Longitudinal Handling Qualities Evaluation for Soft Actor-Critic Deep Reinforcement Learning Flight Control

Jansen, Hidde; van Kampen, E.

DOI
10.2514/6.2025-2794

Publication date
2025

Document Version
Final published version

Published in
Proceedings of the AIAA SCITECH 2025 Forum

AIAA SciTech Forum
6-10 January 2025, Orlando, FL
AIAA SCITECH 2025 Forum

10.2514/6.2025-2794

Longitudinal Handling Qualities Evaluation for Soft Actor-Critic Deep Reinforcement Learning Flight Control

H. Jansen *, Erik-Jan van Kampen†
Delft University of Technology, Kluyverweg 1 2629HS, Delft, The Netherlands

Reinforcement Learning applied to flight control has shown to have several benefits over classical, linear flight controllers, as it eliminates the need for gain scheduling and it could provide fault-tolerance. The application to civil aviation in practice, however, is non-existent as there are multiple safety concerns. This research demonstrates the evaluation of longitudinal Handling Qualities of the Soft Actor-Critic Deep Reinforcement Learning framework with the aim to translate the unpredictable black box of Reinforcement Learning into classical flight control terminology. The framework is applied to a pitch rate command system of a jet aircraft and shows robustness to off-nominal flight conditions, center of gravity shifts and biased sensor noise. Accurate tracking performance is achieved, while adhering to Level 1 longitudinal Handling Qualities for all conditions.

Nomenclature

$\mathbf{s}, \mathbf{a}, \mathbf{u}, \mathbf{x}$	=	observed state, action, input and state vectors
$t, \Delta t, N$	=	time step, sampling time and total number of time steps
$Q^{\pi}, Q_{\boldsymbol{\theta}}$	=	Q-value function and parameterized Q-value function
$\pi, \pi_{\boldsymbol{\phi}}$	=	policy and parameterized policy
$\boldsymbol{\theta}, \bar{\boldsymbol{\theta}}, \boldsymbol{\phi}$	=	critic, target critic and actor parameters
r, γ	=	scalar reward and discount factor
$\mathcal{H}, \bar{\mathcal{H}}$	=	entropy and target entropy
$\mathcal{D}, \mathcal{B}$	=	replay buffer and mini-batch taken from replay buffer
η, τ, κ	=	entropy coefficient, target critic smoothing factor and reward scaling factor
σ, μ	=	standard deviation and mean
λ_T, λ_S	=	temporal and spatial smoothing factors
$L_{Q_{\boldsymbol{\theta}}}, L_{\pi_{\boldsymbol{\phi}}}, L_{\eta}$	=	loss functions for critic, actor and entropy coefficient
L_T, L_S	=	temporal and spatial loss functions
CAP, CAP_e	=	Control Anticipation Parameter and equivalent Control Anticipation Parameter [$g^{-1}s^{-2}$]
$n_{z_{ss}}, q_{ss}$	=	steady state normal load factor [g] and steady state pitch rate [deg/s]
q, q_{cmd}, q_{ref}	=	pitch rate, pitch rate command and pitch rate reference, all in [deg/s]
$\dot{q}, \dot{q}_0, \dot{q}_{nd}$	=	pitch acceleration, instantaneous pitch acceleration and attenuation factor, all in [deg/s^2]
V, g	=	velocity [m/s] and gravitational acceleration [m/s^2]
ζ_{sp}, ω_{sp}	=	short period damping ratio and natural frequency [rad/s]
$T_{\theta_2}, K_{\theta}, \tau_e$	=	incidence lag, equivalent gain and equivalent time delay
α, θ	=	angle of attack [deg] and pitch angle [deg]
$\delta_e, \dot{\delta}_e, \delta_{e,act}$	=	elevator deflection angle [deg], elevator rate [deg/s] and elevator activity [deg/s]
N_{ω}, ω_k	=	number of logarithmically spaced natural frequency points and discrete frequency
$G(\omega_k), \phi(\omega_k)$	=	gain [dB] and phase angle [deg] sampled at discrete frequencies
κ_G, κ_{ϕ}	=	gain and phase angle scaling factors

*MSc Student, Faculty of Aerospace Engineering, Delft University of Technology
†Associate Professor, Faculty of Aerospace Engineering, Delft University of Technology. E.vanKampen@TUDelft.nl

I. Introduction

In the rapidly developing world of civil aviation, the demand for safety is crucial. Flight control systems of conventional aircraft heavily rely on gain scheduling, where the control gains are carefully selected for each flight regime within the operating envelope [1]. This requires complete knowledge of the dynamical aircraft model, obtained from costly wind tunnel tests and simulations. Even though the flight control systems are designed to ensure stable and safe behaviour, the majority of civil aviation accidents are still due to in-flight loss of control, often related to off-nominal flight conditions [2]. In the meanwhile, more unconvential aircraft are being developed, like vertical take-off and landing (VTOL) designs [3], morphing wing structures [4], v-shaped flying wings [5] and tilt-rotor aircraft [6]. These developments bring more challenges, as the aerodynamic models include nonlinearities and become more complex, showing that the need for model-free and fault-tolerant flight control is evident.

Reinforcement Learning (RL), relying on learning by interaction, is currently being actively researched and has been demonstrated to be a promising candidate for intelligent flight control. Originally it was developed in a discrete form using tabular methods, but the development of Neural Networks (NNs) as powerful function approximators provided a solution for the curse of dimensionality and enabled RL for continuous state and action spaces [7]. Several state-of-the-art frameworks can be found within the field of Approximate Dynamic Programming (ADP) and more specifically Adaptive Critic Designs (ACDs) [8]. Most of these methods, where an actor-critic structure is used for the selection of actions based on value functions, require an offline learning phase to learn an approximation of the dynamical model. Recent developments, however, have led to incremental ADP (iADP), where an incremental model is used that eliminates the need for offline learning. Within this field, Incremental Dual Heuristic Programming (IDHP) and Incremental Global Dual Heuristic Dynamic Programming (IGDHP) are considered state-of-the-art and have been successfully applied to control the longitudinal motion of a fighter jet [9], nonlinear missile model [10] and a business jet aircraft [11]. Although these methods provide high adaptive capabilities, there are concerns about reliability and safety when these methods are applied to fully control the inner and outer loops of flight control systems, as action policies change quickly, making it somewhat unpredictable.

The advancing research on Deep Neural Networks (DNNs) made Deep Reinforcement Learning (DRL) possible and shows potential for the application to flight control, since it is capable in dealing with high-dimensional state and action spaces and is characterized by its generalization power. Deep Deterministic Policy Gradient (DDPG) methods make use of the actor-critic structure to estimate policy and value functions and apply sampling from a replay buffer, making it an off-policy framework [12]. State-of-the-art methods like Twin-Delayed DDPG (TD3) [13] [14] and the Soft Actor-Critic (SAC) framework [15] are built upon DDPG and use target networks and double Q-value functions to improve learning stability and decrease sensitivity to hyperparameters. SAC exploits a stochastic policy and adds an entropy term to benefit exploration during training. It is a model-free RL method that has been proven to be robust to several failure cases, including center of gravity shifts and reduced control effectiveness, for a nonlinear coupled business jet aircraft, shown in Figure 1 [16]. DRL frameworks mainly address fault-tolerance due to their robustness and high generalization power.

Fig. 1 TU Delft Cessna Citation-II research aircraft PH-LAB. *

However, despite all the benefits, the real-world application of RL to flight control in civil aviation remains non-existent. The reason for this is that it becomes increasingly more complex to understand what an RL agent is doing with the development of state-of-the-art frameworks. Research has been performed on the "black box" analysis of state-of-the-art RL flight controllers under the name of explainable reinforcement learning, revealing that RL agents behave quasi-linear in non-linear flight regimes [17]. There is, however, an alternative approach that could aid in getting more insight in the underlying working mechanism of RL applied to flight control. Handling Qualities (HQ) describe

*https://cs.lr.tudelft.nl/citation/

the way an aircraft responds to pilot's inputs [18]. An extensive amount of literature exists on the desired HQ for flight control systems, providing guidelines and requirements that were originally developed to assess the performance of classical flight controllers [19]. The evaluation of HQ of RL flight control can assist in translating the complex black box structure of RL algorithms into well-known flight control terminology. At first glance, HQ evaluation of fully automatic RL flight control systems seems redundant as there is no pilot present, but it ensures that the aircraft is controlled as if a pilot were flying the aircraft. Furthermore, it is more likely that the implementation of RL in flight control occurs gradually and the inner control loops are replaced by RL while the outer loops remain to be controlled by the pilot or linear controllers, which further indicates the relevance of HQ evaluation.

The contribution of this research is to stimulate civil aviation to move towards RL flight control, by showing a proof-of-concept of the evaluation of longitudinal HQ for the state-of-the-art SAC framework applied to the Cessna Citation II PH-LAB research aircraft of the TU Delft. The research builds upon earlier work done on the development of a SAC controller for the same aircraft [16], but instead of developing an autonomous controller for the entire aircraft, this work centers on a pitch rate command system to make the implementation of RL flight control more realizable.

The theoretical background of the SAC framework and an overview of longitudinal HQ will be provided in section II. The implementation of the SAC framework for a pitch rate command system will be discussed in section III. The results will be presented and analyzed in section IV and the main conclusions of the research will be drawn in section V.

II. Background

This section contains background information on the selected RL algorithm and includes the approach of estimating the relevant longitudinal HQ for nonlinear flight control applications.

A. Soft Actor-Critic Framework

The underlying principle of RL is an agent acting in an environment and learning from its interactions by receiving feedback through rewards. More specifically, at time t the state of the environment $\mathbf{s}_t \in \mathbb{R}^n$ and the action of the agent $\mathbf{a}_t \in \mathbb{R}^m$ result in a scalar reward r_{t+1} and new state $\mathbf{s}_{t+1}$. The state transition function, represented by Equation 1, relies on the Markov property, i.e., the current state and action contain all the required information from history to estimate the subsequent state [7]. The goal of the RL agent is to find a policy π, that maximizes the cumulative reward over time. The SAC framework is based on a stochastic policy, meaning that actions are sampled from a policy distribution as specified by Equation 2.

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

$$\mathbf{a}_t \sim \pi(\cdot \mid \mathbf{s}_t) \tag{2}$$

The expected sum of future rewards, as a results of following the policy π and starting from the current state $\mathbf{s}_t$ and action $\mathbf{a}_t$, is incorporated in the Q-value function as outlined in Equation 3. A discount factor γ is included to provide the ability to adapt the balance between short- and long-term future rewards of the learning episode consisting of N time steps. The Q-value function gives an indication of how valuable the state-action pair is when following the current policy. The recursive property of Equation 3 is visible in Equation 4, better known as the Bellman equation.

