CEAS-GNC-2024-062

EuroGNC Navigation Guidance Control CEAS Bristol, UK June 11th-13th 2024
University of BRISTOL ROYAL AERONAUTICAL SOCIETY AIAA

Verification & Validation (V&V) of Reinforcement Learning based Online Adaptive Flight Control Laws on CS-25 Class Aircraft

Ramesh Konatala Research Associate, German Aerospace Center (DLR), Institute of System Dynamics and Control, 82234, Weßling, Germany. ramesh.konatala@dlr.de

Reiko Müller Research Associate, German Aerospace Center (DLR), Institute of System Dynamics and Control, 82234, Weßling, Germany. reiko.mueller@dlr.de

Marc May Research Associate, German Aerospace Center (DLR), Institute of System Dynamics and Control, 82234, Weßling, Germany. marc.may@dlr.de

Gertjan Looye Head of Department, German Aerospace Center (DLR), Institute of System Dynamics and Control, 82234, Weßling, Germany. gertjan.looye@dlr.de

E. van Kampen Associate Professor, Delft University of Technology, Faculty of Aerospace Engineering, Kluyverweg 1, 2629 HS Delft, Netherlands. E.vanKampen@tudelft.nl

ABSTRACT

Unforeseen faults during flight can lead to Loss of Control In-Flight (LOC-I), a significant cause of fatal aircraft accidents worldwide. Current offline synthesized, model based flight control methods have limited capability to adapt to unforeseen situations. From a fault tolerance perspective, the Incremental Approximate Dynamic Programming (iADP) controller serves as an ideal model-agnostic, online adaptive control method. This method 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 which limits the adoption of these self-learning control methods for flight control is V&V through flight testing. This study addresses the problem by exploring tools, methods and framework for V&V of the online adaptive Flight control law on a CS-25 class Citation-II passenger aircraft. These flight tests mark world's first demonstration of an online RL based automatic Flight Control System (FCS) for this aircraft category, demonstrating real-time learning and adaptation capabilities.

Keywords: Verification and Validation (V&V), Reinforcement Learning, Online Adaptive Flight Control, Flight Testing

1 Introduction

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 trend towards autonomous and complex aerospace systems, one can only expect an increase in such LOC-I incidents unless proactive measures are taken. Factors contributing to LOC-I include adverse onboard conditions (such as faults, failures, and crew errors), external hazards (like icing, wind shear, and turbulence), and abnormal flight conditions (such as unusual attitudes) [2]. 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. Some main challenges in designing such a controller include degradation of performance of model based controllers due to low confidence in the aircraft model post-failure, non-linearities in the model post-failure, and the need for rapid adaptation of the controller to restore the aircraft to the safe flight envelope.

Self-learning Adaptive FCS algorithms were initially tested in the 1960s on the X-15 research aircraft [3]. Some of the open challenges in realizing adaptive FCS include sample efficiency, convergence, robustness of controller, and interpretability of the controller adaptive mechanism [4]. RL is a machine learning technique, where an agent could learn the optimal strategy(control scheme) to achieve a defined goal, using a scalar feedback reward signal, using a Dynamic Programming based optimization technique [5–7]. Although several variants of RL based FCS design are being developed [8, 9], practical V&V constraints guided the choice of this RL based FCS [10, 11] design. An essential part of certifying these resilient adaptive FCS for safety-critical operations involves establishing a V&V process. This process ensures that the FCS undergoes testing to identify system weaknesses and vulnerabilities.

Although adaptive FCS holds potential, none have received Federal Aviation Administration (FAA) certification for use in the National Air Space (NAS) [12, 13]. Some of the observed gaps in extending V&V for adaptive control include undefined specifications, instability-proof challenges, lack of high-fidelity simulations incorporating nonlinear effects, absence of effective monitoring tools, the need for a certification plan addressing robustness and control architecture sensitivity, and non-determinism of intelligent and reasoning algorithms [14]. EASA has developed guidelines to ensure the safe integration of AI in aviation, including frameworks for V&V to meet stringent safety standards, crucial for the certification of adaptive control systems using Machine Learning (ML) [15]. However, these guidelines are confined to ML algorithms that assist humans and facilitate human-machine collaboration, excluding more autonomous systems, which is the focus of this research work.

The contribution of this paper is V&V of RL-based online adaptive FCS, encompassing the tools and methods. A background study on methodologies used to certify model based control laws to satisfy FAA airworthiness requirements is conducted. An overlap in these methods and tools with FCS being tested is first identified and then adapted along with some custom V&V methods. The technical scope of this study is limited to V&V excluding the interaction of human with the adaptive system. Firstly, the control design specifications are formulated, taking into account the possibilities and limitations arising from the testing platform, available software tools, aircraft models, and required hardware. The procedure to verify that the designed flight controller has met the flight control objectives through desktop, and real-time simulations is presented. Lastly, the procedure to validate the functionality of the FCS through successful flight test campaigns is detailed.

The structure of the paper is organized as follows: Section 2 briefly covers the background information on the control methodology, FCS architecture design, control design specifications, available hardware, software, and models used for V&V. Section 3 provides information on different tools, methods, and frameworks used to conduct V&V. Section 4 delves into verification tests conducted to ensure compliance with the defined controller design requirements, followed by validation of controller functionality, focusing on the results obtained from the flight test campaign. Lastly, Section 5 includes outcome, concluding remarks, and potential future work.

2 Background: Flight Control Law Design for Cessna Citation II

2.1 Incremental Approximate Dynamic Programming

The Incremental Approximate Dynamic Programming (iADP) algorithm provides a framework for optimal control design, well-suited for nonlinear dynamical systems like aircraft. iADP formulates the optimal control design problem within the framework of RL, where an agent interacts with the environment to minimize discounted infinite horizon cost or cost-to-go. However, traditional RL methods, such as Dynamic Programming (DP), face challenges with high-dimensional systems(Curse of Dimensionality). To handle this issue, iADP employs an approximate value function, quadratic in the state, ensuring computational tractability for online implementation. Utilizing an incremental model representation, the algorithm linearizes the nonlinear system to provide a locally linearized incremental model at each timestep using a Recursive Least Squares (RLS) method. Online incremental model identification, facilitated by sample efficient Approximate Dynamic Programming(ADP) algorithm, ensures adaptability and real-time implementation, making iADP a suitable algorithm for designing fault-tolerant FCLs for nonlinear systems. This algorithm has two variants: one for systems with full state feedback and another for systems with partial observability, where input-output measurements are utilized for controller design [10, 16]. iADP was used to design FCL for a nonlinear F-16 aircraft model and validated for online adaptation in case of failures in [17]. iADP based FCL design for rate control on a CS-25 class aircraft, along with control architecture, control law evaluation strategy, controller integration, and a summary of the flight test results [11], which forms the foundation for this study.

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

2.2 PH-LAB for Flight Control Law V&V

The Cessna Citation II (PH-LAB) research aircraft (fig. 1), jointly operated by TU Delft and the Netherlands Aerospace Center (NLR) is certified under CS-25 specifications for large aircraft. The standard configuration of the Cessna Citation includes a reversible FCS connected to the pilot's yoke and pedals, along with an asymmetric pair of ailerons, a symmetric pair of elevators, and a single rudder. Additionally, it features an autopilot system, activated by clutch-engaged servo motors linked to the control surface cables. For testing the experimental flight control functions, the aircraft is equipped with an experimental fly-by-wire (FBW) system, as detailed in [18], which has undergone exhaustive testing 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. Additionally, the aircraft had an upgrade of several sensor systems and is equipped with a Flight Test Instrument System that includes a data acquisition computer and a signal conditioning unit for processing information from sensors, with more information found in [18–20]. This setup makes this research aircraft an ideal experimental platform for iADP-based FCS Design and subsequent V&V [21–25].

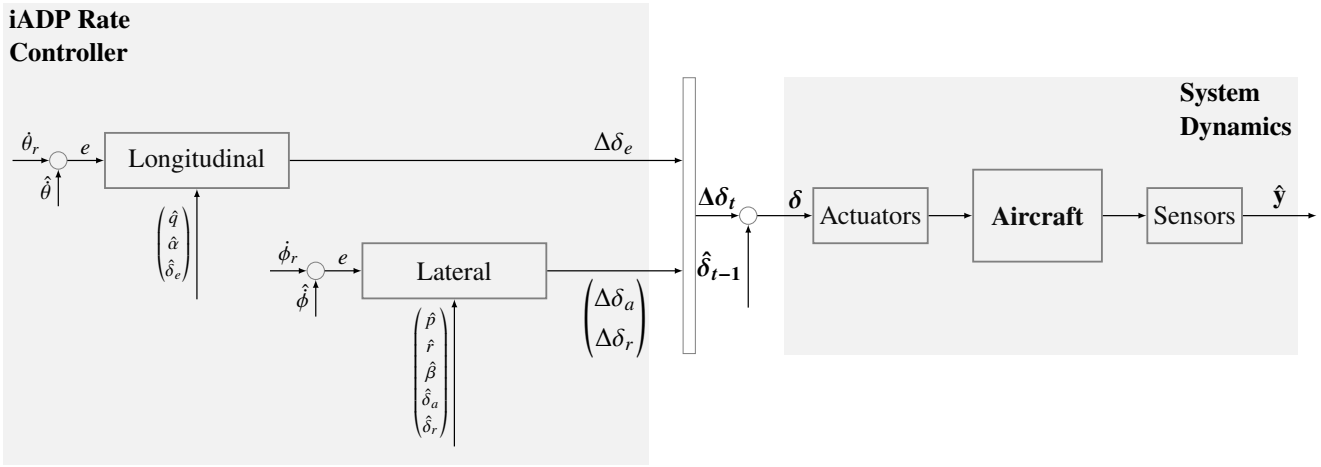

Fig. 2 Incremental Approximate Dynamic Programming based Flight Control Law Architecture for Inner Loop Rate Tracking. Decoupled Longitudinal and Lateral Reinforcement Learning Controllers for PH-LAB Aircraft.

2.3 iADP based Flight Control System Design

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. 2 (iADP Rate Controller). 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 and/or rudder. 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. This architecture ensures a model-free and aircraft-independent inner loop control structure. The iADP controller computes the required incremental control input at each time step, with the total control input determined by adding the previous control surface measurement to the evaluated control increment.

