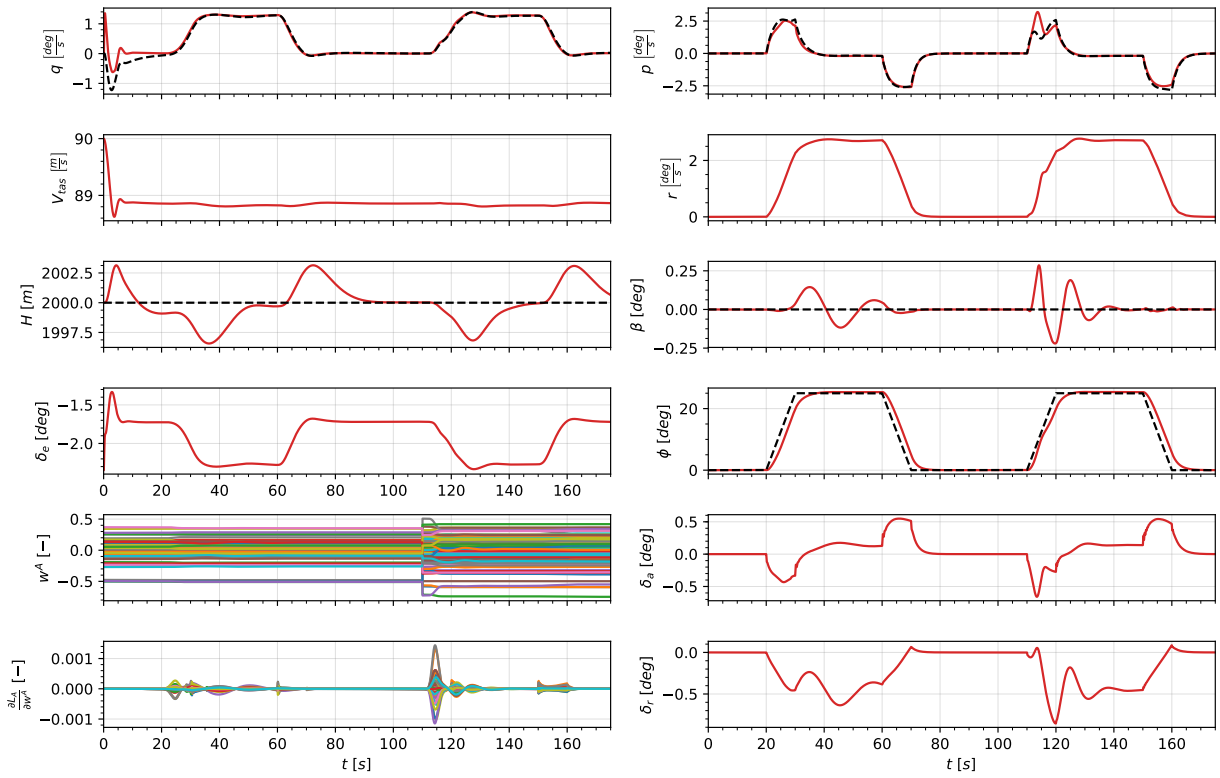

Fig. 11 Control adaption of the lateral controller during a Gaussian disturbance to the actor's parameters at 110 seconds. The plant is initiated at a trimmed state at $(V_{tas}, H) = (90\ \frac{m}{s}, 2\ km)$ and the longitudinal and lateral controller at their state after the initial online training phase. Reference signals are represented by black, dashed lines.

V. Conclusion

The design and analysis of a RL adaptive flight controller for a CS-25 class aircraft are presented. The adaptive, learning controller is successfully implemented for a full-scale, high-fidelity aircraft simulation. It is demonstrated that the proposed framework is able to learn a near-optimal control policy online without a priori knowledge of the system dynamics nor an offline training phase. The results reveal that the proposed target critic increases the learning stability, which is vital for the reliable operation of the aircraft. Through the simulation of representative flight profiles, the results indicate that the learning controller is able to generalize and operate the aircraft in not previously encountered flight regimes as well as identify and adapt to unforeseen changes to the aircraft's dynamics. The framework proposed in this study mitigates the current limitation of ACDs that require an offline training phase expanding the application of such algorithms to applications for which no accurate system model is available, nor readily identifiable. This is especially interesting for future autonomous applications. Further investigation is recommended into the design of information-based online system identification and the execution of a flight test with the proposed framework on the PH-LAB research aircraft.

References

[1] Woods, D. D., "The risks of autonomy: Doyle's catch," *Journal of Cognitive Engineering and Decision Making*, Vol. 10, No. 2, 2016, pp. 131–133.

[2] Sghairi, M., Bonneval, A., Crouzet, Y., Aubert, J. J., and Brot, P., "Challenges in Building Fault-Tolerant Flight Control System for a Civil Aircraft," *IAENG International Journal of Computer Science*, Vol. 35, No. 4, 2008.