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathop{\mathbb{E}}_{\pi}\left[\sum_{k=0}^{N} \gamma^k r_{t+k+1} \mid \mathbf{s}_t, \mathbf{a}_t\right] \tag{3}$$

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathop{\mathbb{E}}_{\pi}\left[r_{t+1} + \gamma Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})\right] \tag{4}$$

A key characterizing feature of the SAC framework is the use of entropy $\mathcal{H}$, which is computed with the log-likelihood function according to Equation 5. It gives an indication of the randomness of the policy π and therefore introduces the ability to make a trade-off between exploration and exploitation while learning. Finding a policy which exploits high rewards while also incorporating randomness to a high degree is beneficial to avoid converging quickly to local-optima and contributes to the robustness of the agent.

$$\mathcal{H}(\pi(\cdot \mid \mathbf{s}_t)) = \mathop{\mathbb{E}}_{\mathbf{a}\sim\pi}\left[-\log \pi(\mathbf{a} \mid \mathbf{s}_t)\right] \tag{5}$$

The SAC frameworks evolves around an actor-critic structure, where the actor and critic generate estimates of the policy and Q-value function respectively, using function approximators in the form of Deep Neural Networks (DNNs). Furthermore, the learning process is offline and state transition samples $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_{t+1})$ are stored in a replay buffer $\mathcal{D}$. The off-policy learning property of the SAC framework enables the agent to learn from the past by using a mini-batch $\mathcal{B}$ with state transition samples obtained from the replay buffer [15].

1. Critic

The critic estimates the Q-value function with the parameter vector $\boldsymbol{\theta}$. Equation 4 is modified with an entropy term to account for the randomness of the policy distribution, estimated by the actor with parameter vector $\boldsymbol{\phi}$, to form Equation 6. The entropy term includes a minus sign, since the log-likelihood of the policy distribution generally outputs negative values. Furthermore, the entropy is multiplied with the entropy coefficient η, which essentially indicates the weight of the contribution of the entropy term. The state transitions $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ are sampled from the mini-batch $\mathcal{B}$, whereas the next action is sampled from the parameterized policy distribution $\pi_{\boldsymbol{\phi}}$.

$$Q_{\boldsymbol{\theta}}(\mathbf{s}_t, \mathbf{a}_t) = \mathop{\mathbb{E}}_{\substack{(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \sim \mathcal{B} \\ \mathbf{a}_{t+1} \sim \pi_{\boldsymbol{\phi}}}} \left[r_{t+1} + \gamma Q_{\boldsymbol{\theta}}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - \eta \log \pi_{\boldsymbol{\phi}}(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) \right] \tag{6}$$

To stabilize the learning process, a target critic is introduced with parameters $\bar{\boldsymbol{\theta}}$. A soft update of the target critic parameters is performed with a smoothing factor τ according to the exponentially moving average: $\bar{\boldsymbol{\theta}}_{t+1} = \tau \bar{\boldsymbol{\theta}}_t + (1-\tau)\boldsymbol{\theta}_t$. The target critic prevents the Q-value function estimate to be updated in an aggressive and potentially unstable manner. Next to that, double Q-value function estimates are used for both the normal and target critic, to further improve stability. With a single Q-value function approximation, there is the risk of having an overestimation bias of the Q-value estimates, which is partially mitigated by introducing additional parallel Q-value function estimates and always selecting the lowest value for learning.

The loss function for each of the two critics consists of the squared difference between the critic Q-value estimate at time t and the target critic Q-value at time $t+1$, derived from the Bellman equation. In essence, the loss function is a form of the Temporal Difference (TD) error, modified to the SAC framework.

$$L_{Q_{\boldsymbol{\theta}_i}} = \mathop{\mathbb{E}}_{\substack{(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \sim \mathcal{B} \\ \mathbf{a}_{t+1} \sim \pi_{\boldsymbol{\phi}}}} \left[\left(Q_{\boldsymbol{\theta}_i}(\mathbf{s}_t, \mathbf{a}_t) - \left(r_{t+1} + \gamma \left(\min_{i=1,2} Q_{\bar{\boldsymbol{\theta}}_i}(\mathbf{s}_{t+1}, \mathbf{a}_{t+1}) - \eta \log \pi_{\boldsymbol{\phi}}(\mathbf{a}_{t+1} \mid \mathbf{s}_{t+1}) \right) \right) \right)^2 \right] \tag{7}$$

2. Actor

The actor resembles the stochastic policy, where the DNN with parameters $\boldsymbol{\phi}$ outputs the mean $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ and standard deviation $\boldsymbol{\sigma}_{\boldsymbol{\phi}}$ of a Gaussian policy distribution. In order to make the output of the DNN differentiable for parameter updates, $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ and $\boldsymbol{\sigma}_{\boldsymbol{\phi}}$ are reparameterized by sampling an action with a Gaussian noise vector $\boldsymbol{\epsilon}_t$ according to: $\mathbf{a}_t = \boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{s}_t) + \boldsymbol{\epsilon}_t \cdot \boldsymbol{\sigma}_{\boldsymbol{\phi}}(\mathbf{s}_t)$. A hyperbolic tangent squashing function is used to ensure that the action remains bounded. It should be noted that the action can be made deterministic, which is for instance necessary when evaluating the SAC agent, by taking the mean $\boldsymbol{\mu}_{\boldsymbol{\phi}}$ of the policy distribution. The loss function of the policy is implemented such that a combination of maximizing the expected return and entropy of the policy distribution is reached, while using the lowest Q-value approximation from the target critics:

$$L_{\pi_{\boldsymbol{\phi}}} = \mathop{\mathbb{E}}_{\substack{\mathbf{s}_t \sim \mathcal{B} \\ \mathbf{a}_t \sim \pi_{\boldsymbol{\phi}}}} \left[\eta \log \pi_{\boldsymbol{\phi}}(\mathbf{a}_t \mid \mathbf{s}_t) - \min_{i=1,2} Q_{\bar{\boldsymbol{\theta}}_i}(\mathbf{s}_t, \mathbf{a}_t) \right] \tag{8}$$

3. Entropy Adjustment

As the policy develops and its approximation improves while the agent is learning, the entropy should be adjusted during training as well. In the early stages of learning, a high degree of exploration is desired to find regions within the environment that yield a high return, whereas the emphasis of learning should be put on exploitation when the agent gets more experienced. It was therefore proposed to automatically adjust the entropy coefficient η, in a way that the entropy remains above a minimum threshold, i.e., the target entropy $\bar{\mathcal{H}}$. The loss function for entropy coefficient is defined by Equation 9 and it was empirically found that when the target entropy is set to the negative of the dimension of the action space ($\bar{\mathcal{H}} = -m$), it leads to stable results [20].

$$L_{\eta} = \mathop{\mathbb{E}}_{\substack{\mathbf{s}_t \sim \mathcal{B} \\ \mathbf{a}_t \sim \pi_{\boldsymbol{\phi}}}} \left[\eta \log \pi_{\boldsymbol{\phi}}(\mathbf{a}_t \mid \mathbf{s}_t) - \eta \bar{\mathcal{H}} \right] \tag{9}$$

4. Overview

Figure 2 illustrates the interactions between the main components of the SAC framework, where the notation $\{\cdot\}$ indicates a batch of samples. In the figure, dashed lines correspond to the parameter updates, specified by the gradients $\nabla_{\theta_i} L_{Q_{\theta_i}}$, $\nabla_{\phi} L_{\pi_{\phi}}$ and $\nabla_{\eta} L_{\eta}$ for the critics, actor and entropy coefficient respectively.

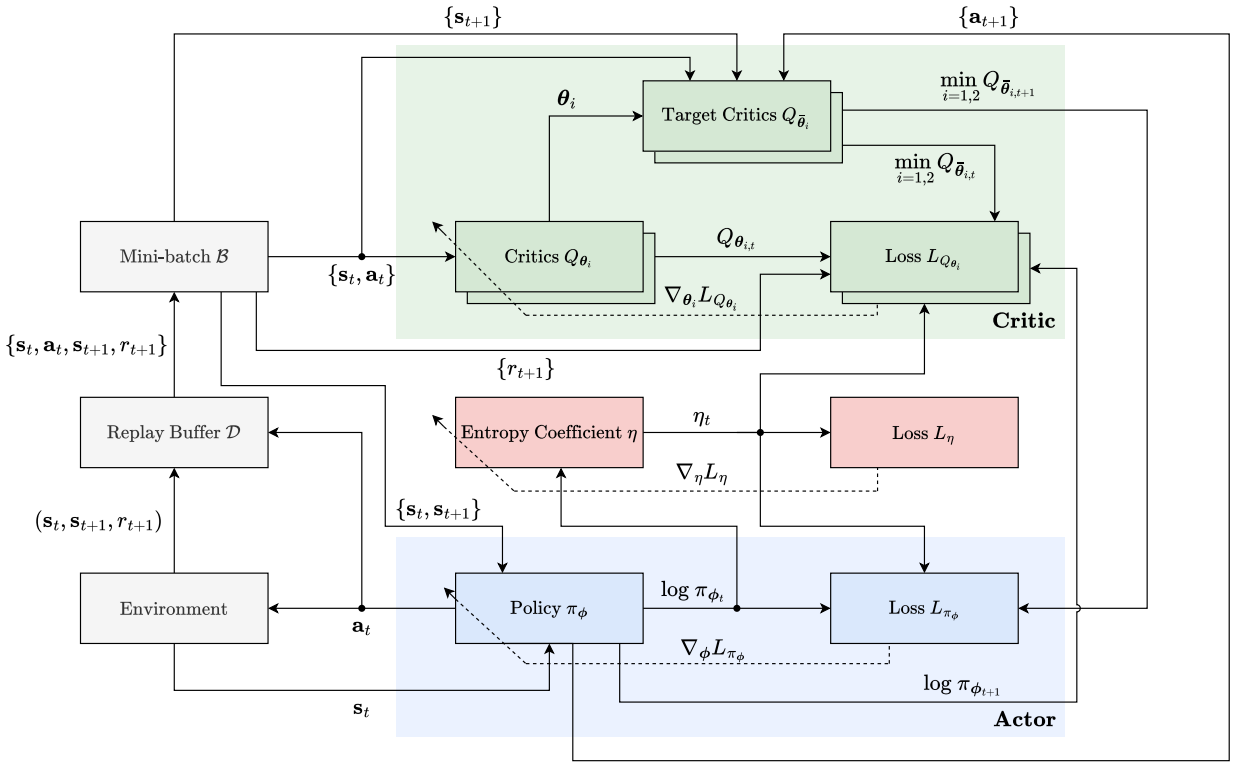

Fig. 2 Overview of the SAC framework showing the interaction between the actor, critic, environment and entropy. Adapted from [16].

