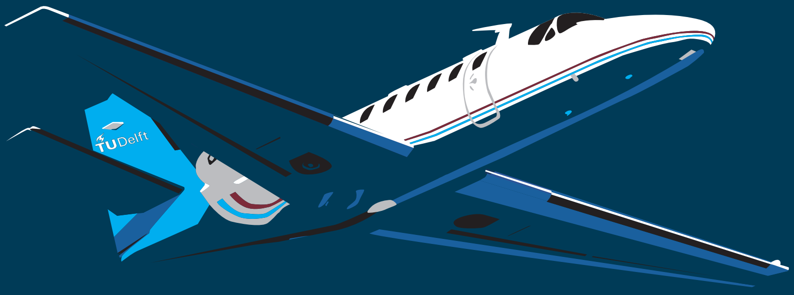
Reinforcement Learning for Flight Control

Evaluating Handling Qualities and Stability Properties of the PH-LAB

Hidde Jansen



TUDelft

Reinforcement Learning for Flight Control

Evaluating Handling Qualities and Stability Properties of the PH-LAB

by

Hidde Jansen

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on April 12, 2024 at 14:00

Thesis committee:	Dr. M.D. Pavel	TU Delft, chair
	Dr. Ir. E. van Kampen	TU Delft, supervisor
	Dr. Ir. E. Mooij	TU Delft, external examiner
	Ir. R. Konatala	DLR, additional member
Project Duration:	December, 2022 - April, 2024	
Student number:	4551141	

Faculty of Aerospace Engineering · Delft University of Technology

TU Delft Delft University of Technology

Preface

As a result of the rise of artificial intelligence, the world around us is developing at an unprecedented pace. Machine learning, more specifically reinforcement learning, has found its way into the world of aviation, bringing countless challenges and opportunities. With this research, I embarked on the exciting journey of implementing reinforcement learning to flight control. I am proud to say that with this research I contributed to the development of reinforcement learning for flight control by evaluating handling qualities and stability properties, thereby bringing the real-world implementation one step closer. It is a promising field of research and I truly believe that reinforcement learning could contribute to safer aviation. I sincerely hope that more students will continue to further develop this work in the future.

This project completes my studies in Aerospace Engineering and thereby my time here as a student in Delft. It was a wonderful time, where I learned a great deal about fascinating technology and also about myself. I would like to express my gratitude to the people who supported me during the thesis; I could not have done it without them. First of all, I would like to thank my supervisor Erik-Jan van Kampen. From the first meeting we had, you were very supportive and kept me on track by providing constructive feedback. Even though I encountered numerous setbacks, the weekly progress meetings were very useful and I could not wish for a better supervisor. I want to thank my family for their unconditional support. Without my parents, finishing this project would not have been possible. Laura, you were the one whom I could share all joyous moments with, but also during difficult times you were there for me. And of course the 2.40 squad, I enjoyed the coffee breaks, inspiring talks and sharing the daily struggles we encountered. Thank you Hajo, Tim and Max; it was a memorable time to conclude our studies here in Delft together.

Hidde Jansen
Delft, March 2024

Contents

Nomenclature

List of Abbreviations

ACD	Adaptive Critic Design
ADP	Approximate Dynamic Programming
ANN	Artificial Neural Network
BW	Bandwidth
CAP	Control Anticipation Parameter
DB	Dropback
DDPG	Deep Deterministic Policy Gradient
DHP	Dual Heuristic Dynamic Programming
DNN	Deep Neural Network
DP	Dynamic Programming
DRL	Deep Reinforcement Learning
FQ	Flying Qualities
GDHP	Global Dual Heuritstic Dynamic Programming
GM	Gain Margin
HDP	Heuristic Dynamic Programming
HOS	Higher Order System
HQ&S	Handling Qualities and Stability Properties
IDHP	Incremental Dual Heuristic Dynamic Programming
IGDHP	Incremental Global Dual Heuristic Dynamic Programming
INDI	Incremental Nonlinear Dynamic Inversion
LOES	Low Order Equivalent System
MC	Monte Carlo
ML	Machine Learning
MUAD	Maximum Unnoticable Added Dynamics
PIO	Pilot Induced Oscillations
PM	Phase Margin
RL	Reinforcement Learning
RMF	Reference Model Following
RMR	Reference Model Reward Modification
SAC	Soft Actor-Critic
TD	Temporal Difference
TD3	Twin-Delay Deep Deterministic Policy Gradient

List of Symbols

α	Angle of attack
Δt	Sampling time
δ_e	Elevator deflection
η	Learning rate
γ	Discount rate
ω_{sp}	Short period natural frequency
π	Policy
τ_e	Equivalent time delay
θ	Pitch angle
ζ_{sp}	Short period damping ratio
a	Action
f	Frequency
g	Gravitational constant
G_t	Expected return
K_θ	Short period gain
$n_{z_{ss}}$	Steady state normal load factor
q	Pitch rate
$Q(s,a)$	Action-value function
r	Reward
s	State
t	Time
T_{θ_2}	Short period time constant
V	Velocity
$V(s)$	State value function

List of Figures

List of Tables

1 Introduction

In recent times, the development of nonlinear and adaptive flight control systems has become increasingly successful, but its application in the civil aviation industry remains limited. Currently, civil aircraft still use classical flight control systems, based on gain scheduling for altering flight conditions [1]. The major drawback of gain scheduling, besides that the procedure requires elaborate and time demanding testing, is that in case of unexpected changes in the aerodynamic model of the aircraft or when it flies outside the safe flight conditions of the operating regime, it could result into performance degradation or even failure. The so called ”intelligent flight control systems”, including techniques such as Incremental Nonlinear Dynamic Inversion and Incremental Backstepping, that are currently being developed can cope with these unexpected changes by using online system identification, making them adaptive flight control systems [2].

Reinforcement Learning (RL) has shown a lot of potential for the flight control application in several studies and already has been applied successfully to a 6-degree-of-freedom simulation model of the Cessna Citation II [3]. There are multiple frameworks within RL that can either quickly adapt to sudden changes (online learning) or are robust to failures (offline learning) [4]. The reason, however, that these advanced control systems have rarely been applied in the civil aviation industry has paradoxically to do with safety. Especially when looking at the nonlinear flight controllers that make use of machine learning, it is often abstract what happens inside the controller and can be considered a black box. Therefore, it is deemed risky to apply such a controller on an aircraft and perform flight tests in reality [5].

For classical flight control systems, there are numerous regulations and guidelines for the Handling Qualities and Stability (HQ&S) requirements that the systems need to comply with [6]. These properties help in the design of stable and well controllable aircraft, but have never been applied to RL flight control systems. The main goal of this thesis is therefore to develop a proof-of-concept, showing that the HQ&S requirements can be integrated in a RL flight control system to stimulate and aid the civil aviation industry in moving towards intelligent flight control.

1.1. Research Objective and Questions

The aim of this research is further defined in this section and supported by research questions. This report aims to provide answers to all the research questions presented in this section.

Research Objective

The aim of this research is to contribute to the development of Reinforcement Learning for continuous flight control, by assessing handling qualities and stability properties and integrating them in the control loop.

Research Question 1

RQ 1 Which RL framework is the most suitable for continuous flight control and the integration of handling qualities and stability properties?

RQ 1.1 What are the state-of-the-art RL frameworks for continuous flight control?

RQ 1.2 What flight control frameworks will be used for the analysis?

RQ 1.3 How will the RL framework be integrated with the flight control framework?

Research Question 2

RQ 2 How will the performance of the RL flight controller be assessed?

RQ 2.1 Which handling qualities and stability requirements will be considered as performance criteria?

RQ 2.2 How can the selected performance criteria be obtained from flight data?

RQ 2.3 How will the selected performance criteria be included in the optimization process?

Research Question 3

RQ 3 How can the performance criteria be integrated in the RL flight control loop?

RQ 3.1 How can the reward function be modified such that the stability and handling qualities requirements are complied with?

RQ 3.2 What is the relation between the flight control framework structure and the selected performance parameters?

RQ 3.3 How can the RL framework and flight control framework structures be adapted to reach the best integration of the performance parameters?

Research Question 4

RQ 4 What is the performance of the adapted RL flight controller?

RQ 4.1 How can the RL controller be verified and validated?

RQ 4.2 How does the adapted RL flight controller compare with the same RL flight controller that uses only the tracking error as performance parameter?

1.2. Structure of the Report

The report will be structured as follows. The methodology and main results of the research project will be presented in the scientific article in Part I. In Part II, the preliminary study is presented, including a literature review on state-of-the-art RL frameworks and their application to flight control in Chapter 3, a literature review on HQ&S and their derivation from nonlinear flight control systems in Chapter 4 and a preliminary analysis in Chapter 5. Additional results will be presented in Part III, where a robustness analysis to off-nominal flight conditions is performed in Chapter 6, a threshold sensitivity study on the successful training run criteria will be included in Chapter 7 and the developed flight controller will be validated in Chapter 8. The report will be concluded in Part IV, where main conclusions and answers to the research questions are summarized in Chapter 9 and recommendations for future work are provided in Chapter 10.

Part I

Scientific Article

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

H. Jansen [*]
Delft University of Technology, P.O. Box 5058, 2600GB 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, Δt, N	=	time step, sampling time and total number of time steps
Q^{π}, $Q_{\boldsymbol{\theta}}$	=	Q-value function and parameterized Q-value function
π, $\pi_{\boldsymbol{\phi}}$	=	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_{η}	=	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_{θ_2}, K_{θ}, τ_e	=	incidence lag, equivalent gain and equivalent time delay
α, θ	=	angle of attack [deg] and pitch angle [deg]
δ_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, Control & Simulation Division, Delft University of Technology

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.

$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t) \quad (1) \qquad \mathbf{a}_t \sim \pi(\cdot \mid \mathbf{s}_t) \quad (2)$$

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.

$$Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathop{\mathbb{E}}_{\pi}\left[\sum_{k=0}^{N} \gamma^k r_{t+k+1} \mid \mathbf{s}_t, \mathbf{a}_t\right] \quad (3) \qquad Q^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathop{\mathbb{E}}_{\pi}\left[r_{t+1} + \gamma Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})\right] \quad (4)$$

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.

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

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 $\pi_{\boldsymbol{\phi}}$.

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

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.

