Flight Testing Reinforcement Learning based Online Adaptive Flight Control Laws on CS-25 Class Aircraft

Konatala, R.; Milz, Daniel ; Weiser, Christian ; Looye, Gertjan H.N.; van Kampen, E.

DOI
10.2514/6.2024-2402

Publication date
2024

Document Version
Final published version

Published in
Proceedings of the AIAA SCITECH 2024 Forum

AIAA SciTech Forum
8-12 January 2024, Orlando, FL
AIAA SCITECH 2024 Forum

10.2514/6.2024-2402

Flight Testing Reinforcement Learning based Online Adaptive Flight Control Laws on CS-25 Class Aircraft

Ramesh Konatala[*], Daniel Milz[†], Christian Weiser [‡], Gertjan Looye [§], and E. van Kampen [¶]
German Aerospace Center (DLR), 82234 Weßling, Germany
Delft University of Technology, P.O. Box 5058, 2626HS Delft, The Netherlands

Unforeseen failures during flight can lead to Loss of Control In-Flight, a significant cause of fatal aircraft accidents worldwide. Current offline synthesized flight control methods have limited capability to recover from failures, due to their limited adaptability. Incremental Approximate Dynamic Programming (iADP) control is a model-agnostic online adaptive control method, which integrates an online identified locally linearized incremental model, with a Reinforcement Learning (RL) based optimization technique to minimize an infinite horizon quadratic cost-to-go. A key challenge for adopting these self-learning flight control methods is validation through flight testing. This paper presents the iADP flight control law design for CS-25 class aircraft to achieve rate control. It outlines the controller evaluation strategy, controller integration, verification & validation procedures, and a discussion on flight test results. To the author's understanding, this flight test marks the world's first demonstration of an online RL based automatic flight control system for this aircraft category, demonstrating real-time learning and adaptation capabilities to aircraft configurations.

Nomenclature

$\hat{\bullet}$	=	Measured Value
$\tilde{\bullet}$	=	Estimated Value
$\bullet^T$	=	Matrix Transform
$\bullet^\dagger$	=	Matrix Pseudo Inverse
$\mathbf{X}, x$	=	Vector, Scalar
$\otimes$	=	Kronecker Product
$\tilde{F}$	=	State Transition Matrix estimate
$\tilde{G}$	=	Control Effectiveness Matrix estimate
Φ	=	Covariance Matrix
γ_{RLS}	=	Forgetting factor of the RLS algorithm
c_t	=	One-step quadratic cost in error and control input
$\tilde{P}$	=	Kernel Matrix
γ	=	Discount factor in Bellman's Equation
V_π	=	Bellman's Value Function estimate
π	=	Control Policy
$\delta_a, \delta_e, \delta_r$	=	Aileron, Elevator and Rudder deflections, respectively
p, q, r	=	Roll, Pitch and Yaw Angular Rates
ϕ, θ, ψ	=	Roll, Pitch and Yaw Angles
α, β	=	Angle of attach and Sideslip Angle
V_{TAS}, h	=	True Airspeed and Altitude

[*] Research Associate, ramesh.konatala@dlr.de, Department of Aircraft System Dynamics

[†] Research Associate, daniel.milz@dlr.de, Department of Aircraft System Dynamics

[‡] Research Associate, christian.weiser@dlr.de, Department of Aircraft System Dynamics

[§] Head of Department, gertjan.looye@dlr.de, Department of Aircraft System Dynamics

[¶] Assistant Professor, E.vanKampen@tudelft.nl, Faculty of Aerospace Engineering, Control and Simulation Division, Delft University of Technology

I. Introduction

Loss of Control in-flight(LOC-I) is an off-nominal flying condition, where the aircraft deviates from the normal flight envelope and is a leading cause of accidents in commercial aviation[1]. With the increasing trend towards autonomous and complex systems, one can expect an increasing trend of such LOC-I incidents unless proactive measures are taken. Developing an integrated fault-tolerant resilient Flight Control Law (FCL) is imperative to enhance safety under off-nominal conditions, addressing parametric failures and abnormal flight scenarios. Main challenges in designing such a controller include: low confidence on the aircraft model post failure, which degrades the model dependent controller performance, non-linearities in the model post failure, and the need for rapid adaptation of the controller to restore the aircraft within the safe flight envelope.

Nonlinear Dynamic Inversion(NDI) based control design involves inverting the nonlinear dynamics, rendering a system which can then be controlled with a linear controller, thus addressing the non-linearity issue for flight control. Incremental Nonlinear Dynamic Inversion(INDI) is another popular method, which is less dependent on a complete aerodynamic model, but needs local control derivatives with feedback of accelerations and actuator positions[2][3][4]. Adaptive version of these dynamic inversion methods holds promise in addressing fault recovery, but they are limited during off-nominal conditions due to their reliance on an accurate state model for inversion and a control effectiveness model. These models might be difficult to estimate with confidence during off-nominal flight conditions, necessitating a model agnostic approach. Self-learning Adaptive Flight Control System (FCS) algorithms were initially tested in the 1960s on the X-15 research aircraft[5]. Some of the open challenges in realizing adaptive FCS include sample efficiency and convergence, controller robustness, interpretability of the controller's adaptive mechanism[6].

Fig. 1 Cessna Citation II (PH-LAB) Research Aircraft
Captured by Alan Wilson. Image licensed under CC BY-SA 2.0.

RL, a bio-inspired machine learning approach, has been used for adaptive flight control since the early 2000s, e.g., the work from Enns and Si on helicopter control using Neuro-dynamic programming [7] or the work from Ferrari and Stengel on applying RL for control of a business jet type of aircraft [8]. An advantage of RL is that it can be used as model-free controller, meaning that no information about the plant that is to be controlled has to be known before the start of training. Another advantage is that it is by definition an adaptive controller, and hence can be used when online adaptation is required, for example after a fault or failure of a part of the aircraft. More recent applications of RL to flight control can be seen in [9] and [10], where RL controllers are designed that make use of an incremental model of the plant, which is identified online. A key challenge which limits the adoption of RL based methods for flight control is validation through flight testing on a CS-25 class aircraft, which could aid in certification of RL based FCL for Fault Tolerance [11].

Although several variants of RL based FCL's were developed[12][10], practical Verification & Validation (V&V) constraints guided the choice of this RL based FCL design. The following desirable features for the control method are defined as follows: a simpler control design strategy with less and interpretable learning parameters, ability to adapt in real-time to fast changing nonlinear dynamics of aircraft in case of a failure, an algorithm which is sample efficient [13] [14] and fast to converge on the conventional Flight Control Computer (FCC). In this regard, iADP based FCS[15] is a good choice amongst the RL based FCS with practical interests. This algorithm identifies a local linearized incremental model online, to estimate and minimize an infinite horizon quadratic cost-to-go, exclusively using the collected aircraft state data [16]. The iADP algorithm demonstrated effective online adaptation for a F-16 aircraft model, with good tracking performance both in normal and failure conditions [17].

The main contributions of this paper are, RL based iADP FCS Design for a CS-25 Class aircraft and controller's validation through flight tests on the PH-LAB Research Aircraft shown in fig. 1. The outcome from the flight test campaigns is detailed viz., the ability of the controller to capture pitch and roll rate tasks without a priori knowledge of aircraft model or any pretraining of the controller, stable continuous learning of the controller, adaptability of the controller to aircraft configurations assessed by comparing the adapting parameters across different configurations, and finally a discussion on observed challenges of some flight test trials. The technical scope of this study is limited to V&V excluding the interaction of pilot with the adaptive system.

The structure of the paper is organized as follows: Section II briefly covers the background information on RL and the methodology of iADP algorithm, and extensions of this algorithm for flight control. Section III contains the iADP FCL Architecture Design for the Cessna Citation-II Aircraft, FCL Evaluation Strategy, Software and Hardware Integration of the controller, and a summary of V&V procedures for the FCL Clearance for Flight tests. Section IV delves into the validation of controller functionality, focusing extensively on the results obtained from the flight test campaign. This section details the defined Experimental Objectives and Setup, a summary of the Control Law Flight Performance followed by discussion on flight testing experience, challenges faced, features and limitations of the iADP FCL . Lastly, section V includes concluding remarks, and a note on how this research could aid aviation safety.

II. Theory of Incremental Approximate Dynamic Programming

Consider a nonlinear dynamical system, which obeys the memoryless Markov property, the concept that a future state is independent of the preceding states given the current state. This system is represented in discrete general form as,

$$\begin{aligned} \mathbf{x}_{t+1} &= f(\mathbf{x}_t, \boldsymbol{\delta}_t) \\ \mathbf{y}_t &= h(\mathbf{x}_t) \end{aligned} \tag{1}$$

Here, $\mathbf{x}_t$ is the system state, $\boldsymbol{\delta}_t$ is the control input, and $\mathbf{y}_t$ represents output measurements. The behavior of the system is determined by knowing the states and control inputs. The controller's objective is to generate the control input for any system state $\mathbf{x}_t$ in a way that steers the system to a desired state. In other words, the controller aims to find the mapping from the current state and a desired state to the control actions, where the desired state depends on the control objective (e.g., 0 for regulation or a desired trajectory $\mathbf{x}_t^r$ for tracking). This control objective can be formulated as a RL problem, involving an agent interacting with an environment. The agent acts on the environment, and in return, the environment provides feedback through observations or a state, along with a reward signal or cost. The objective of the agent is to gradually improve its decision-making to maximize rewards or minimize costs over time. This mapping from the system's states to actions is denoted by π as follows.

$$\boldsymbol{\delta}_t = \pi(\mathbf{x}_t) \tag{2}$$

The RL agent's goal is to find an optimal policy π^*, minimizing the expected future cost (or maximizing reward) based on the underlying objective. The discounted infinite horizon cost or cost-to-go measures the sum of future costs incurred by the dynamic system over time and is given by,

$$\begin{aligned} V_\pi(\mathbf{x}_t) &= \sum_{i=t}^{\infty} \gamma^{i-t} c_i \\ &= c_t + \gamma V_\pi(\mathbf{x}_{t+1}) \end{aligned} \tag{3}$$

Here, c_t is the one-step cost observed from the environment, and $\gamma \in (0, 1)$ is the discount factor that discounts future costs. The Cost-to-go ($V_\pi(\mathbf{x}_t)$) is referred to as the value function, providing a measure of the value of being in a state $\mathbf{x}_t$ and the system choosing actions from $t \longrightarrow \infty$ according to the control policy (π) as per eq. (3). Bellman's Equation from Dynamic Programming[18] is used to evaluate this value function estimate in a step known as Policy Evaluation. The policy is then improved with respect to this current value estimate, aiming to minimize this value function. This improvement step is called Policy Improvement. The ultimate goal is to converge to the optimal value function(V^*) and an optimal policy (π^*) using Dynamic Programming.