B. Longitudinal Handling Qualities

Aircraft HQ are defined as how the pilot experiences the way the aircraft responds to the pilot's input. An extensive amount of literature has been written on HQ and several versions of guidelines exist to aid the design of aircraft and flight control systems. Qualitative methods for determining the HQ through pilot opinion ratings are often used in real flight experiments and in simulators, but since evaluation of HQ has not yet been performed for RL flight control, this paper will focus on quantitative HQ determination through simulations. The longitudinal motion of aircraft consist of two eigenmodes; the short period and phugoid. Since the latter is usually slow, with low frequency oscillations, the pilot is often capable of controlling and stabilizing the motion. The short period mode, on the other hand, involves higher frequencies and has a significant impact on the manoeuvrability of the aircraft. Therefore, adequate short period HQ are crucial for the longitudinal controllability of the aircraft [18]. Several civilian aircraft standards exist, but focus more on qualitative assessment [18]. The Military Standards provide quantitative guidelines for the assessment of short period HQ and strongly recommend to include the Control Anticipation Parameter as the key requirement, due to the fact that it captures the majority of the short period dynamics [19]. These standards can be applied to civilian aircraft as well, as the requirements in the standards are specified per aircraft category.

1. Control Anticipation Parameter

The CAP, originally defined in a study by Bihrle [21], is the main HQ criterion of this research and is defined by the instantaneous pitch acceleration $\dot{q}_0$ over the steady state load factor $n_{z_{ss}}$. Alternatively, the steady state pitch rate q_{ss}, in combination with the velocity V and gravitational acceleration g could be used for the computation of the CAP, as

shown in Equation 10. The parameter is a metric that indicates to what extent the pilot can anticipate on the aircraft's response after a step input is exerted on the control stick, based on the initial pitch acceleration. A CAP that is too low results in a response that tends to feel sluggish which in turn can generate Pilot Induced Oscillations (PIO). When the CAP is too high on the other had, the aircraft feels sensitive and the pilot might overcompensate resulting in PIO as well.

$$\text{CAP} = \frac{\dot{q}_0}{n_{z_{ss}}} = \frac{\dot{q}_0}{\frac{V}{g} q_{ss}} \tag{10}$$

2. Low Order Equivalent System

Additional parameters like the damping ratio and natural frequency are often used for the assessment of the short period HQ. These parameters, however, are related to a second order model of the aircraft, whereas modern aircraft are generally highly augmented and include sensor, control and actuator dynamics. Next to that, the aircraft dynamics could be nonlinear, which complicates the analysis of second order short period parameters. Typically, the nonlinear aircraft model is linearized around the operating point such that a Higher Order System (HOS) is obtained. It was found that the linear HOS can be represented by a second order model with an equivalent time delay to account for the higher order dynamics [22]. The resulting Low Order Equivalent System (LOES), defined by Equation 11, contains the short period damping ratio ζ_{sp}, natural frequency ω_{sp}, incidence lag T_{θ_2}, equivalent gain K_θ and equivalent time delay τ_e. It relates the pitch rate q to the pitch rate command q_{cmd} exerted by the pilot through the control stick. The LOES parameters are acquired through frequency matching at the bandwidth where the pilot is the most sensitive [23].

$$\frac{q(s)}{q_{cmd}(s)} = \frac{K_\theta \left(s + 1/T_{\theta_2}\right) e^{-\tau_e s}}{s^2 + 2\zeta_{sp}\omega_{sp} s + \omega_{sp}^2} \tag{11}$$

An alternative approach for the computation of the CAP is to use the short period model parameters derived from the LOES as shown in Equation 12. The equivalent CAP_e includes an attenuation factor $\dot{q}_{nd}$ to compensate for the difference in instantaneous accelerations of the LOES and the full aircraft model. More specifically, it is the ratio between the maximum pitch acceleration of the full aircraft model $\dot{q}_{max}$ and the instantaneous pitch acceleration of the LOES $\dot{q}_{0,sp}$, which is equal to the equivalent gain K_θ. The underlying reason for the inclusion of this factor is that the maximum pitch acceleration of the full aircraft model occurs not exactly at the instant at which the step input is exerted by the pilot, but with a short delay due to actuator dynamics. It is generally of lower magnitude than $\dot{q}_{0,sp}$ and thus the CAP obtained from the LOES requires compensation for this phenomenon [24].

$$\text{CAP}_e = \frac{\omega_{sp}^2}{\frac{V}{g}\frac{1}{T_{\theta_2}}} \dot{q}_{nd} \tag{12}$$

3. Requirements

The aforementioned CAP and short period parameters were used to develop requirements for longitudinal HQ by the Military Standards [19]. The requirements are divided according to the intensity of the pilot workload, with Level 1 being the lowest and thus desired workload intensity and Level 3 being the highest. The specific requirements that apply to the aircraft category that the Cessna Citation II falls under are presented in Table 1. Note that the CAP is usually assessed in combination with the damping ratio ζ_{sp} and requires both parameters to be in Level 1.

Table 1 Longitudinal HQ requirements of the aircraft's short period reponse for the three levels of pilot workload [19].

Parameter	Level 1	Level 2	Level 3
Damping ratio short period [-]	$0.35 \leq \zeta_{sp} \leq 1.3$	$0.25 \leq \zeta_{sp} \leq 2.0$	$0.15 \leq \zeta_{sp}$
Natural frequency short period [rad/s]	$\omega_{sp} \geq 1.0$	$\omega_{sp} \geq 0.6$	-
Time delay [s]	$\tau_e < 0.1$	$\tau_e < 0.2$	$\tau_e < 0.25$
Control Anticipation Parameter [$g^{-1}s^{-2}$]	$0.28 \leq \text{CAP} \leq 3.42$	$0.15 \leq \text{CAP} \leq 9.85$	-

III. Flight Control System Design

The implementation of the SAC framework for flight control and the evaluation of longitudinal HQ during training will be discussed in this section.

A. Flight Control Framework

The proposed flight control framework is a Command and Stability Augmentation System (CSAS) with the aim to stabilize the inner flight control loop while providing adequate HQ [25]. It is realized by including a reference model on the command path, i.e., the feedforward path between the pilot input and controller, that can be designed to shape the response for the desired HQ. As the scope of this research is developing a proof-of-concept, rather than optimizing a RL flight controller, it was decided to implement the CSAS in the form of a pitch rate command system, as visualized in Figure 3. This improves the ease of implementation in practice as it does not fully take over the pilot, but merely the inner control loop. Furthermore, short period HQ are directly related to pitch rate control.

1. High-Fidelity Longitudinal Aircraft Model

For the implementation of the SAC controller, the Cessna Citation II PH-LAB research aircraft of the TU Delft was selected, visualized in Figure 1. As the aim of this research is to stimulate aviation to move towards intelligent flight control with RL, by means of evaluating HQ to get more insight on how such a controller would behave, the PH-LAB provides a platform to implement the proposed controller in practice in the future. Moreover, a high-fidelity simulation model, validated for the PH-LAB [26], created with the Delft University Aircraft Simulation Model and Analysis Tool (DASMAT) is available for performing simulations.

For the development of a pitch rate command system, the full DASMAT simulation model is reduced to a longitudinal model with state vector $\mathbf{x}$, shown in Equation 13, containing the pitch rate q, velocity V, angle of attack α and pitch angle θ. An auto-throttle is applied to maintain constant velocity, hence the only degree of freedom for the controller is the elevator deflection δ_e, which is also referred to as the control input $\mathbf{u}$. Furthermore, the altitude of the aircraft h is assumed to remain constant for the simulations. The sampling rate of the model is 100 Hz and the sensors are assumed to be ideal, hence no additional sensor dynamics are included. The actuator is modelled as a first order transfer function with deflection angle limits in the range of [-17, 15] deg and rate limits of [-20, 20] deg/s [27].

$$\mathbf{x} = [q, V, \alpha, \theta]^T \tag{13}$$

$$\mathbf{u} = [\delta_e] \tag{14}$$

2. Reference Model

As mentioned before, for CSAS controllers, a reference model is used to shape the aircraft's response to adhere to the desired HQ. A second order reference model is selected for the implementation of the pitch rate command controller as presented in Equation 15, as the parameters of such a model are directly related to the short period HQ. Note that the reference model is very similar to Equation 11, but there is no time delay as the model prescribes the ideal behaviour of the aircraft. The desired CAP_{ref}, damping ratio ζ_{ref} and incidence lag T_{ref} can be selected by the designer of the control system and the natural frequency ω_{ref} and gain K_{ref} follow from Equation 12 (with $\dot{q}_{nd} = 1$) and the DC gain. The underlying theory is that when the controller is able to follow the reference model commands accurately, the HQ of the full control system will be close to the ones set by the reference model [28].

$$\frac{q_{ref}(s)}{q_{cmd}(s)} = \frac{K_{ref}\left(s + 1/T_{ref}\right)}{s^2 + 2\zeta_{ref}\omega_{ref}s + \omega_{ref}^2} \tag{15}$$

B. Controller Implementation

Two SAC flight controllers were developed for this research. Additionally, a linear controller was designed with the purpose of performance comparison.

1. Baseline SAC Flight Controller

An overview of the pitch rate command system, showing the interactions of the SAC controller with the reference model and high-fidelity aircraft model, is presented in Figure 3. The goal of the SAC controller is to track the reference pitch rate q_{ref} and therefore the reference error $q_{ref} - q$ is used in the observed state vector $\mathbf{s}$ as shown in Equation 16.

The reward is the negative of the squared tracking error, where a reward scaling factor κ is included, as given in Equation 17. The scaling factor is also used in the observed state vector and preliminary research showed that the SAC controller performance is sensitive to the selection of this factor. Furthermore, the pitch acceleration $\dot{q}$ is included in the observed state vector, as it provides information to the controller on the transient response. The inclusion of the pitch acceleration $\dot{q}$ does require the aircraft to be equipped with an angular accelerometer.

The elevator deflection angle δ_e is controlled in an incremental manner. The reparameterized output of the actor is squashed with a hyperbolic tangent function such that values of the action **a** remain between [-1,1]. Subsequently, the action is scaled with the elevator rate limits, to get the current elevator rate $\dot{\delta}_{e,t}$. It is multiplied with the sampling time Δt to get the control input increment and then added to the previous state of the elevator deflection: $\delta_{e,t} = \delta_{e,t-1} + \Delta t \dot{\delta}_{e,t}$. The incremental control causes smoother and less aggressive changes of the elevator deflection angle. The elevator deflection angle needs to be added to the observed state vector, however, in order for the SAC agent to know its current position. In general, the training of the SAC controller takes longer and becomes more complex as states are added to the observation vector. The states that are included are therefore the minimum required states for satisfactory performance.

$$\mathbf{s} = [\dot{q}, \kappa(q_{ref} - q), \delta_e]^T \tag{16}$$

$$r = -(\kappa(q_{ref} - q))^2 \tag{17}$$

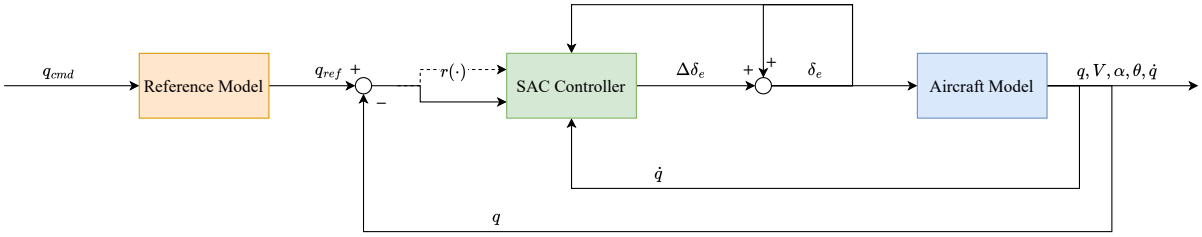

Fig. 3 Overview of the SAC pitch rate control system, with the interactions between the controller, aircraft model and reference model.