2.4 Aircraft Model preparation

While there are no established analytical methods to assess the quality of adaptive control laws, simulation-based investigations are a cheap and safe, thus preferable alternative. To achieve insights into the closed-loop behavior on a level comparable to experimental flight tests, a high-fidelity simulation model, covering relevant real-world phenomena and nonlinear effects, is required. With controller V&V in mind, these shall also include the boundaries of the flight envelope and off-nominal conditions. With a long service as a research platform, the Cessna Citation II PH-LAB has been thoroughly investigated throughout the flight envelope [26], and derived models of the aircraft and its sub-components were validated against flight test data. The Delft University Aircraft Simulation Model and Analysis Tool (DASMAT) [27] in the MATLAB/Simulink environment serves as the core for the high-fidelity rigid body 6-DoF flight dynamic model of the Cessna Citation II. The aerodynamic model is based on the Cessna Citation 500 and was adapted to the real geometry using flight test data [28]. From a flight control perspective, actuator and sensor models are obviously of special interest, as they define the interaction with the plant. In addition, flap and landing gear models are considered briefly, as they are deployed in flight for validation of FCS adaptability to aircraft configurations.

FBW system

Allowing the interaction between flight controller and aircraft, actuators and their modeling fidelity take an important role in the development and validation of flight control algorithms. On the other hand, verification of FCS with a high fidelity actuator model, before flight tests is mandatory to perform tests under realistic circumstances. Next to passthrough behavior, a LoFi PT1-model of actuator with integrator saturation is used. These linear models assume that commanded deflections are perfectly met by the FBW-controller. In contrast, the HiFi model takes a physical approach to consider the interaction of aerodynamic hinge moment, cable dynamics, and other nonlinear FCS-components like column weights and springs. It includes the FBW-controller as implemented in the real aircraft and a model of the servo and auto trim system. A detailed description can be found in [29].

Sensor Models

The PH-LAB flight test instrumentation provides numerous signals from sensor systems like Attitude Heading and Reference System (AHRS), Digital Air Data Computer (DADC), air data boom, and control surface deflections. The hardware setup is described in more detail in [18–20].

Aircraft Configuration

An indirect way to assess control law adaptability to failure, is to check for its adaptability to deal with aircraft configuration changes like deploying either flaps or landing gear. Their main impact is on the aerodynamics, thereby significantly altering the aircraft behavior. This effect is considered by the aerodynamic look-up tables used within the flight dynamic model. Especially for large flap deflections at higher flight speeds, noticeable fluctuations are present due to flow separation.

2.5 Flight Control Design Specifications

Table 1 iADP FCS Design Specifications. The last column indicates if V&V was conducted for the requirement.

Spec. ID	Summary	Description	V&V
S01-MT	Minimize Rate Tracking Error	Ensure controller commands aircraft to follow pitch and roll rate commands	✓
S02-AC	Adaptability to Aircraft Config.	Ensure rate tracking task in different aircraft configurations	✓
S03-OP	Robust against Operating Conditions	Ensure rate tracking task in different operating conditions (Alt. and Vel. changes)	✓
S04-RR	Reproducible results	Consistent controller behaviour under similar conditions	✓
S05-MF	Agnostic to Model	Ensure acceptable controller behaviour without a priori knowledge of the model	✓
S06-RM	Robustness to Model Uncertainties	Acceptable Controller behaviour subjected to model uncertainties	✓
S07-SN	Sensitivity to Noise, Bias & Delays	Sensitivity of controller against sensor dynamics to be minimum	✓
S08-SP	Low Sensitivity to hyperparameters	Ensure stable controller learning over a wider controller tuning parameter range	✓
S09-DR	Disturbance Rejection	Acceptable controller learning process under turbulence and wind	×
S10-CR	Acceptable Controller Response	Minimize steady state error, rise time and overshoot	×
S11-CL	Continuous Learning	Ensure Stable continuous learning of controller over longer manoeuvres	✓
S12-IN	Explainability of Learning	Ensure Learning process can be explainable for post flight critical analysis	✓
S13-OM	Online Monitoring	Ensure real-time monitoring of critical parameters/data	✓
S14-FT	Fault Tolerance	Assess online adaptation for fault tolerance in engine failure scenarios.	×
S15-HQ	Handling Qualities	Quantitative Handling Quality Assessment Using Time Domain Methods	×

Flight Control Design specifications serve as a blueprint for control engineers, guiding them in the design process to verify that the controller meets the desired criteria. These specifications aided in the systematic validation of the controller's performance and reliability, ensuring the FCS meets the intended goals. Table 1 outlines the formulated FCS design requirements.

3 Tools, methods and framework to conduct V&V

3.1 Tools for V&V assessment

MATLAB/Simulink has been selected as the preferred software for developing the iADP FCL. The availability of the PH-LAB nonlinear aircraft model, the ability to produce C-code suitable for hardware implementation, compatibility with real-time target machines like Speedgoat, rapid prototyping capabilities, and established toolchains (MOPS) [30] for hyperparameter tuning serve as primary motivations for this decision. Another advantage is the utilization of the Embedded Coder toolbox, facilitating the generation of C-code compliant with aviation standards such as DO-178C [13].

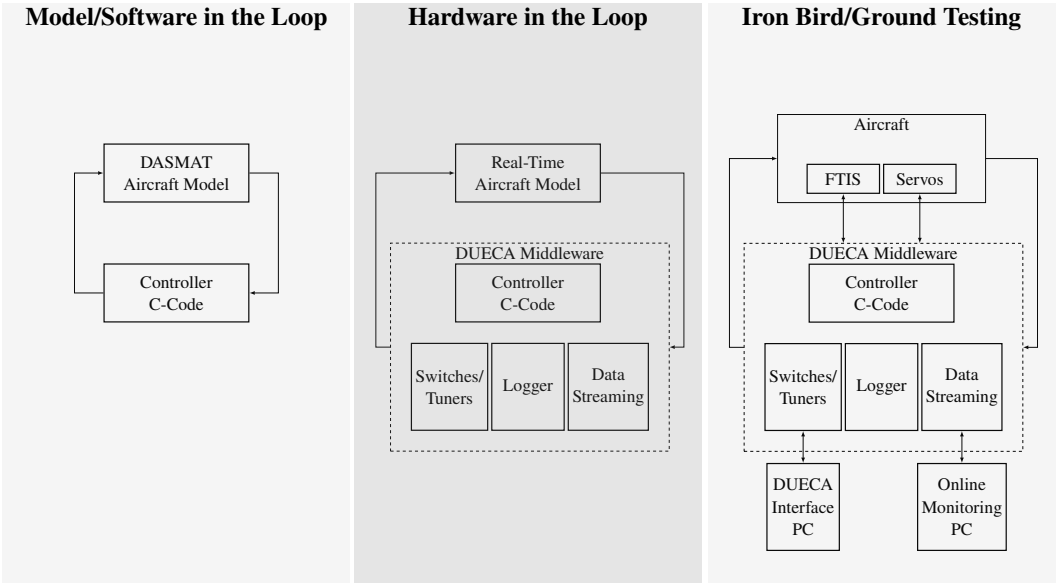

Fig. 3 Simulation setup indicating various fidelity levels of the simulation setup from Model/Software in the Loop, Processor in the Loop to final Iron-bird/Ground testing of the flight control laws.

During the initial development phase, a Model-in-the-Loop environment was utilized for rapid prototyping and controller development, employing the DASMAT aircraft model, as depicted in fig. 3. A modified version of this environment, referred to as Software-in-the-Loop, was created, where the controller code was substituted with C-Code. This intermediate verification step ensures that the generated C-code performs similarly to the original Simulink version of the controller. In the Hardware-in-the-Loop phase, the DASMAT aircraft model is replaced with a real-time compatible model by converting the original nonlinear Simulink-based aircraft model to C-code with defined sampling times. This real-time model is then interfaced with the Delft University Environment for Communication and Activation (DUECA), a middleware facilitating real-time implementation and communication of distributed systems [31]. DUECA encourages modular design and ensures real-time synchronization and exchange of signals. After revising the automatically generated code to ensure the correct export of control functions, the code is integrated into the real-time framework.

The controller code is encapsulated in a separate DUECA module, which manages the necessary interfaces. DUECA ensures real-time synchronization via a master module overseeing all concurrently operating modules at the base frequency, which, in this flight test scenario, was set at 1000 Hz. The remaining DUECA modules, including switches/tuners for controller interfacing, are configured using a graphical user interface (GUI). Another module of DUECA is designated for logging purposes. This logger module accommodates two types of logging: first, it captures raw data from the aircraft sensors via FTIS, while another logging system records data from the controller. The third DUECA module serves as a data streaming module, where a subset of the controller logger data is selected and streamed

in real time through a UDP sender. This real-time monitoring enables the observation of critical data, providing insights into the health of controller subsystems during flight trials.

During the Iron bird/ground testing phase, the real-time aircraft model is replaced by the actual aircraft, which is interfaced with sensors via the FTIS system and actuators via the Servos DUECA module. The DUECA module operates on the flight control computer, equipped with an Intel® Core™ i5-3550S quad-core processor and runs Ubuntu 20.04 LTS with a real-time kernel (PREEMPT_RT) [18, 32]. In this phase, the controller has access to the actual hardware setup of the aircraft, albeit on the ground. This phase confirms the correct interfacing between the controller via the DUECA module and sensors/actuators. This is verified by observing sensor data through logs/data streaming and issuing dummy actuator commands. Two different PCs are set up to communicate with DUECA. One is used to adjust switches and tuners, while the other PC is configured to receive live data via UDP communication. Real-time data is monitored using the Simulink data inspector.

Another toolchain for automating flight data processing has been developed. This tool stores information of the controller/aircraft hardware interfaces and converts raw flight log data into MATLAB-supported .mat files, which can be easily read via the Simulink data inspector. This automated data processing tool chain enabled quick identification of any issues in the controller post-flight and facilitated the comparison of simulation/HIL data with flight test data. During the rapid controller prototyping stage, MATLAB/Simulink inbuilt testing tools such as static unit testing frameworks, automated report generation tools, code coverage analysis, and custom-made scripts are extensively utilized for verification.