To address the "Curse of Dimensionality" issue [19], the approach involves generalizing the value function from a restricted subset of visited states to an approximate value function applicable across a broader range. One method to achieve this generalization is by considering the value function to be quadratic in the state, as outlined below,

$$V_\pi(\mathbf{x}_t) = \mathbf{x}_t^T P \mathbf{x}_t \tag{4}$$

The kernel matrix P is defined to be positive definite to ensure that the cost function remains convex and positive. The one-step cost function c_t can be defined based on the operational requirements of the system. For instance, it could be a quadratic function in tracking error or a quadratic function in states for a regulation task [20]. Drawing inspiration from optimal control theory, one might include a quadratic term in the control input to operate the system with minimal cost expenditure. For tracking a reference command, the following one-step cost function is defined,

$$c_t = (\mathbf{y}_t - \mathbf{y}_t^r)^T Q \underbrace{(\mathbf{y}_t - \mathbf{y}_t^r)}_{\mathbf{e}_t} + \boldsymbol{\delta}_t^T R \boldsymbol{\delta}_t \tag{5}$$

where $Q = Q^T \geq 0$, $R = R^T \geq 0$ are respectively the state weighting matrix and input weighting matrix, and $\boldsymbol{\delta}_t$ is the control input, $\mathbf{y}_t$ is the output and $\mathbf{y}_t^r$ is a reference to be tracked by the system. For the value function to predict the cost related to reference dynamics, the state vector $\mathbf{x}_t$ is augmented with the state of reference dynamics as follows,

$$\mathbf{X}_t = \begin{pmatrix} \mathbf{x}_t \\ \mathbf{x}_t^r \end{pmatrix} \tag{6}$$

Using eq. (4), eq. (5), eq. (6), and eq. (3), the Bellman Equation for Linear Quadratic Tracking task is written as,

$$\mathbf{X}_t^T P \mathbf{X}_t = (\mathbf{y}_t - \mathbf{y}_t^r)^T Q (\mathbf{y}_t - \mathbf{y}_t^r) + \boldsymbol{\delta}_t^T R \boldsymbol{\delta}_t + \gamma \mathbf{X}_{t+1}^T P \mathbf{X}_{t+1} \tag{7}$$

The prediction of $\mathbf{X}_{t+1}$ or the knowledge of the system and reference dynamics is necessary to estimate $V_\pi(\mathbf{X}_{t+1})$ [21]. To maintain the model-free nature of the controller methods, one can estimate this prediction using an online Recursive Least Squares (RLS) approach, thus enabling online adaptability [22] [23]. The discount factor γ in eq. (7) provides regularization, by discounting costs incurred further in the future, accommodating uncertainty in the future and model prediction. This makes the value function estimate less sensitive to errors in model estimates.

Equation (7) is valid for linear systems, making it less suitable for nonlinear flight control. The value function can be modified to handle non-linearities using an incremental model representation of the original model. Linearizing the nonlinear discrete system in eq. (1) around $(\mathbf{X}_{t-1}, \boldsymbol{\delta}_{t-1})$, by taking the first-order Taylor series expansion, the nonlinear discretized system is expressed as eq. (8):

$$\mathbf{X}_{t+1} \approx \overbrace{f(\mathbf{X}_{t-1}, \boldsymbol{\delta}_{t-1})}^{\mathbf{X}_t} + \overbrace{\frac{\partial f(X_t, \delta_t)}{\partial \mathbf{X}_t}}^{F_t}\Bigg|_{\mathbf{X}_{t-1}, \boldsymbol{\delta}_{t-1}} (\mathbf{X}_t - \mathbf{X}_{t-1}) + \overbrace{\frac{\partial f(X_t, \delta_t)}{\partial \boldsymbol{\delta}_t}}^{G_t}\Bigg|_{\mathbf{X}_{t-1}, \boldsymbol{\delta}_{t-1}} (\boldsymbol{\delta}_t - \boldsymbol{\delta}_{t-1}) \tag{8}$$

$$\Rightarrow \Delta \mathbf{X}_{t+1} \approx F_t \Delta \mathbf{X}_t + G_t \Delta \delta_t \tag{9}$$

F_t is the system transition matrix and G_t is the control effectiveness matrix in incremental form. This regression model represented by $(\tilde{F}_t, \tilde{G}_t)$ can be identified using RLS techniques, providing a Linear Time Variant(LTV) incremental model approximation. An additional benefit of employing the incremental model is its reduced sensitivity to sensor biases and trim values, compared to the linear model.

By combining the Incremental model in eq. (9) and the Bellman Equation using value function approximation in eq. (7), the Bellman value function is rewritten as eq. (10) for nonlinear systems.

$$\begin{aligned} \mathbf{X}_t^T P \mathbf{X}_t = &(\mathbf{y}_t - \mathbf{y}_t^r)^T Q (\mathbf{y}_t - \mathbf{y}_t^r) + (\boldsymbol{\delta}_{t-1} + \Delta \boldsymbol{\delta}_t)^T R (\boldsymbol{\delta}_{t-1} + \Delta \boldsymbol{\delta}_t) \\ &+ \gamma (\mathbf{X}_t + \tilde{F}_t \Delta \mathbf{X}_t + \tilde{G}_t \Delta \boldsymbol{\delta}_t)^T P (\mathbf{X}_t + \tilde{F}_t \Delta \mathbf{X}_t + \tilde{G}_t \Delta \boldsymbol{\delta}_t) \end{aligned} \tag{10}$$

By Bellman's optimality principle, an optimal control law or policy can be obtained by minimizing this cost-to-go with respect to the incremental control input($\Delta \boldsymbol{\delta}_t$) as in eq. (11),

$$\Delta \boldsymbol{\delta}_t = -(R + \gamma \tilde{G}_t^T \tilde{P}^{j+1} \tilde{G}_t)^{-1} [R \boldsymbol{\delta}_{t-1} + \gamma \tilde{G}_t^T \tilde{P}^{j+1} \mathbf{X}_t + \gamma \tilde{G}_t \tilde{P}^{j+1} \tilde{F}_t \Delta \mathbf{X}_t] \tag{11}$$

The Pseudo Code of the iADP method including the RLS algorithm for a full state feedback system is shown in fig. 2. This algorithm is suitable for a nonlinear system with the availability of full states for feedback. The extensions of iADP Control Algorithm for a nonlinear system, to follow time-varying references and handling system with partial observability of full state can be found in [20] and [16].

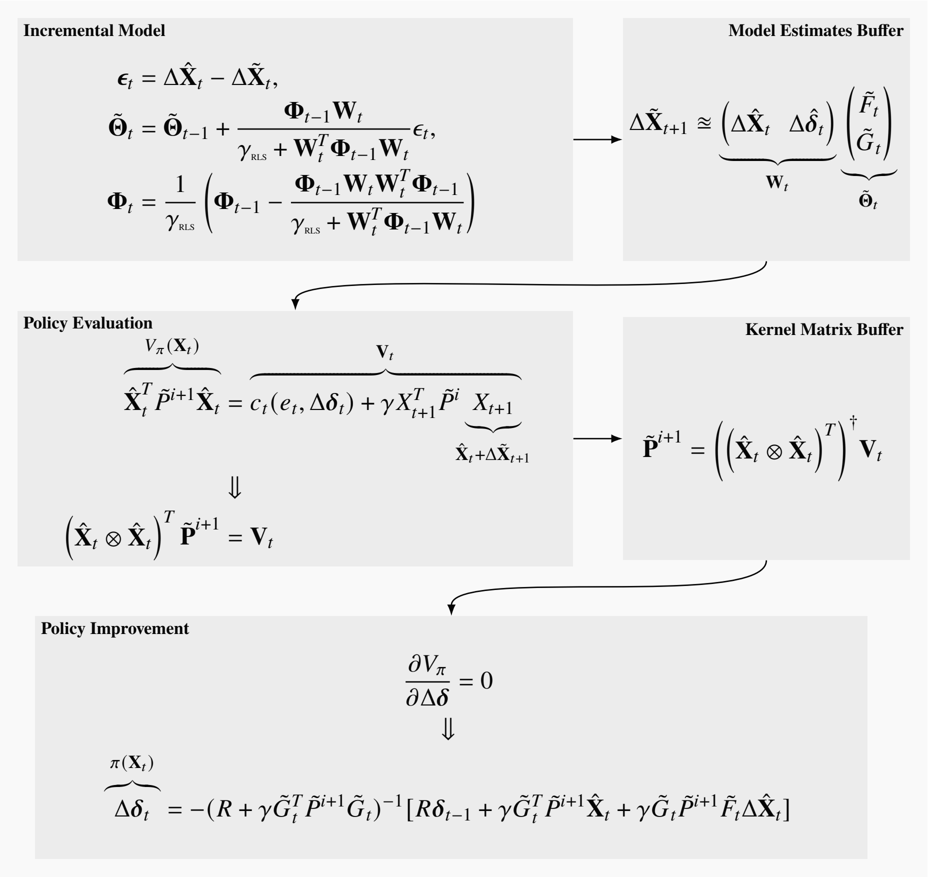

Incremental Model
$\boldsymbol{\epsilon}_t = \Delta\hat{\mathbf{X}}_t - \Delta\tilde{\mathbf{X}}_t,$
$\tilde{\mathbf{\Theta}}_t = \tilde{\mathbf{\Theta}}_{t-1} + \frac{\mathbf{\Phi}_{t-1}\mathbf{W}_t}{\gamma_{\text{RLS}} + \mathbf{W}_t^T\mathbf{\Phi}_{t-1}\mathbf{W}_t}\epsilon_t,$
$\mathbf{\Phi}_t = \frac{1}{\gamma_{\text{RLS}}}\left(\mathbf{\Phi}_{t-1} - \frac{\mathbf{\Phi}_{t-1}\mathbf{W}_t\mathbf{W}_t^T\mathbf{\Phi}_{t-1}}{\gamma_{\text{RLS}} + \mathbf{W}_t^T\mathbf{\Phi}_{t-1}\mathbf{W}_t}\right)$
Model Estimates Buffer
$\Delta\tilde{\mathbf{X}}_{t+1} \cong \underbrace{\begin{pmatrix}\Delta\hat{\mathbf{X}}_t & \Delta\hat{\boldsymbol{\delta}}_t\end{pmatrix}}_{\mathbf{W}_t}\underbrace{\begin{pmatrix}\tilde{F}_t \\ \tilde{G}_t\end{pmatrix}}_{\tilde{\mathbf{\Theta}}_t}$
Policy Evaluation
$\overbrace{\hat{\mathbf{X}}_t^T\tilde{P}^{i+1}\hat{\mathbf{X}}_t}^{V_\pi(\mathbf{X}_t)} = \overbrace{c_t(e_t, \Delta\boldsymbol{\delta}_t) + \gamma X_{t+1}^T\tilde{P}^i\underbrace{X_{t+1}}_{\hat{\mathbf{X}}_t+\Delta\tilde{\mathbf{X}}_{t+1}}}^{\mathbf{V}_t}$
$\Downarrow$
$\left(\hat{\mathbf{X}}_t \otimes \hat{\mathbf{X}}_t\right)^T\tilde{\mathbf{P}}^{i+1} = \mathbf{V}_t$
Kernel Matrix Buffer
$\tilde{\mathbf{P}}^{i+1} = \left(\left(\hat{\mathbf{X}}_t \otimes \hat{\mathbf{X}}_t\right)^T\right)^\dagger\mathbf{V}_t$
Policy Improvement
$\frac{\partial V_\pi}{\partial\Delta\boldsymbol{\delta}} = 0$
$\Downarrow$
$\overbrace{\Delta\boldsymbol{\delta}_t}^{\pi(\mathbf{X}_t)} = -(R + \gamma\tilde{G}_t^T\tilde{P}^{i+1}\tilde{G}_t)^{-1}[R\boldsymbol{\delta}_{t-1} + \gamma\tilde{G}_t^T\tilde{P}^{i+1}\hat{\mathbf{X}}_t + \gamma\tilde{G}_t\tilde{P}^{i+1}\tilde{F}_t\Delta\hat{\mathbf{X}}_t]$

Fig. 2 Pseudo Code for the iADP Algorithm. Refer to Nomenclature for the corresponding notation of the symbols used.

III. iADP based Flight Control System Design

A. Control Law Architecture

The iADP algorithm is used to design the FCL for the inner loop, tracking desired pitch and roll rate commands using three control surfaces: aileron, elevator, and rudder, as shown in fig. 3. As this FCL is solely sensor-based, signal processing of sensor measurements is performed to reduce the impact of noisy sensor signals. Smooth sensor measurements are obtained by low-pass filtering relevant aircraft states and actuator position measurements. Aerodynamic angles, namely angle of attack and sideslip angle, are acquired through a boom with attached vanes on the aircraft. Complementary filtering of these angles is executed by combining them with inertial-reference sensors. The signal processing block does not contain any knowledge of the aircraft model, ensuring a model-free and aircraft-independent inner loop control structure. The iADP controller computes the required incremental control input at each timestep, with the total control input determined by adding the previous control surface measurement to the evaluated control increment. Pilot-in-the-loop studies are excluded from this flight control design, and thus, the desired pitch and roll rate commands are generated automatically. However, the iADP controller is not aware of these reference commands a priori.