2. SAC Flight Controller with Conditioning for Action Policy Smoothness

Preliminary experiments have shown that even though incremental elevator control is used, the SAC agent still shows high gain tracking behaviour. Therefore, a second SAC controller was developed which includes Conditioning for Action Policy Smoothness (CAPS). With this approach, two additional loss terms are added to the actor loss function to further smoothen the SAC agent's policy [29]:

$$L_{\pi_{\boldsymbol{\phi}}}^{CAPS} = L_{\pi_{\boldsymbol{\phi}}} + \lambda_T L_T + \lambda_S L_S \tag{18}$$

The temporal loss term L_T is computed as the L2-norm of the deterministic actions at time t and time $t+1$. It is scaled with the temporal smoothing factor λ_T, which is a new hyperparameter. Likewise, the spatial loss term L_T is weighted with a spatial smoothing factor λ_S and calculated with the L2-norm of the deterministic action of the policy and the action for a normally sampled state $\bar{\mathbf{s}}$ with a standard deviation of $\sigma = 0.035$.

$$L_T = ||\pi_{\boldsymbol{\phi}}(\mathbf{s}_t) - \pi_{\boldsymbol{\phi}}(\mathbf{s}_{t+1})||_2 \tag{19}$$

$$L_S = ||\pi_{\boldsymbol{\phi}}(\mathbf{s}) - \pi_{\boldsymbol{\phi}}(\bar{\mathbf{s}})||_2 \tag{20}$$

3. Linear Flight Controller

To compare the SAC agents to a classical controller, a Linear Controller (LC) was developed for the same flight control framework. For fair comparison, pitch rate reference error $q_{ref} - q$ and pitch acceleration $\dot{q}$ are used for selecting the elevator deflection angle δ_e. It is not considered necessary to use the incremental control approach, because the controller is linear and aggressiveness of the controller can be adapted with the control gains. The elevator deflection angle is determined with the gains K_p and K_d by the following equation:

$$\delta_e = -K_p(q_{ref} - q) - K_d\dot{q} \tag{21}$$

C. Training Approach

The selection of the hyperparameters and training signal is an important aspect of the design of the SAC controllers. This section will explain why certain design choices are made with respect to these parameters.

1. Hyperparameters

The hyperparameters of the SAC agents are of significant influence on the tracking performance and HQ. As the scope of this research is limited to the proof-of-concept of HQ evaluation for RL flight control, instead of optimizing an RL agent itself, the hyperparameters are based on earlier research as they were already tuned succesfully for the SAC framework [16]. All the hyperparameters are presented in Table 2 and it can be observed that the actor and critic network architectures are similar in terms of hidden layers, as well as the initial learning rates. Similar to earlier work, the two hidden layers of the actor and critic DNNs contain normalization layers with ReLu activation functions and the gradient-descent parameter updates are performed with the Adam optimizer [16]. It should be noted that the temporal smoothing factor λ_T is set to 0 for the SAC with CAPS, because temporal smoothness was found to lead to poor short period HQ in preliminary experiments, as it makes the controller very sluggish.

Table 2 Hyperparameters of the SAC agents, partially adapted from [16].

Parameter	Symbol	Value
Discount factor	γ	0.99
Target critic smoothing factor	τ	0.005
Actor and critic hidden layer sizes	l_1, l_2	64,64
Actor and critic initial learning rates	$\eta_a \eta_c$	9.4e-4, 9.4e-4
Replay buffer batch size	$\lvert\mathcal{B}\rvert$	256
Replay buffer maximum size	$\lvert\mathcal{D}\rvert$	50000
Initial entropy coefficient	η_0	1.0
Number of episodes	N_e	200
Reward scaling factor	κ	$\frac{1}{4}\frac{180}{\pi}$
Temporal smoothing factor	λ_T	0.0 (CAPS only)
Spatial smoothing factor	λ_S	100.0 (CAPS only)

2. Simulation Strategy

The simulations during training were performed with episodes that last 30 seconds, where every 5 seconds a random step input on the pitch rate command q_{cmd} between [-2,2] degrees is fed to the two SAC agents. Using random step inputs ensures that the SAC agents can explore the full state-action domain within the given range, as long as there are enough episodes to learn. Training is performed with the linearized aircraft model and does not contain the saturation limits. In the context of the SAC framework, this training phase is often referred to as offline learning. This is done with a simulation model of the aircraft and crashes are permitted. After the offline learning phase, online evaluation is performed where the agent is controlling the aircraft in real-time and crashes are not tolerated. Since all experiments in this research are carried out through simulations, the online evaluation requires a simulation model as well, but it mimics the situation as if the SAC controller were controlling the aircraft in reality.

After 200 episodes of offline learning, the SAC agents are evaluated online with a 3-2-1-1 step input signal for the nonlinear aircraft model. This signal was selected as is commonly used for system identification. The magnitudes of the step inputs are selected between [-1,1] degrees, such that the evaluation is done within the domain that the SAC has covered while training. During the 30 seconds evaluation, the normalized Mean Absolute Error (nMAE) is monitored as well as the elevator activity $\delta_{e,act}$. The former provides information on the tracking performance, whilst the latter gives insight in the degree of aggressiveness of the SAC controllers. The elevator activity is calculated with the integral of the elevator rate, divided by the simulation time T [30]:

$$\delta_{e,act} = \frac{\int_0^T |\dot{\delta}_e| \Delta t}{T} \tag{22}$$

3. Handling Qualities Evaluation

During training, the longitudinal HQ are evaluated after the completion of each episode. The SAC controllers are linearized with the perturbation approach and combined with the reference model and linearized aircraft model they form the HOS of the full pitch rate command system. To obtain the LOES, the frequencies of the HOS and LOES are matched between 0.1 and 10 rad/s, as this is the region where the pilot is the most sensitive [19]. The LOES fit is optimized by minimizing the cost function specified by Equation 23, where N_ω is the number of logarithmically spaced frequency datapoints, ϕ the phase angle, G the gain and ω the frequency. The optimization was performed using the Scipy Nelder-Mead algorithm in Python*. The Maximum Unnoticable Added Dynamics (MUAD) bounds determine the scaling factors κ_G and κ_ϕ. These bounds were developed to further specify the frequencies at which the pilot feels the most, and at which frequency additional dynamics could be added without the pilot noticing it [31]. A successful LOES fit is defined as a fit where the fit error for all frequencies, for both the gain and phase, remain within the MUAD bounds.

$$J = \frac{20}{N_\omega} \sum_{k=1}^{N_\omega} \left[\kappa_{G(\omega_k)} (G(\omega_k)_{HOS} - G(\omega_k)_{LOES})^2 + \kappa_{\phi(\omega_k)} (\phi(\omega_k)_{HOS} - \phi(\omega_k)_{LOES})^2 \right] \tag{23}$$

The parameters of the LOES are related to the shortperiod HQ as explained in subsection II.B. The attenuation factor $\dot{q}_{nd}$ that compensates the CAP for higher order dynamics, not captured by the LOES, is determined with a time response simulation of the pitch rate command system. A step input is given on the system and the resulting maximum pitch acceleration is used for computing the equivalent CAP_e.

IV. Results and Discussion

In this section the results of the offline training phase and online evaluation will be presented. The results for multiple flight conditions with forward and aft Center of Gravity (CG) shifts and the effect of biased sensor noised will be discussed as well. The results of the SAC controllers will be compared to the LCs.

A. Offline Training

The offline training phase was performed for both SAC controllers for the nominal flight condition, which is at an altitude of H = 2000 m and velocity of V = 90 m/s. This was done for multiple realizations of random parameter initializations, to assess the robustness to the initialization of the DNNs. Two criteria were applied for determining whether a training run was successful or not. A run is labelled successful when the trained controller evaluated on the 3-2-1-1 evaluation signal has a nMAE smaller or equal to 5% and an elevator activity $\delta_{e,act}$ of no more than 0.5 deg/s. For a total of 76 training runs, the SAC baseline controller was successful 26% of the time, whereas for the SAC controller with CAPS a success rate of 53% was reached.

Figure 4 shows the episode return, which is the sum of rewards for one episode, during training for all successful runs for both SAC controllers. It can be observed that the median of the SAC baseline controller reaches a better average return than the SAC controller with CAPS. The SAC baseline controller, however, contains runs that had very low values of episode returns during the training, but climb to higher values only at the very end of training. The SAC controller with CAPS shows considerably more stable behaviour during training; after around 50 episodes the average return has stabilized to around a value of approximately -50. The probable cause for this effect is that the SAC baseline controller tracks the reference signal more aggressively, resulting in smaller tracking errors and higher final returns.

This effect can also be observed in Figure 5, where the CAP_e during training is shown for both controllers. The successful runs of the SAC baseline controller show more widely spread values of the CAP_e, whereas the values for the SAC controller with CAPS lie within a more compact region. Even though both controllers reach a L1 HQ rating at the final stage of training, the median of the SAC baseline controller is almost exactly equal to the CAP of the reference model, CAP_{ref}. Again, this is probably caused by the aggressive tracking of the SAC baseline controller. The SAC controller with CAPS has a slightly more sluggish value of the CAP_e, which could be the cause of the spatial smoothening of the policy.