[3] Lombaerts, T., Oort, E. V., Chu, Q. P., Mulder, J. A., and Joosten, D., "Online Aerodynamic Model Structure Selection and Parameter Estimation for Fault Tolerant Control," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 3, 2010, pp. 707–723.

[4] Balas, G. J., "Flight Control Law Design: An Industry Perspective," *European Journal of Control*, Vol. 9, No. 2-3, 2003, pp. 207–226.

[5] Lane, S. H., and Stengel, R. F., "Flight control design using non-linear inverse dynamics," *Automatica*, Vol. 24, No. 4, 1988, pp. 471–483.

[6] da Costa, R. R., Chu, Q. P., and Mulder, J. A., "Reentry Flight Controller Design Using Nonlinear Dynamic Inversion," *Journal of Spacecraft and Rockets*, Vol. 40, No. 1, 2003, pp. 64–71.

[7] Sonneveldt, L., van Oort, E. R., Chu, Q. P., De Visser, C. C., Mulder, J. A., and Breeman, J. H., "Lyapunov-based Fault Tolerant Flight Control Designs for a Modern Fighter Aircraft Model," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Chicago, Illinois, 2009.

[8] Farrell, J., Sharma, M., and Polycarpou, M., "Backstepping-Based Flight Control with Adaptive Function Approximation," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 6, 2005, pp. 1089–1102.

[9] Sonneveldt, L., Chu, Q. P., and Mulder, J. A., "Nonlinear Flight Control Design Using Constrained Adaptive Backstepping," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 2, 2007, pp. 322–336.

[10] Sonneveldt, L., van Oort, E. R., Chu, Q. P., and Mulder, J. A., "Comparison of Inverse Optimal and Tuning Functions Designs for Adaptive Missile Control," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 1176–1182.

[11] Sonneveldt, L., van Oort, E. R., Chu, Q. P., and Mulder, J. A., "Nonlinear Adaptive Trajectory Control Applied to an F-16 Model," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 25–39.

[12] van Oort, E. R., Sonneveldt, L., Chu, Q. P., and Mulder, J. A., "Full-Envelope Modular Adaptive Control of a Fighter Aircraft Using Orthogonal Least Squares," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp. 1461–1472.

[13] Lu, P., van Kampen, E., de Visser, C., and Chu, Q. P., "Aircraft fault-tolerant trajectory control using Incremental Nonlinear Dynamic Inversion," *Control Engineering Practice*, Vol. 57, 2016, pp. 126–141.

[14] Sieberling, S., Chu, Q. P., and Mulder, J. A., "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 6, 2010, pp. 1732–1742.

[15] Simplício, P., Pavel, M. D., van Kampen, E., and Chu, Q. P., "An acceleration measurements-based approach for helicopter nonlinear flight control using incremental Nonlinear Dynamic Inversion," *Control Engineering Practice*, Vol. 21, No. 8, 2013, pp. 1065–1077.

[16] Acquatella, P., Falkena, W., Van Kampen, E., and Chu, Q. P., "Robust Nonlinear Spacecraft Attitude Control using Incremental Nonlinear Dynamic Inversion," *AIAA Guidance, Navigation, and Control Conference*, Minneapolis, Minnesota, 2012.

[17] Smeur, E. J. J., Chu, Q. P., and de Croon, G. C. H. E., "Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2016, pp. 450–461.

[18] Acquatella, P., van Kampen, E., and Chu, Q. P., "Incremental Backstepping for Robust Nonlinear Flight Control," *Proceedings of the EuroGNC*, Delft, The Netherlands, 2013.

[19] Wang, X., van Kampen, E., Chu, Q. P., and Lu, P., "Stability Analysis for Incremental Nonlinear Dynamic Inversion Control," *AIAA Guidance, Navigation, and Control Conference*, Kissimmee, Florida, 2018.

[20] Grondman, F., Looye, G., Kuchar, R. O., Chu, Q. P., and van Kampen, E., "Design and Flight Testing of Incremental Nonlinear Dynamic Inversion-based Control Laws for a Passenger Aircraft," *AIAA Guidance, Navigation, and Control Conference*, Kissimmee, Florida, 2018.

[21] Schultz, W., Dayan, P., and Montague, P. R., "A Neural Substrate of Prediction and Reward," *Science*, Vol. 275, No. 5306, 1997, pp. 1593–1599.

[22] Sutton, R. S., and Barto, A. G., *Reinforcement learning: An introduction*, 2nd ed., A Bradford Book, 2018.

[23] Powell, W. B., *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, John Wiley & Sons, 2007.

[24] Si, J., Barto, A. G., Powell, W. B., and Wunsch, D., *Handbook of Learning and Approximate Dynamic Programming*, Wiley-IEEE Press, 2004.

[25] Bertsekas, D. P., Homer, M. L., Logan, D. A., Patek, S. D., and Sandell, N. R., "Missile defense and interceptor allocation by neuro-dynamic programming," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 30, No. 1, 2000, pp. 42–51.