3.2 Methods for V&V assessment

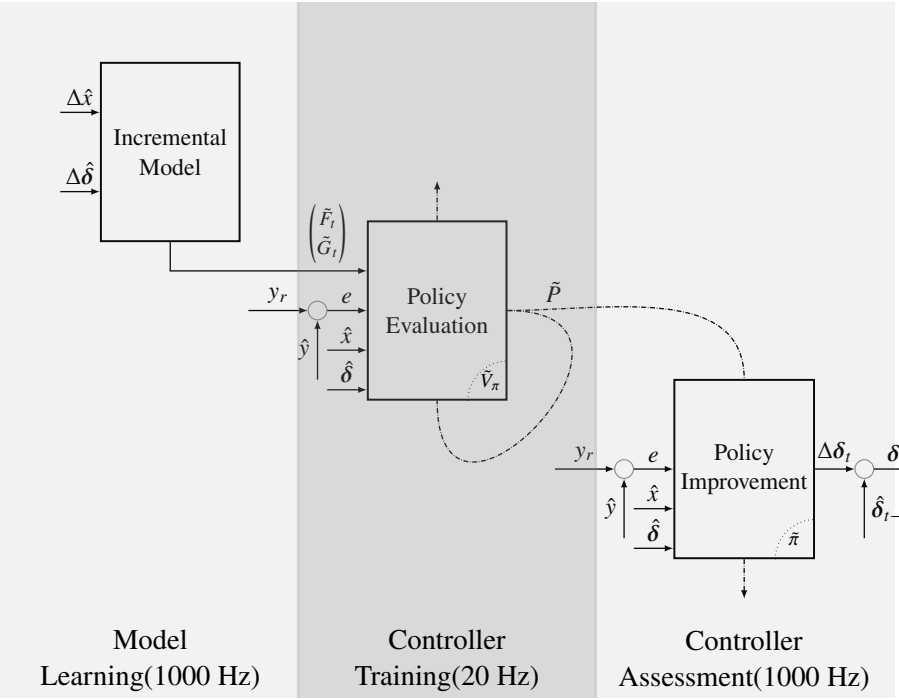

Fig. 4 Structure of the Reinforcement Learning Agent of iADP Flight Control Law. Model Learning provides the latest model estimates using the RLS algorithm. Controller Training evaluates(V_π) the Control Policy using incremental model estimates and one-step Cost. Controller Assessment takes actions and improves Control Policy (π) based on policy evaluation. The frequency at which each subsystem on the Flight Control Computer runs is indicated below. $\hat{\bullet}$ denotes measured value, $\tilde{\bullet}$ denotes estimated value, $\tilde{F}$ denotes state transition matrix estimate and $\tilde{G}$ denotes controller effectiveness matrix estimate.

The V&V process of the iADP algorithm is done systematically, involving three sequential phases: model learning, controller training, and controller evaluation, as illustrated in fig. 4. Two distinct imple-

mentation approaches have been considered: sequential learning and continuous learning approaches. In sequential learning, the controller undergoes three consecutive stages. Initially, a model learning phase, during which the model parameters are determined and then fixed. Subsequently, these parameters are passed on to the controller training phase, where, using the model parameters from the earlier phase, and the observed error in rate tracking, the controller estimates the cost-to-go. Upon completion of the controller training phase, the controller parameters are fixed and forwarded to the subsequent controller evaluation phase. Here, the objective is to track the reference command based on these converged controller parameters. In continuous learning, apart from the initial 20 seconds of the open-loop model learning phase, all three stages run concurrently.

Optimization

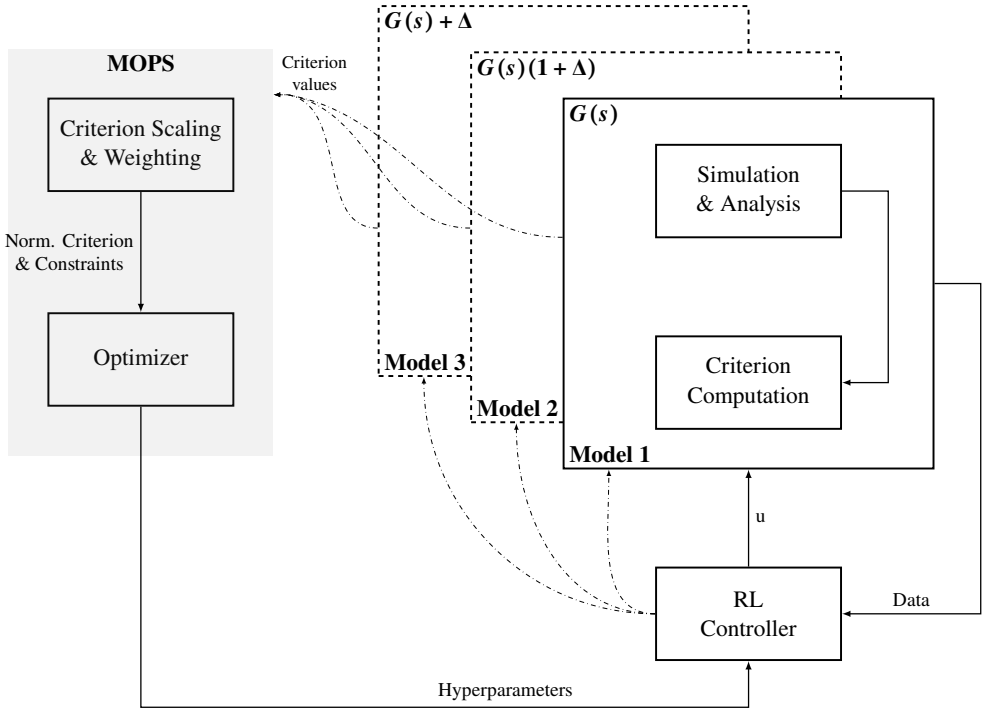

Fig. 5 Tuning hyperparameters of RL based control law for robustness using multi- model approach. MOPS denotes Multi-Objective Parameter Synthesis, Norm. denotes Normalized, RL denotes Reinforcement Learning, $G(s)$ denotes nominal plant, $G(s)+\Delta$ denotes nominal plant with additive uncertainties, $G(s)(1+\Delta)$ denotes nominal plant with multiplicative uncertainties.

The DLR optimization tool Multi-Objective Parameter Synthesis (MOPS), is a software written in MATLAB for solving general purpose parameter optimization problems but also features modules for optimal control, system identification, design of experiments, and performing Monte Carlo simulation. A parameterized run-script works as the central element for performing all of the previous tasks and allows to set up analysis chains. Verification in the context of online adaptive control necessitates the use of optimization algorithms, for example, to find a set of controller parameters that are robust with respect to uncertainties, via anti-optimization [33, 34]. For instance, a development loop within MOPS for robust control law design including verification would be:

1) Perform an optimization of the controller parameters for a set of cases. The cases could for example be different models, operating points or environmental conditions as shown in fig. 5.
2) To check whether the found optimal parameter set is viable, perform an anti-optimization of disturbances, (model) errors, failures, etc. with respect to the requirements given to the controller. The found worst cases per criterion are added to the controller optimization of the previous step,

forming an iteration loop. This loop should be run through until the anti-optimization cannot find non-satisfactory cases anymore.

3) In the final step, a controller clearance can be performed, i.e. a Monte-Carlo simulation for verification of the final controller design. This is similar to the anti-optimization step, as the controller is again subjected to external sources of error and disturbances, but yields a statistical measure for the safety/performance/robustness/... of the controller. Again, if requirements are violated, the process should be started anew until an optimal and admissible set of parameters is found.

Typically, control design requirements refer to mission performance (tracking error and disturbance rejection), stability robustness with respect to key system parameters (mass, center of gravity) as well as unmodelled dynamics (e.g. gain and phase margins), ride quality (passenger and pilot comfort by bounds on allowable acceleration and minimum damping), safety (envelope safeguards), and control activity (power consumed by the controls, control rates). In the context of RL, MOPS can be used to tune the hyperparameters of the iADP algorithm by assessing the adaptive controller's performance against data simulated from models representative of design requirements. This methodology enables faster control development as time-consuming manual tuning of hyperparameters of the learning algorithm is automated. The design objective is to generate pareto-optimal design alternatives and to negotiate the best-possible compromise solutions based on user priorities.

3.3 Framework for V&V assessment

A typical simplified FCL design process is outlined in fig. 6. Initially, the underlying problem statement is transformed into control design requirements. These requirements are then translated into method-specific design objectives, followed by the development of a control architecture capable of achieving these objectives. At this point, the overarching design philosophy of the control law is established, leading to the subsequent phase of control law software and hardware integration and testing, which is an iterative process involving V&V. The process of testing the performance of the FCL against the intended function defined through control design requirements is known as validation. Verification involves the analysis and testing processes to ensure that the control algorithms perform as intended and are correctly implemented in both software and hardware. This step also involves decomposing higher system-level requirements into subsystem-level requirements. The verification step ensures that the subsystem performance meets the requirements, followed by subsystem integration and an integration test. This adheres to a classical - V framework for conducting V&V as shown in [35], where verified subsystems are integrated to form the complete system, and final system-level verification tests are conducted. In the context of the iADP control law, the iADP rate loop is segmented into subsystems, the iADP longitudinal rate loop and iADP lateral rate loop. Individual iADP control loops are further broken down into atomic subsystems of model learning, policy evaluation, and policy improvement subsystems as shown in fig. 4, and are verified and validated against requirements. Common controller subsystems such as the logger subsystem, switches/tuners, and data streaming subsystems are also verified and validated for functionality, both offline and during the ground testing phase. The validation of the controller is carried out through HIL ground tests and ultimately flight tests. Validation tasks include ensuring that the FCC can manage the computational load of the recursive algorithms, the accuracy of the controller commands on the ground, controller logging and real-time monitoring.

4 Verification & Validation of FCS

Verification involves analyzing and testing processes to confirm that FCS operates as intended, ensuring accurate implementation in both software and hardware. Validation, on the other hand, assesses the performance of FCS against defined requirements, using a set of criteria [35]. The V&V plan for iADP FCS is designed to ensure that the control design specifications mentioned in the table are fulfilled.

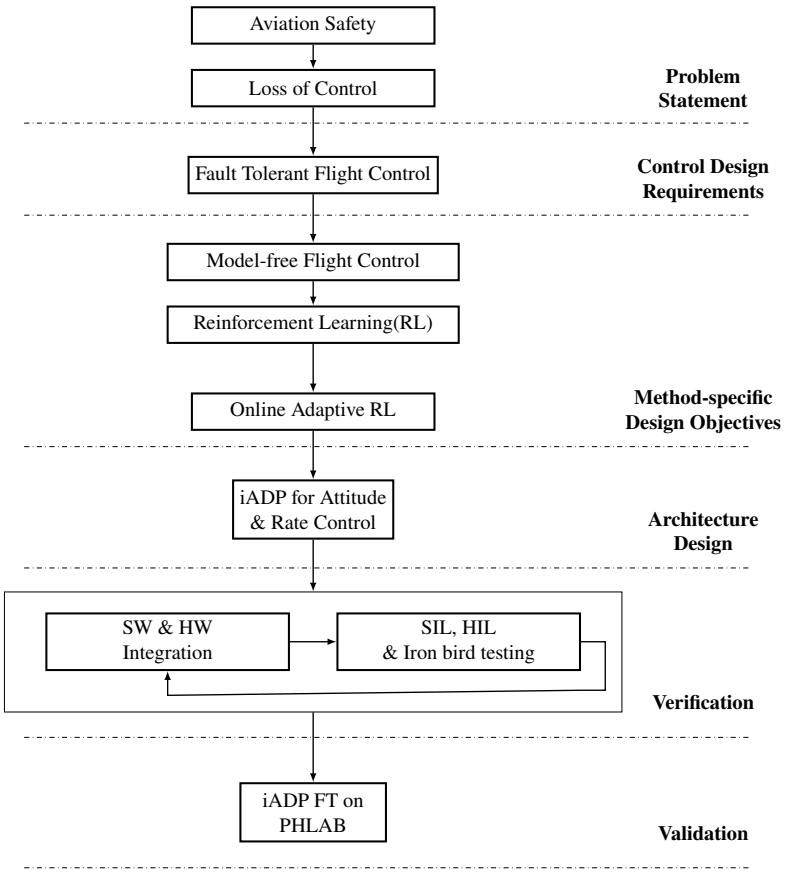

Fig. 6 V&V framework for iADP FCS Design for Fault Tolerant Flight Control.