Direct control over the aircraft's attitude angles is absent, as the cost function does not consider attitude angle errors, as only the one step rate errors are fed back as the reward signal. Additionally, airspeed and altitude information is excluded from the state vector due to slower local variations, which could impact incremental model identification and subsequent value function approximation.

To mitigate the effect of airspeed and altitude variations, reference commands are chosen to restrict variations in airspeed to smaller values. Decoupled longitudinal and lateral controllers are designed: the longitudinal control loop tracks a pitch rate command using the elevator, while the lateral control loop tracks a roll rate command using the aileron

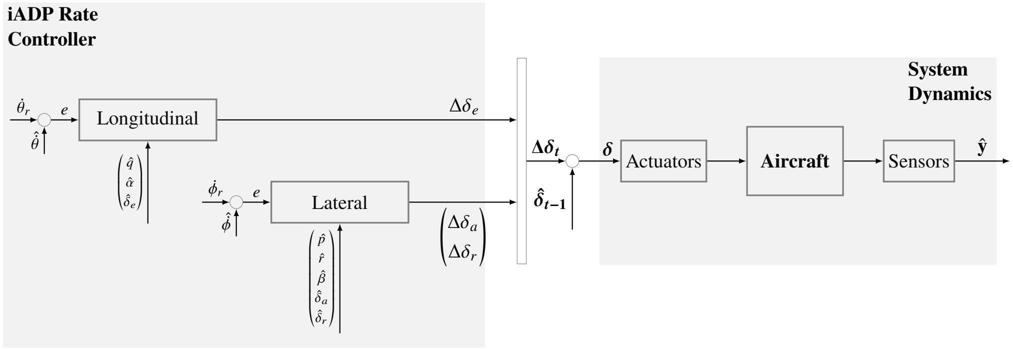

iADP Rate Controller
$\dot{\theta}_r$
e
Longitudinal
$\hat{\theta}$
$\begin{pmatrix}\hat{q}\\ \hat{\alpha}\\ \hat{\delta}_e\end{pmatrix}$
$\Delta\delta_e$
$\dot{\phi}_r$
e
Lateral
$\hat{\phi}$
$\begin{pmatrix}\hat{p}\\ \hat{r}\\ \hat{\beta}\\ \hat{\delta}_a\\ \hat{\delta}_r\end{pmatrix}$
$\begin{pmatrix}\Delta\delta_a\\ \Delta\delta_r\end{pmatrix}$
$\boldsymbol{\Delta\delta_t}$
$\boldsymbol{\hat{\delta}_{t-1}}$
$\boldsymbol{\delta}$
System Dynamics
Actuators
Aircraft
Sensors
$\hat{\mathbf{y}}$

Fig. 3 Incremental Approximate Dynamic Programming based Flight Control Law (FCL) Architecture for Inner Loop Rate Tracking. Decoupled Longitudinal and Lateral Reinforcement Learning (RL) Controllers for PH-LAB Aircraft.

and/or rudder. Only one axis is controlled during a maneuver, which means during the evaluation of the longitudinal task, aileron and rudder maintain the trimmed control input and vice versa.

B. iADP Control Law Evaluation Strategy

The iADP controller operates in real-time through three phases: Model Learning, Controller Training, and Controller Assessment, as illustrated in fig. 4.

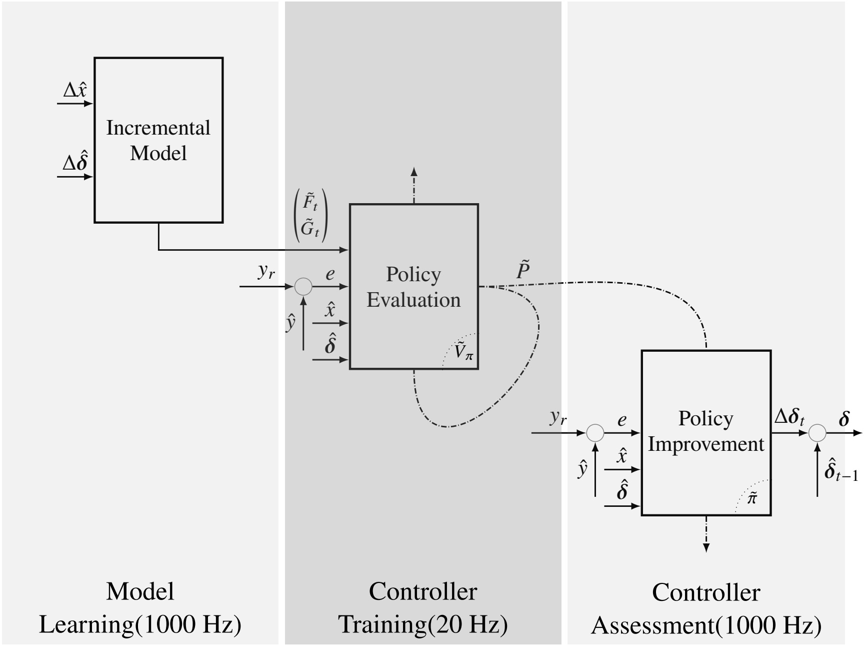
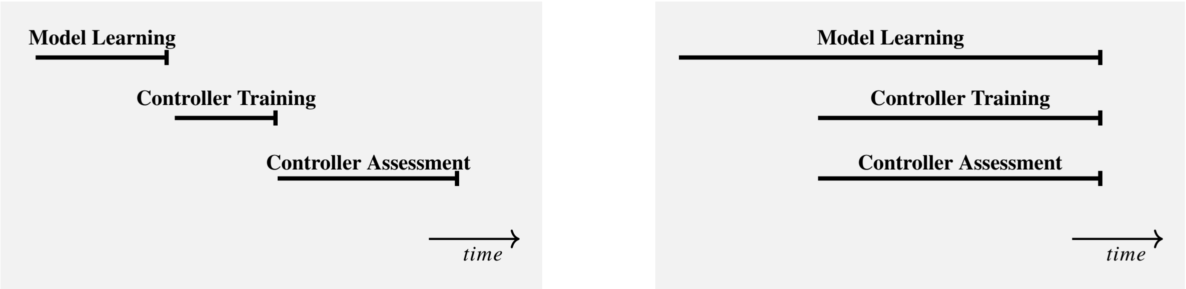

$\Delta\hat{x}$
$\Delta\hat{\boldsymbol{\delta}}$
Incremental Model
$\begin{pmatrix}\tilde{F}_t\\ \tilde{G}_t\end{pmatrix}$
y_r
e
$\hat{y}$
$\hat{x}$
$\hat{\boldsymbol{\delta}}$
Policy Evaluation
$\tilde{V}_\pi$
$\tilde{P}$
y_r
e
$\hat{y}$
$\hat{x}$
$\hat{\boldsymbol{\delta}}$
Policy Improvement
$\tilde{\pi}$
$\Delta\boldsymbol{\delta}_t$
$\boldsymbol{\delta}$
$\hat{\boldsymbol{\delta}}_{t-1}$
Model Learning(1000 Hz)
Controller Training(20 Hz)
Controller Assessment(1000 Hz)

Fig. 4 Structure of the Reinforcement Learning Agent of iADP Flight Control Law (FCL). Model Learning provides the latest model estimates using the RLS algorithm. Controller Training evaluates(V_π) the Control Policy using incremental model estimates and one-step Cost. Controller Assessment takes actions and improves Control Policy (π) based on policy evaluation. The frequency at which each subsystem on the Flight Control Computer runs is indicated below. Refer to Nomenclature for the corresponding notation of the symbols used.

During the Model Learning phase, the controller estimates incremental model parameters using the RLS approach. These estimates are fed to the controller training phase, where, using the latest available model estimates and control policy, the value function(cost-to-go) estimate is improved. This value function provides a measure of goodness of the underlying control policy. To enable smoothness in parameter update, the value function update is done batch wise considering data over a window of 'x' no of samples. The optimal window length is determined from offline simulation analysis and based on how much of computational load could the FCC handle using HIL ground tests. The Controller Assessment phase evaluates the controller against a commanded reference signal, using the converged control parameters, obtained from the Controller Training phase.

Two experimental approaches are considered for controller evaluation as shown in fig. 5a and fig. 5b:

1) **Sequential Learning Approach (SLA)**: This approach runs each phase sequentially, starting with the Model Learning phase, followed by the Controller Training and Controller Assessment phases. The Model Learning phase is open loop and lasts for 20-25 seconds.
2) **Continuous Learning Approach (CLA)**: In this approach, all three phases run concurrently. The Controller updates its policy at every timestep using the latest model parameters from the Model Learning loop and controller parameters from the Controller Training loop. The Controller Assessment loop runs concurrently, allowing the controller to update the model and control along the commanded trajectory defined for the Controller Assessment. To facilitate the learning process, the model learning phase operates in an open loop for the first 20-25 seconds of the trial. Subsequently, it adjusts the model parameters, taking into account the influence of the controller in the loop.



(a) Sequential Learning

(b) Continuous Learning

Fig. 5 Comparison of Controller Learning Approaches

C. Controller Integration

Once the controller was designed, tuned, tested and validated in simulation, it is necessary to transfer the developed algorithms to the aircraft for in-flight testing. Accordingly, the iADP controller is initially exported into compatible C code to be embedded into the test framework and compiled on the test hardware. Subsequently, the generated code was transferred and validated on the FCC.

1. Software

Prior to exporting the control functions, the controller interfaces were ensured to match with the ones from the experimental FCC. Since the flight controller is developed in Simulink, the proprietary Mathworks tool Embedded Coder was used to generate standalone C code according to given specifications. After revising the automatically generated code to ensure correct export of the control functions, the code is embedded into the Delft University Environment for Communication and Activation (DUECA) real-time framework. DUECA is a middle-ware layer written in C++ and allows real-time implementation and communication of distributed systems [24]. It motivates modular design and ensures real-time synchronization and exchange of signals.

The code for the controller is put into a separate DUECA module, which handles the needed interfaces. Other connected modules include the servo position controller, sensor data acquisition, and the UDP sender. This last one is crucial during flight tests, facilitating real-time access to aircraft data and auxiliary controller outputs. This enabled direct evaluation of experiment results and changes to the flight test plan if needed.

DUECA ensures real-time synchronization through a master that controls all concurrently running modules at the base frequency. During this flight test, the system ran at 1000 Hz. This means, the controller has a window of at most 1 ms for performing one step in order to stay synchronized. This was a challenge for the iADP FCL. Especially the learning procedure, which includes large matrix operations, needed to be optimized to stay within this frame.

2. Hardware

The Cessna Citation 550 PH-LAB aircraft (shown in fig. 1) is jointly operated by the TU Delft and the Dutch Aerospace Center (NLR) and serves as a multi-functional research platform. The aircraft is certified in accordance with the EASA CS-25 specifications for large aircraft. The nominal Cessna Citation comes with a reversible FCS realized between the pilot's joke and pedals and an asymmetrically deflecting pair of ailerons, a symmetrically deflecting pair of elevators, and one rudder. Furthermore, an autopilot system is available which is realized by clutch-coupled servo motors on the control surface cables.

For testing the experimental flight control functions, the airplane is equipped with an experimental fly-by-wire (FBW) system, as detailed in [25], which has undergone exhaustive testing in order to be certified under EASA CS-25. This FBW system ensures that the flight control surfaces follow the desired actuator commands through feedback signals from actuator servos. The movements of the control surfaces are converted to electronic signals and the FCC determines these signals based on expected actuator response. FCC provides these signals to the servo amplifiers of the actuators that deflect the control surfaces. The experimental system runs on a dedicated computer, equipped with an Intel® Core™ i5-3550S quad-core processor, and running Ubuntu 20.04 LTS with real-time kernel (PREEMPT_RT). More details are given in [25, 26].