[26] Ferrari, S., and Stengel, R. F., "Online Adaptive Critic Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 5, 2004, pp. 777–786.

[27] Enns, R., and Si, J., "Helicopter trimming and tracking control using direct neural dynamic programming," *IEEE Transactions on Neural Networks*, Vol. 14, No. 4, 2003, pp. 929–939.

[28] van Kampen, E., Chu, Q. P., and Mulder, J. A., "Continuous Adaptive Critic Flight Control Aided with Approximated Plant Dynamics," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, Virigina, 2006.

[29] Zhou, Y., van Kampen, E., and Chu, Q. P., "Launch Vehicle Adaptive Flight Control with Incremental Model Based Heuristic Dynamic Programming," *68th International Astronautical Congress (IAC)*, Adelaide, Australia, 2017.

[30] Zhou, Y., van Kampen, E., and Chu, Q. P., "Incremental model based online dual heuristic programming for nonlinear adaptive control," *Control Engineering Practice*, Vol. 73, 2018, pp. 13–25.

[31] Zhou, Y., van Kampen, E., and Chu, Q., "Nonlinear adaptive flight control using incremental approximate dynamic programming and output feedback," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2018, pp. 493–496.

[32] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D., "Human-level control through deep reinforcement learning," *Nature*, Vol. 518, No. 7540, 2015, pp. 529–533.

[33] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D., "Continuous Control with Deep Reinforcement Learning," *International Conference on Learning Representations (ICLR)*, 2016.

[34] Zhou, Y., van Kampen, E., and Chu, Q. P., "Nonlinear Adaptive Flight Control Using Incremental Approximate Dynamic Programming and Output Feedback," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 493–496.

[35] Sadhukhan, D., and Feteih, S., "F8 neurocontroller based on dynamic inversion," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 1, 1996, pp. 150–156.

[36] Napolitano, M. R., and Kincheloe, M., "On-line learning neural-network controllers for autopilot systems," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 5, 1995, pp. 1008–1015.

[37] Kim, B. S., and Calise, A. J., "Nonlinear Flight Control Using Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 26–33.

[38] Calise, A. J., "Neural networks in nonlinear aircraft flight control," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 11, No. 7, 1996, pp. 5–10.

[39] Ha, C. M., "Neural networks approach to AIAA aircraft control design challenge," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 4, 1995, pp. 731–739.

[40] Balakrishnan, S. N., and Biega, V., "Adaptive-Critic-Based Neural Networks for Aircraft Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, 1996, pp. 893–898.

[41] Prokhorov, D. V., and Wunsch, D. C., "Adaptive critic designs," *IEEE Transactions on Neural Networks*, Vol. 8, No. 5, 1997, pp. 997–1007.

[42] Srouji, M., Zhang, J., and Salakhutdinov, R., "Structured Control Nets for Deep Reinforcement Learning," *arXiv preprint arXiv:1802.08311*, 2018.

[43] van den Hoek, M. A., de Visser, C. C., and Pool, D. M., "Identification of a Cessna Citation II Model Based on Flight Test Data," *Advances in Aerospace Guidance, Navigation and Control*, Springer International Publishing, Cham, 2018, pp. 259–277.

[44] Hägglund, T., "Recursive Estimation of Slowly Time-Varying Parameters," *IFAC Proceedings Volumes*, Vol. 18, No. 5, 1985, pp. 1137–1142.

[45] Kulhavý, R., "Restricted exponential forgetting in real-time identification," *Automatica*, Vol. 23, No. 5, 1987, pp. 589–600.

[46] Cao, L., and Schwartz, H. M., "A novel recursive algorithm for directional forgetting," *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251)*, Vol. 2, IEEE, 1999, pp. 1334–1338.

[47] Saelid, S., and Foss, B., "Adaptive controllers with a vector variable forgetting factor," *The 22nd IEEE Conference on Decision and Control*, IEEE, 1983, pp. 1488–1494.

[48] Parkum, J. E., Poulsen, N. K., and Holst, J., "Selective Forgetting in Adaptive Procedures," *IFAC Proceedings Volumes*, Vol. 23, No. 8, 1990, pp. 137–142.

[49] Klein, V., and Morelli, E. A., *Aircraft System Identification: Theory and Practice*, American Institute of Aeronautics and Astronautics, 2006.

[50] Mannucci, T., van Kampen, E., de Visser, C., and Chu, Q., "Safe exploration algorithms for reinforcement learning controllers," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 4, 2018, pp. 1069–1081.

[51] Lu, P., Van Eykeren, L., van Kampen, E., de Visser, C., and Chu, Q., "Double-model adaptive fault detection and diagnosis applied to real flight data," *Control Engineering Practice*, Vol. 36, 2015, pp. 39–57.

[52] Looye, G., and Joos, H., "Design of robust dynamic inversion control laws using multi-objective optimization," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, Virigina, 2001, p. 4285.