*https://docs.scipy.org/doc/scipy/reference/optimize.minimize-neldermead.html

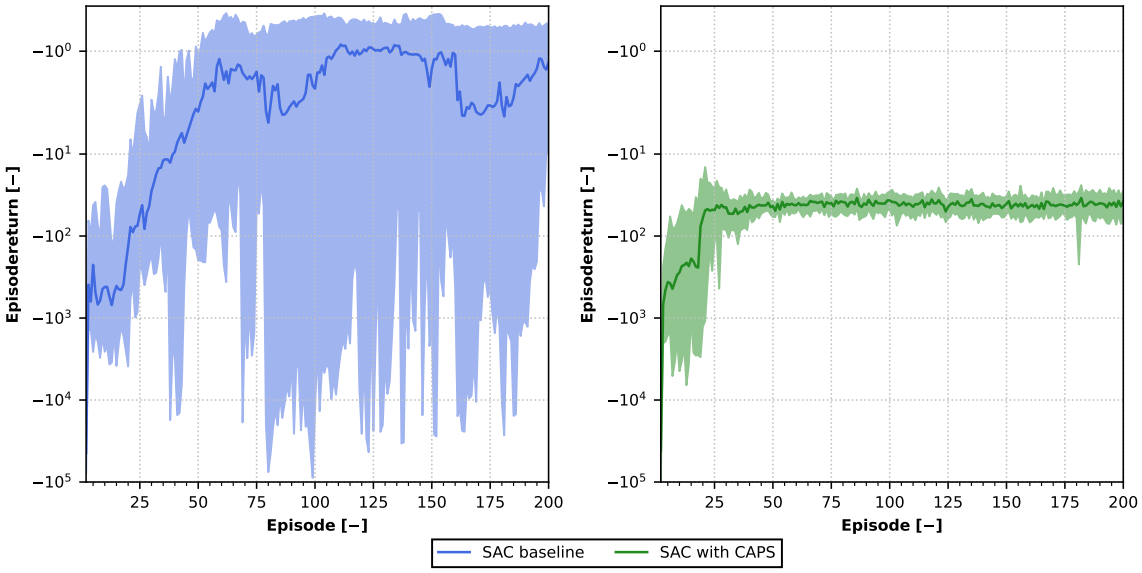

Fig. 4 Training curves, showing the episode return for the SAC baseline controller and SAC controller with CAPS during offline learning. Solid blue and green lines present the median and shaded regions in blue and green show all successful runs.

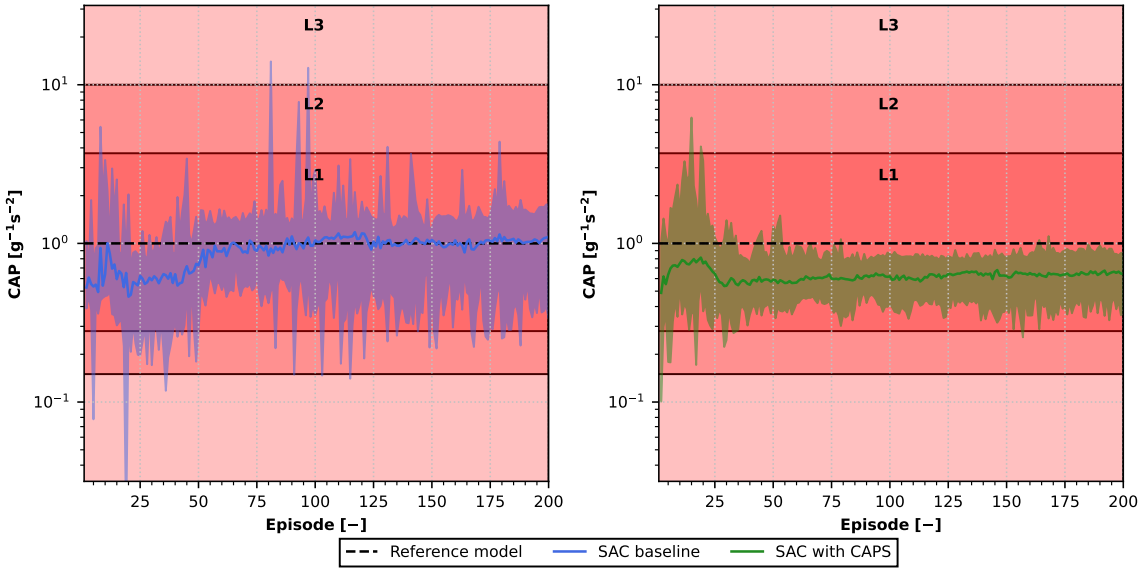

Fig. 5 The development of the equivalent CAP_e for the SAC baseline controller and SAC controller with CAPS during offline learning. Solid blue and green lines present the median and shaded regions in blue and green show all successful runs. The levels of HQ ratings are indicated with the red shaded areas.

Next to the CAP, the other short period HQ were also monitored during the training of both SAC controllers. The short period parameters obtained from LOES fits are shown in Figure 6 and Figure 7 for the SAC baseline controller and SAC controller with CAPS respectively. The main conclusion that can be drawn from the figures is that the SAC baseline controllers show more widely spread results, but the median approaches the short period reference model parameters, whereas the training runs of the SAC controller with CAPS yield results short period parameters with less variance. The SAC controller with CAPS has an offset from the reference for most of the short period parameters, which could be caused by the aggressiveness limitations posed by the action policy smoothness.

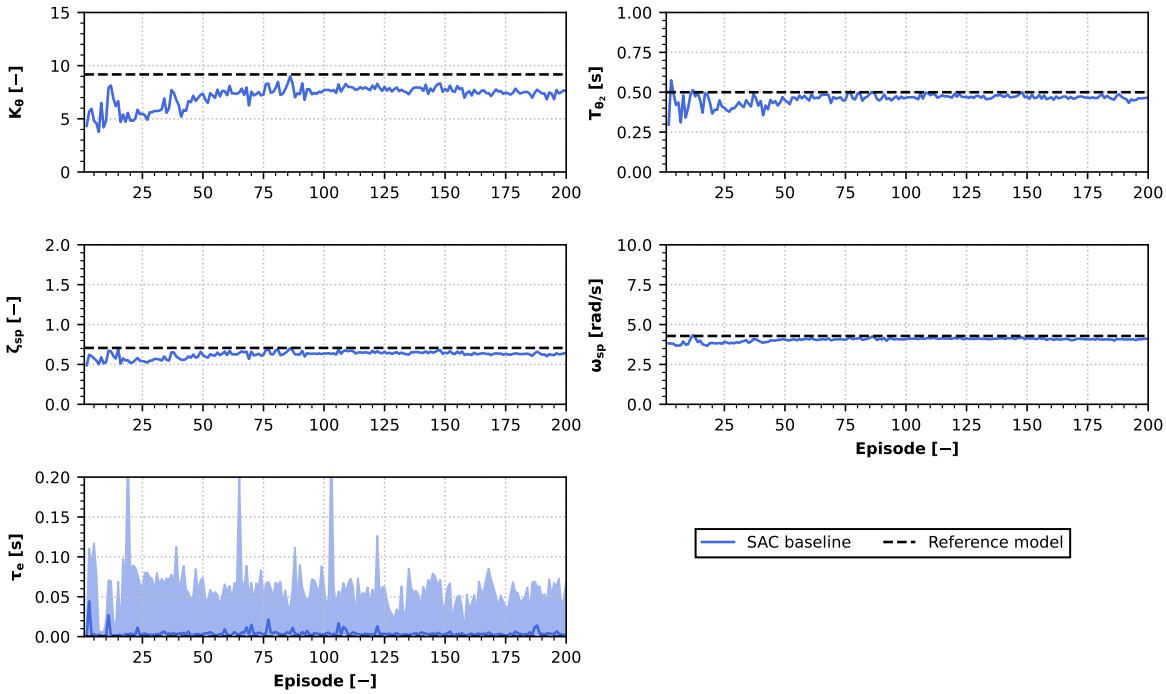

Fig. 6 Short period parameters obtained from LOES fits during offline learning for the SAC baseline controller. Solid blue lines present the median and shaded regions in blue show all successful runs.

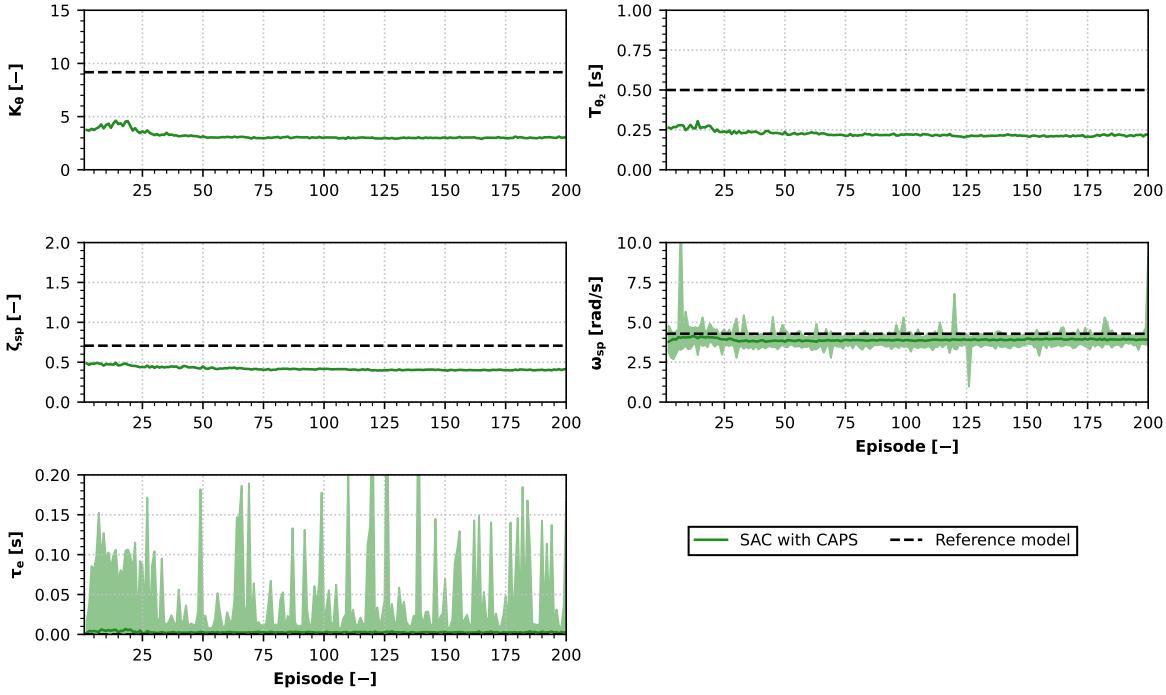

Fig. 7 Short period parameters obtained from LOES fits during offline learning for the SAC controller with CAPS. Solid green lines present the median and shaded regions in green show all successful runs.

B. Online Evaluation

For the online evaluation of the SAC controllers, a 3-2-1-1 step input signal is used. Figure 8 shows the time responses of realizations of a successful run for both SAC controllers as well as the desired behaviour posed by the reference model. From the figure, it can be observed that the SAC baseline controller tracks the reference model more accurately, which is also supported by the low nMAE of 0.95%. The SAC controller with CAPS, however, seems to have a minor steady state error which integrates over time to a nMAE of 3.53%, worse than the baseline. The figure also shows that the SAC baseline controller reaches the elevator rate saturation limits (-20 and 20 deg/s) at some instances in time. This further indicates more aggressive tracking. Overall, both controllers are able to track the reference model succesfully and there are no major differences between the other states (V, α and θ). The figure also shows the Power Lever Angle (PLA), which indicates the thrust setting and it can be seen that the autothrottle actively keeps the velocity more or less constant around the nominal flight condition of V = 90 m/s.