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

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:

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

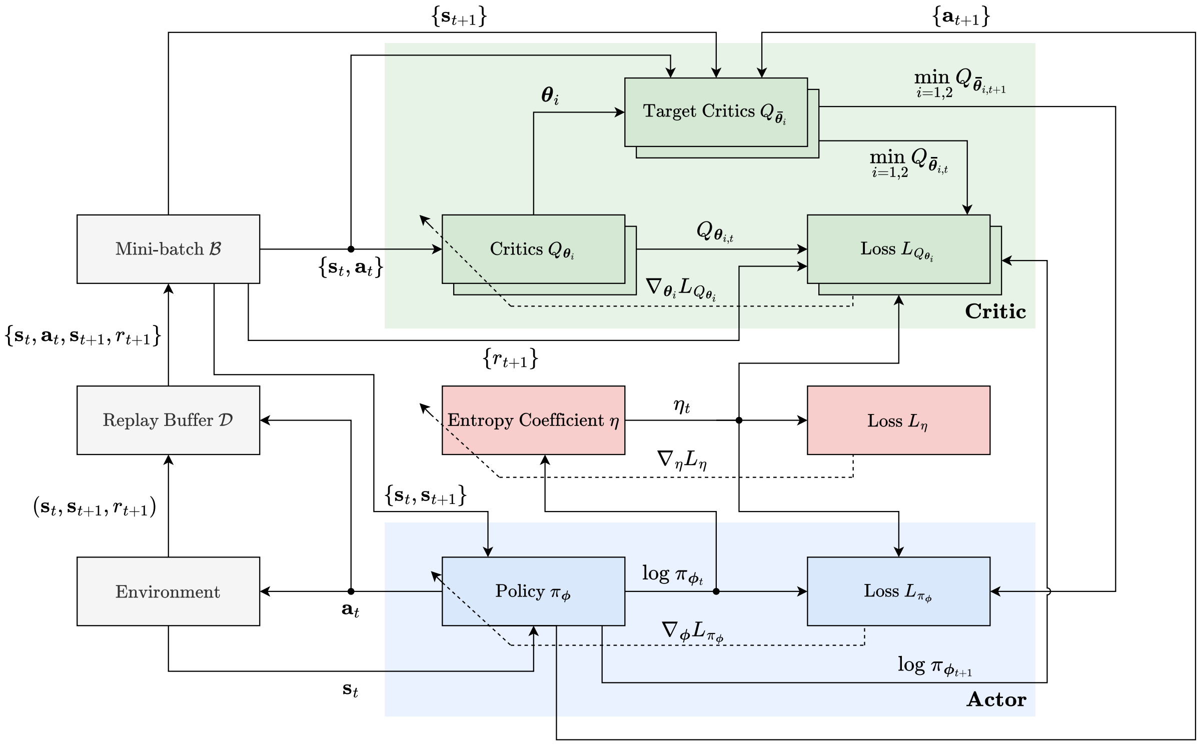
3. Entropy Adjustment

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].

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

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.



$\{\mathbf{s}_{t+1}\}$
$\{\mathbf{a}_{t+1}\}$
$\boldsymbol{\theta}_i$
Target Critics $Q_{\bar{\boldsymbol{\theta}}_i}$
$\min_{i=1,2} Q_{\bar{\boldsymbol{\theta}}_{i,t+1}}$
$\min_{i=1,2} Q_{\bar{\boldsymbol{\theta}}_{i,t}}$
Mini-batch $\mathcal{B}$
Critics $Q_{\boldsymbol{\theta}_i}$
$Q_{\boldsymbol{\theta}_{i,t}}$
Loss $L_{Q_{\boldsymbol{\theta}_i}}$
$\{\mathbf{s}_t, \mathbf{a}_t\}$
$\nabla_{\boldsymbol{\theta}_i} L_{Q_{\boldsymbol{\theta}_i}}$
Critic
$\{\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_{t+1}\}$
$\{r_{t+1}\}$
Replay Buffer $\mathcal{D}$
Entropy Coefficient η
η_t
Loss L_η
$\nabla_\eta L_\eta$
$(\mathbf{s}_t, \mathbf{s}_{t+1}, r_{t+1})$
$\{\mathbf{s}_t, \mathbf{s}_{t+1}\}$
Environment
Policy π_ϕ
$\log \pi_{\phi_t}$
Loss L_{π_ϕ}
$\mathbf{a}_t$
$\nabla_{\boldsymbol{\phi}} L_{\pi_{\boldsymbol{\phi}}}$
$\log \pi_{\phi_{t+1}}$
$\mathbf{s}_t$
Actor

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].

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

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].

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

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.

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

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

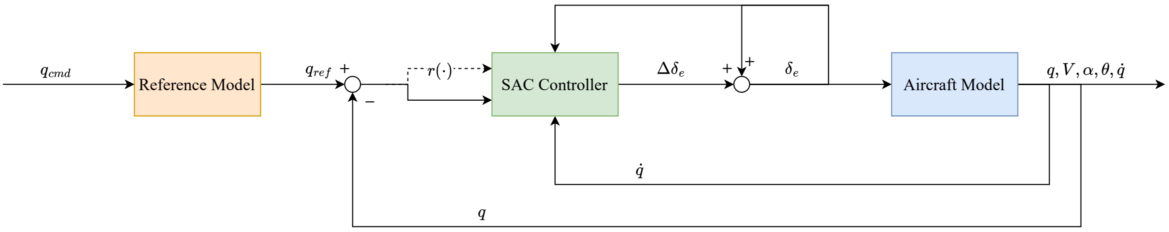


Fig. 3 Overview of the SAC pitch rate control system, with the interactions between the controller, aircraft model and reference model.

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]:

$$L_{\pi_\phi}^{CAPS} = L_{\pi_\phi} + \lambda_T L_T + \lambda_S L_S \qquad (18)$$

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 $\sigma = 0.035$.

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

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

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:

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

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

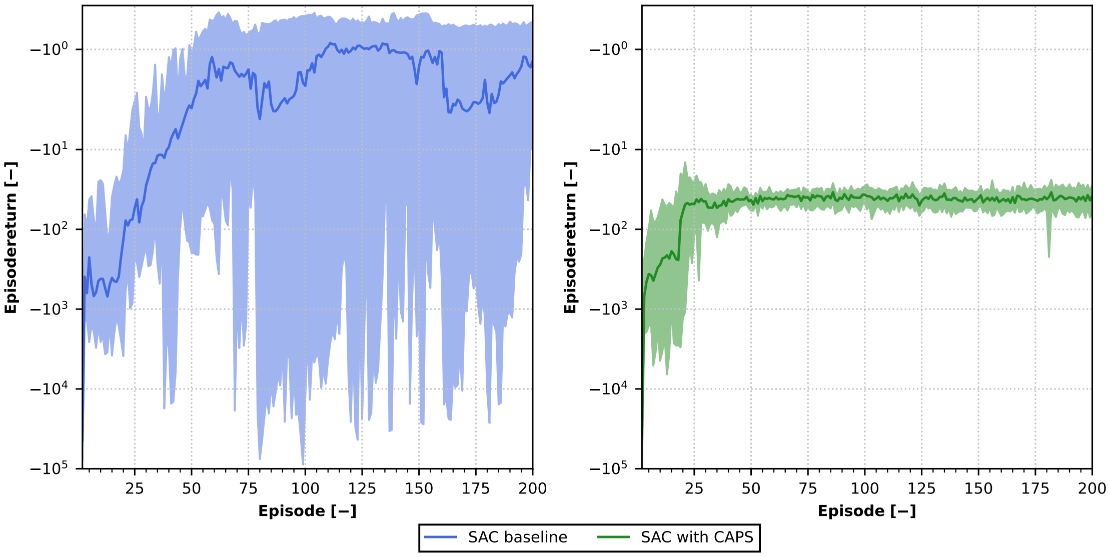


Fig. 4 Training curves, showing the episode return for the SAC baseline controller and SAC controller with CAPS during offline learning. Solid blue and green lines present the median and shaded regions in blue and green show all successful runs.

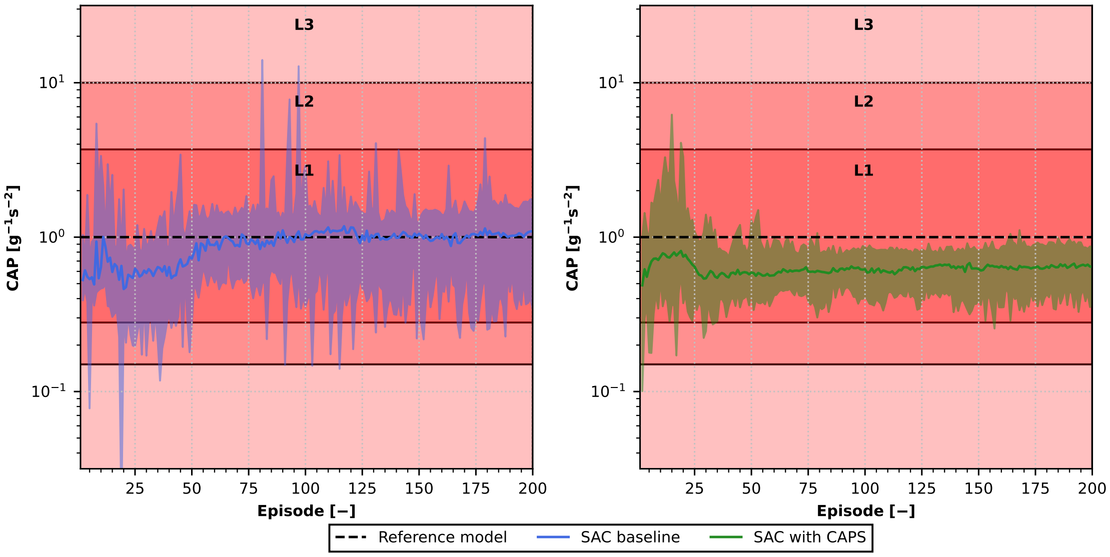


Fig. 5 The development of the equivalent CAP$_e$ for the SAC baseline controller and SAC controller with CAPS during offline learning. Solid blue and green lines present the median and shaded regions in blue and green show all successful runs. The levels of HQ ratings are indicated with the red shaded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

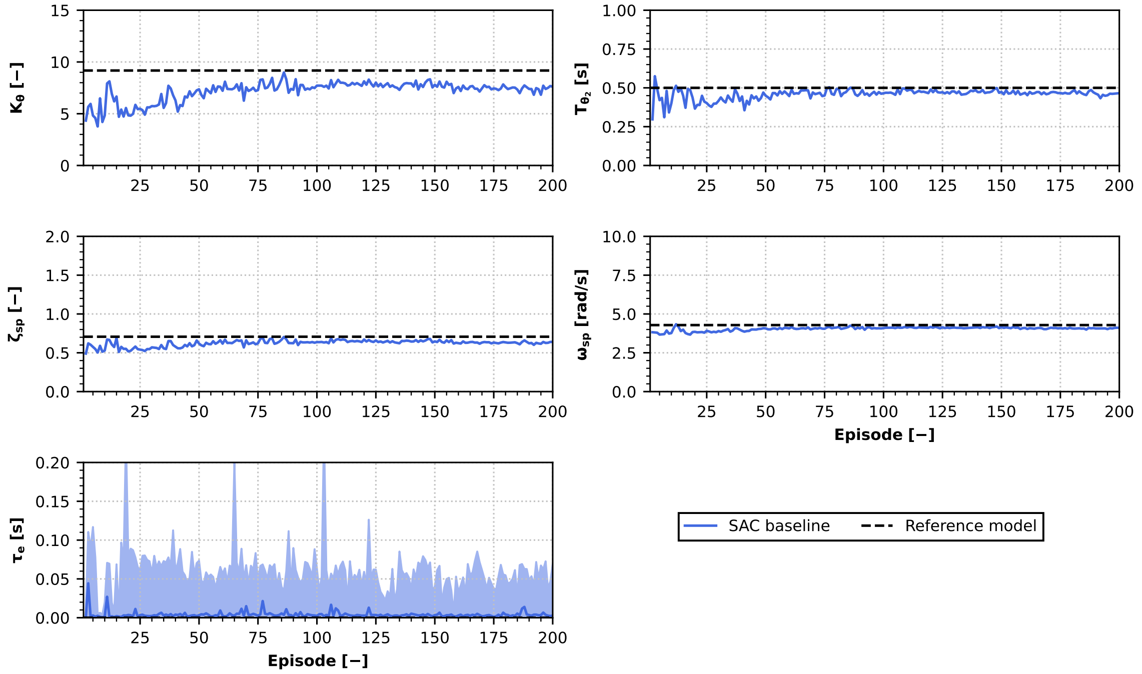


Fig. 6 Short period parameters obtained from LOES fits during offline learning for the SAC baseline controller. Solid blue lines present the median and shaded regions in blue show all successful runs.