In addition to the experimental FBW, the research aircraft had an upgrade of several sensor systems. The Flight Test Instrument System consists of a data acquisition computer and a signal conditioning unit for processing information from sensors. For further details, the reader is advised to refer to [4, 25, 27]. The sensors used for the iADP testing are shown in table 1.

Table 1 Sensors used for the iADP flight control testing, with values from [4, 28, 29]

Sensor	Signal	Sampling freq. [Hz]	Est. Delay [ms]
AHRS	ϕ, θ	52	90
Hall Sensor	$\delta_a, \delta_e, \delta_r$	100	<1
Air data boom	α, β	100	100
Air data sensor	$V_{\text{TAS}}, V_{\text{CAS}}, h, \dot{h}$	16	300
IMU	$p, q, r, \dot{\phi}, \dot{\theta}, \dot{\psi}$	1000	15

Prior to conducting the flight tests, the generated code is examined through HIL testing, performed with the aircraft and experimental system on ground. The flight testing procedure with the PH-LAB aircraft is detailed in [30]. During these tests, both the aircraft hardware and FBW system are evaluated, along with the iADP controller code that is generated. The flight tests of the iADP controller are conducted with solely automatically generated references and without pilot commands.

D. Clearance of Control Laws through Verification & Validation (V&V)

Verification involves analyzing and testing processes to confirm that flight control algorithms operate as intended, ensuring accurate implementation in both software and hardware. Validation assesses the performance of FCL's against defined requirements, utilizing a set of criteria. The V&V plan for iADP FCS is designed to ensure that control design specifications are fulfilled. Verification primarily utilized offline simulation analysis, while validation involved Hardware-in-the-loop (HIL) and flight testing methods.

The verification process employs the PH-LAB aircraft model, with the controller operating without any prior knowledge of the model. It accomplishes tasks based solely on data observed from the aircraft model collected along simulated system trajectories in offline desktop simulations. The PH-LAB aircraft model used is a high-fidelity simulation model, considering relevant real-world phenomena and nonlinear effects, validated against flight test data from previous works[31][32]. This high-fidelity model includes the effects of different aircraft configurations, such as landing gear down and extended flaps. iADP flight controller adaptation to different aircraft configurations is verified using these models through simulations, before validation through flight test experiments.

iADP hyper-parameter tuning($\gamma_{\text{RLS}}, \gamma, Q, R$) is conducted using DLR's optimization tool Multi-Objective Parameter Synthesis (MOPS), a Matlab-based software for solving general-purpose parameter optimization problems[33]. The controller's robustness to model uncertainties is assessed through an anti-optimization routine using MOPS. This involves determining parameter combinations of model uncertainties leading to the worst-case performance (rate tracking error) of the iADP controller. Hyper-parameters are tuned against this worst-case model obtained from the previous step, and this loop is run iteratively. Furthermore, several unit tests are done through offline desktop simulation to ensure low sensitivity of the iADP controller to hyper-parameters, changes in operating conditions (velocity and altitude changes), aircraft configuration changes, and sensitivity to sensor dynamics (bias, noise, and delays).

The validation of the controller is performed through HIL ground tests and finally flight tests. Validation tasks include ensuring the FCC can handle the computational load of the recursive algorithms, verifying the signs of the controller commands on the ground, and conducting controller logging and real-time monitoring tool chain tests.

During flight tests, different subsystems of the iADP FCL are validated one after the other. The open-loop model learning phase is first validated by running only the RLS algorithm during preliminary flight test trials. The identified incremental model parameters are then validated against the model through post-flight data analysis. Once the model learning subsystem is validated, the controller training subsystem is validated by running model learning and controller training phases during another flight test trial. The converged controller parameters are verified through post-flight test data analysis. After clearing the model learning and controller training subsystems, the final control assessment validation took place. The V&V of the iADP FCL for this flight test campaign is detailed in [34]. This step-by-step clearance of different subsystems of iADP FCL ensured the analysis and rectification of any issues found with individual subsystems. This procedure ensured that the learning process can be interpretable for post-flight critical analysis, crucial for V&V of online adaptive FCL's.

IV. Flight Tests

Controller validation involves assessing the performance of the FCL's against predefined criteria. iADP FCL validation occurred through flight tests on the PH-LAB research Aircraft, conducted in November 2022 and August 2023, departing from Rotterdam The Hague Airport.

A. Experimental Objectives and Setup

Flight Control Design specifications act as a guide for control engineers, directing them in the design process to verify that the controller meets specific criteria [35]. For iADP FCL validation, the following FCS design specifications are formulated:

1) **Minimize Attitude Rate Tracking error** - Ensure the controller commands the aircraft to follow pitch and roll rates.
2) **Low Sensitivity to changes in Operating Conditions** - Ensure rate tracking in different operating conditions (Altitude and Velocity changes).
3) **Reproducible Results** - Ensure consistent controller behavior under similar conditions.
4) **Continuous Learning** - Ensure stable continuous learning over longer maneuvers.
5) **Adaptability to Aircraft Configurations** - Ensure rate tracking in different aircraft configurations.

Several factors influenced how the flight trial could be conducted. The dimensions of the available airspace for the tests were decided preflight, in consultation with pilots. Operating conditions for which the aircraft equipment is certified, in consultation with technicians, provided estimates on the boundaries of the flight envelope. These factors determined the higher limit on the trial's duration. The lower limit on trial duration was determined by the minimum time/samples required for the iADP algorithm to converge, determined through offline desktop simulations and HIL tests. To test adaptability, a set of feasible aircraft configurations were decided preflight. These configurations should be observable by the iADP controller and sufficiently alter the aircraft's dynamics to assess FCL's adaptability. For these tests, the higher limit on the airspeed is imposed by the fact that landing gear and flap maximum extension configurations can be deployed only below a certain airspeed.

Subsequently, flight test cards were prepared to facilitate in-flight communication. The command control station laptop served as an interface for communication with the FCC, enabling tasks like switching between controllers, engaging/disengaging, resetting, and changing tunable controller parameters. Another control station laptop monitored real-time controller performance by reading data from the FCC, enabling critical analysis and intervention if abnormalities were observed. The Interface Control Document (ICD) was updated to include parameters requiring tuning during flight, essential controller switches, signals to be logged onto the FCC.

B. iADP based Control Law Flight Performance

The table 2 provides a summary of all the flight test trials, offering a concise overview of the outcomes along with the operating condition at which the aircraft is trimmed before the beginning of the flight trial.

Table 2 Overview of the iADP Controller Flight Testing Campaigns. The trials are listed chronologically, and the trial ID follows the notation: N22 for November 2022, A23 for August 2023, F# indicates the flight test day, and T# indicates the trial number. The *Axis* indicates the actively controlled channel of the aircraft via the FBW during the trial. V_{TAS} represents True Airspeed, and h represents altitude. The *Outcome* column reflects whether the controller response aligned with the design specifications. *Config.* denotes Aircraft Configuration.

Trial ID	Axis	V_{TAS} [m/s]	h [m]	Brief Description	Outcome
N22-F2-T1	Pitch	101	3600	Oscillatory response; Convergence in Model prediction	✓
N22-F2-T2	Pitch	106	3650	Off-nominal Flight; Inverted Incremental Model Parameters	×
N22-F2-T3	Pitch	104	3650	First Success; Decent tracking; Slight Elevator oscillations	✓
N22-F2-T4	Pitch	105	3550	Inverted Controller Commands; Inverted Model Parameters	×
N22-F2-T5	Pitch	94	3500	Better tracking; Increased Elevator oscillations	✓
N22-F3-T1	Pitch	102	2100	Decent tracking; High Model Prediction error	✓
N22-F3-T2	Roll	91	2150	Oscillatory response; High Model Prediction error	×
N22-F3-T3	Roll	96	2000	Aircraft deviated from level flight post Model Learning phase	×
A23-F1-T1	Pitch	99	2750	Oscillatory response; Model Learning duration too short	✓
A23-F1-T2	Roll	101	2800	Deviated from level flight; Model Learning duration too short	×
A23-F1-T3	Pitch	102	2750	First success with Continuous Learning; Decent tracking	✓
A23-F1-T4	Pitch	101	2800	Reproducible Continuous Learning; Better tracking	✓
A23-F1-T5	Roll	100	2800	First success in Lateral with Continuous Learning	✓
A23-F1-T6	Roll	101	2800	Reproducible Continuous Learning; Good tracking response	✓
A23-F2-T1	Pitch	97	3050	Stable Continuous Learning; Decent tracking response	✓
A23-F2-T2	Roll	101	3050	Nominal Config.; Reproducible Continuous Learning	✓
A23-F2-T3	Roll	95	3100	Nominal Config.; Reproducible Continuous Learning	✓
A23-F2-T4	Roll	97	3100	Nominal Config.; Reproducible Continuous Learning	✓
A23-F2-T5	Roll	97	3050	Landing Gear Down Config. ; Stable Continuous Learning	✓
A23-F2-T6	Roll	97	3050	Flaps 15° Config. ; Stable Continuous Learning	✓
A23-F2-T7	Roll	98	3100	Flaps 40° Config. ; Slightly Oscillatory tracking	✓

SLA is adopted from trial 'N22-F2-T1' to 'A23-F1-T2' and all subsequent trials have adopted CLA. SLA takes precedence to identify and assess converged parameters, addressing issues such as the ideal open-loop aircraft identification input signal, determining acceptable phase durations for parameter convergence, refining the reference signals to ensure the aircraft stays within airspace limits. The controller parameters, including model parameters, are reset after each trial. Consequently, no information about the aircraft model or controller parameters from previous flight tests is retained or transferred to subsequent attempts. For stable parameter convergence in the RLS algorithm, we combine frequency-rich persistently exciting signals, with the calculated control input as shown in eq. (12). The need for the persistent excitation is further explained in [36] [37]. The selected persistent excitation signal for the iADP FCL is a sinusoidal signal with a small amplitude.

$$(\Delta\delta_t)_{cmd} = \Delta\delta_t + PE \tag{12}$$

Longitudinal Controller Assessment - Sequential Learning Approach

The first successful trial of the iADP algorithm for longitudinal rate control is presented in fig. 6. The left-aligned plots illustrate the model learning phase, an open-loop period where a 3211 maneuver is commanded by the elevator. The online RLS algorithm updates model parameters at each time step during this 20 second phase, with fixed parameters subsequently passed to the controller training phase. The choice of the 3211 signal is based on its proven effectiveness in previous system identification flight tests on the Citation aircraft. Control over the system's excited frequency can be achieved by adjusting the duration of individual step commands in the 3211 signal. The 3211 signal also serves the functionality of persistently exciting signal as mentioned in eq. (12). The forgetting factor (γ_{RLS}) is tuned using the Multi-Objective Parameter Synthesis tool (MOPS). Results comparing measured longitudinal states against predictions from the RLS algorithm show a good fit. Although model parameters seem to converge quickly, some fluctuations are observed, which are discussed later in this paper.