For the sake of comparison, similar simulations of the 3-2-1-1 step input signal have been performed for two LCs. The pitch acceleration gain K_d was set to 0.15 for both LCs and the pitch rate reference gain K_p was set to 0.07 (low gain LC) and 0.7 (high gain LC). The gains were used to control the elevator deflection as specified by Equation 21. The gains were selected by manual tuning and the low and high gains were chosen to demonstrate the effect on the aggressiveness of the tracking. Figure 9 shows the results for both LCs, where it can be observed that the high gain LC tracks the reference model better than the low gain LC, which also supported by the nMAEs of 1.27% and 7.69% respectively. The high gain LC reaches the elevator rate saturation limits several times while the low gain LC is too slow and diverges away from the reference signal (visible between t = 8 and t = 13 seconds).

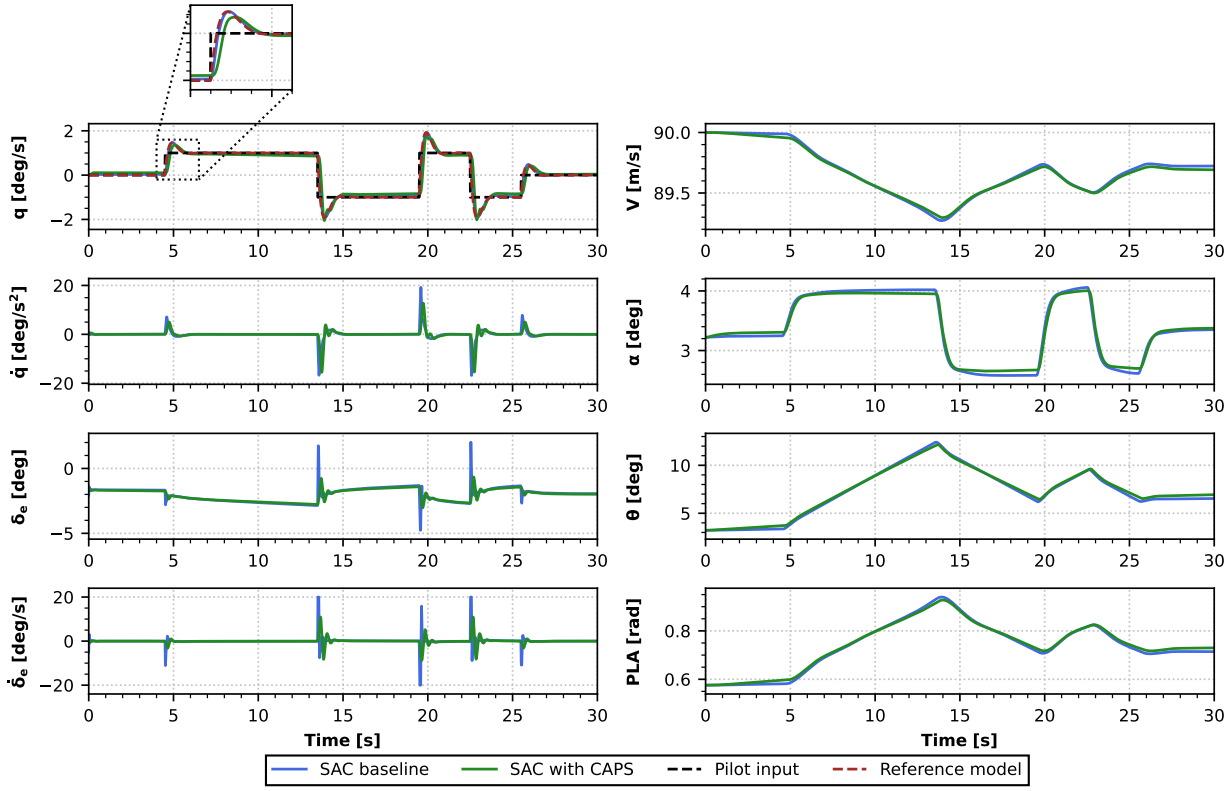

Fig. 8 Time response of the SAC baseline controller and SAC controller with CAPS for the 3-2-1-1 evaluation signal.

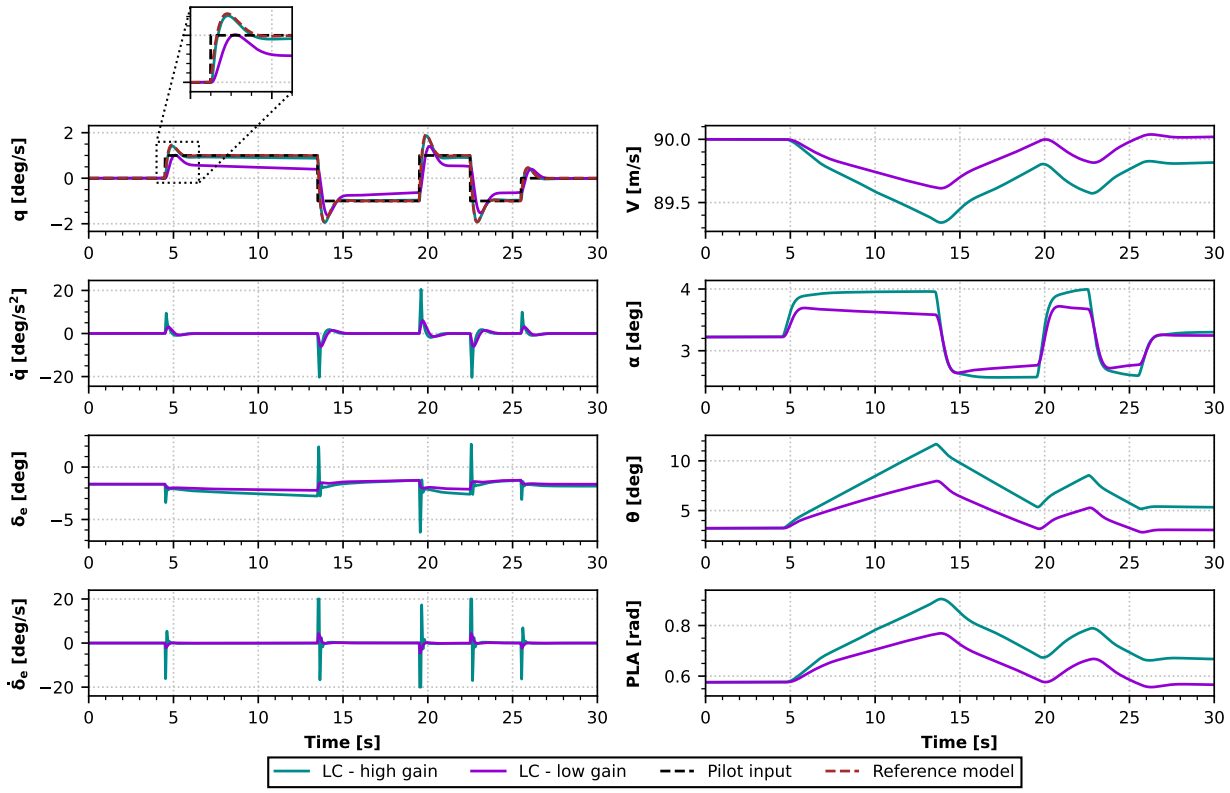

Fig. 9 Time response of the LCs with low and high gains for the 3-2-1-1 evaluation signal.

Table 3 The nMAE and elevator activity for various flight conditions and CG shifts, for both SAC controllers and LCs.

FC	CG	SAC baseline		SAC with CAPS		LC - low gain		LC - high gain	
		nMAE [%]	$\delta_{e,act}$ [deg/s]	nMAE [%]	$\delta_{e,act}$ [deg/s]	nMAE [%]	$\delta_{e,act}$ [deg/s]	nMAE [%]	$\delta_{e,act}$ [deg/s]
H = 2000 m V = 90 m/s	Normal	1.26	0.44	3.61	0.38	7.69	0.12	1.27	0.45
H = 2000 m V = 140 m/s	Normal	10.80	2.30	4.61	0.29	5.67	0.10	0.86	0.28
H = 5000 m V = 90 m/s	Normal	10.13	1.08	4.46	0.59	8.35	0.17	1.48	0.67
H = 5000 m V = 140 m/s	Normal	16.82	1.91	4.30	0.38	6.03	0.11	0.96	0.43
H = 2000 m V = 90 m/s	Aft	9.49	1.75	5.48	0.47	5.76	0.16	0.96	0.5
H = 2000 m V = 90 m/s	Fwd	11.21	1.66	4.43	0.45	9.36	0.14	1.68	0.47

Online evaluation was performed for four different flight conditions. Additionally, simulations were performed for the nominal flight condition with CG shifts of 0.25 m forward and aft. The average nMAE and elevator activity

values for all online evaluations are presented in Table 3, for both SAC controllers and LCs. Although the SAC baseline controller performs well for the nominal flight condition, the tracking is significantly worse for all other conditions. The conditions for successful runs are not met, therefore indicating poor robustness properties of the baseline controller. The SAC controller with CAPS on the other hand, shows very similar performance for all flight conditions and CG shifts. The nMAE is worse than the one for the nominal flight condition for the SAC baseline controller, but the requirements for succesful runs are almost all met (with some values just above the threshold for a successful run). When looking at the LCs, it can be seen that the low gain LC does not meet the requirements for successful tracking, but the high gain LC is successful and as a matter of fact the best of all of the four controllers.