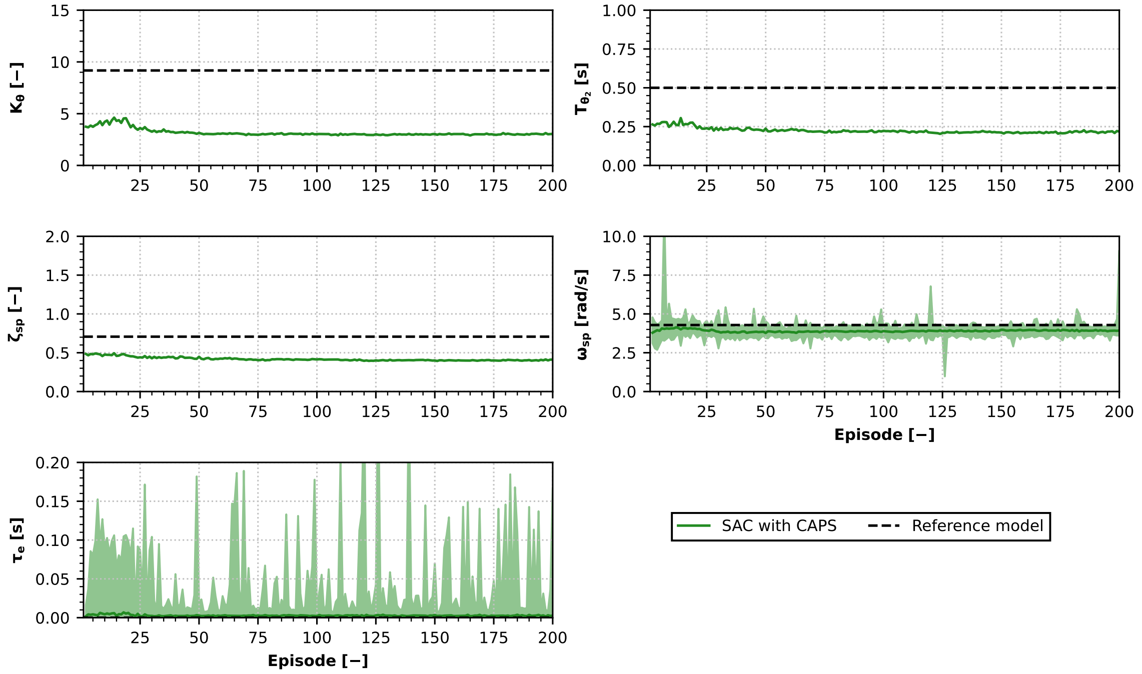


Fig. 7 Short period parameters obtained from LOES fits during offline learning for the SAC controller with CAPS. Solid green lines present the median and shaded regions in green show all successful runs.

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).

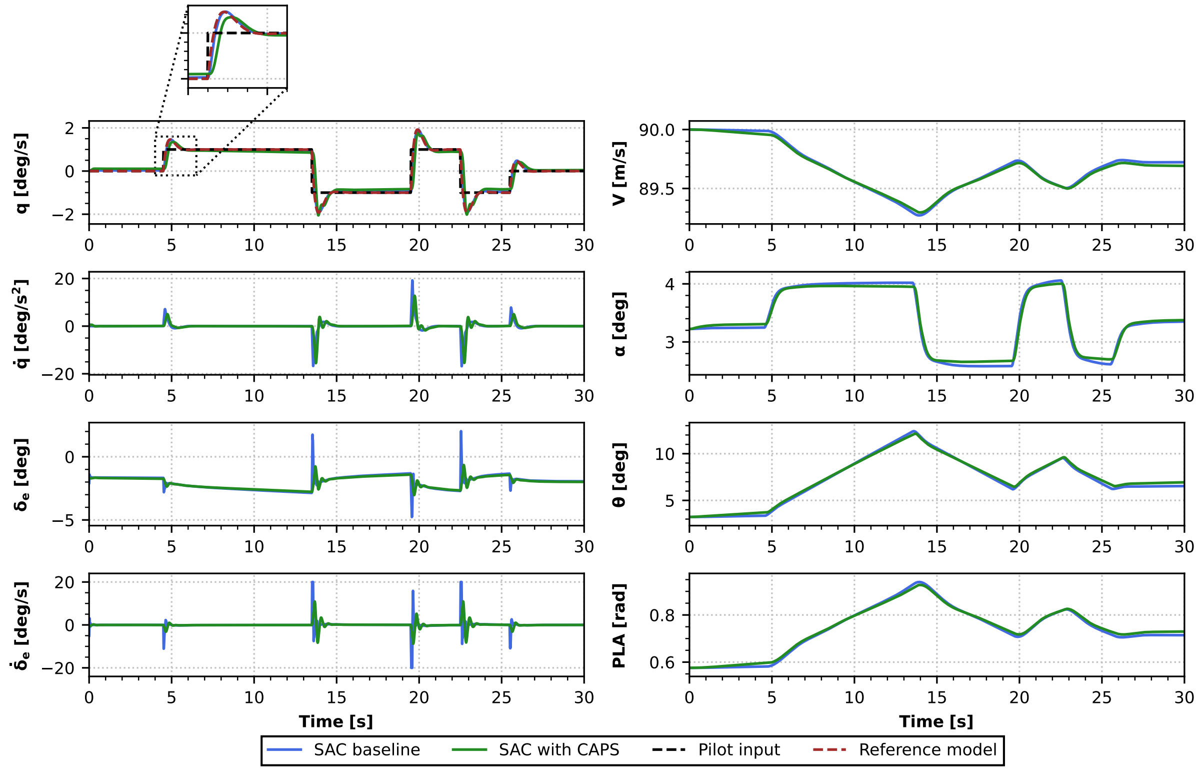


Fig. 8 Time response of the SAC baseline controller and SAC controller with CAPS for the 3-2-1-1 evaluation signal.

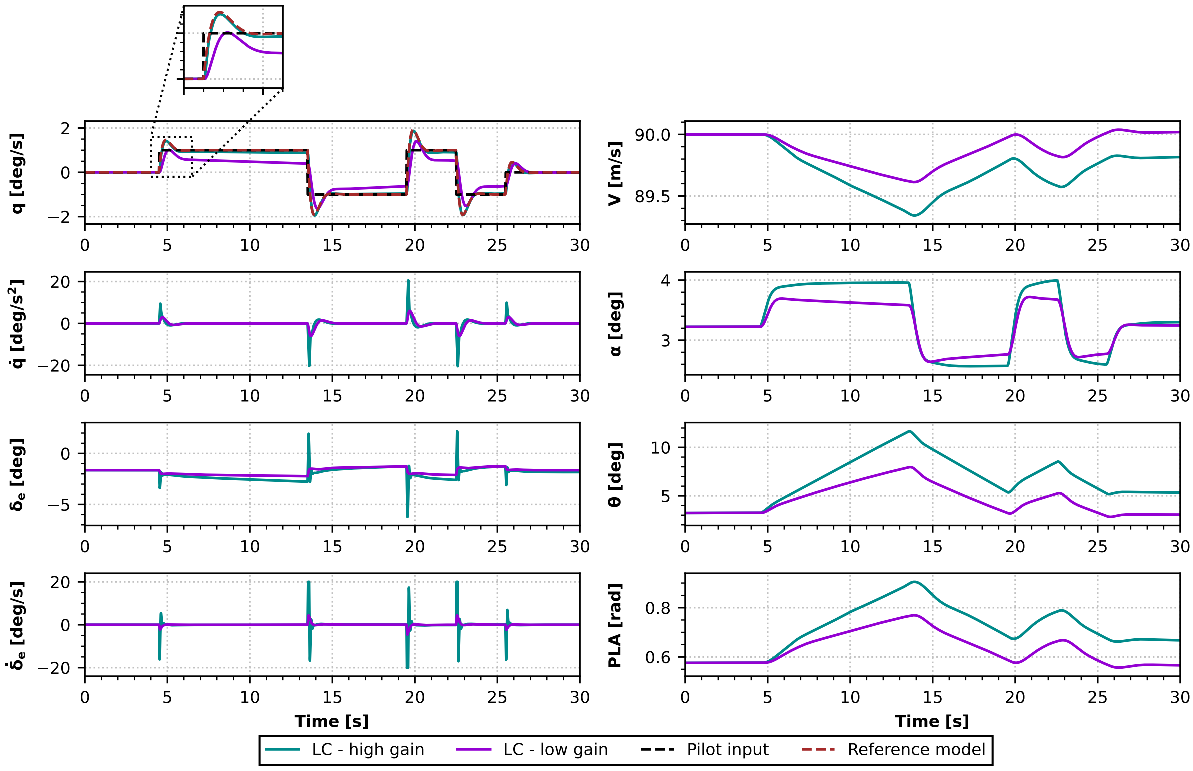


Fig. 9 Time response of the LCs with low and high gains for the 3-2-1-1 evaluation signal.

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.

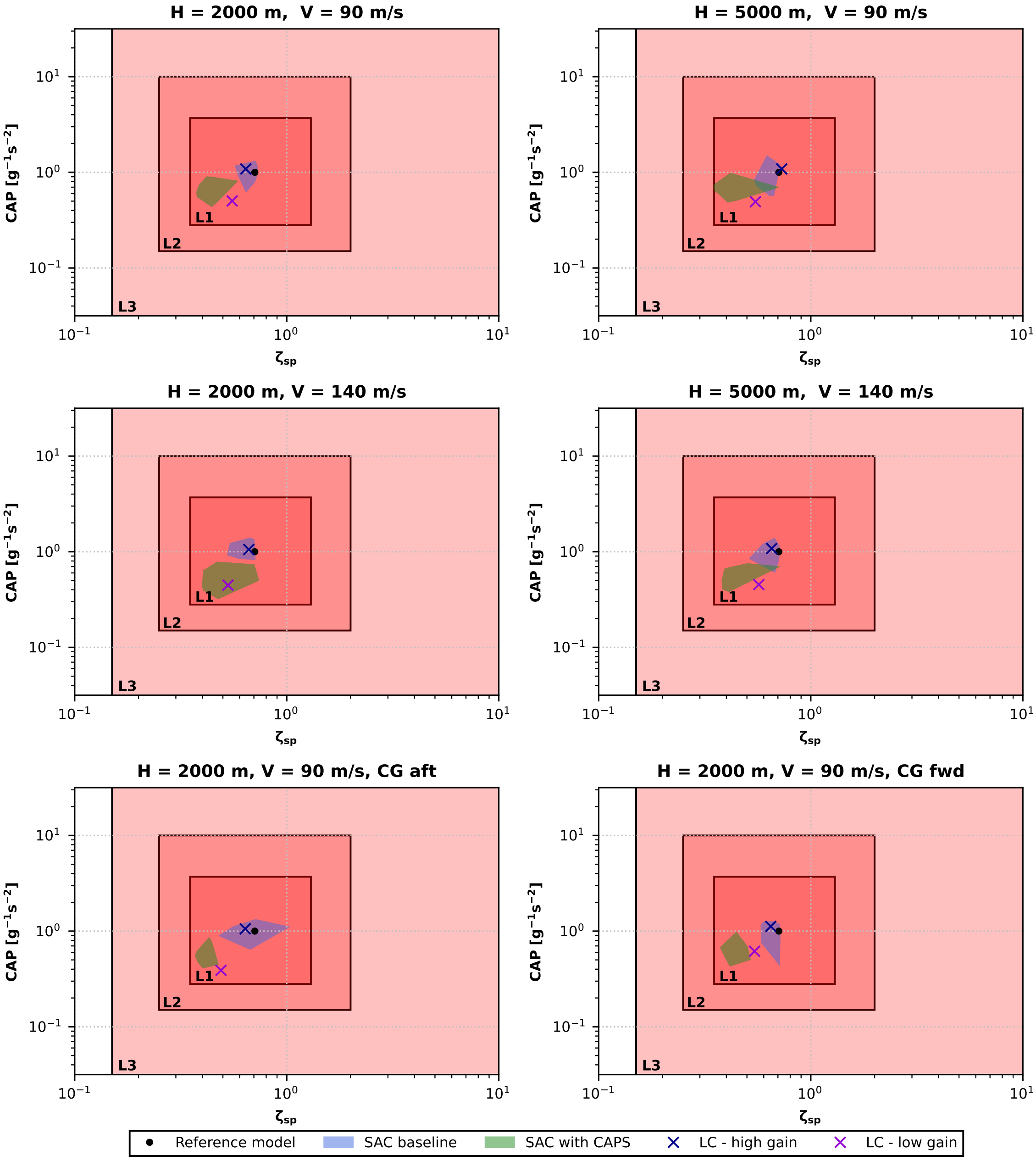


Fig. 10 The equivalent CAP$_e$ for various flight conditions and CG shifts obtained from online evaluation. Shaded blue and green areas show the all successful runs for the SAC baseline controller and SAC controller with CAPS respectively. The levels of HQ ratings are indicated with the red shaded areas.