Verification primarily utilizes offline simulation analysis, while validation involves Hardware in the Loop and flight testing methods. Due to the decoupled nature of the iADP controller, every V&V procedure is performed separately for each control loop. However, for the sake of brevity, the following sections present the results from the V&V of only one loop(either longitudinal or lateral).

4.1 Simulation Analysis

To verify the FCS against the control design requirement S03-OP (Robust against operating conditions) mentioned in table 1, we consider the flight conditions listed in table 2. This requirement ensures that the iADP controller maintains rate tracking across different operating conditions, such as varying velocities and altitudes. Assessing this requirement is crucial because aircraft dynamics change with flight conditions, which are typically managed using gain scheduling. Robustness against varying

Table 2 Flight Conditions for iADP FCS verification for robustness against operating conditions. FC denotes Flight Condition, V denotes True Airspeed and h denotes altitude.

	FC1	**FC2**	**FC3**
V [m/s]	85	95	105
h [m]	2000	2000	2000

operating conditions is essential due to the unpredictability of flight conditions before a flight test. The flight level (altitude) for testing depends on available airspace, which air traffic controllers communicate to pilots a few hours before the trial. Although pilots can manually adjust airspeed, the controller must not be overly sensitive to significant airspeed variations, given the lack of an auto-throttle on PH-LAB to maintain minimal deviations in airspeed during the flight trial. The current iADP controller cannot

handle large deviations in operating conditions because it does not incorporate velocity and altitude information into its cost function. Including airspeed and altitude data from sensors in the cost function could increase the size of state space in cost-to-go approximation, requiring more computations and necessitating exploration of these new states during maneuvers.

The left-aligned time plots from fig. 7 show the iADP longitudinal controller's pitch rate tracking performance across different flight conditions listed in table 2, using the PH-LAB aircraft model in a model-in-the-loop simulation setting. The controller is assessed in a continuous learning setting, meaning the parameters are updated at every time step. All flight conditions are evaluated with fixed hyperparameters. Despite being model-free, the absence of velocity and altitude information impacts the tracking performance as expected, with lower tracking error observed at higher velocities and vice versa.

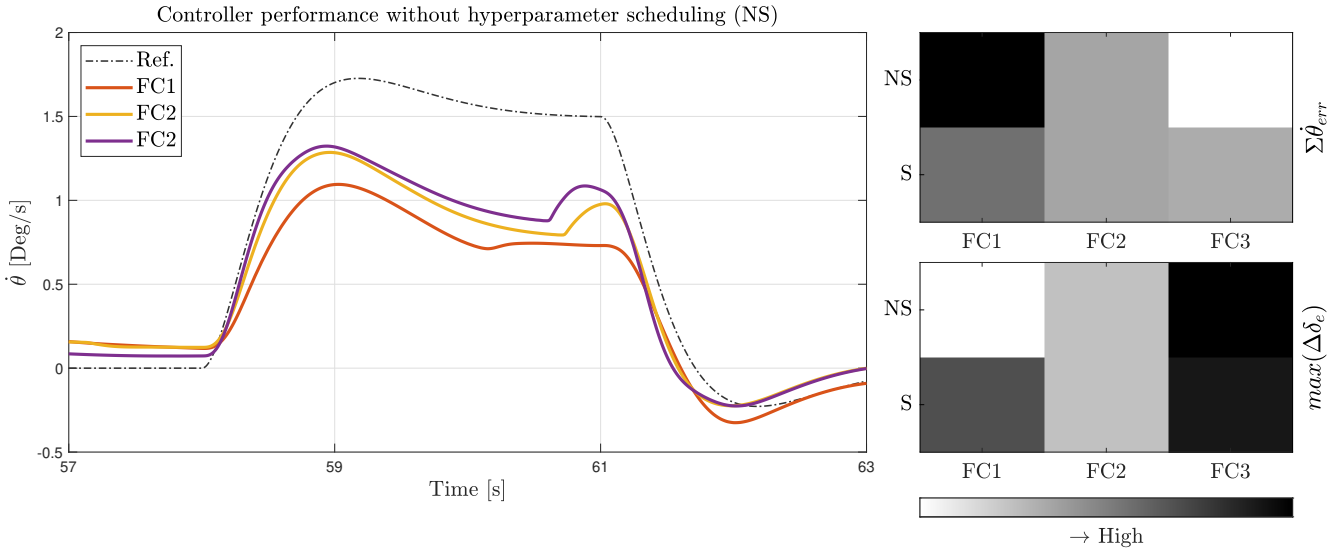

Fig. 7 Verification of Controller Robustness to Operating Conditions through Simulations (V&V of requirements S03-OP). NS refers to No Scheduling of hyperparameter Q with dynamic pressure, S refers to with Scheduling of hyperparameter Q with dynamic pressure. FC refers to Flight Condition.

To ensure consistent tracking performance across different flight conditions, another set of simulations was performed by scheduling the hyperparameter (the Q weighting matrix related to one-step tracking error) with dynamic pressure. These results are shown on the right-aligned heat map plots in fig. 7. The cumulative pitch rate tracking error is reduced with this hyperparameter scheduling, leading to more consistent tracking errors across the three flight conditions, compared to the variability observed without scheduling. Based on these results, the Q weighting matrix is scheduled with dynamic pressure for flight tests in both longitudinal and lateral control loops. This scheduling maintains the rate control loop as model-free since airspeed and altitude sensors are available on the aircraft, and these values are accessible to the controller.

Table 3 Nominal hyperparameters for iADP longitudinal rate control(referred to as Lon.) and iADP lateral rate control(referred to as Lat.). γ_{RLS} refers to forgetting factor of the Recursive Least Squares (RLS) algorithm, γ refers to discount factor in Bellman's equation, Q and R refer to weighing matrices in one step cost function.

	γ_{RLS}	γ	Q	R
Lon.	0.99593	0.4	50	1
Lat.	0.99544	0.4	40	$\begin{pmatrix} 1 & 0 \\ 0 & 200 \end{pmatrix}$

Another key design requirement for learning-based controllers is to ensure low sensitivity to the hyperparameters of the algorithms, defined as control design requirement S08-SP (Low sensitivity to hyperparameters). To conduct the V&V for this requirement, a set of nominal hyperparameters for the iADP algorithm are defined, as listed in table 3. The hyperparameters include γ_{RLS}, the forgetting factor of the Recursive Least Squares (RLS) algorithm, which affects the model prediction error and indirectly influences the cost-to-go estimate. γ is the discount factor in Bellman's equation, which discounts future costs. Q and R are weighting matrices in the one-step cost function, balancing the trade-off between the tracking and the control effort required.

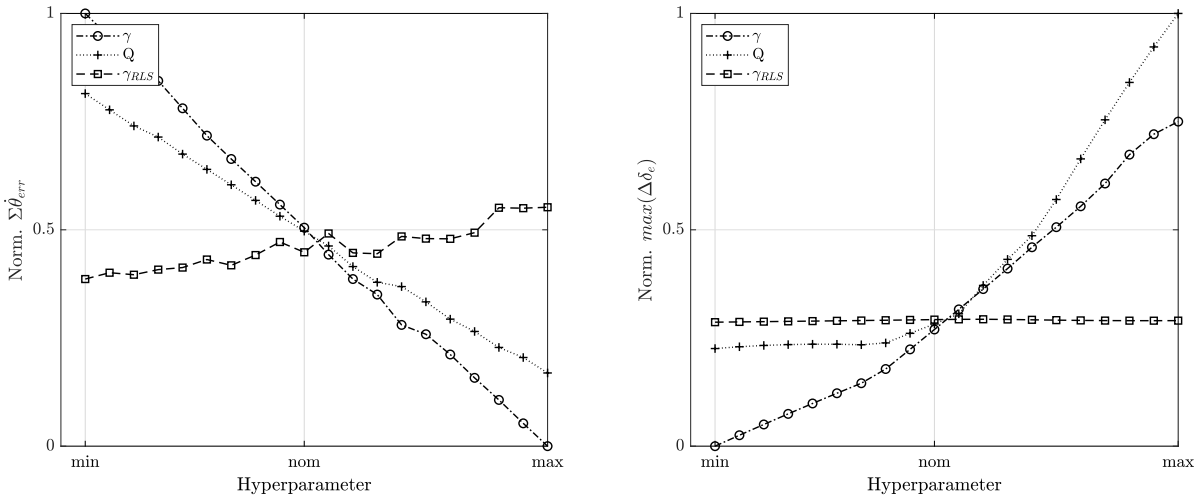

Fig. 8 Verification of Controller's low sensitivity to iADP hyperparameters using Simulations (V&V of requirements S08-SP). Norm. refers to Normalized values. min, nom, and max refer to the Minimum, Nominal, and Maximum values of the hyperparameter.

Figure 8 plots two key metrics for assessing the controller performance: $\Sigma\dot{\theta}err$, the cumulative pitch rate tracking error for the trial, and $max(\Delta\delta_e)$, the maximum value of the commanded increment to the elevator deflection. These plots illustrate how these metrics vary with changes in the hyperparameters, whose minimum and maximum values are defined w.r.t nominal value as $[nom(1 \pm \frac{\Delta}{100})]$. The Δ values are (10, 10, 1) for (γ, Q, γ_{RLS}) respectively.

Key observations from the plots include: The metrics vary smoothly with the hyperparameters. An increase in γ and Q positively correlates with reduced cumulative tracking error and negatively correlates with maximum elevator deflection. Sensitivity to R is not tested because the Q/R ratio is the primary influence on the metrics, which can be assessed by varying either Q or R. The correlation of γ_{RLS} with the metrics is weak, as this parameter directly affects incremental model prediction and only indirectly influences tracking error. Nonetheless, the controller performance remains smooth with respect to the forgetting factor. These results suggest that the controller performance is robust to variations in the hyperparameters, satisfying the low sensitivity of the hyperparameters requirement.