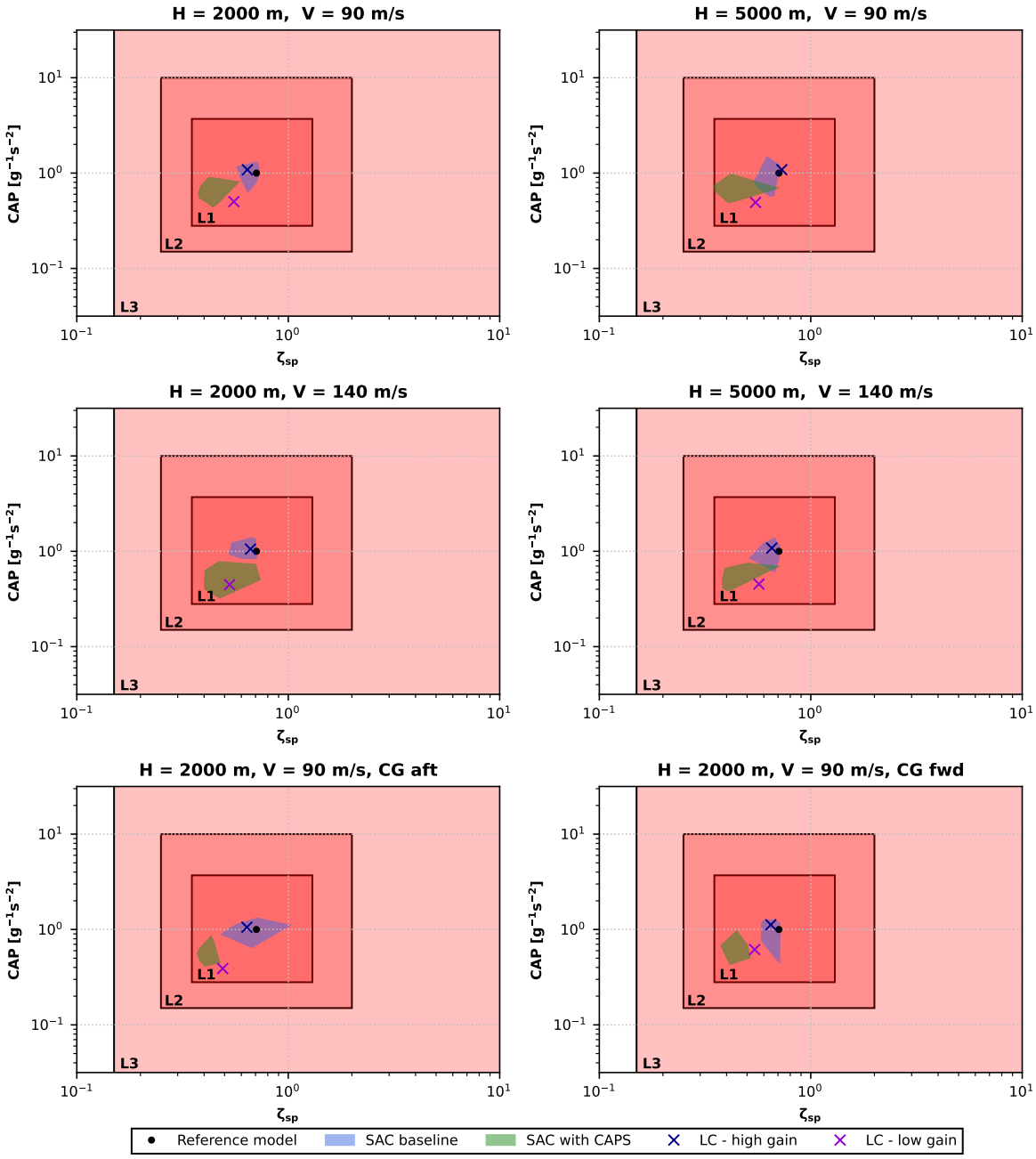

Fig. 10 The equivalent CAP$_e$ for various flight conditions and CG shifts obtained from online evaluation. Shaded blue and green areas show the all successful runs for the SAC baseline controller and SAC controller with CAPS respectively. The levels of HQ ratings are indicated with the red shaded areas.

Additionally, the equivalent CAP$_e$ in combination with the short period damping ratio ζ_{sp} is presented for all flight conditions, CG shifts and the four controllers in Figure 10. The main conclusion that can be drawn from this figure is that all controllers satisfy the Level 1 HQ ratings for all flight conditions, but the low gain LC and SAC controller with CAPS are slightly more sluggish and less damped. The high gain LC and SAC baseline controller have HQ that are closer to the reference model. It should be noted however, that only the successful LOES fits are shown in this figure, hence the poor tracking characteristics of the SAC baseline controller for off-nominal flight conditions are not reflected here.

C. Online Evaluation with Biased Sensor Noise

To demonstrate how the developed controllers operate in reality, biased sensor noise is added to the measurements used by the controllers. For the PH-LAB, the pitch rate sensor has a bias of 3.0e-5 deg/s and variance of 4.0e-7 deg/s and the angular accelerometer is set to have a bias of 0.04 deg/s^2 and variance of 1.5e-6 deg/s^2, both implemented as Gaussian noise [27]. The results of online evaluation for the nominal flight condition for the SAC controllers and LCs are shown in Figure 11 and Figure 12 respectively. It can be immediately seen that the SAC baseline controller and high gain LC get an extreme level of oscillation in the elevator deflection due to the presence of noise. The reason for this is that both controllers are too aggressive and become infeasible in practice because of elevator wear.

The low gain LC and SAC controller with CAPS are significantly less affected by the oscillatory component of the noise, as they are less aggressive. The bias of the sensors, however, does increase the steady-state errors of both controllers. In further research, a potential solution for this problem could be including an integral term. This could be realized by adding the pitch angle θ to the state observation vector of the SAC controller with CAPS.

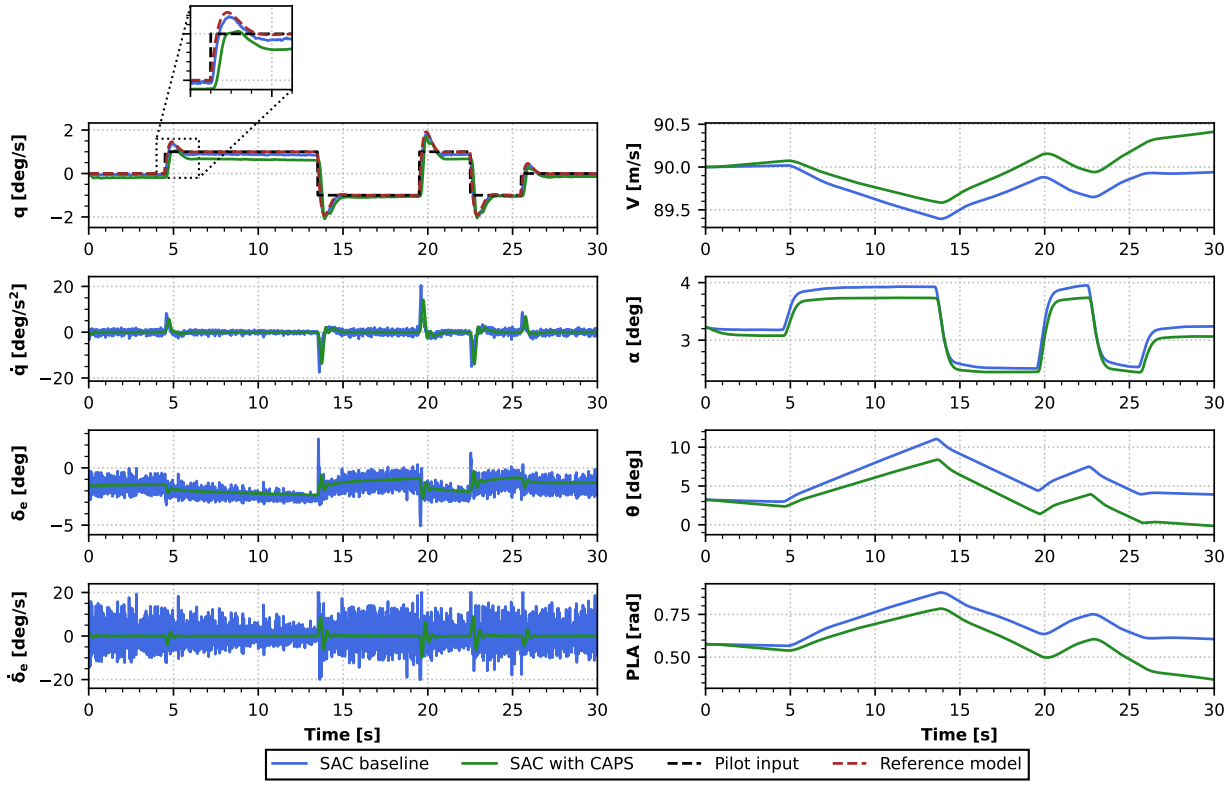

Fig. 11 Time response of the SAC baseline controller and SAC controller with CAPS for the 3-2-1-1 evaluation signal, subject to biased sensor noise.

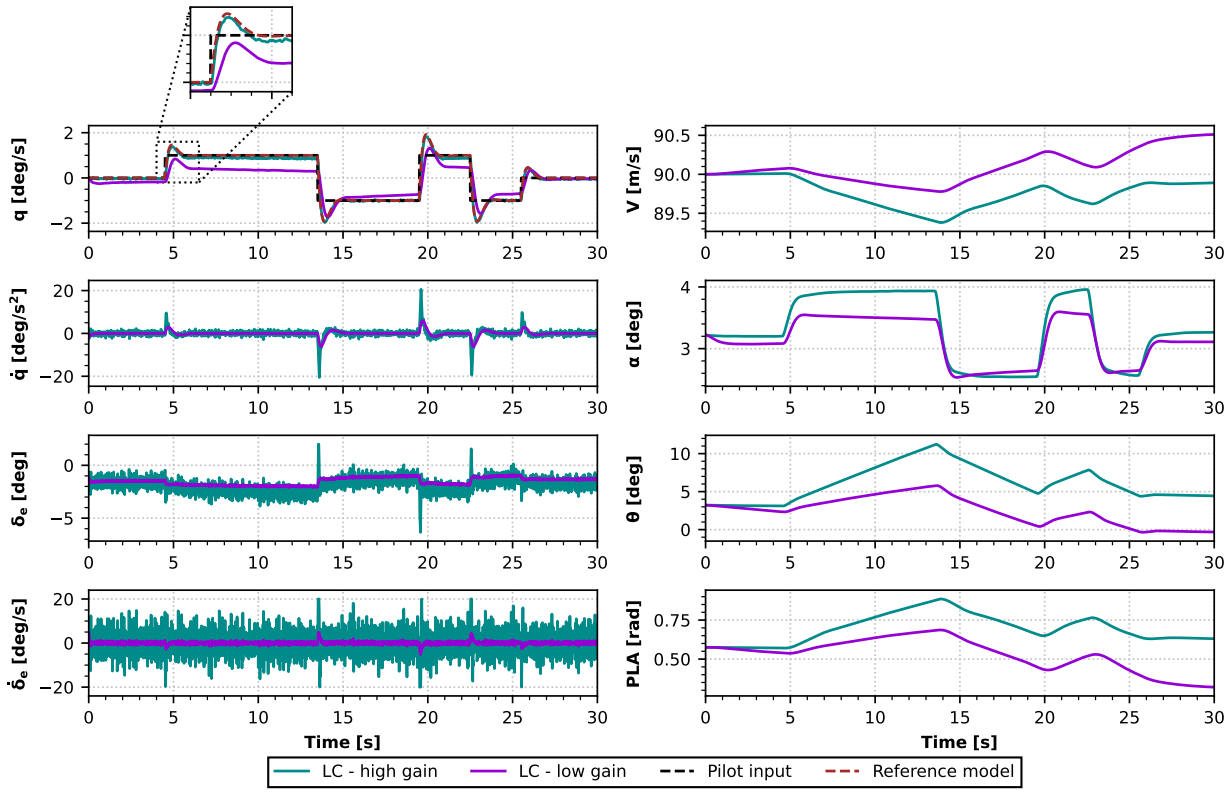

Fig. 12 Time response of the LC controllers with low and high gains for the 3-2-1-1 evaluation signal, subject to biased sensor noise.