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.

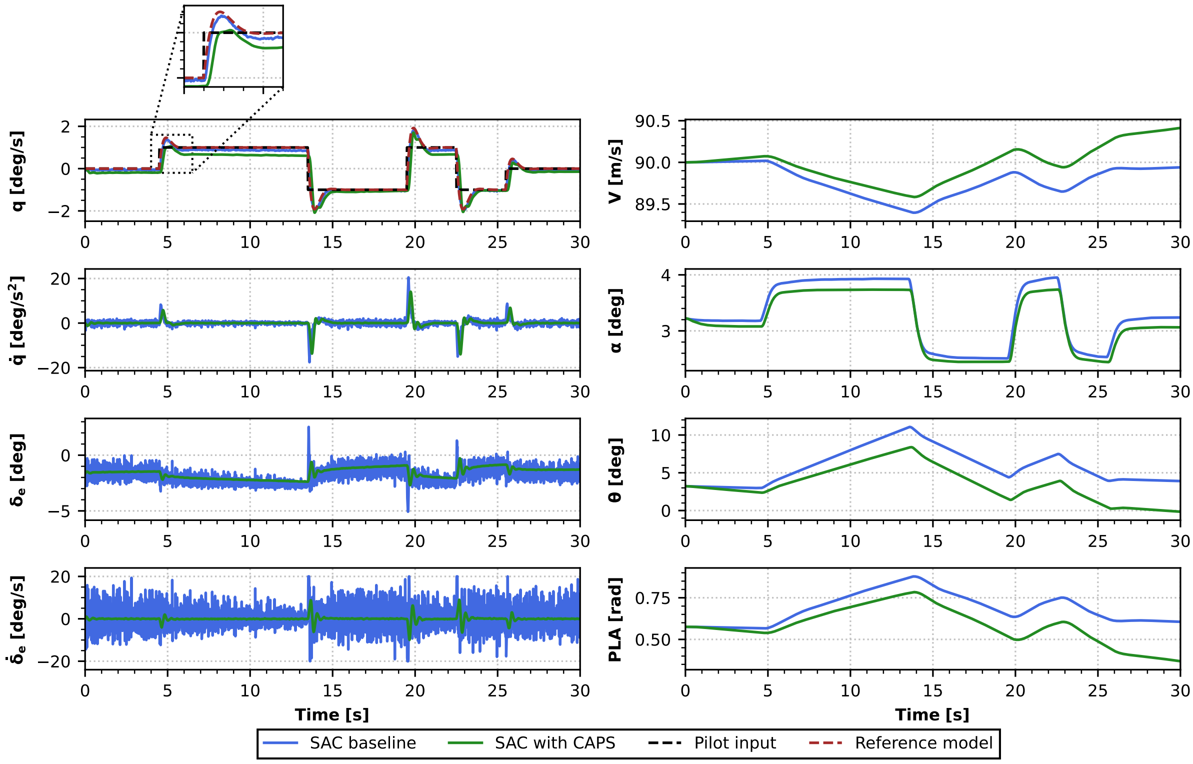


Fig. 11 Time response of the SAC baseline controller and SAC controller with CAPS for the 3-2-1-1 evaluation signal, subject to biased sensor noise.

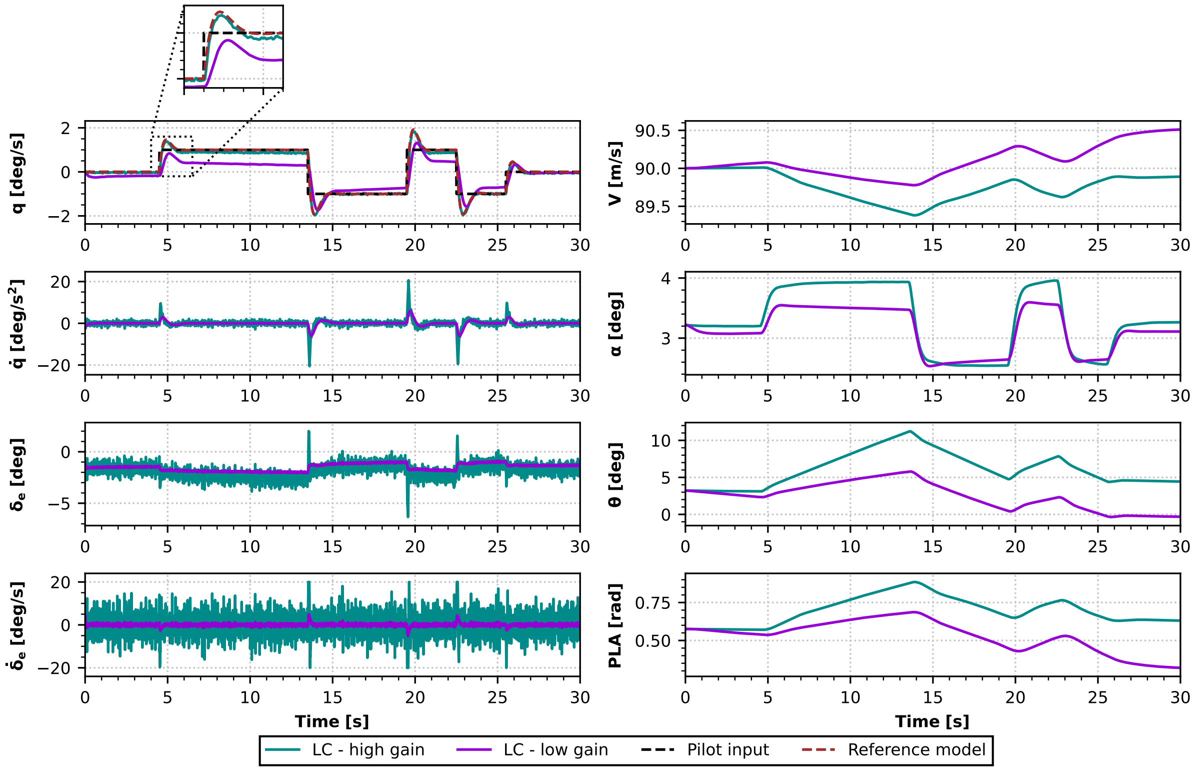


Fig. 12 Time response of the LC controllers with low and high gains for the 3-2-1-1 evaluation signal, subject to biased sensor noise.

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*, 2$^{\text{nd}}$ 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.

Part II

Literature Review and Preliminary Research

*This part has been assessed for the course AE4020 Literature Study.

3

Reinforcement Learning for Flight Control

This chapter will cover the application of Reinforcement Learning (RL) to flight control. The aim of this research is to evaluate and integrate Handling Qualities and Stability properties (HQ&S) in RL flight control applications and therefore it is not considered desired a RL framework from scratch. The focus of this research fill rather be selecting an existing state-of-the-art framework and use it to apply HQ&S analysis. Therefore, this chapter will provide an analysis of the core principle of RL in Section 3.1 for the understanding of RL in general. Subsequently, the key characterizing feature of RL frameworks will be discussed in Section 3.2 after which three state-of-the-art RL frameworks and their application to flight control will be presented in Section 3.3. To conclude, a summary with the main findings and answers to the first research questions will be given in Section 3.4.

3.1. Core Principle of Reinforcement Learning

The general idea behind RL is that the learning process is based on interaction with the environment, primarily inspired by how animals learn through trial and error. RL can be distinguished from the two other main fields within Machine Learning (ML); i.e. supervised and unsupervised learning. Where supervised learning uses labeled data from the training set and generalizes for unseen cases during testing, RL rather focusses on learning from its own experiences while interacting with the environment. The distinction between unsupervised learning and RL is more delicate, as both methods use unlabeled data. The difference, however, is that unsupervised learning is more about classification and structuring of the unlabeled data, whereas the main goal of RL is to maximize the reward with the help of unlabeled data [7].

This section will provide an overview of the key concepts and core principle behind RL, necessary to understand the application of RL to flight control. The fundamental theory discussed in this section is based on the approach developed by Sutton and Barto [7].

3.1.1. Agent and Environment Concept

As mentioned before, RL evolves around the principle of learning from interaction. Figure 3.1 shows schematically how the interaction between the learning agent and the environment is realized. The agent performs an action a and in turn receives a state s and reward r from the environment. The goal in RL is to take the best possible set of sequential actions in order to maximize the cumulative reward over time.

Note that the selected action a is dependent on the state s received by the agent. When reinforcement learning is applied to a control problem, the agent, environment and action correspond to the controller, plant and control input respectively.

3.1.2. Markov Decision Processes

When the agent interacts with the environment at a time step t, it receives a state $s_t \in \mathcal{S}$ and based on that state it selects the corresponding action $a_t \in \mathcal{A}(s)$, where $\mathcal{S}$ and $\mathcal{A}(s)$ are the state and action spaces respectively. As a result, the environment provides a new state s_{t+1} and returns a reward $r_{t+1} \in \mathcal{R}$. The agent has to make a decision about which action to take at every time step t, hence the process becomes a sequential decision making problem. The state and reward are random variables and the mathematical representation of the probability of reaching a state and reward at time $t+1$, given all previous states the agent has observed and actions it has taken is given by Equation 3.1.