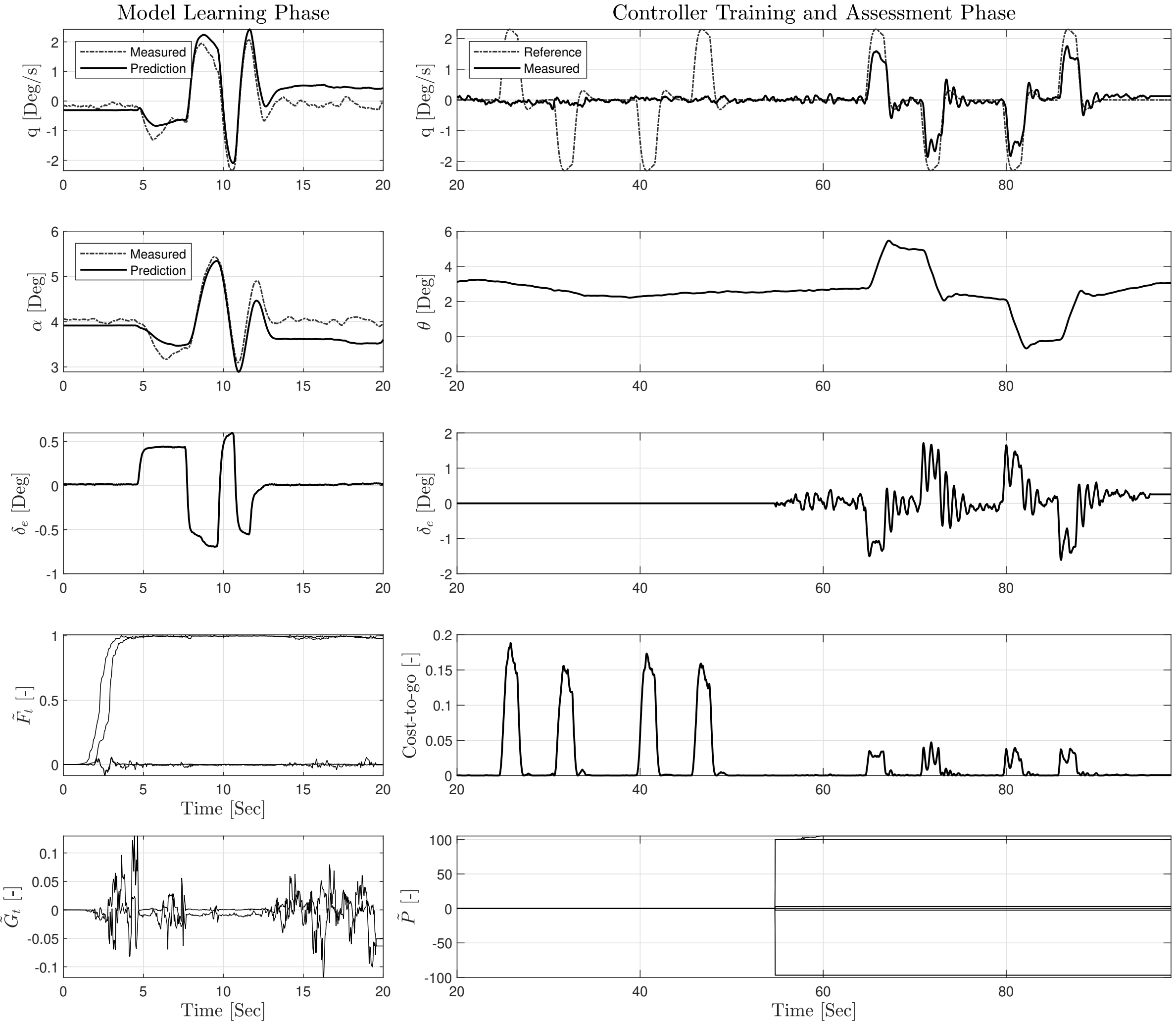


Fig. 6 Flight Test Data(Trial ID : N22-F2-T3), PH-LAB performing a Longitudinal maneuver: iADP Flight Control Law (FCL) designed for Pitch Rate Capture. Sequential Learning Approach (SLA) with fixed parameters post Model Learning and Controller Training.

The right-aligned plots depict results from the controller training and assessment phase. Parameters for this phase, including the discount factor (γ) and weighting matrices (Q and R), are tuned using MOPS tool [33]. The controller training phase lasted for 40 seconds(from 20 to 60 seconds), during which the controller loop is closed and internally a pitch rate reference command to evaluate the policy is generated. The controller along with model parameters estimated from model learning phase and observed one step error in pitch rate tracking, has to improve its estimates of cost-to-go function. Controller parameters are updated during a brief 5 second phase (55 to 60 seconds). This computationally intensive phase updates kernel matrix parameters, using data collected over a 20 second window(data from 35-55 seconds is used for the update from 55$^{\text{th}}$ second onwards). This loop is running at a much lower 20 Hz due to real-time constraints. After the controller training phase concludes, the parameters are fixed and passed to the subsequent Controller Assessment phase.

During the Controller Assessment phase, an internal pitch rate command is generated, and the controller's objective is to track this reference command. The results, from 60 to 100 seconds, show the aircraft effectively tracking a pitch rate command, which can also be interpreted as a reduction in the cost-to-go plot. The tracking performance improved with higher values of Q, but this made the controller response more oscillatory(from flight trial N22-F2-T5 in table 2).

Lateral Controller Assessment - Sequential Learning Approach

A similar strategy is planned to validate the lateral controller, to achieve roll rate captures using aileron and/or rudder control surfaces. Initially, a SLA is planned for validation. However, both trials N22-F3-T2 and N22-F3-T3, using a SLA to achieve roll rate capture, were unsuccessful. Post-flight test analysis revealed that the 3211 maneuver commanded by the aileron during the model learning phase did not yield a symmetric lateral response, causing the aircraft to deviate from level flight before controller parameters could converge. To mitigate this, the 3211 maneuver is replaced with doublets at the aileron, and the duration of the model learning and controller training phases were reduced. Another SLA was planned for trial A23-F1-T2 considering these changes. During this trial, although symmetric commands helped aircraft to maintain level flight post the model learning phase, the controller parameters could not converge during the controller training phase. This lead to the aircraft steering away from level flight, after the controller training phase. Although the model learning phase yielded good model predictions, a shorter controller training duration meant the controller parameters were fixed before they converged, affecting the controller's performance. No successful lateral controller validation tests were conducted with SLA, and all subsequent lateral controller tests were conducted only with CLA.

Lateral Controller Assessment - Continuous Learning Approach

Figure 7 and fig. 8 depict the flight test results from trial A23-F2-T2, focusing on achieving a roll rate tracking task with a CLA.

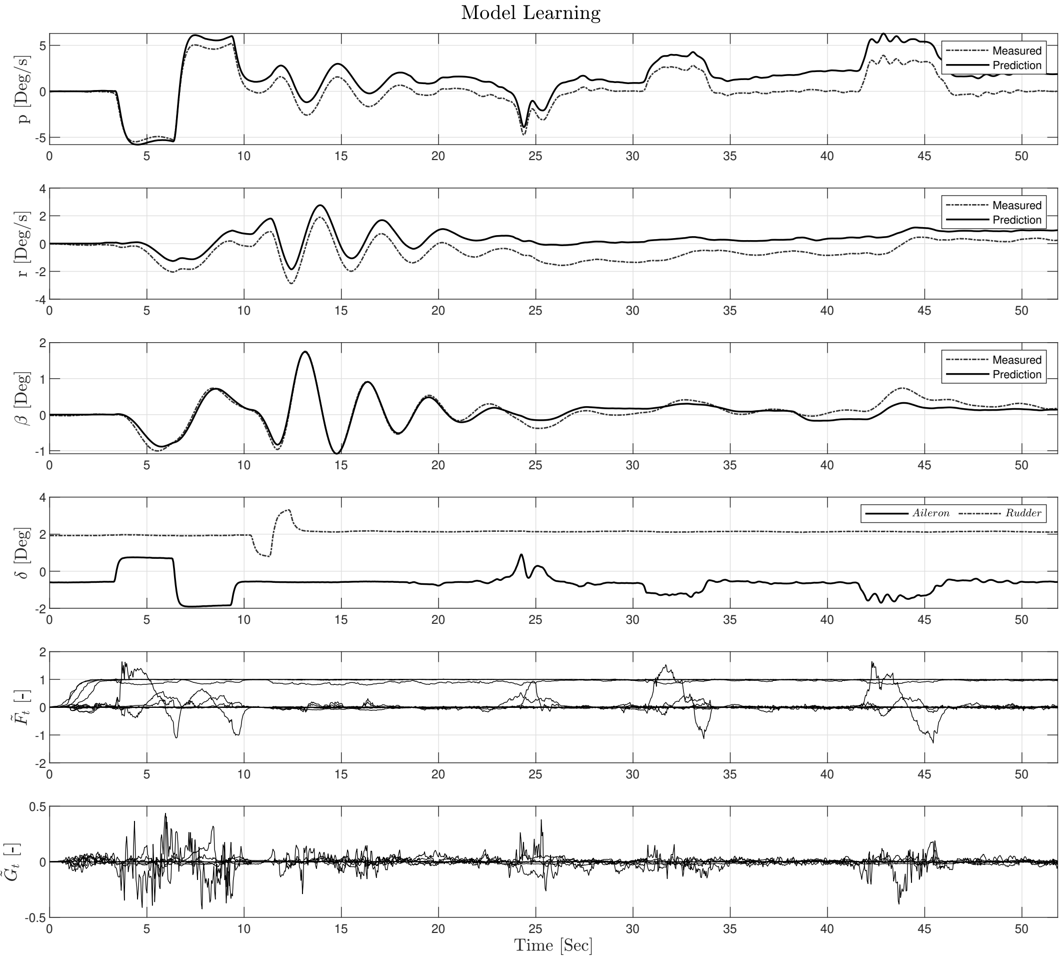


Fig. 7 Flight Test Data(Trial ID : A23-F2-T2), PH-LAB performing a Lateral maneuver: Plots compare the predictions from the Incremental model to the measured states. iADP Flight Control Law (FCL) designed for Roll Rate Capture. Continuous Learning Approach (CLA) with real-time parameter adaptation.

The controller's objective is to command the aircraft to follow a roll rate reference and demonstrate stable continuous learning capability. Figure 7 illustrates the performance of the incremental model identification stage. Doublets are initially commanded at the Aileron and Rudder to aid the identification process. A sinusoidal signal is superimposed on commanded aileron and rudder, to ensure RLS parameter convergence. When comparing the measured state variable with the prediction, the RLS algorithm appears to offer a good fit in state prediction throughout the maneuver. However, after 30 seconds, error in predicting certain states was observed to have an increasing trend, likely attributed to sharp control inputs occurring around the moment the control loop was closed at 20 seconds.

The performance of the controller training and assessment stage is presented in fig. 8. Comparing the reference to the measured roll rate output in the first row, a good tracking response is observed. The bottom two plots show the reduction in the cost-to-go estimate and the evolution of the controller parameters respectively.

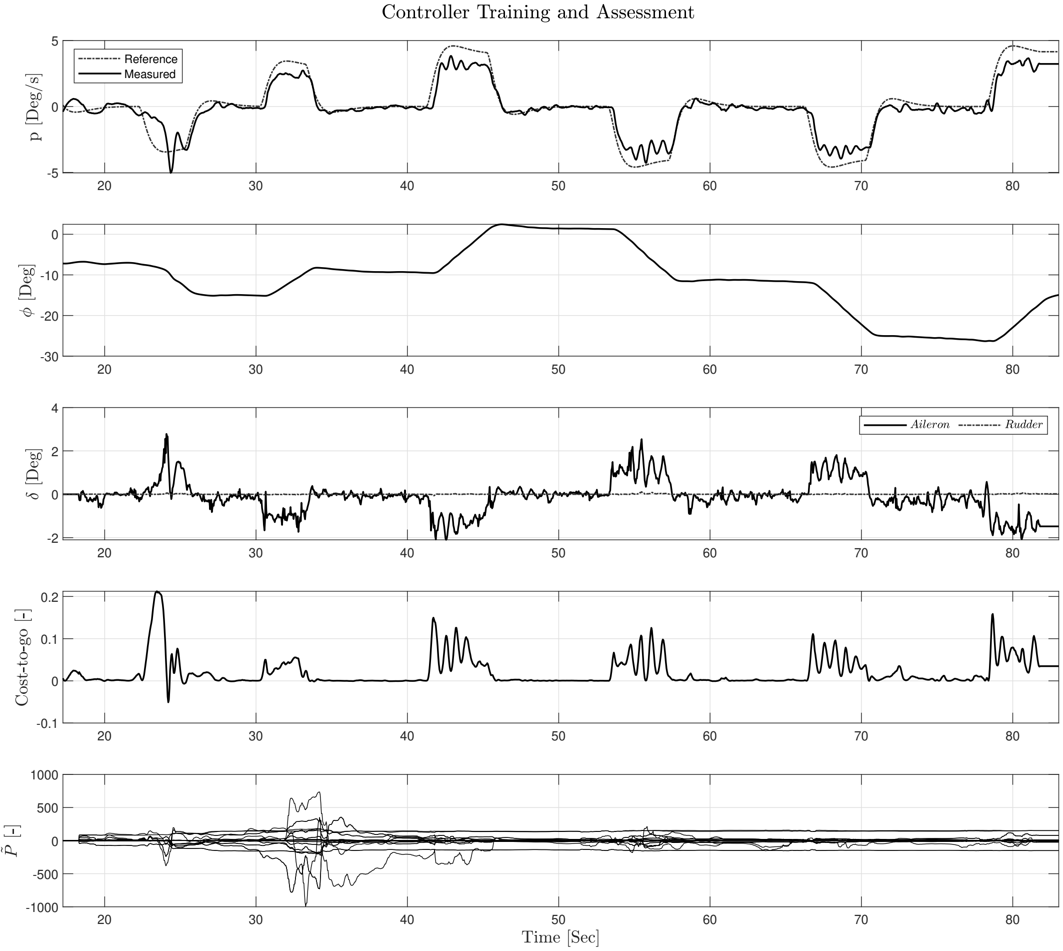


Fig. 8 Flight Test Data(Trial ID : A23-F2-T2), PH-LAB performing a Lateral maneuver: Plots show performance from Controller Training and Assessment Phase. iADP Flight Control Law (FCL) designed for Roll Rate Capture. Continuous Learning Approach (CLA) with real-time parameter adaptation.