An overview of the nMAE and elevator activity of the four controllers for the nominal flight condition with and without biased sensor noise is presented in Table 4. For the SAC controllers, the average nMAE and elevator activity are taken from the successful training runs. Again, it can be observed that the high gain LC and SAC baseline controller produce unrealistically high elevator activity values when sensor noise is included, which further demonstrates that these controllers are not feasible in practice. The SAC controller with CAPS and low gain LC have acceptable levels of elevator activity in the presence of noise. The SAC controller with CAPS performs better in terms of tracking compared to the low gain LC, which is indicated by the nMAE. This was also shown in Table 3 for the different flight conditions, making the SAC controller with CAPS the best controller in terms of robustness, while maintaining Level 1 longitudinal HQ. In the future, the SAC controller with CAPS could be further optimized in terms of hyperparameters and the pitch angle θ could be included in the state observation vector. This can increase tracking performance and ensure even better HQ, while keeping the fault-tolerant property of the controller.

Table 4 The nMAE and elevator activity for both SAC controllers and LCs, subject to biased sensor noise.

FC	Sensor noise	SAC baseline		SAC with CAPS		LC - low gain		LC - high gain	
		nMAE [%]	$\delta_{e,act}$ [deg/s]	nMAE [%]	$\delta_{e,act}$ [deg/s]	nMAE [%]	$\delta_{e,act}$ [deg/s]	nMAE [%]	$\delta_{e,act}$ [deg/s]
H = 2000 m V = 90 m/s	No	1.26	0.44	3.61	0.38	7.69	0.12	1.27	0.45
H = 2000 m V = 90 m/s	Yes	10.75	6.87	8.36	0.44	9.30	0.45	1.79	4.27

V. Conclusion

In this research, the evaluation of longitudinal Handling Qualities (HQ) was applied to Reinforcement Learning (RL) flight control. The Soft Actor-Critic (SAC) framework was implemented in a Control and Stability Augmentation System (CSAS) to control the pitch rate of the TU Delft Cessna Citation-II research aircraft, the PH-LAB. Several longitudinal HQ were evaluated like the Control Anticipation Parameter (CAP) and other second order short period parameters. The HQ were evaluated during and after training for a regular baseline SAC controller and one were Conditioning for Action Policy Smoothness (CAPS) was applied. Training was successful for the SAC baseline controller 26% of the time and for the SAC controller with CAPS 53% of the time. For the nominal flight condition, which was used for training, the SAC baseline controller outperformed the SAC controller with CAPS in terms of tracking performance (nMAE of 1.26% versus 3.61%) and approximated the reference model with the desired short period HQ more closely. The SAC controller with CAPS showed more stable behaviour during training.

Both controllers were evaluated online for off-nominal flight conditions and Center of Gravity (CG) shifts and results showed that the SAC controller with CAPS is more robust to these altering conditions, while both controllers maintained Level 1 short period HQ. When biased sensor noise was introduced to the nominal flight condition, the SAC baseline controller showed too aggressive behaviour leading to actuator wear and making the implementation in practice infeasible. A comparison for both controllers was made with two classical Linear Controllers (LCs), one with a high and one with a lower gain. The high gain LC showed comparable aggressive behaviour as the SAC baseline controller, whereas the low gain LC contained similarities with the SAC controller with CAPS.

This paper contributes to moving civil aviation towards RL flight control, as a fault-tolerant pitch rate command system, using SAC with CAPS, was shown to be robust to off-nominal flight conditions and biased sensor noise while maintaining Level 1 longitudinal HQ. The controller outperforms LCs within the same CSAS flight control framework in terms of tracking performance. It is therefore a step towards the implementation of RL flight control in practice and eliminates the need for gain scheduling. For future research, it is recommended to spend more time on optimizing the hyperparameters of the controller to increase the performance even further. Additionally, the pitch angle could be added to the state observation vector as an integral term, such that the controller is able remove the steady state errors.

References

[1] Balas, G. J., "Flight Control Law Design: An Industry Perspective," *European Journal of Control*, Vol. 9, 2003, pp. 207–226. https://doi.org/10.3166/ejc.9.207-226.

[2] Belcastro, C. M., Foster, J. V., Shah, G. H., Gregory, I. M., Cox, D. E., Crider, D. A., Groff, L., Newman, R. L., and Klyde, D. H., "Aircraft Loss of Control Problem Analysis and Research Toward a Holistic Solution," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 4, 2017, pp. 733–775. https://doi.org/10.2514/1.G002815.

[3] Bauersfeld, L., Spannagl, L., Ducard, G. J. J., and Onder, C. H., "MPC Flight Control for a Tilt-Rotor VTOL Aircraft," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 57, No. 4, 2021, pp. 2395–2409. https://doi.org/10.1109/TAES.2021.3061819.

[4] Sofla, A. Y. N., Meguid, S. A., Tan, K. T., and Yeo, W. K., "Shape morphing of aircraft wing : Status and challenges," *Materials and Design*, Vol. 31, No. 3, 2010, pp. 1284–1292. https://doi.org/10.1016/j.matdes.2009.09.011.

[5] Faggiano, F., Vos, R., Baan, M., and Van Dijk, R., "Aerodynamic Design of a Flying V Aircraft," *AIAA Aviation Technology, Integration, and Operations Conference*, Denver, Colorado, 2017. https://doi.org/10.2514/6.2017-3589.

[6] Cook, J., and Gregory, I., "A Robust Uniform Control Approach for VTOL Aircraft," *VFS Autonomous VTOL Technical Meeting and Electric VTOL Symposium*, 2021. URL https://ntrs.nasa.gov/citations/20210000418.

[7] Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, 2nd ed., The MIT Press, Cambridge, Massachusetts, 2018.

[8] Phrokhorov, D., and Wunsch, D. C., "Adaptive Critic Designs," *IEEE Transactions on Neural Networks*, Vol. 8, No. 5, 1997, pp. 997–1007. https://doi.org/10.1109/72.623201.

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

[10] Sun, B., and Van Kampen, E., "Incremental Model-Based Global Dual Heuristic Programming for Flight Control," *IFAC PapersOnline*, Vol. 52, No. 29, 2019, pp. 7–12. https://doi.org/10.1016/j.ifacol.2019.12.613.

[11] Heyer, S., Kroezen, D., and Van Kampen, E., "Online Adaptive Incremental Reinforcement Learning Flight Control for a CS-25 Class Aircraft," *AIAA Scitech 2020 Forum*, Orlando, Florida, 2020. https://doi.org/10.2514/6.2020-1844.

[12] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D., "Continuous control with deep reinforcement learning," *International Conference on Learning Representations*, 2016. https://doi.org/10.48550/arXiv.1509.02971.

[13] Fujimoto, S., Van Hoof, H., and Meger, D., "Addressing Function Approximation Error in Actor-Critic Methods," *35th International Conference on Machine Learning*, Stockholm, 2018. https://doi.org/10.48550/arXiv.1802.09477.

[14] Völker, W., Li, Y., and Van Kampen, E., "Twin-Delayed Deep Deterministic Policy Gradient for altitude control of a flying-wing aircraft with an uncertain aerodynamic," *AIAA SciTech Forum*, National Harbor, 2023. https://doi.org/10.2514/6.2023-2678.

[15] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S., "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," *35th International Conference on Machine Learning*, 2018. https://doi.org/10.48550/arXiv.1801.01290.

[16] Dally, K., and Van Kampen, E., "Soft Actor-Critic Deep Reinforcement Learning for Fault-Tolerant Flight Control," *AIAA SciTech Forum*, San Diego, California, 2022. https://doi.org/10.2514/6.2022-2078.

[17] De Haro Pizarroso, G., and Van Kampen, E., "Explainable Artificial Intelligence Techniques for the Analysis of Reinforcement Learning in Non-Linear Flight Regimes," National Harbor, Maryland, 2023. https://doi.org/10.2514/6.2023-2534.

[18] Cook, M. V., *Flight Dynamics Principles*, 2nd ed., Elsevier Ltd, 2007. https://doi.org/10.1016/B978-0-7506-6927-6.X5000-4.

[19] Department of Defence, *Flying Qualities of Piloted Aircraft MIL-STD-1797A*, 1997.

[20] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S., "Soft Actor-Critic Algorithms and Applications," 2019. https://doi.org/10.48550/arXiv.1812.05905.

[21] Bihrle, W., "A Handling Qualities Theory For Precise Flight Path Control," Tech. rep., Air Force Flight Dynamics Laboratory Research and Technology Division, Air Force Systems Command, US Air Force, 1966.

[22] DiFranco, D. A., "In-flight Investigation of the Effects of Higher-order Control System Dynamics on Longitudinal Handling Qualities," Tech. rep., Air Force Flight Dynamics Laboratory, Air Force Systems Command,Wright-Patterson Air Force Base, Ohio, 1968.

[23] Hodgkinson, J., and Lamanna, W. J., "Equivalent system approaches to handling qualities analysis and design problems of augmented aircraft," Tech. rep., McDonnel Aircraft Company, 1977.

[24] Bischoff, D. E., "The Control Anticipation Parameter for Augmented Aircraft," Tech. rep., Naval Air Development Center, Warminster, PA, 1981.

[25] Cook, M. V., "On the design of command and stability augmentation systems for advanced technology aeroplanes," *Transactions of the Institute of Measurement and Control*, Vol. 21, No. 2/3, 1997, pp. 85–98. https://doi.org/10.1177/014233129902100205.

[26] Van den Hoek, M., De Visser, C., and Pool, D., "Identification of a Cessna Citation II Model Based on Flight Test Data," *Advances in Aerospace Guidance, Navigation and Control*, April, Springer, 2017, pp. 259–277.

[27] Konatala, R. B., Van Kampen, E., and Looye, G. H. N., "Reinforcement Learning based Online Adaptive Flight Control for the Cessna Citation II(PH-LAB) Aircraft," *AIAA Scitech 2021 Forum*, 2021. https://doi.org/10.2514/6.2021-0883.

[28] Sun, L., Shi, L., Tan, W., and Liu, X., "Flying qualities evaluation based nonlinear flight control law design method for aircraft," *Aerospace Science and Technology*, Vol. 106, 2020. https://doi.org/10.1016/J.AST.2020.106126.

[29] Mysore, S., Mabsout, B., Mancuso, R., and Saenko, K., "Regularizing Action Policies for Smooth Control with Reinforcement Learning," *IEEE International Conference on Robotics and Automation*, 2021, pp. 1810–1816. https://doi.org/10.1109/ICRA48506.2021.9561138.

[30] Stougie, J., "INDI with Flight Envelope Protection for the Flying-V," , 2022. URL http://resolver.tudelft.nl/uuid:5d0a883c-bf58-4507-b688-6abccdca4842.

[31] Wood, J., and Hodgkinson, J., "Definition of Acceptable Levels of Mismatch for Equivalent systems of Augmented CTOL Aircraft," Tech. rep., McDonnell Aircraft Coroporation, Saint Louis, Missouri, 1984.