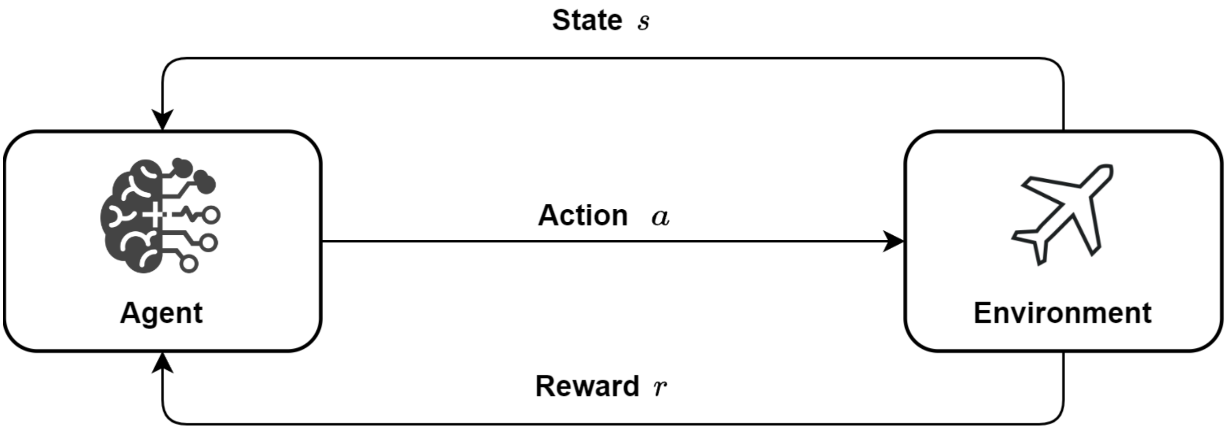


Figure 3.1: Interface between the agent and the environment in Reinforcement Learning with the action, state and reward.

$$\mathcal{P}\{s_{t+1}, r_{t+1} | s_t, a_t, ...s_0, a_0\} \tag{3.1}$$

Equation 3.1 is said to have the Markov property when the state at time $t+1$ is independent of all states and actions up until time $t-1$ given the states and actions at time t. The resulting probabilistic equation then becomes:

$$\mathcal{P}\{s_{t+1}, r_{t+1} | s_t, a_t\} \tag{3.2}$$

Equation 3.2 represents the full dynamics of a Markov Decision Process (MDP). When the sets $\mathcal{S}$, $\mathcal{A}(s)$ and $\mathcal{R}$ have a finite number of elements, the process becomes a finite MDP.

3.1.3. Reward and Return

The goal of the agent is to maximize the sum of expected future rewards over time, which is often referred to as the expected return G_t in RL terminology. Maximizing the expected return does not necessarily mean taking all the actions that yield the best instantaneous rewards, as future rewards should also be taken into account. The expected return for a task with finite time, i.e. an episodic task, can be computed with Equation 3.3.

$$G_t = r_{t+1} + r_{t+2} \dots r_{t+n} = \sum_{k=1}^{n} r_{t+k} \tag{3.3}$$

When there is a task with infinite duration, in other words a continuing task, Equation 3.3 might reach infinity as well. Therefore a discount factor γ is introduced, ensuring that the total expected return remains definite. The newly formulated expected return can now be described by Equation 3.4, where the discount factor takes a value within the range $0 \leq \gamma < 1$.

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \dots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{3.4}$$

The recursive property of the expected return can already be noted:

$$G_t = r_{t+1} + \gamma G_{t+1} \tag{3.5}$$

The reward signal that the agent achieves from the environment is up to the designer of the RL framework. For instance, it is possible to only provide a positive reward to the agent when the final goal of the task is reached or give the agent rewards for intermediate goals to aid in reaching the final goal faster. The latter of course takes some freedom away and might lead to the designer supervising the RL agent in the end after all.

3.1.4. Policy and Value Functions

The policy of a RL agent, denoted by π, is defined by the probability of taking an action, given a state at time step t, as represented by Equation 3.6. In other words, it is a mathematical representation of telling the agent what to do at each state it encounters.

$$\pi(a_t|s_t) = \mathcal{P}\{a_t|s_t\} \tag{3.6}$$

In order for the agent to know what policy yields the maximum expected return, which is in the end the goal of RL, a value function is required. Equation 3.7 shows the definition of the state-value function V^π; the expected return, given the current state while taking policy π. It describes how favorable the current state is based on the policy of the agent and the resulting expected return.

$$V^\pi(s_t) = \mathbb{E}_\pi\left[G_t|s_t\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t\right] \tag{3.7}$$

In a similar fashion, an action-value function Q^π can be formulated by not merely including the current state but also the current action. Equation 3.8 shows the relationship between the action-value, also known as Q-value, and the expected return.

$$Q^\pi(s_t,a_t) = \mathbb{E}_\pi\left[G_t|s_t,a_t\right] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t,a_t\right] \tag{3.8}$$

3.1.5. Bellman Optimality Equations

The recursive property of the expected return, as introduced by Equation 3.5, can be combined with the relation for the state-value function of Equation 3.7 to demonstrate that the state-value function experiences a similar recursive property as shown by Equation 3.9. Exactly the same could be done for the Q-value function, resulting in Equation 3.10.

$$\begin{aligned} V^\pi(s_t) &= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t\right] \\ &= \mathbb{E}_\pi\left[r_{t+1} + \gamma G_{t+1}\right] \\ &= r_{t+1} + \gamma V^\pi(s_{t+1}) \end{aligned} \tag{3.9}$$

$$Q^\pi(s_t,a_t) = r_{t+1} + \gamma Q^\pi(s_{t+1},a_{t+1}) \tag{3.10}$$

These relations are better known as the Bellman equations and are the foundation for update laws in RL, due to the recursive property. The expected return can be maximized by taking an optimal policy π_*, i.e. a policy with higher expected return than all other policies. It is possible that several optimal policies exist that yield exactly the same return. The optimal value functions are reached when the optimal policy is used as shown in Equation 3.11 and Equation 3.12 for the state- and action-value functions respectively. These equations, combined with the recursive properties from Equation 3.9 and Equation 3.10 are known as the Bellman optimality equations. Updating and learning the optimal versions of the policies and value functions is an iterative process, for which multiple methods exist.

$$V^*(s_t) = \text{max}_\pi V^\pi(s_t) \tag{3.11}$$

$$Q^*(s_t,a_t) = \text{max}_\pi Q^\pi(s_t,a_t) \tag{3.12}$$

3.2. Key Characterizing Features

In this section an overview of the key characterizing features in RL will be presented. These features distinguish RL frameworks from each other and will aid in the selection of the framework best suited for this research. The theory described in this section is based on the concepts defined by Dong et al. [8].

3.2.1. Update Laws

Within RL, there are three main approaches for providing a solution to the RL problem through the use of update laws. These are Monte Carlo (MC) methods, Dynamic Programming (DP) and Temporal Difference (TD) learning.

The working mechanism behind MC methods in RL is that they rely on running through a series of states and actions, during which rewards are received. After an episode is completed, the policy of the RL agent is updated. This already highlights the weakness of MC methods, because the parameters are updated episodically and thus not at every time step. The benefit of MC methods, however, is that no prior knowledge of the environment is required. Furthermore, bootstrapping, which is the reliance of the current estimate on the previous estimate, does not occur.

In contradiction with MC methods, the framework of DP does require full knowledge of the environment and heavily relies on it. DP generally uses policy evaluation and policy improvement, where the Bellman equations are used as update rule. The value function is iteratively improved at every time step by evaluating the current policy and subsequently the policy is improved using the improved value function. In this way bootstrapping occurs, as the current estimates of the value functions and policies rely on the previous estimates.

TD learning is a balance between MC methods and DP; it does rely on bootstrapping but does not require complete knowledge of the environment it interacts with. Equation 3.13 expresses the TD update rule for the current value function $V(s_t)$, based on its own estimate, the reward, the target value function $V(s_{t+1})$ and the TD learning rate λ.

$$V(s_t) \leftarrow V(s_t) + \lambda \left[r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \right] \tag{3.13}$$

3.2.2. Exploration vs. Exploitation

One of the common trade-offs that needs to be made in RL is the one between exploration and exploitation. When the agents takes random actions it has a higher chance of exploring unseen or unknown areas of the environment. This could be beneficial for finding an optimal policy. An agent that relies fully on exploitation on the other hand, always takes the greedy action, i.e. the action that leads to highest immediate reward according to the value function. The downside of this is that if the value function is not accurate, it will never find the optimal value function.

In practice, the trade-off between exploration and exploration is not binary. It is for example possible to add some noise to a greedy agent for exploration when the policy is not already stochastic. An alternative approach is to apply a ratio between exploration and exploitation with a factor ϵ, where exploration is multiplied with ϵ and exploitation with $(1-\epsilon)$. In such an update law the factor ϵ can be decayed over time, such that the agent has an exploring nature in the beginning whilst becoming more greedy the longer the learning process takes.

3.2.3. On-policy vs. Off-policy Learning

On-policy and off-policy learning can be distinguished by the way the executed action is used for learning. With on-policy learning the agent uses the executed action directly for updating its policy. Off-policy learning, however, is based on improving a policy that is not used for executing the actions. This is approach has the benefit that it can use more exploration for better generalization, but has the disadvantage of slow convergence.

3.2.4. Model-based vs. Model-free

The knowledge of the environment significantly influences the way RL agents learn. When a state-transition model is required for the agent, the learning framework is called model-based. DP methods fall under this category as they require full knowledge of the environment.

For model-free RL, there is no need for a state-transition model and the agent treats the environment as a black box. The agent can directly try to optimize its policy or develop a model by itself through the use of model identification. The latter utilizes a model of the environment, but it is not required before the learning process starts and therefore it is still considered to be model-free.

3.2.5. Offline vs. Online Learning

The learning process of the agent can be either performed offline or online. The latter requires very sample-efficient algorithms, as there is usually no room for mistakes while learning during interaction. When online learning is applied to fault tolerant flight control, it is often referred to as adaptive control.

On the other side, for offline learning it is not required to learn as quickly as possible and mistakes are allowed. It can however be used for fault-tolerant flight control, in the form of robust control. This is because methods using offline learning, like Deep Reinforcement Learning (DRL) frameworks commonly have a higher generalization power. It is not uncommon to use both offline and online learning during training. A hybrid form could be used where offline learning can be implemented to learn basic knowledge about the system after which online learning can be applied during real-time interaction with the environment.

3.2.6. Discrete vs. Continuous Learning

Most of the theory developed so far holds for discrete RL problems. The flight control application, however, contains continuous state and action spaces and when these are discretized the problem can grow very large resulting in the curse of dimensionality. Nevertheless, function approximation can provide a solution for this issue. Numerous methods of function approximation exist, like least-square methods, nearest-neighbour and multivariate splines.

Most state-of-the-art RL frameworks implement Artificial Neural Networks (ANNs) as function approximators due to their nonlinear nature and high generalization power. Recent advances have led to ANNs with multiple hidden layers known as Deep Neural Networks (DNN), resulting even better generalization power, but also a higher complexity and thus longer learning time.

3.3. State-of-the-art Reinforcement Learning Frameworks

Now that the basic principles of and key characterizing features of RL have been discussed, the state-of-the-art algorithms will be outlined in this section. Since the goal of this research is not to develop or optimize a RL framework itself, but rather using an existing framework for higher-level analysis in terms of Handling Qualities and Stability properties (HQ&S), only a brief analysis of the work mechanism behind each framework will be provided. Furthermore, an example of the flight control application for each of the frameworks will be presented based on recently performed research.

3.3.1. Approximate Dynamic Programming Algorithms

In essence, Approximate Dynamic Programming (ADP) is a class of RL, where DP is used in combination with function approximators such that it can be applied in continuous state and action spaces, by avoiding the curse of dimensionality [9]. ADP algorithms frequently use actor-critic structures, where the actor maps the state to the action, resembling the policy, and the critic approximates the value function. Several on-policy Adaptive Critic Designs (ACDs) have been developed and can be distinguished by multiple parameters [10]. Heuristic Dynamic Programming (HDP) uses the state-value function as an input for the critic, whereas Dual Heuristic Programming (DHP) uses the gradient of the state-value function. Global Dual Heuristic Programming (GDHP) uses both the state-value and its gradient for as input for the critic. These three versions can become action-dependent when the action is added to the critic input, or incremental when the environment is approximated by an incremental model to remove the model-dependency.