Adaptability to different Aircraft Configurations

To assess the adaptability of the iADP controller, five flight test trials were conducted, with varying aircraft configurations. The controller's objective is to track roll rate commands using a CLA. The operating conditions (trimmed aircraft velocity and altitude) are consistent across all trials. The controller objective, approach, operating conditions, and hyper-parameters of the iADP controller are kept constant, enabling an evaluation of controller performance solely against changes in aircraft configuration. Four aircraft configurations are considered:

1) **Nominal(N)**: Landing gear up, flaps completely retracted (0° extension), similar to a normal cruise flight.

2) **Landing Gear Down(G-D)**: Landing gear down, flaps completely retracted (0° extension).
3) **Flaps 15° Extension(F-15)**: Landing gear up, flaps extended to 15°.
4) **Flaps 40° Extension(F-40)**: Landing gear up, flaps extended to 40°.

Each configuration change is introduced one at a time, allowing an assessment of adaptability to individual changes. Pilots change the aircraft configuration by deploying landing gear or extending flaps first, and the aircraft is trimmed in each configuration before engaging the iADP controller. Following each configuration change, the iADP FCL undergoes three phases: model learning, controller training, and controller assessment phase.

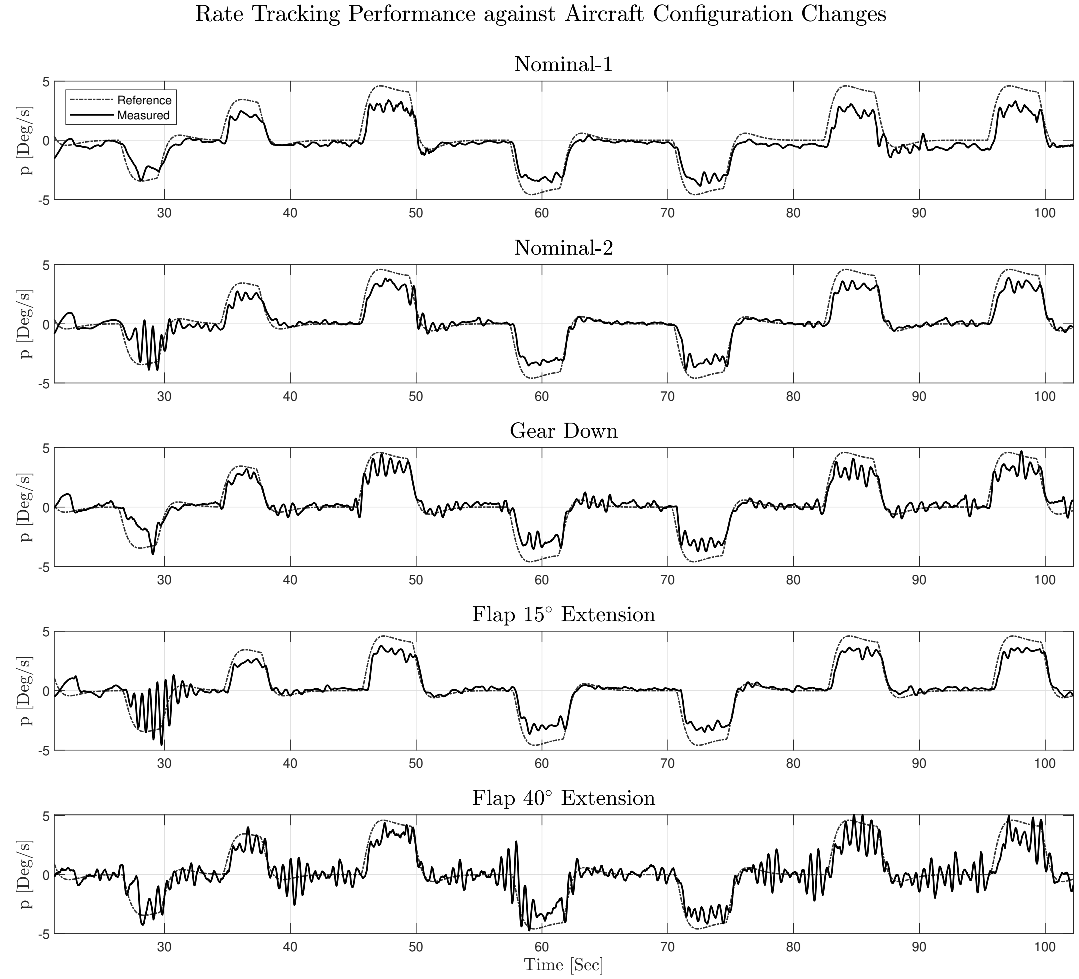


Fig. 9 Flight Test Data(Trial ID : A23-F2-T3 to A23-F2-T7), PH-LAB performing Lateral maneuvers: Experimental setup to test the controller adaptability to change in aircraft configurations. Plots show performance from Controller Training and Assessment Phase across different aircraft configurations. iADP Flight Control Law (FCL) designed for Roll Rate Capture. Continuous Learning Approach (CLA) with real-time parameter adaptation.

Two trials, designated as N-1 and N-2, were conducted in the nominal configuration, followed by the deployment of landing gear (G-D) and flap extensions (F-15 and F-40). The adaptability test aims to evaluate differences in model/controller parameters with respect to configuration changes. Figure 9 illustrates the controller's performance across these configurations. Model and controller parameters undergo continuous updates at each time step, i.e., CLA. When comparing Nominal-1 configuration with others, the tracking performance exhibits the most significant difference with landing gear and 40° flaps extension, while showing similar performance with Nominal-2 and 15° Flaps extension configurations. The controller response appears oscillatory when flaps are at their maximum (40°) extension, likely due to the substantial alteration in the aircraft's aerodynamic properties. Flap deployment modifies the aerodynamic forces on the wings, influencing the rolling moment of the aircraft. The landing gear-down configuration also results in a different controller response, possibly due to the changes in the moment of inertia about the roll axis due to mass redistribution, consequently affecting the lateral stability characteristics of the aircraft. Aileron effectiveness is expected to be impacted by both flap extension and landing gear down configurations.

Despite the variations in aircraft configurations, it is interesting to observe that the controller, despite being unaware of these model changes, effectively guides the aircraft in the lateral axis throughout the entire maneuver. Additionally, the controller parameter updates exhibit stability throughout the entire maneuver across all four configurations.

To quantify the adaptability of the controller further, time-evolving parameters are compared against different configurations. Four different metrics are considered for comparison,

1) **Tracking Error:** Evaluates controller tracking performance, assessing the control objective.
2) **Incremental Model State Matrix ($\tilde{F_t}$):** Measures identified incremental model parameters related to state transitions, containing state derivatives.
3) **Incremental Model Control Effectiveness Matrix ($\tilde{G_t}$):** Measures identified incremental model parameters related to control effectiveness, containing control derivatives.
4) **Kernel Matrix ($\tilde{P}$):** Measures learned control policy parameters.

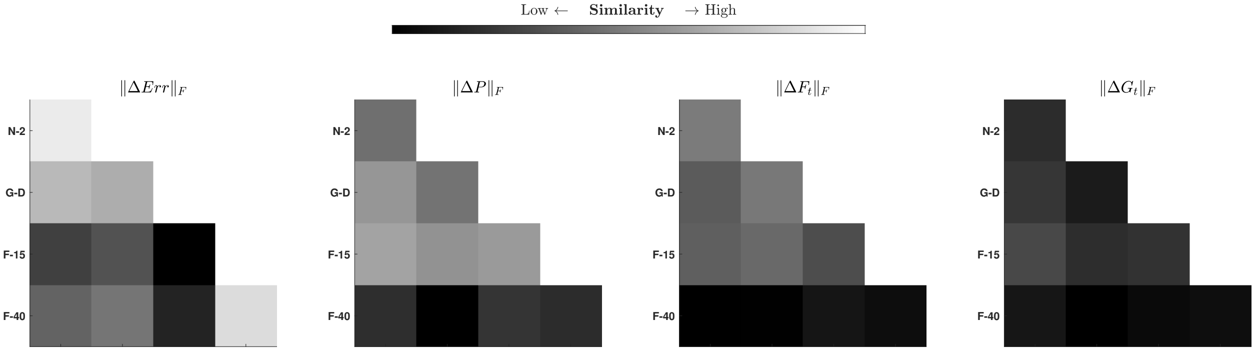


Fig. 10 Quantifying similarity index of Adaptive control parameters using Frobenius Norm across Aircraft Configurations. Tracking error is less sensitive to aircraft configuration changes in Continuous Learning Approach (CLA). Kernel Matrix Parameters $\tilde{P}$ are correlated to Aircraft Configurations, indicating adaptation of control policy by the Reinforcement Learning agent. State transition matrix parameters($\tilde{F_t}$) from the Incremental Model are correlated to configurations, high sensitivity to Flaps 40° Extension. High variance in control effectiveness matrix parameters($\tilde{G_t}$) of the Incremental Model. Configuration labels: N-1 (1st trial in Nominal Configuration), N-2 (2nd trial in Nominal Configuration), G-D (Landing Gear Down), F-15 (Flaps 15° Extension), F-40 (Flaps 40° Extension)

For an accurate comparison, flight test data is aligned, and values are smoothed using a Gaussian-weighted moving average filter to remove noise artifacts. Additionally, only data from 35 to 100 seconds is utilized in this analysis to mitigate the impact of transients. The Frobenius norm of the difference in matrices is selected to assess the similarity of these values. Although nuclear and spectral norms are considered as alternatives for comparison, the results appear insensitive to the choice of the norm, and only Frobenius norm based evaluation is presented. For example, the norm for comparing N-1 configuration data with Flap 15 configuration is defined as follows:

$$\|\Delta P\|_F = \|P_{N1} - P_{F15}\|_F$$

The evolving matrix $\|\Delta P\|_F$, sized according to each trial's duration, represents a time-dependent parameter. A cumulative sum of this value serves as a metric indicating the similarity between P_{N1} and P_{F15} parameters. This similarity measure is depicted for various values and compared across different aircraft configurations in fig. 10. Comparing the first column of all four plots, i.e., comparing N-1 to {N-2, G-D, F-15, and F-40}, the difference in the tracking error seems to be minimum between N-1 and N-2. However, examining the $\|\Delta\tilde{P}\|_F$ plot indicates that the similarity is least between N-1 and F-40, contrasting with the N-1 and N-2 comparison. This observation suggests that controller parameters undergo updates to accommodate aircraft configuration alterations. The $\|\tilde{F_t}\|_F$ plot indicates the greatest difference in the state derivative matrix during the F-40 configuration, as expected due to significant changes in the aircraft's aerodynamic properties during maximum flap extension. The variability of the control effectiveness matrix $\tilde{G}_t$ appears high, making comparisons challenging. But $\|\tilde{G}_t\|_F$ plot indicates lowest similarity when comparing {N-1, N-2} with F-40, and G-D. This suggests the model learning phase adapting to identify the control derivative parameters specific to F-40 and G-D configurations.