To ensure the V&V of control design requirement S06-RM (robustness to model uncertainties), we employ the MOPS tool's anti-optimization technique as detailed in Section 3.2. A significant challenge in verifying robustness for online adaptive control methods like iADP lies in assessing the necessary safety margins, given the limited theoretical foundations for adaptive control and stability margins. An alternative approach involves generating multiple system models with uncertainties [36] and evaluating the controller's performance across these models.

This involves a two-step procedure:

1) **Anti-Optimization Task:** For a given set of flight conditions and a nominal iADP controller, global search techniques such as genetic algorithm was used to find the combination of model

Table 4 Uncertain parameters of PH-LAB aircraft for worst case optimization. Parameters for longitudinal and lateral control are considered separately due to the decoupled iADP control structure. Physical units are not shown, because the uncertainties are expressed in terms of percentages (e.g. +0.2 refers to +20% of the nominal value). Center of gravity shifts are in meters. The symbols adhere to the conventional aircraft coefficient notation.

Longitudinal				Lateral			
Parameter	Min.	Nom.	Max.	Parameter	Min.	Nom.	Max.
$\Delta\delta_e$	-0.03	0	0.03	$\Delta\delta_a$	-0.03	0	0.03
ΔX_{cg}	-0.05	0	0.05	ΔY_{cg}	-0.03	0	0.03
ΔI_{yy}	-0.05	0	0.05	ΔI_{xx}	-0.1	0	0.1
				ΔI_{zz}	-0.08	0	0.08
ΔC_{m_q}	-0.1	0	0.1	ΔC_{l_p}	-0.1	0	0.1
				ΔC_{l_r}	-0.1	0	0.1
$\Delta C_{m_{\delta_e}}$	-0.1	0	0.1	$\Delta C_{l_{\delta_a}}$	-0.1	0	0.1
				$\Delta C_{l_{\delta_r}}$	-0.1	0	0.1

uncertainties that most severely violate the performance criteria, referred to as anti-optimization. This results in a worst-case model for the given controller.

2) **Hyperparameter Tuning:** This worst-case model is then used to tune the iADP controller's hyperparameters using another optimization routine that searches the hyperparameter space to optimize the defined criteria.

To perform anti-optimization, a set of critical uncertain parameters are defined (see table 4). These parameters, which differ for longitudinal and lateral control, are selected based on their relevance to the clearance of FCL. The uncertainty ranges are based on data from [19] and [37]. For example, uncertainties in the center of gravity position directly affect longitudinal stability and are linked to gain margin. Changes in inertia can also impact both longitudinal and lateral stability and gain margins, with these uncertainty effects varying depending on the aircraft type. Control effectiveness uncertainties can alter gain margins in the respective axes and are therefore considered. Changes in damping derivatives can affect the dynamic stability characteristics of the aircraft. A combination of the most relevant parameters is used to generate worst-case models.

Using these parameters, multiple nonlinear aircraft models are generated. The GA2 algorithm, an alternate implementation of a genetic algorithm based on enhanced techniques for selection, crossover, and mutation most documented in [38], implemented within the MOPS tool, searches for the combination of uncertain parameters and the nominal iADP controller, that results in the worst nonlinear performance criterion. Figure 9 shows the anti-optimization results for lateral control, with the left plot illustrating the lateral controller's response to a roll rate command under different parameter uncertainties. The top right plot shows how the criterion worsens with the number of evaluations as the genetic algorithm seeks the model that maximizes the tracking error. The bottom right plot displays the parameter combinations for different evaluations. The colors of all three plots correspond to the evaluation number. The worst-case model is highlighted with a thick red line, and this combination of uncertain parameters is then used to define the worst-case model for the subsequent iADP hyperparameter tuning.

Figure 10 illustrates the subsequent optimization of iADP hyperparameters to minimize the criteria for the identified worst-case model. This is also done using the MOPS tool, and SQP(Sequential Quadratic Programming) gradient-based local search algorithm is applied for this optimization procedure. Two criteria are defined as shown in the right-aligned plot and the objective of this optimization algorithm is to optimize the hyperparameter Q which minimizes these criteria. The left-aligned plot shows the roll

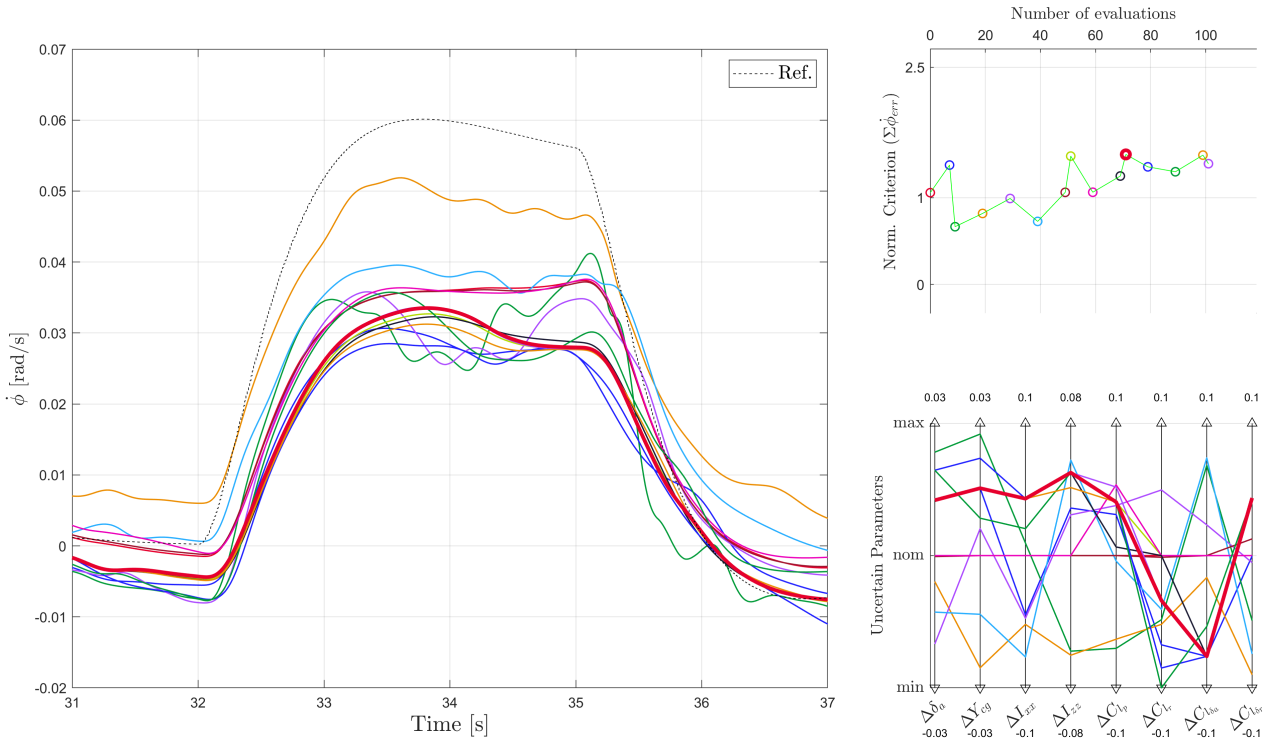

Fig. 9 Verification of robustness of the Controller Performance to Model Uncertainties based on worst case model search with genetic algorithm using Multi-Objective Parameter Synthesis (MOPS) (V&V of requirements S06-RM).

rate tracking performance for different evaluations of the optimization routine. The color of the plots is mapped to different evaluations as the algorithm searches for the best Q value to minimize the criteria.

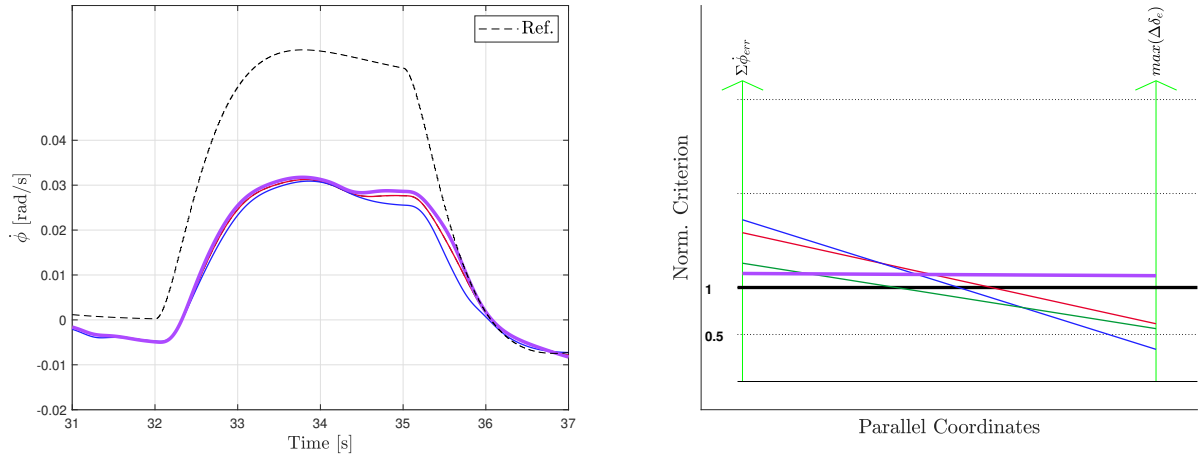

Fig. 10 Verification of robustness of the Controller Performance to Model Uncertainties based on hyperparameter(Q) tuning for the worst-case model using Multi-Objective Parameter Synthesis (MOPS) (V&V of requirements S06-RM).

An advantage of this model-based clearance of control laws for robustness to model uncertainties is its applicability to model-free control methods like iADP. This is possible because data can be generated from various models that incorporate these parametric uncertainties. Another advantage is the ability to define nonlinear criteria and use a nonlinear model for this process. While this study focused on a limited set of criteria and hyperparameters, the procedure can be scaled to fine-tune controllers for additional criteria like rise time and overshoot, and can optimize multiple loops simultaneously. A similar approach was adopted for iADP longitudinal control, considering the parametric uncertainties for longitudinal dynamics in table 4, thereby meeting control design requirement S06-RM as outlined in table 1.

4.2 Flight Test Results

Table 5 provides a summary of the flight test campaigns conducted in November '22, August '23, and December '23, as part of the iADP-based FCS validation process. It details a concise overview of the flight trials conducted, including a brief description and the outcome of each trial.

The first successful trial of the iADP algorithm for longitudinal rate control is presented in fig. 11. 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. The 3211 signal also serves the functionality of a persistently exciting signal. Results comparing measured longitudinal states against predictions from the RLS algorithm show a good fit.