Flight Control Application

The concept of ACDs has been successfully applied to flight control, for example to the Cessna Citation II [3]. In the study, an Incremental Dual Heuristic Programming (IDHP) is used to control a 6-degree-of-freedom model in a decoupled fashion, where a target critic is added to decrease learning instability. As can be seen in Figure 3.2, PID controllers are used to give rate commands and the IDHP agent is charge of the control surface deflections. The agent is trained fully online and it is demonstrated that the controlled aircraft shows fault tolerant behaviour.

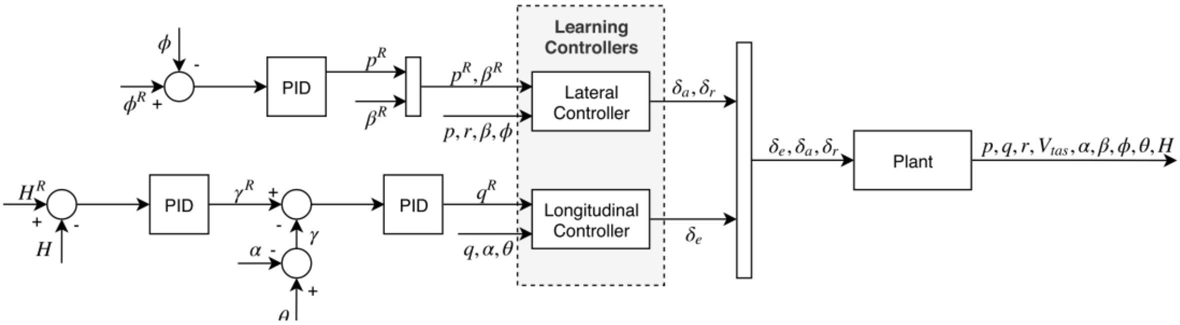


Figure 3.2: Block diagram of decoupled IDHP agent for altitude and attitude control applied to the Cessna Citation II [3].

Another example of the implementation of ACDs is the application of Incremental Global Dual Heuristic Programming (IGDHP) to a nonlinear longitudinal model of the F-16 [11]. It shows that the IGDHP has superior tracking performance on an online attitude tracking problem over regular GDHP. Next to that, it shows better fault-tolerance for experiments with a decrease in elevator bandwith, control effectiveness and partial horizontal stabilizer damage.

3.3.2. Twin-Delayed Deep Deterministic Policy Gradient

The working mechanism behind Deep Deterministic Policy Gradient (DDPG) algorithms is an actor-critic structure, similar to ACDs, where stochastic policies are integrated over the state space to form a deterministic policy. It is a model-free, off-policy learning algorithm, where DNNs are used for function approximation. Furthermore, experience replay is used for learning, meaning that samples are taken from a replay buffer that stores states, actions and rewards from the past. DDPG algorithms use target networks for the actor and critic to avoid divergence [12].

Twin-Delay DDPG (TD3) builds upon the approach of DDPG and is adapted with multiple improvements, one of which is the use of two critic networks. This helps reducing the overestimation of the value function, as the minimum of the two networks is selected for updating the target networks. Next to that, some noise is added to the target action for updating the actor such that exploitation of minor improvements is avoided. As a last improvement, TD3 uses delayed policy updates, meaning that the policies are updated at a lower rate than the Q-values with aim of improving stability [13].

Flight Control Application

TD3 has been applied to the Flying-V aircraft developed by the TU Delft recently [14]. The main goal of the research was to investigate how a TD3 agent could be used to cope with aerodynamic model uncertainty, using offline learning. The control diagram of the developed longitudinal controller with the purpose of tracking altitude is shown in Figure 3.3. The outcome of the research was that TD3 is able to track reference signals related to the altitude and showed robustness to aerodynamic model uncertainty of 25%.

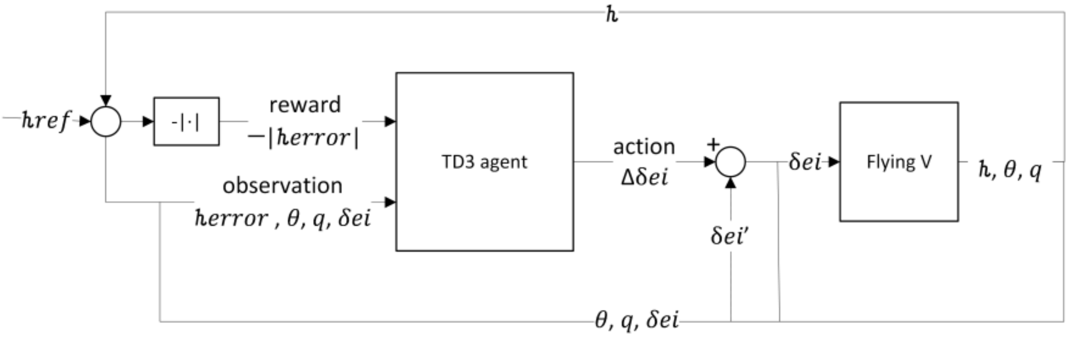


Figure 3.3: Block diagram of TD3 agent for altitude control applied to the Flying-V [14].

3.3.3. Soft Actor-Critic

The Soft Actor-Critic (SAC) is another improved version of the DDPG algorithm explained in the previous subsection. It is an off-policy learning algorithm with a stochastic policy, which is favourable for exploration. The SAC algorithm adds an entropy term to the Q-function, which is a metric of the randomness in the policy at the given state. This ensures that states with high randomness and thus uncertainty are explored, which makes the probability of reaching a superior solution more likely. Similar to TD3, the SAC algorithm learns two Q-functions to avoid overestimating the value. When the SAC agent is used for evaluation, the mean of the stochastic policy is used, such that randomness is removed [15].

Flight Control Application

The SAC controller has been implemented to the Cessna Citation II, in the form of a cascaded control system shown in Figure 3.4 [4]. Offline training was performed for the nonlinear coupled aircraft model. Results of the research show that the algorithm is robust to several failure cases, related to reduced control effectiveness, center of gravity shifts and structural failures.

Three state-of-the-art RL algorithms have been presented in this brief analysis. For the implementation of this research, the preferred method of training is offline, as the evaluation and integration of HQ&S in RL flight control is proof-of-concept. This leaves the TD3 and SAC algorithms, as the strength of ACDs is its online adaptability. The SAC is preferred over the TD3 algorithm, as it has a better sampling efficiency than TD3. Furthermore, the code used for the SAC flight control implementation is readily available, which means more time can be spent on the HQ&S research.

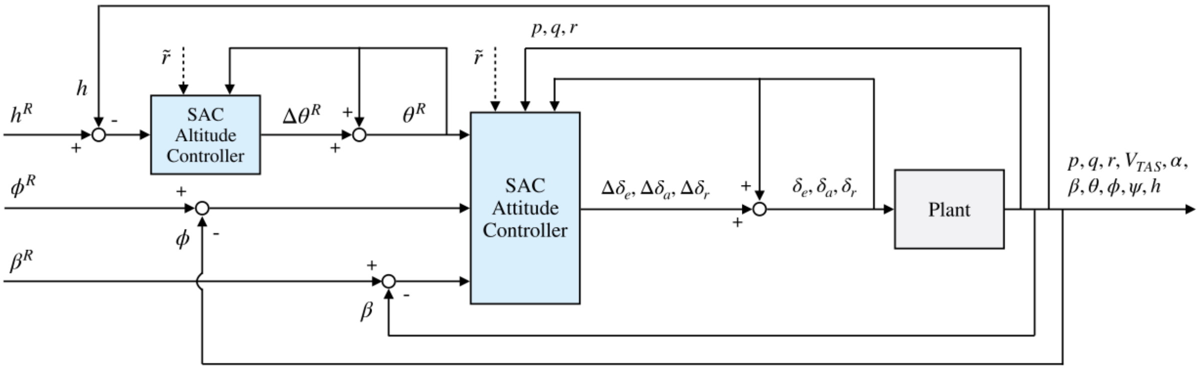


Figure 3.4: Block diagram of SAC agent for altitude and attitude control applied to the Cessna Citation II [4].

3.4. Synopsis

This chapter provided an analysis of the application of RL in flight control. It started with the core principle of RL, which forms a foundation for the understanding of the state-of-the-art algorithms. Furthermore, an overview of the key characterizing features of RL was presented, with the aim of clarifying the distinction between individual RL frameworks. Lastly, three state-of-the-art RL frameworks were presented, together with their implementation to flight control problems. The SAC algorithm was selected as state-of-the-art RL framework due its high generalization power, robustness, sample efficiency and ease of implementation as the code is readily available. The latter is an important criterion, as this research puts emphasis on the evaluation of HQ&S instead of designing a RL framework. With the selection of the SAC algorithm, **RQ 1.1** has been answered.

As the code presented in the research by Dally et al. will be used, some design choices are already made [4]. The algorithm is applied to the Cessna Citation II, hence the aircraft model and flight control framework is selected. Furthermore, the integration of the RL and flight control frameworks are already done. This provides answers to **RQ 1.2** and **RQ 1.3** and thus **RQ 1** is answered entirely.

4

Handling Qualities and Stability Properties

The state-of-the-art Reinforcement Learning (RL) frameworks applied to flight control that were analyzed in Chapter 3 have one key element in common, their performance was assessed primarily on aspects related to the tracking error. There is however the need for a more general description of the performance of RL flight control systems, including how a pilot will experience flying such systems. This is where the Handling Qualities (HQ) can come into play, as they describe how the pilot experiences the (augmented) aircraft's response to pilot inputs [16]. Next to that, the stability properties of an aircraft give more insight in how an aircraft responds to disturbances and if it is able to return to a steady state. Furthermore, it might also aid in the design of RL flight control systems when using the Handling Qualities and Stability properties (HQ&S) in the learning process, e.g. by reward signal adaptation.

Before diving deeper into the world of HQ&S, two aspects have to be discussed. First of all, it should be noted that in literature the terms Flying Qualities (FQ) and Handling Qualities (HQ) are sometimes used interchangeably, but there is a clear distinction between them. The FQ are related to how the pilot experiences the aircraft's behaviours with respect to the mission task, whereas the HQ address the response. Figure 4.1 illustrates the distinction between FQ and HQ for a fly-by-wire controlled aircraft, where the pilot gives commands to the flight control system. FQ are therefore higher-level performance metrics, but are harder to quantify as it often involves the pilot's opinion.

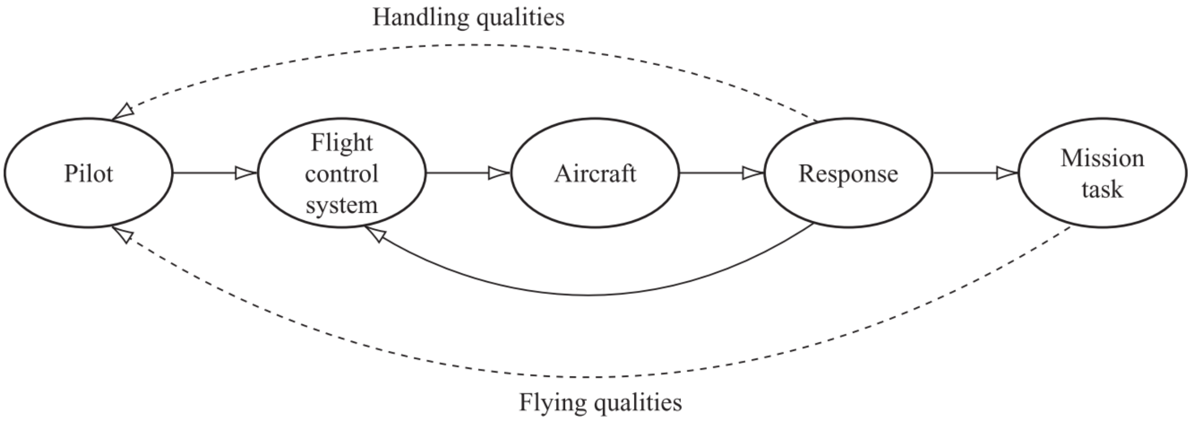


Figure 4.1: Schematic representation of a pilot performing a flight mission task. The relation of the pilot to the Flying and Handling Qualities is shown [16].