C. Discussion

Controller Effectiveness Matrix parameter estimation Issues

During one of the iADP pitch rate tracking controller validation experiments, an abnormal elevator command by the controller led to the aircraft entering an off-nominal flight condition(trials N22-F2-T2 and N22-F2-T4). Upon investigation, it was observed that the model parameters are not stable post 3211 phase, specifically the identified parameters of the controller effectiveness matrix had the wrong signs. These erroneously frozen parameters were used in the subsequent controller training phase. The kernel matrix $\tilde{P}$, which provides a measure of how good the policy is, failed to retain positive symmetric definiteness properties, due to inaccurate Incremental Model parameters. The pitch rate tracking response indicated that the controller initially attempted to command in the wrong direction relative to the reference, a result of the sign inversion in the model parameters.

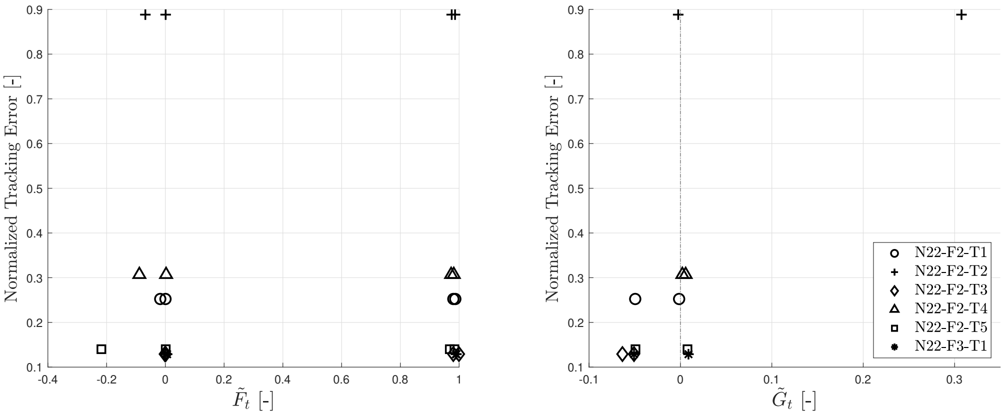


Fig. 11 Flight Test Data across different Trials: Sensitivity Assessment of iADP Controller Tracking Error to Converged Incremental Model Parameters. The normalized integral tracking error is correlated with variance in control effectiveness matrix parameters estimates, provided by RLS algorithm. PH-LAB performing a Longitudinal maneuver. iADP Flight Control Law (FCL) designed for Pitch Rate Capture. Sequential Learning Approach (SLA) with fixed parameters post Model Learning and Controller Training.

To further investigate this issue, the fixed incremental model parameters i.e., both state transition matrix and control effectiveness matrix entries, along with integral tracking error from different longitudinal control flight test trials are considered. Only the fixed parameters at the end of model learning phase are considered. The data from successful and off-nominal flight test trials is compared in fig. 11. The error shown in y-axis is an integral error, normalized across different flight test trials. Only the flight trials for longitudinal pitch rate tracking task with SLA are considered for this analysis. The scatter plot denotes an element in Ft or G_t from a flight trial, and these are compared against the observed tracking error from this flight trial. Comparing G_t value from trials (N22-F2-T2 and N22-F2-T4) with other trials, we see that the high tracking error was linked to the large deviation in G_t estimates. Hence, the positive correlation between normalized tracking error and variability in the control effectiveness estimates is evident. Additionally, the tracking error is also correlated with sign inversion of the control effectiveness matrices, which ideally should have negative values. These experiments offered important insights, highlighting the need to design controller validation experiments to maximize the interpretability of parameters for online adaptive FCL's.

Significant variability in estimating the controller effectiveness parameters of the Incremental model through the RLS algorithm is evident. Flight trial data indicates incorrect sign estimation, a physically implausible scenario(e.g., a positive elevator deflection should induce a negative pitch rate). The Incremental model is running at 1000 Hz, making it prone to noise. The noise and delays from the sensors, including actuator position measurements needs to be investigated and filtering of these measurements should be reevaluated. Unavailability of accurate sensor models of PH-LAB led to difficulty in verifying the sensitivity of the algorithm to sensor dynamics through desktop simulations. However, the flight test data from CLA is interesting, which indicate that the controller could compensate for this error in estimating the control effectiveness matrix parameters and was still able to control aircraft. Potential solutions to address variance in estimating control effectiveness matrix parameters include: running the incremental model at a lower frequency or introducing L1 constraints in the optimization routine to penalize sign inversion.

Fine-Tuning Controller Performance

The current iADP controller cannot handle large deviations in operating conditions, due to the absence of velocity and altitude information in the cost. Notably, the controller leaves a residual attitude angle at the end of maneuvers, as it is designed exclusively to track the rate commands, and lacks integral error information. The errors in attitude and position can be embedded into the iADP controller cost function. Including attitude angle/position error and/or airspeed/altitude in the cost function could increase the state space size in cost-to-go approximation, requiring more computations, and necessitating the exploration of these new states during maneuvers.

Alternatively, the control structure could be redesigned by augmenting the inner rate control loop with a middle attitude loop and subsequently position loops, following a cascaded controller architecture[38]. This can be achieved through an NDI-based attitude loop. As this is purely a kinematic inversion, an NDI based attitude loop could be designed to convert attitude commands to rate commands, which serve as reference signals to the iADP controller [38]. This provides a solution that retains the advantages of handling non-linearities and being model-free.

Controlling Stable and Unstable dynamics

Validation tests of the iADP FCL demonstrate its ability to handle both stable longitudinal dynamics through pitch rate capture, and also unstable lateral dynamics through roll rate captures. The aircraft is unstable in lateral spiral mode, and without corrective control input, the aircraft slowly deviates from wings level flight. The instability in the lateral axis imposed limitations on the open-loop model identification duration, as the aircraft deviates from level flight, and by the time the controller took the corrective action, the error is too large. The controller training needs to be sufficiently fast to have an acceptable transient response, before errors become too significant.

It is worth noting that the controller being less dependent on the model, resulted in quicker controller design adoption for lateral axis, from the one which was initially designed for longitudinal rate control. Being model agnostic meant, there are fewer variables to consider during controller design process and integration, saving time and effort—an advantage, difficult to achieve with conventional non-adaptive control methods.

Sequential Learning vs. Continuous Learning Approach

Observed issues with sequential learning include, the choice of *when* to fix the model parameters was found to be a key parameter. Fixing parameters assumes consistent aircraft and operating conditions, potentially causing mismatches when deviations occur. Any deviation from this assumption means that the learned controller parameters may not accurately represent the current state of the aircraft being controlled. CLA overcame this issue by continuously adjusting both the model and the controller parameters. This might be the reason for CLA producing better results compared to SLA. Nevertheless, SLA aided in validating each subsystems of the iADP Controller, as it runs only one loop at a time.

Curiosity vs. Sensitivity Trade-off

The control input acting on the system serves dual purposes: firstly, to stimulate the system for improved identification, and secondly, to ensure a stable control action [39] [40]. Consequently, the policy must exhibit a curiosity for acquiring novel information while maintaining a resistance to disturbances or noise in the loop, as these can impact steady-state control performance. This dual requirement presents a trade-off between exploration and exploitation, in the context of RL. Various factors contribute to shaping this trade-off: including the forgetting factor in the RLS algorithm, the discount factor in the Bellman equation, sensor sampling frequency, update frequency of controller learning phases, and parameters of sensor filters. Techniques such as variable forgetting factors [41] can effectively strike a balance, allowing the policy to remain curious about new information while not being reactive to turbulence or sensor noise.

Tools for Monitoring Real-time Critical Parameters During Online Adaptation

Real-time monitoring of key adaptive parameters is found to be key during the conducting the flight test trials. An incremental model watchdog is implemented, which provided a real time monitoring of predicted aircraft states by the Incremental model and the measured aircraft states. This served as a tool to intervene in the middle of the flight, in case the health of this incremental model degrades. A comparable approach could be applied to monitor the health of the covariance matrix of the RLS algorithm, Kernel matrix P, and other critical adaptive control indicators.

On the Complexity of the Cost Function Approximation

The iADP algorithm takes advantage of the fact that, when the state space of the controlled system is confined to a region near a nominal optimum path, the Dynamic Programming problem can often be well approximated by a linear-quadratic (LQ) problem, i.e., a problem with linear(time-varying) dynamics and a quadratic performance index with respect to the states and the controls [42] [43]. Alternatively, if a quadratic value function approximation proves inadequate for representing the cost-to-go, neural network-based function approximations for value and/or policy functions can be chosen, leading to an actor-critic like structure [8] [44] [45]. However, this choice involves a trade-off, as neural networks demand more computational effort and are less sample efficient compared to the iADP algorithm.

The iADP algorithm can be modified to include actuator constraints as detailed in [46]. High frequency inputs could be reduced through inclusion of cost ($\Delta\delta_t^T R_\Delta \Delta\delta_t$) in the one-step cost function. The control input command could be rate limited by including rate limits on the calculated $\Delta\delta_t$.

Handling Non-linearities

The iADP algorithm addresses nonlinear dynamics using a first-order Taylor series approximation of the local time-varying model. This approximation remains valid when the sensors, used in recursive model identification, are sampled at a high frequency. While the incremental model is not directly used in control design, it aids in estimating the infinite horizon cost-to-go through policy evaluation, making the controller less sensitive to incremental model estimation inaccuracies. While the Taylor series approximation holds true only within its convergence radius, the iADP algorithm relies on the concept that, if the controller can respond quickly enough to keep the system on the expected optimal path, this approximation holds true for the entire nonlinear dynamics landscape.

The predominantly linear behavior of the PH-LAB aircraft model in a significant part of its flight envelope makes assessing the nonlinear control capability of the FCL challenging. Nevertheless, the iADP FCL is designed and successfully verified on a nonlinear F-16 aircraft model, including tests for online adaptation following a control surface failure [17].

Model Free Controller Adaptation for Fault Tolerance

A controller, which can control the aircraft without a priori knowledge of the aircraft, in theory, could be tolerant to aircraft failures, provided the aircraft remains controllable, and states are observable. iADP FCL's model agnostic property was validated in flight tests, demonstrating its ability to track commands across various aircraft configurations, without hyper-parameter adjustments or awareness of configuration changes. Currently, configuration changes occurred before the controller is engaged. It would be interesting to assess the controllers adaptability to handle these changes occurring while the controller is learning. For a direct fault tolerance assessment, scenarios like engine failure or stuck control surfaces could be introduced during flight to evaluate failure recovery capability of the FCL. Another advantage of iADP FCL is that it is tolerant to sensor signal mismatches or signs, enabled by the model free control structure, making it less sensitive to these type of sensor failures. The algorithm only fails in case of complete sensor breakdown, a concern addressable through hardware redundancy.

V. Conclusion

This paper presented the RL based iADP FCL design for a CS-25 class aircraft and reported the findings from the maiden flight test campaign of this controller. The methodology of the iADP controller was detailed, including an assessment of its features and limitations, along with a concise overview of the FCL 's V&V process. The controller demonstrated its capability to control the aircraft without prior knowledge of the aircraft model or any offline pre-training, relying solely on the data collected along the system trajectories of the aircraft, with real-time parameter adaptation. The flight tests validated the FCL for the stable longitudinal axis, through pitch rate captures and unstable lateral axis, through roll rate captures. Post-flight comparison of adaptive parameters indicated, the capability of the controller to adapt to different configurations, while retaining parameter interpretability. The outcome of flight tests was reviewed, encompassing discussions on challenges faced, potential improvements to the FCL and scenarios for future fault tolerance validation through flight tests. This model free and adaptive FCS could work as a potential lifeline in flight emergencies, complementing traditional FCS and aiding aircraft in fault recovery when facing controllable failures.