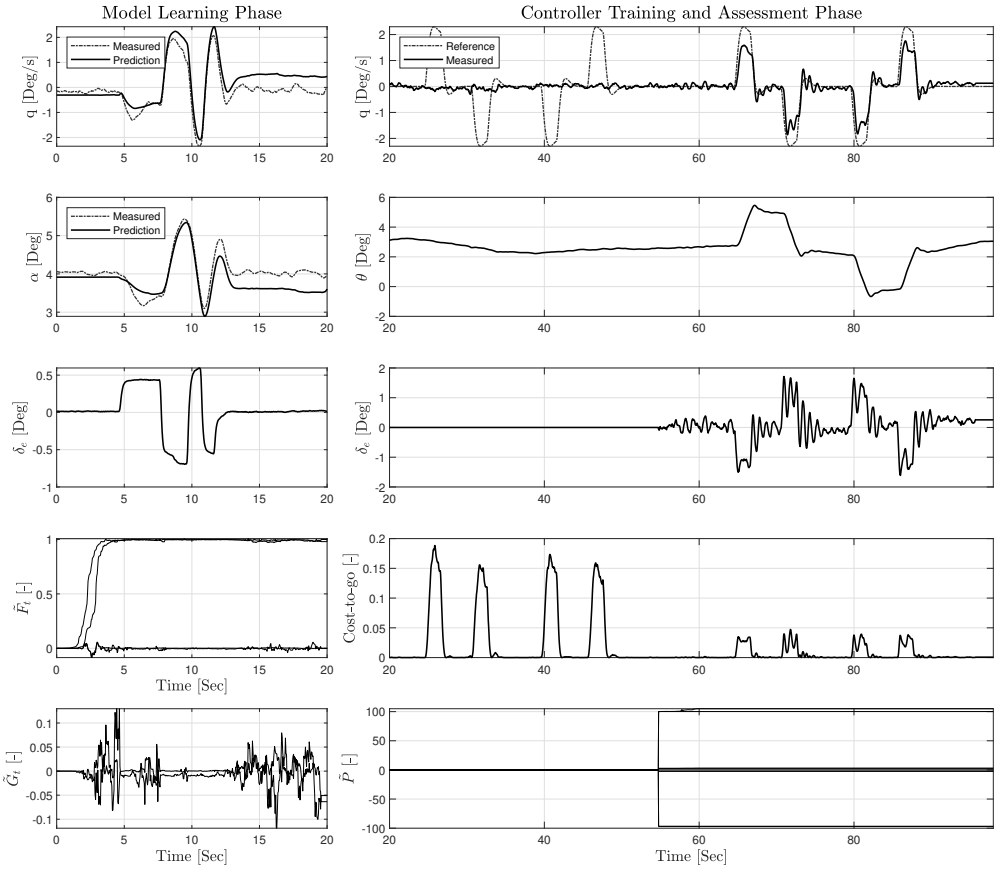

Fig. 11 Flight Test Data(Trial ID : N22-F2-T3), PH-LAB performing a Longitudinal maneuver: iADP Flight Control Law (FCL) designed for Pitch Rate Capture. Sequential Learning Approach (SLA) with fixed parameters post model learning and controller training.

The right-aligned plots depict results from the controller training and assessment phase. The controller training phase lasted for 40 seconds(from 20 to 60 seconds), during which the controller loop is closed and an automatic pitch rate reference command to evaluate the policy is generated. The controller along with model parameters estimated from the 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^{th} second onwards). After the controller training phase concludes, the parameters are fixed and passed to the subsequent Controller Assessment phase. During the Controller Assessment phase, the controller's objective is to track the reference command. The results, from 60 to 100 seconds,

Table 5 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, D23 for December 23, 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	✓
D23-F1-T1	Roll	95	3050	Nominal Config.; Initial checks	✓
D23-F1-T2	Roll	95	3050	Nominal Config.; Initial checks	✓
D23-F1-T3	Roll	95	3050	Nominal Config.; Reproducible Continuous Learning	✓
D23-F1-T4	Roll	95	3050	Nominal Config.; Reproducible Continuous Learning	✓
D23-F1-T5	Roll	95	3050	Nominal Config.; Reproducible Continuous Learning	✓
D23-F1-T6	Roll	95	3050	Nominal Config.; Reproducible Continuous Learning	✓
D23-F1-T7	Roll	94	3050	Flaps 15° Config. ; Stable Continuous Learning	✓
D23-F1-T8	Roll	95	3050	Landing Gear Down Config. ; Stable Continuous Learning	✓
D23-F1-T9	Roll	95	3050	Flaps 40° Config. ; Slightly Oscillatory tracking	✓
D23-F1-T10	Roll	95	2800	Active Config. change; N -> GD -> N	✓
D23-F1-T11	Roll	95	2750	Active Config. change; N -> F15° -> N	✓
D23-F1-T12	Roll	95	2800	Active Config. change; N -> F15° -> F40° -> F15° -> N	×
D23-F1-T13	Roll	95	3000	Active Config. change; N -> F15° -> N	✓
D23-F1-T14	Roll	95	3020	Active Config. change; N -> F15° -> N	✓
D23-F1-T15	Roll	95	3050	Active Config. change; N -> F15° -> F40° -> F15° -> N	×

show the aircraft tracking a pitch rate command, which can also be interpreted as a reduction in the cost-to-go plot.

Figure 12 present the flight test results from trial A23-F2-T2. In this trial, the controller's objective is to command the aircraft to follow a roll rate reference and also to demonstrate the capability of stable continuous learning. Ft and Gt plots illustrate the performance of the incremental model identification stage throughout the entire maneuver. Given the algorithm's lack of prior knowledge about the model, doublets are commanded at the Aileron and Rudder initially to assist the identification process. The first plot shows reference and measured roll rate output, the second plot shows the roll angle and the third plot shows the aileron and rudder output commands to achieve the rate tracking task. Comparing reference to the measure roll rate output, a good tracking response is observed. The bottom plot shows the evolution of the controller parameters throughout the maneuver, respectively.

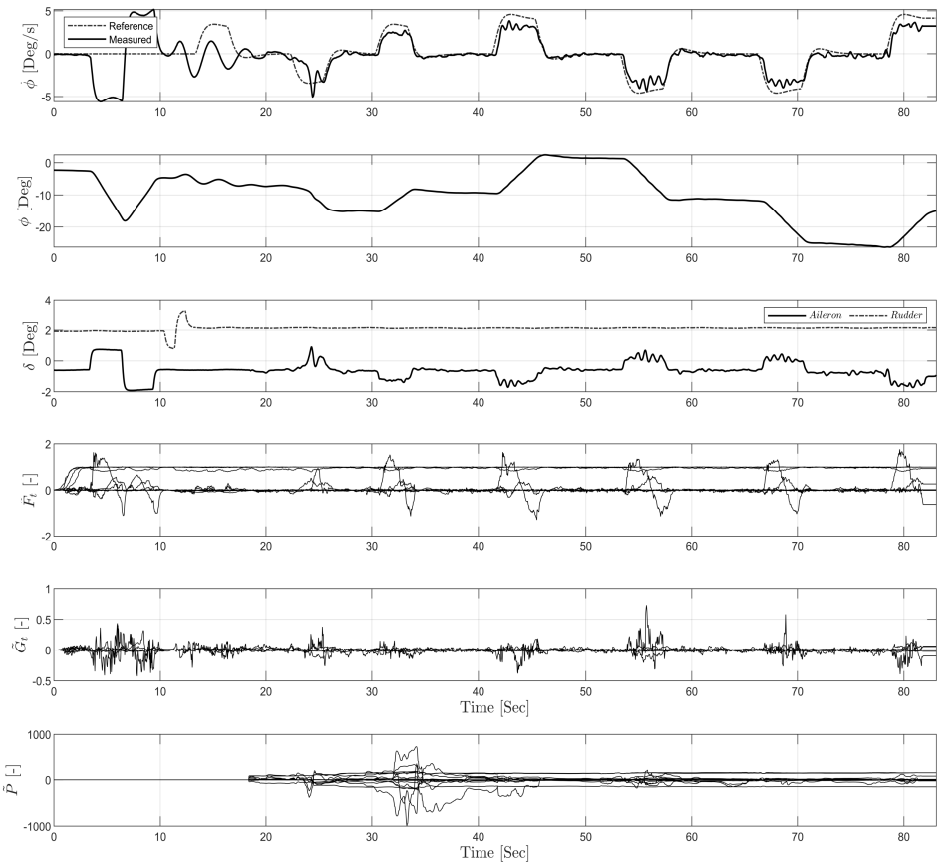

Fig. 12 Flight Test Data(Trial ID: A23-F2-T2), PH-LAB performing a Lateral maneuver: Plots show performance from Controller Training and Assessment Phase. iADP Flight Control Law (FCL) designed for Roll Rate Capture. Continuous Learning Approach (CLA) with real-time parameter adaptation.

The flight test data presented in fig. 11 and fig. 12 validates that the FCS fulfills the design specifications outlined in Table 1, namely S01-MT (Minimize Rate Tracking Error), S05-MF (Agnostic to model), S07-SN(Sensitivity to Noise, Bias & Delays) and S11-CL (Continuous Learning).

To conduct V&V for the control design requirement S07-SN (sensitivity to sensor dynamics and other hardware effects), a comparison was made to assess the controller's performance with various fidelities of aircraft models. This step is crucial for identifying any issues with the flight control software, interfaces, and the controller's sensitivity to real hardware and software constraints. Although Hardware-in-the-Loop (HIL) simulations, as detailed in fig. 3, were planned before the flight tests, the unavailability of the aircraft model to be interfaced as a DUECA module made this verification step not possible. The

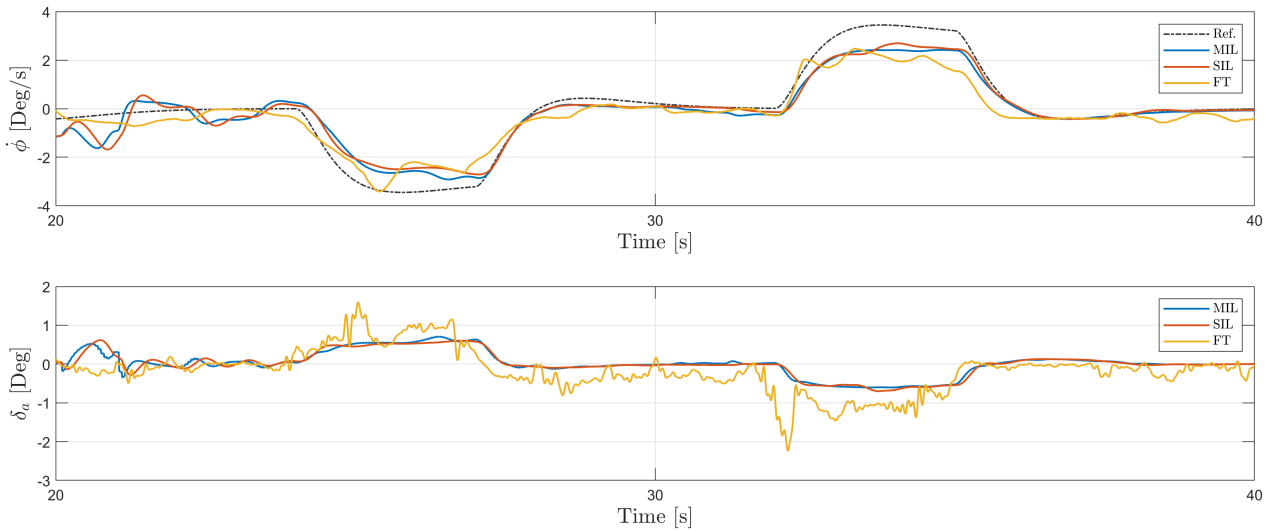

Fig. 13 Comparison of Controller Performance: Model in the Loop(MIL) vs. Software in the Loop(SIL) vs. Flight Test(FT) Data considering a Continuous Learning Approach (CLA) (V&V of requirements S01-MT and S07-SN).

controller's performance is shown in fig. 13. The data compare the responses across different simulation fidelities. The plots show similarity in the controller's response for Model-in-the-Loop (MIL), Software-in-the-Loop (SIL) simulations, and flight test data.