Secondly, this chapter will address the longitudinal motion of the aircraft and the short period response in particular, because there is an extensive amount of literature written on this subject. Besides that, to the best of the author's knowledge, it is one of the first researches in which the HQ&S are evaluated and adapted for a RL flight control system and therefore it is considered to be a proof-of-concept.

This chapter will start with an overview of the most relevant HQ&S that might be applied to a RL flight control system in Section 4.1. The analysis of HQ&S of nonlinear or intelligent flight control systems is not trivial. As a RL flight control system falls inside this category, Section 4.2 is dedicated to this matter. Several potential methods for placing the HQ&S at certain design points, will be discussed in Section 4.3. Finally, the main take-aways of this chapter will be summarized in Section 4.4.

4.1. Overview of Relevant Handling Qualities and Stability Properties

Civil aviation authorities have set a wide range of requirements for the safety of the aircraft, but there are no strict requirements or elaborate guidelines of the FQ and HQ [16]. The military standards (MIL-STD-1797A), however, provide detailed descriptions of the FQ and HQ&S requirements and on their acceptable means of compliance to aid the design of aircraft [6]. These standards are commonly used and will therefore be applied to this thesis as well.

The military standards apply to different aircraft classes and flight phase categories, for which the flying qualities are rated with three individual levels. The aircraft are subdivided into classes according to their weight and manoeuvrability as specified by Table 4.1.

Table 4.1: The four aircraft classes for FQ and HQ&S qualification according to the military standards [6, 16].

Class I	Small light aeroplanes.
Class II	Medium weight, low to medium manoeuvrability aeroplanes.
Class III	Large, heavy, low to medium manoeuvrability aeroplanes.
Class IV	High manoeuvrability aeroplanes.

The FQ and HQ requirements are also dependent on the flight phase. As one can imagine, the requirements during normal operating conditions in cruise alter from the ones that apply during a terminal flight phase. Table 4.2 shows the three different categories related to the flight phase.

Table 4.2: The three categories of flight phases for FQ and HQ&S qualification according to the military standards [6, 16].

Category A	Non-terminal flight phases that require rapid manoeuvring, precision tracking, or precise flight path control.
Category B	Non-terminal flight phases that require gradual manoeuvring, less precise tracking and accurate flight path control.
Category C	Terminal flight phases that require gradual manoeuvring and precision flight path control.

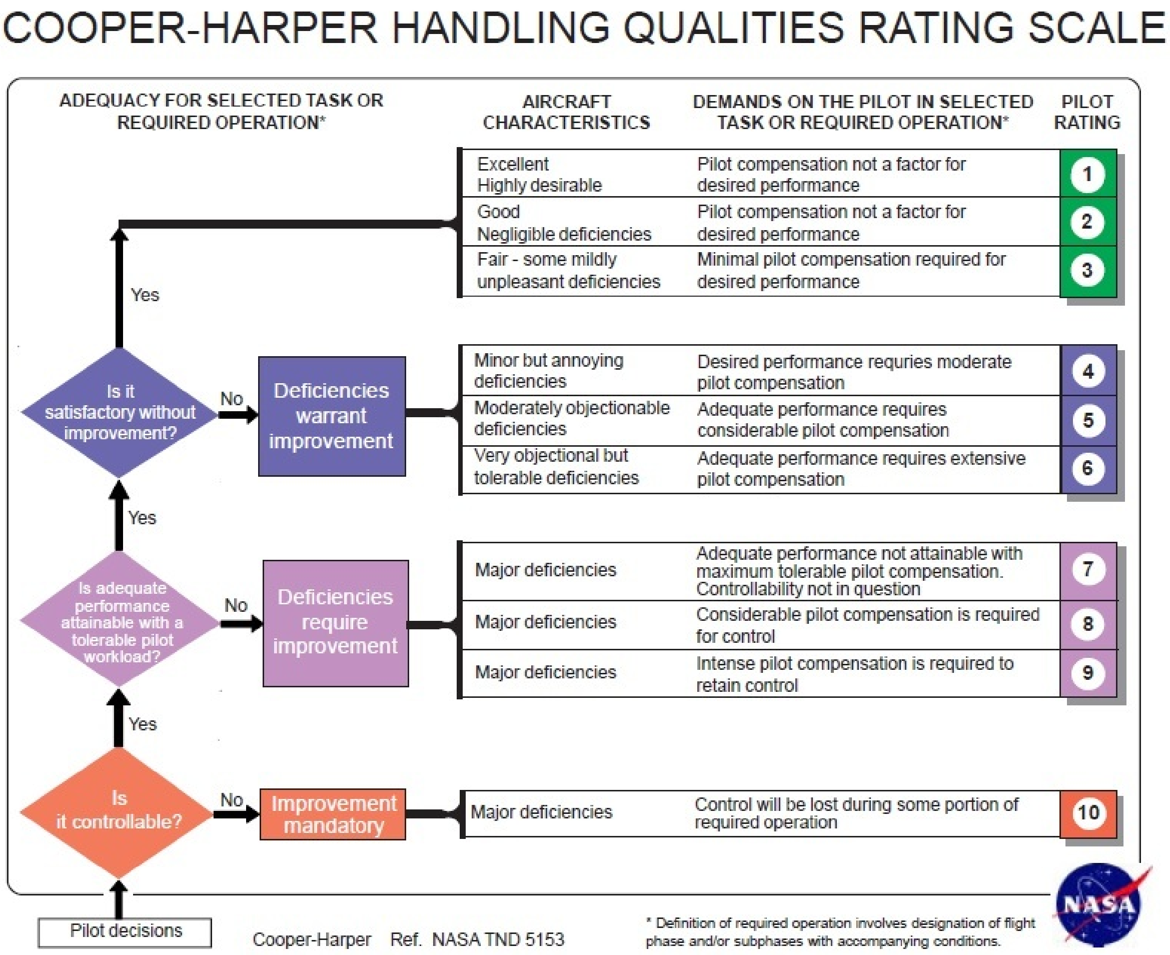
Finally, the FQ and HQ can be divided into three different levels that specify the pilot workload as shown in Table 4.3.

Table 4.3: Three levels of FQ and HQ&S according to the military standards [6, 16].

Level 1	Flying qualities clearly adequate for the mission flight phase.
Level 2	Flying qualities adequate to accomplish the mission flight phase, but with an increase in pilot workload and, or, degradation in mission effectiveness.
Level 3	Degraded flying qualities, but such that the aeroplane can be controlled, inadequate mission effectiveness and high, or, limiting, pilot workload.

4.1.1. Cooper-Harper rating scale

Translating the pilot workload into mathematically quantifiable guidelines or requirements is not trivial. A rating scale was developed which had the aim to express the pilot's experience into a number between 1 to 10, with 1 being the value for the lowest workload and 10 for the highest [17]. The full scale is visualized in Figure 4.2. Note that the FQ and HQ&S levels expressed in Table 4.3 correspond to the ratings of the Cooper-Harper rating scale as follows: Level 1 is equal to Cooper-Harper ratings 1-3, Level 2 corresponds to 4-6 and Level 3 to 7-9 [16].


COOPER-HARPER HANDLING QUALITIES RATING SCALE
ADEQUACY FOR SELECTED TASK OR REQUIRED OPERATION*
Is it satisfactory without improvement?
Yes
No
Deficiencies warrant improvement
Is adequate performance attainable with a tolerable pilot workload?
Yes
No
Deficiencies require improvement
Is it controllable?
Yes
No
Improvement mandatory
Pilot decisions
Cooper-Harper Ref. NASA TND 5153
NASA

AIRCRAFT CHARACTERISTICS	DEMANDS ON THE PILOT IN SELECTED TASK OR REQUIRED OPERATION*	PILOT RATING
Excellent Highly desirable	Pilot compensation not a factor for desired performance	1
Good Negligible deficiencies	Pilot compensation not a factor for desired performance	2
Fair - some mildly unpleasant deficiencies	Minimal pilot compensation required for desired performance	3
Minor but annoying deficiencies	Desired performance requries moderate pilot compensation	4
Moderately objectionable deficiencies	Adequate performance requires considerable pilot compensation	5
Very objectional but tolerable deficiencies	Adequate performance requires extensive pilot compensation	6
Major deficiencies	Adequate performance not attainable with maximum tolerable pilot compensation. Controllability not in question	7
Major deficiencies	Considerable pilot compensation is required for control	8
Major deficiencies	Intense pilot compensation is required to retain control	9
Major deficiencies	Control will be lost during some portion of required operation	10

* Definition of required operation involves designation of flight phase and/or subphases with accompanying conditions.

Figure 4.2: Cooper-Harper rating scale quantifying pilot's workload ratings [17].

The Cooper-Harper rating scale is especially useful for experiments with pilots, but for an initial evaluation of the HQ&S of RL flight control systems it is recommended to start testing with simulations that do not involve a pilot (yet). Hence, from this point onwards, HQ&S that can be obtained from numerical simulation will be discussed.

4.1.2. Longitudinal Modes

The aircraft longitudinal motion is subject to two characteristic eigenmodes; the phugoid and the short period [18]. The pitch rate q to a pilot control-deflection in $\delta_{e,s}$ transfer function in the longitudinal case can be expressed by the following model [6]:

$$\frac{q(s)}{\delta_{e,s}(s)} = \frac{K_\theta s\left(s + 1/T_{\theta_1}\right)\left(s + 1/T_{\theta_2}\right) e^{-\tau_e s}}{\left[s^2 + 2\zeta_p \omega_p + \omega_p^2\right]\left[s^2 + 2\zeta_{sp}\omega_{sp} + \omega_{sp}^2\right]} \tag{4.1}$$

It should be noted that this is a representation of the linear simplified model for the longitudinal motion of an aircraft, which in reality can be of higher order due to added control, sensor or actuator dynamics. For now, the higher order dynamics are captured in the time delay τ_e, but a more deliberate analysis with Low Order Equivalent System (LOES) fits will be provided later on in this chapter. There are several HQ&S requirements posed by the military standards that apply to the transfer function parameters in Equation 4.1, which are provided in Table 4.4 [6]. The requirements are set for the different levels of pilot workload as defined in Table 4.3. It can be observed that the phugoid mode is allowed to be unstable for Level 3 (negative damping ratio), as long as the time constant T_{θ_1} is larger than 55 seconds.

Table 4.4: HQ&S requirements of the aircraft's longitudinal response for the three levels of pilot workload [6].