Acknowledgments

The authors express their gratitude to personnel at Delft University of Technology, including Alexander in't Veld, Hans Mulder, Menno Klaassen, Ferdinand Postema, Olaf Stroosma, René van Paassen, Fred den Toom, and Isabelle El-Hajj, for providing the framework and expertise that facilitated the successful execution of the flight tests. Special acknowledgment also goes to the DLR (German Aerospace Center) team members involved in the flight test campaign, namely Richard Kuchar, Marc May, Reiko Müller, Stefan Langen, and Christina Schreppel, for their valuable support and expertise. A special acknowledgment goes to Carsten Oldemeyer for carefully reviewing the manuscript.

References

[1] "Loss of Control In-Flight Accident Analysis Report," Tech. rep., International Air Transport Association (IATA), Montreal—Geneva, 2019. URL `https://www.iata.org/contentassets/b6eb2adc248c484192101edd1ed36015/loc-i_2019.pdf`.

[2] Ostroff, A. J., and Bacon, B. J., "Enhanced NDI Strategies for Reconfigurable Flight Control," *Proceedings of the American Control Conference*, 2002. https://doi.org/10.1109/acc.2002.1024492.

[3] Bacon, B. J., and Ostroff, A. J., "Reconfigurable Flight Control Using Nonlinear Dynamic Inversion With a Special Accelerometer Implementation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000. https://doi.org/10.2514/6.2000-4565.

[4] Grondman, F., Looye, G., Kuchar, R. O., Chu, Q. P., and Kampen, E.-J. V., "Design and Flight Testing of Incremental Nonlinear Dynamic Inversion-based Control Laws for a Passenger Aircraft," *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2018, p. 0385. https://doi.org/10.2514/6.2018-0385.

[5] Dydek, Z. T., Annaswamy, A. M., and Lavretsky, E., "Adaptive Control and the NASA X-15-3 Flight Revisited," *IEEE Control Systems Magazine*, Vol. 30, No. 3, 2010, pp. 32–48. https://doi.org/10.1109/MCS.2010.936292.

[6] Annaswamy, A. M., and Fradkov, A. L., "A Historical Perspective of Adaptive Control and Learning," *Annual Reviews in Control*, Vol. 52, 2021, pp. 18–41. https://doi.org/10.1016/j.arcontrol.2021.10.014.

[7] Enns, R., and Si, J., "Neuro-Dynamic Programming Applied to Helicopter Flight Control," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA, 2000, p. 4280.

[8] 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.

[9] Helder, B., van Kampen, E.-J., and Pavel, M., "Online Adaptive Helicopter Control Using Incremental Dual Heuristic Programming," *AIAA Scitech 2021 Forum*, 2021, p. 1118.

[10] Heyer, S., Kroezen, D., and van Kampen, E.-J., "Online Adaptive Incremental Reinforcement Learning Flight Control for a CS-25 Class Aircraft," *AIAA Scitech 2020 Forum*, 2020, p. 1844.

[11] Jacklin, S. A., "Closing the Certification Gaps in Adaptive Flight Control Software," *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008. https://doi.org/10.2514/6.2008-6988.

[12] Zhou, Y., Van Kampen, E.-J., and Chu, Q., "Incremental Model-Based Online Dual Heuristic Programming for Nonlinear Adaptive Control," *Control Engineering Practice*, Vol. 73, 2018, p. 13–25. https://doi.org/10.1016/j.conengprac.2017.12.011.

[13] Recht, B., "A Tour of Reinforcement Learning: The View from Continuous Control," *Annual Review of Control, Robotics, and Autonomous Systems*, Vol. 2, No. 1, 2019, p. 253–279. https://doi.org/10.1146/annurev-control-053018-023825.

[14] Dean, S., Mania, H., Matni, N., Recht, B., and Tu, S., "On the Sample Complexity of the Linear Quadratic Regulator," *Foundations of Computational Mathematics*, Vol. 20, No. 4, 2019, p. 633–679. https://doi.org/10.1007/s10208-019-09426-y.

[15] Konatala, R., Kampen, E.-J. V., and Looye, G., "Reinforcement Learning Based Online Adaptive Flight Control for the Cessna Citation II(PH-LAB) Aircraft," *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, 2021, p. 0883. https://doi.org/10.2514/6.2021-0883.

[16] 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, 2017, p. 493–496. https://doi.org/10.2514/1.g001762.

[17] Dias, P. M., Zhou, Y., and Kampen, E.-J. V., "Intelligent Nonlinear Adaptive Flight Control using Incremental Approximate Dynamic Programming," *AIAA Scitech 2019 Forum*, 2019, p. 2339. https://doi.org/10.2514/6.2019-2339.

[18] Bellman, R., "Dynamic Programming," *Science*, Vol. 153, 1957, pp. 34 – 37.

[19] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, A Bradford Book, Cambridge, MA, USA, 2018.

[20] Zhou, Y., van Kampen, E.-J., and Chu, Q., "Incremental Approximate Dynamic Programming for Nonlinear Adaptive Tracking Control With Partial Observability," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 12, 2018, pp. 2554–2567. https://doi.org/10.2514/1.G003472.

[21] Lewis, F. L., and Vrabie, D., "Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control," *IEEE Circuits and Systems Magazine*, Vol. 9, No. 3, 2009, pp. 32–50. https://doi.org/10.1109/MCAS.2009.933854.

[22] Bradtke, S. J., and Ydstie, B. E., "Adaptive Linear Quadratic Control Using Policy Iteration," *American Automatic Control Council*, Vol. 3, 1994, pp. 3475–3479. https://doi.org/10.1109/acc.1994.735224.

[23] Bradtke, S. J., and Barto, A. G., "Linear Least-Squares Algorithms for Temporal Difference Learning," *Machine Learning*, Vol. 22, No. 1–3, 1996, p. 33–57. https://doi.org/10.1007/bf00114723.

[24] van Paassen, M., Stroosma, O., and Delatour, J., "DUECA - Data-driven Activation in Distributed Real-time Computation," *Modeling and Simulation Technologies Conference*, American Institute of Aeronautics and Astronautics, 2000, p. 4503. https://doi.org/10.2514/6.2000-4503.

[25] Zaal, P., Pool, D., in 't Veld, A., Postema, F., Mulder, M., van Paassen, M., and Mulder, J., "Design and Certification of a Fly-by-Wire System with Minimal Impact on the Original Flight Controls," *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2009, p. 5985. https://doi.org/10.2514/6.2009-5985.

[26] Scholten, P. A., van Paassen, M. M., Chu, Q. P., and Mulder, M., "Variable Stability In-Flight Simulation System Based on Existing Autopilot Hardware," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 12, 2020, pp. 2275–2288. https://doi.org/10.2514/1.g005066.

[27] Muis, A., Oliveira, J., and Mulder, J. A., "Development of a Flexible Flight Test Instrumentation System," 2006.

[28] Keijzer, T., Looye, G., Chu, Q. P., and Kampen, E.-J. V., "Design and Flight Testing of Incremental Backstepping based Control Laws with Angular Accelerometer Feedback," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019, p. 0129. https://doi.org/10.2514/6.2019-0129.

[29] van 't Veld, R., Van Kampen, E.-J., and Chu, Q. P., "Stability and Robustness Analysis and Improvements for Incremental Nonlinear Dynamic Inversion Control," *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, 2018, p. 1127. https://doi.org/10.2514/6.2018-1127.

[30] Weiser, C., Milz, D., May, M., Konatala, R., Kuchar, R., Müller, R., Langen, S., Schreppel, C., and Looye, G., "Flight Testing Advanced Control Functions on a Passenger Aircraft," *EuroGNC 2024*, 2024. Manuscript submitted for publication.

[31] Laban, M., "On-Line Aircraft Aerodynamic Model Identification," Doctoral thesis, Delft University of Technology, The Netherlands, 1994.

[32] Mulder, M., Lubbers, B., Zaal, P., van Paassen, M. M., and Mulder, J. A., "Aerodynamic Hinge Moment Coefficient Estimation Using Automatic Fly-By-Wire Control Inputs," *AIAA Modeling and Simulation Technologies Conference*, 2009, p. 5692. https://doi.org/10.2514/6.2009-5692.

[33] Joos, H.-D., "A Methodology for Multi-objective Design Assessment and Flight Control Synthesis Tuning," *Aerospace Science and Technology*, Vol. 3, No. 3, 1999, pp. 161–176. https://doi.org/https://doi.org/10.1016/S1270-9638(99)80040-6.

[34] Konatala, R., May, M., Müller, R., Looye, G., and Van Kampen, E.-J., "Verification & Validation(V&V) of Reinforcement Learning based Online Adaptive Flight Control Laws on CS-25 Class Aircraft," *EuroGNC 2024*, 2024. Manuscript submitted for publication.

[35] Research, and Technology Organization(RTO), N. A. T. O., *Flight Control Design - Best Practices*, ISBN 92-837-1047-9, Defense Technical Information Center, 2000. URL https://apps.dtic.mil/sti/tr/pdf/ADA387777.pdf.

[36] Anderson, B. D. O., "Adaptive Systems, Lack of Persistency of Excitation and Bursting Phenomena," *Automatica*, Vol. 21, No. 3, 1985, p. 247–258. https://doi.org/10.1016/0005-1098(85)90058-5.

[37] Narendra, K. S., and Annaswamy, A. M., “Persistent Excitation in Adaptive Systems,” *International Journal of Control*, Vol. 45, No. 1, 1987, pp. 127–160. https://doi.org/10.1080/00207178708933715.

[38] Lombaerts, T., and Looye, G., “Design and Flight Testing of Nonlinear Autoflight Control Laws,” *AIAA Guidance, Navigation, and Control Conference 2012*, 2012, p. 4982. https://doi.org/10.2514/6.2012-4982.

[39] Feldbaum, A. A., “Dual Control Theory Problems,” *IFAC Proceedings Volumes*, Vol. 1, No. 2, 1963, p. 541–550. https://doi.org/10.1016/s1474-6670(17)69687-3.

[40] Annaswamy, A. M., and Fradkov, A. L., “A Historical Perspective of Adaptive Control and Learning,” *Annual Reviews in Control*, Vol. 52, 2021, pp. 18–41. https://doi.org/https://doi.org/10.1016/j.arcontrol.2021.10.014.

[41] Fortescue, T. R., Kershenbaum, L., and Ydstie, B. E., “Implementation of Self-Tuning Regulators with Variable Forgetting Factors,” *Automatica*, Vol. 17, No. 6, 1981, p. 831–835. https://doi.org/10.1016/0005-1098(81)90070-4.

[42] Jacobson, D., and Mayne, D., *Differential Dynamic Programming*, Modern Analytic and Computational Methods in Science and Mathematics, American Elsevier Publishing Company, 1970.

[43] Bryson, A., “Optimal Control-1950 to 1985,” *IEEE Control Systems Magazine*, Vol. 16, No. 3, 1996, pp. 26–33. https://doi.org/10.1109/37.506395.

[44] Konda, V., and Tsitsiklis, J., “Actor-Critic Algorithms,” *Advances in Neural Information Processing Systems*, Vol. 12, edited by S. Solla, T. Leen, and K. Müller, MIT Press, 1999.

[45] Vamvoudakis, K. G., and Lewis, F. L., “Online Actor–Critic Algorithm to Solve the Continuous-Time Infinite Horizon Optimal Control Problem,” *Automatica*, Vol. 46, No. 5, 2010, p. 878–888. https://doi.org/10.1016/j.automatica.2010.02.018.

[46] Sun, B., and van Kampen, E.-J., “Reinforcement-Learning-Based Adaptive Optimal Flight Control With Output Feedback and Input Constraints,” *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 9, 2021, pp. 1685–1691. https://doi.org/10.2514/1.G005715.