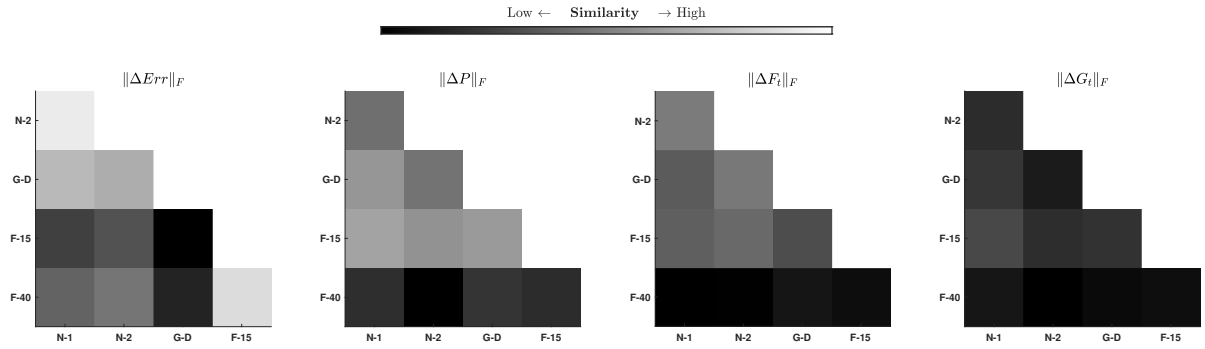

Fig. 14 Quantifying similarity index of Adaptive control parameters using Frobenius Norm across Aircraft Configurations. Tracking error is less sensitive to aircraft configuration changes in Continuous Learning Approach (CLA). Kernel Matrix Parameters $\tilde{P}$ are correlated to Aircraft Configurations, 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). Validation of Controller Adaptability to Aircraft Configurations based on Flight Test Data (V&V of requirements S02-AC, S05-MF, S11-CL).

To quantify the adaptability of the controller to aircraft configurations (control design requirement S02-AC), time-evolving parameters are compared against different configurations. Four different metrics are considered for comparison, **Tracking Error:** Evaluates controller tracking performance, assessing the control objective, **Incremental Model State Matrix ($\tilde{F}_t$):** Measures identified incremental model parameters related to state transitions, containing state derivatives, **Incremental Model Control Effectiveness Matrix ($\tilde{G}_t$):** Measures identified incremental model parameters related to control effectiveness, containing control derivatives, and **Kernel Matrix ($\tilde{P}$):** Measures learned control policy parameters. The Frobenius norm of the difference in matrices is selected to assess the similarity of these values. For example, the norm for comparing N-1 configuration data with Flap 15 configuration is defined as follows:

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

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. 14. 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.

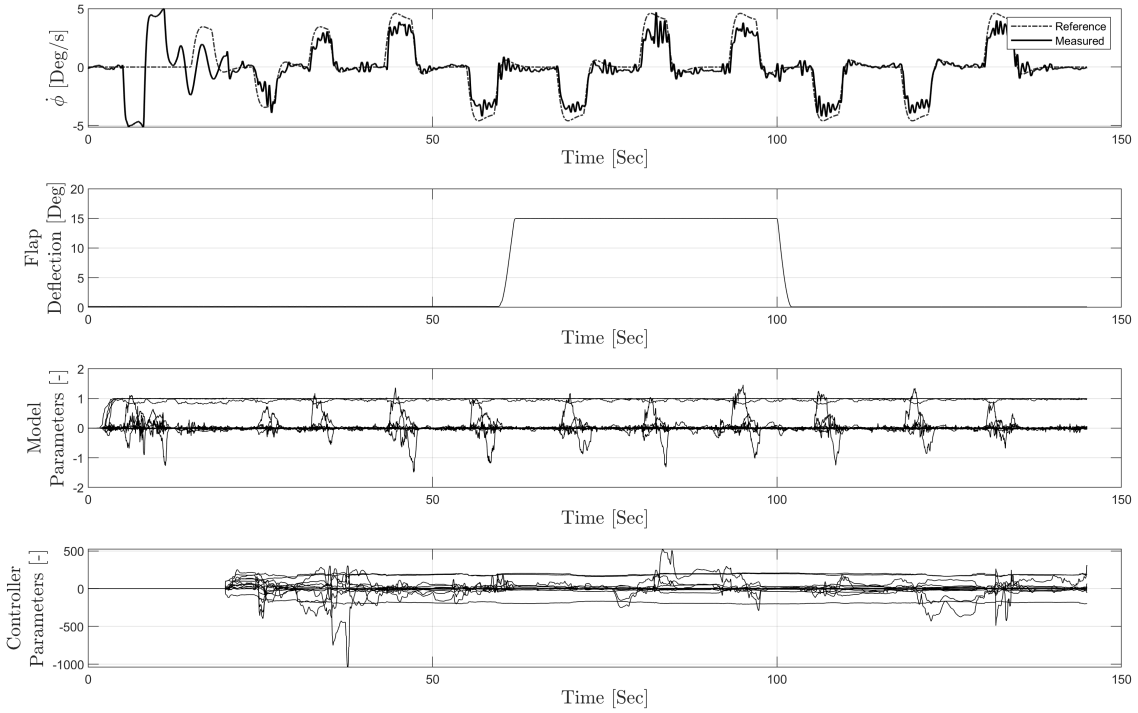

Fig. 15 Flight Test Data(Trial ID: D23-F1-T11), PH-LAB performing a Lateral maneuver: Plots show performance from Controller Training and Assessment Phase. iADP Flight Control Law (FCL) designed for Roll Rate Capture. Continuous Learning Approach (CLA) with real-time parameter adaptation as the aircraft transitions between configurations: from Nominal Flaps 0° Extension to Flaps 15° Extension, and back to the nominal configuration. Validation of Controller Adaptability to Aircraft Configuration Based on Flight Test Data (V&V of requirements S02-AC, S05-MF, S11-CL).

To further validate the control design requirement (S02-AC), multiple flight trials were conducted where the aircraft undergoes configuration changes while the controller is learning (refer to table 5). Flight test data from one of these live configuration change experiments is shown in fig. 15. Around 60 seconds, the flaps were extended from the nominal 0° configuration to 15° and then back to 0° at 100 seconds, while the controller was active and tracking a roll rate command. The configuration change did not influence the tracking response, and the controller and model parameters are being updated in real time. While the analysis in fig. 14 considers configuration changes before controller engagement, the flight trial in fig. 15 involves configuration changes after the controller is engaged and actively updating its parameters. This validates the control design requirement S02-AC.

To assess the reproducibility of the controller performance, flight test data from different trials from different aircraft configurations is summarized in fig. 16. There is a high level of similarity in the

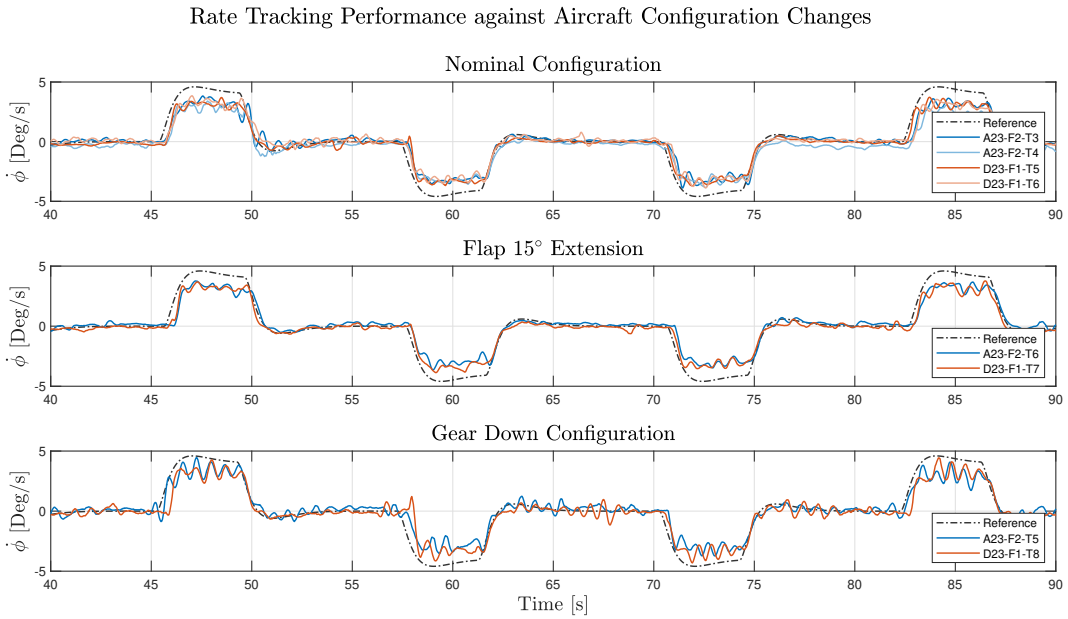

Fig. 16 Validation of reproducibility of the Controller Performance in similar conditions based on Flight Test Data (V&V of requirements S04-RR). The labels of the flight test ID are listed in table 5

controller performance, as flight test data from same configurations appear to be similar. This validates the control design requirement S04-RR for reproducible results.

5 Conclusion

The reuse of V&V procedures, originally designed for model based control methods and adapting them for online adaptive FCL, expedited the clearance of control laws for flight tests. This resulted in successful maiden flight test, where the controller was able to command, without any prior knowledge of the aircraft, relying solely on the real-time data collected on the go. While the V&V procedures detailed in this work are applied for RL-based online adaptive iADP control law, they can be extended to offline training-based RL controllers or other adaptive FCLs. Furthermore, in future, these V&V procedures can be adapted to meet control design requirements such as fine-tuning controller performance, disturbance rejection, direct fault tolerance, and handling qualities using the aforementioned tools, methods, and framework.

The rapid advancement of Machine Learning-based FCS could potentially outpace current V&V procedures. It is crucial to define cost-effective, tailored V&V plans for online learning algorithms, to expedite the development schedules of emerging safety-critical FCS. This research marks a significant step towards V&V of online learning-based adaptive FCS, and the outcome of this controller design process, including V&V, could aid in developing and certifying machine learning based controllers.

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, Coen de Visser, 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, Christian Weiser, Daniel Milz, Stefan Langen, and Christina Schreppel, for their valuable support and expertise.

References

[1] International Air Transport Association (IATA). Loss of control in-flight accident analysis report. Technical report, Montreal, Geneva, 2019.

[2] Christine M. Belcastro. Validation and verification (v&v) of safety-critical systems operating under off-nominal conditions. Technical report, NASA Langley Research Center, 2012. DOI: 10.1007/978-3-642-22627-4_20.

[3] Zachary T. Dydek, Anuradha M. Annaswamy, and Eugene Lavretsky. Adaptive control and the nasa x-15-3 flight revisited. *IEEE Control Systems Magazine*, 30(3):32–48, 2010. DOI: 10.1109/MCS.2010.936292.

[4] Anuradha M. Annaswamy and Alexander L. Fradkov. A historical perspective of adaptive control and learning. *Annual Reviews in Control*, 52:18–41, 1 2021. DOI: 10.1016/j.arcontrol.2021.10.014.

[5] Richard Bellman. Dynamic programming. *Science*, 153:34 – 37, 1957.

[6] Steven Joseph Bradtke. *Incremental Dynamic Programming for On-Line Adaptive Optimal Control*. PhD thesis, University of Massachusetts, USA, 1995. UMI Order No. GAX95-10446.

[7] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[8] Silvia Ferrari and Robert F. Stengel. Online adaptive critic flight control. *Journal of Guidance, Control, and Dynamics*, 27(5):777–786, 2004. DOI: 10.2514/1.12597.

[9] Stefan Heyer, Dave Kroezen, and E. Van Kampen. Online adaptive incremental reinforcement learning flight control for a cs-25 class aircraft. *AIAA Scitech 2020 Forum*, 1 2020. DOI: 10.2514/6.2020-1844.

[10] Ramesh Konatala, Erik-Jan Van Kampen, and Gertjan Looye. Reinforcement learning based online adaptive flight control for the cessna citation II(PH-LAB) aircraft. In *AIAA Scitech 2021 Forum*. American Institute of Aeronautics and Astronautics, 01 2021. DOI: 10.2514/6.2021-0883.

[11] Ramesh Konatala, Erik-Jan Van Kampen, Gertjan Looye, Daniel Milz, and Christian Weiser. Flight testing reinforcement learning based online adaptive flight control laws on cs-25 class aircraft. In *AIAA Scitech 2024 Forum*. American Institute of Aeronautics and Astronautics, 01 2024. DOI: 10.2514/6.2024-2402.

[12] J. Schumann, P. Gupta, and S. Jacklin. Toward verification and validation of adaptive aircraft controllers. In *2005 IEEE Aerospace Conference*, pages 1–6, 2005. DOI: 10.1109/AERO.2005.1559606.

[13] Radio Technical Commission for Aeronautics (RTCA). Software considerations in airborne systems and equipment certification. Technical report, Radio Technical Commission for Aeronautics (RTCA), 12 1992.

[14] Stephen A. Jacklin. Closing the certification gaps in adaptive flight control software. *AIAA Guidance, Navigation and Control Conference and Exhibit*, 6 2008. DOI: 10.2514/6.2008-6988.

[15] EASA. Easa publishes artificial intelligence concept paper issue 2 'guidance for level 1 & 2 machine learning applications', mar 2024.

[16] Ye Zhou, E. Van Kampen, and Q.P. Chu. Nonlinear adaptive flight control using incremental approximate dynamic programming and output feedback. *Journal of Guidance, Control, and Dynamics*, 40(2):493–496, feb 2017. DOI: 10.2514/1.g001762.

[17] Pedro Miguel Dias, Ye Zhou, and Erik-Jan Van Kampen. Intelligent nonlinear adaptive flight control using incremental approximate dynamic programming. In *AIAA Scitech 2019 Forum*, 2019. DOI: 10.2514/6.2019-2339.

[18] Peter Zaal, Daan Pool, Alexander in 't Veld, Ferdinand Postema, Max Mulder, Marinus van Paassen, and Jan Mulder. Design and certification of a fly-by-wire system with minimal impact on the original flight controls. In *AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, jun 2009. DOI: 10.2514/6.2009-5985.

[19] Fabian Grondman, Gertjan Looye, Richard O. Kuchar, Q Ping Chu, and Erik-Jan Van Kampen. Design and flight testing of incremental nonlinear dynamic inversion-based control laws for a passenger aircraft. In *2018 AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, jan 2018. DOI: 10.2514/6.2018-0385.

[20] A. Muis, J. Oliveira, and J. A. Mulder. Development of a flexible flight test instrumentation system. In *17th SFTE (EC) Symposium*, 2006.

[21] Wim van Ekeren, Gertjan Looye, Richard O. Kuchar, Q Ping Chu, and Erik-Jan Van Kampen. Design, implementation and flight-tests of incremental nonlinear flight control methods. In *2018 AIAA Guidance, Navigation, and Control Conference*, 2018. DOI: 10.2514/6.2018-0384.

[22] Christian Weiser, Daniel Ossmann, Richard O. Kuchar, Reiko Müller, Daniel M. Milz, and Gertjan Looye. Flight testing a linear parameter varying control law on a passenger aircraft. In *AIAA Scitech 2020 Forum*. American Institute of Aeronautics and Astronautics, jan 2020. DOI: 10.2514/6.2020-1618.

[23] Twan Keijzer, Gertjan Looye, Q Ping Chu, and Erik-Jan Van Kampen. Design and flight testing of incremental backstepping based control laws with angular accelerometer feedback. In *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, jan 2019. DOI: 10.2514/6.2019-0129.

[24] Tijmen Pollack, Gertjan Looye, and Frans Van der Linden. Design and flight testing of flight control laws integrating incremental nonlinear dynamic inversion and servo current control. In *AIAA Scitech 2019 Forum*. American Institute of Aeronautics and Astronautics, jan 2019. DOI: 10.2514/6.2019-0130.

[25] Daniel Milz, Marc May, and Gertjan Looye. Flight testing air data sensor failure handling with hybrid nonlinear dynamic inversion. In *Proceedings of the 2024 CEAS EuroGNC conference*, 2024.

[26] Y. Zhang, C. C. de Visser, and Q. P. Chu. Aircraft damage identification and classification for database-driven online flight-envelope prediction. *Journal of Guidance, Control, and Dynamics*, 41(2):449–460, 2018. DOI: 10.2514/1.G002866.

[27] C.A.A.M. Van der Linden. Dasmat-delft university aircraft simulation model and analysis tool: A matlab/simulink environment for flight dynamics and control analysis. Technical report, Delft University of Technology, Faculty of Aerospace Engineering, 12 1998.

[28] M. Baarspul, J.A. Mulder, A.H.M. Nieuwpoort, and J.H. Breeman. Mathematical model identification for flight simulation, based on flight and taxi tests. Technical report, Delft University of Technology, Faculty of Aerospace Engineering, 1988.

[29] M. Mulder, B. Lubbers, P.M.T. Zaal, M. M. van Paassen, and J. A. Mulder. Aerodynamic hinge moment coefficient estimation using automatic fly-by-wire control inputs. In *AIAA Modeling and Simulation Technologies Conference*, 2009. DOI: 10.2514/6.2009-5692.

[30] H.-D. Joos. A methodology for multi-objective design assessment and flight control synthesis tuning. *Aerospace Science and Technology*, 3(3):161–176, 1999. ISSN: 1270-9638. DOI: 10.1016/S1270-9638(99)80040-6.

[31] M. Van Paassen, Olaf Stroosma, and J. Delatour. DUECA - Data-driven activation in distributed real-time computation. In *Modeling and Simulation Technologies Conference*, Denver,CO,U.S.A., Aug. 2000. American Institute of Aeronautics and Astronautics. DOI: 10.2514/6.2000-4503.

[32] Pepijn A. Scholten, Marinus M. van Paassen, Q. Ping Chu, and Max Mulder. Variable stability in-flight simulation system based on existing autopilot hardware. *Journal of Guidance, Control, and Dynamics*, 43(12):2275–2288, dec 2020. DOI: 10.2514/1.g005066.

[33] Gertjan Looye and Hans-Dieter Joos. Design of robust dynamic inversion control laws using multi-objective optimization. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics, 2001. DOI: 10.2514/6.2001-4285.

[34] Hans-Dieter Joos. Worst-case parameter search based clearance using parallel nonlinear programming methods. In Andreas Varga, Anders Hansson, and Guilhem" Puyou, editors, *Optimization Based Clearance of Flight Control Laws: A Civil Aircraft Application*, volume 416, pages 149–159. Springer Berlin Heidelberg, 01 2012. ISBN: 978-3-642-22626-7. DOI: 10.1007/978-3-642-22627-4_8.

[35] G.S. Tallant, P. Bose, J.M. Buffington, V.W. Crum, R.A. Hull, T. Johnson, B. Krogh, and R. Prasanth. Validation & verification of intelligent and adaptive control systems. In *2005 IEEE Aerospace Conference*, pages 1–11, 2005. DOI: 10.1109/AERO.2005.1559594.

[36] D. Moormann A. Varga, G. Looye and G. Gräbel. Automated generation of lft-based parametric uncertainty descriptions from generic aircraft models. *Mathematical and Computer Modelling of Dynamical Systems*, 4(4):249–274, 1998. DOI: 10.1080/13873959808837082.

[37] Christopher Fielding, Andras Varga, Samir Bennani, and Michiel Selier. *Selected Clearance Criteria for HIRM+RIDE, Advanced Techniques for Clearance of Flight Control Laws*, pages 151–165. Springer Berlin Heidelberg, 2002. DOI: 10.1007/3-540-45864-6.

[38] A. Eiben and Jim Smith. *Introduction To Evolutionary Computing*, volume 45. Computing Reviews, 01 2003. ISBN: 978-3-642-07285-7. DOI: 10.1007/978-3-662-05094-1.