	Level 1	**Level 2**	**Level 3**
Damping ratio short period [-]	$0.35 \leq \zeta_{sp} \leq 1.3$	$0.25 \leq \zeta_{sp} \leq 2.0$	$0.15 \leq \zeta_{sp}$
Damping ratio phugoid [-]	$\zeta_p > 0.04$	$\zeta_p > 0.0$	$T_{\theta_1} > 55$ s
Natural frequency short period [rad/s]	$\omega_{sp} \geq 1.0$	$\omega_{sp} \geq 0.6$	-
Time delay [s]	$\tau_e < 0.1$	$\tau_e < 0.2$	$\tau_e < 0.25$

For most aircraft, the short period mode can be isolated from the phugoid mode, as it is a much quicker response. The model for the short period is then represented by the following equation [6]:

$$\frac{q(s)}{\delta_{e,s}(s)} = \frac{K_\theta \left(s + 1/T_{\theta_2}\right) e^{-\tau_e s}}{s^2 + 2\zeta_{sp}\omega_{sp} + \omega_{sp}^2} \tag{4.2}$$

4.1.3. Control Anticipation Parameter

The guideline for the design and analysis of the aircraft's short period behaviour that has the highest priority according to the military standards is the Control Anticipation Parameter (CAP) [6]. It was defined in a study by Bihrle, as a metric that describes the pilot's anticipation on how the aircraft responds in the future based on the initial angular pitch acceleration after a longitudinal stick deflection [19]. More concretely, the CAP is the instantaneous angular pitch acceleration $\ddot{\theta}_0$ over the steady state normal load factor $n_{z_{ss}}$ as represented by Equation 4.3.

$$CAP = \frac{\ddot{\theta}_0}{n_{z_{ss}}} \tag{4.3}$$

When a change in flight path is required and the CAP is too low the aircraft tends to feel sluggish and the pilot might increase the stick deflection which in turn could result into Pilot Induced Oscillations (PIO). On the other hand, when the CAP is too high the aircraft tends to feel much more sensitive and could result into oscillatory behaviour as well. Therefore, boundaries were set on the CAP and short period damping ratio ζ_{sp} for the various flight phase categories [6]. Figure 4.3 shows the CAP and damping ratio bounds for an aircraft in flight phase category A with Level 1, 2 and 3 pilot workload ratings, as defined in Section 4.1.

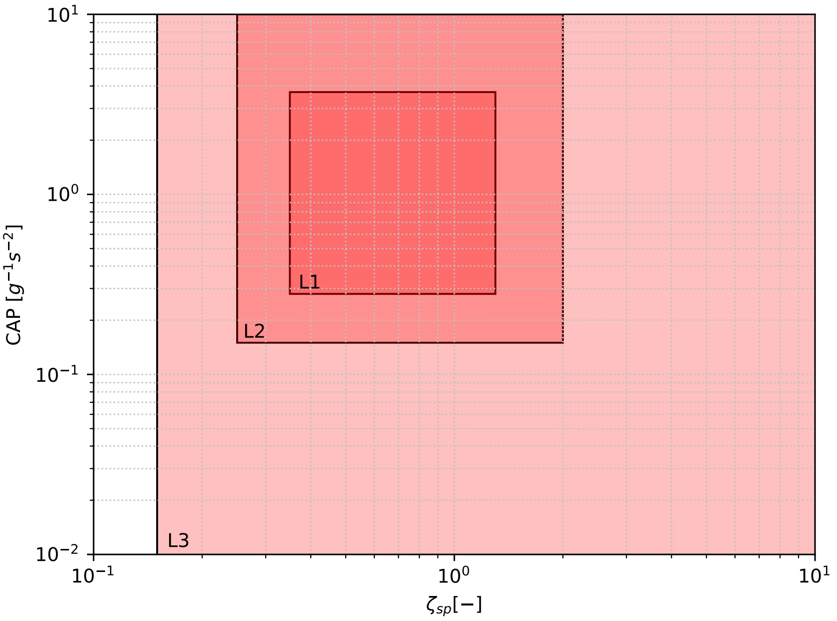


Figure 4.3: CAP boundaries for the three levels of pilot workload [6].

The CAP as specified in Equation 4.3 can be obtained in the time domain from a step response analysis. It was found, however, that the CAP is also related to the short period undamped natural frequency ω_{sp} and the acceleration sensitivity parameter n_α [19]:

$$CAP = \frac{\omega_{sp}^2}{n_\alpha} \tag{4.4}$$

From a frequency domain analysis and the Laplace final value theorem the CAP could be related to the short period time constant T_{θ_2} as described in Equation 4.5. Here the constant aircraft velocity V and gravitational constant g are used. It should be noted, however, that this relationship is only valid for second order short period models.

$$CAP = \frac{\omega_{sp}^2}{\frac{V}{g}\frac{1}{T_{\theta_2}}} \tag{4.5}$$

Bischoff further extended the CAP definition to higher order systems and augmented aircraft, where control dynamics are no longer neglected [20]. Figure 4.4 illustrates the pitch acceleration step response of a Higher Order System (HOS) and clearly shows that the maximum pitch acceleration is reached at some time t after $t = 0$, which is not the case for second order systems. To account for this effect, an attenuation factor was developed. Initially, DiFranco added this factor for a feel system [21] and thereafter it was further developed to account for additional higher order dynamics as well by Bischoff [20]. In essence, the attenuation factor merely scales the CAP found from the second order model with the ratio between the maximum acceleration of the HOS $\ddot{\theta}_{mas}$ and the instantaneous acceleration of the second order short period model $\ddot{\theta}_{0,sp}$. Hence, the resulting CAP of the HOS, also referred to as CAP', becomes:

$$\ddot{\theta}_{nd} = \frac{\ddot{\theta}_{max}}{\ddot{\theta}_{0,sp}} \tag{4.6}$$

$$CAP' = \frac{\ddot{\theta}_{max}}{n_{z_{ss}}} = \frac{\omega_{sp}^2}{\frac{V}{g}\frac{1}{T_{\theta_2}}}\ddot{\theta}_{nd} \tag{4.7}$$

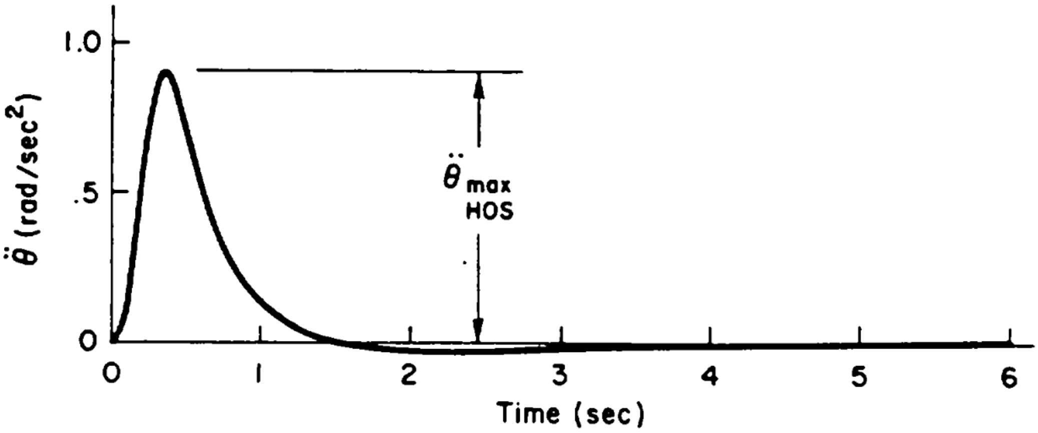


Figure 4.4: Pitch acceleration of a HOS as the result of a step input by the pilot [6].

Finally, from linear analysis it can be shown that the CAP is a function of the center of gravity $\bar{x}_{cg}$ and aerodynamic center $\bar{x}_{ac}$. Roskam translates these parameters, together with other aircraft parameters into the so-called maneuver margin [22]. Hence, the CAP regions specified by Figure 4.3 can be converted to the maneuver margins of the airplane, which be useful for a sensitivity analysis.

4.1.4. Bandwidth and Phase Delay Criteria

The bandwidth ω_{BW} is defined as the highest frequency at which the pilot can close the loop, without the possibility of unstable behaviour [23]. It is the frequency at which the Gain Margin (GM) is at least 6 dB and the Phase Margin (PM) is no less than 45 degrees. The bandwidth is therefore either gain- or phase-limited, in general it could be said that the larger the bandwidth the better, because the pilot is able to follow all commands below this frequency. Nevertheless, it does not necessarily mean that aircraft with the same bandwidth have similar HQ. The shape of the phase diagram significantly influences the HQ, as a steeper roll-off at the point of neutral stability (-180 degrees) results in a more rapid decrease of the PM when the pilot increases its gain [6]. Therefore, a phase delay parameter τ_p was introduced [23]:

$$\tau_p = \frac{\Delta\phi_{2\omega_{-180^\circ}}}{57.3(2\omega_{-180^\circ})} \tag{4.8}$$

In this equation, $\Delta\phi_{2\omega_{-180^\circ}}$ denotes the phase at twice the frequency where the phase ϕ is -180 degrees and $2\omega_{-180^\circ}$ is the corresponding frequency. Figure 4.5 shows the effect of the bandwidth and phase delay on the HQ.

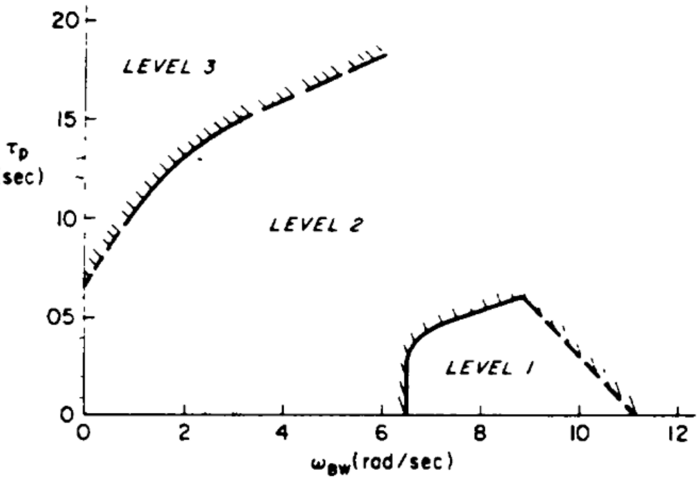


Figure 4.5: HQ shown as pilot workload levels for a category A flight phase, as a function of bandwith and phase delay [6].

4.1.5. Neal-Smith Criterion

The Neal-Smith criterion was developed for pitch angle tracking tasks and used as a metric for Pilot Induced Oscillation (PIO) [24]. A fixed bandwidth ω_{BW} and pilot time delay tau_p are set for the tracking task illustrated in Figure 4.6, where the goal is to find the pilot compensation parameters to meet a ”droop” of no more than 3 dB. The Neal-Smith criterion can be rated on HQ levels as a function of the pilot compensation and the maximum closed loop resonance, for a bandwidth of 3.5 rad/s, according to Figure 4.7 and Figure 4.8, where the latter includes qualitative pilot ratings [23]. The pilot time delay τ_p was originally set at 0.3 seconds, but later it was adapted to 0.25 [6].

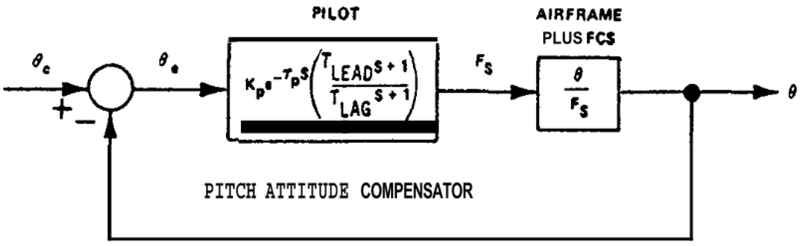

PILOT
AIRFRAME PLUS FCS
θc
+
−
θe
$K_p e^{-\tau_p s}\left(\frac{T_{LEAD}s+1}{T_{LAG}s+1}\right)$
Fs
$\frac{\theta}{F_s}$
θ
PITCH ATTITUDE COMPENSATOR

Figure 4.6: Pitch rate tracking control loop for Neal-Smith criterion [23].

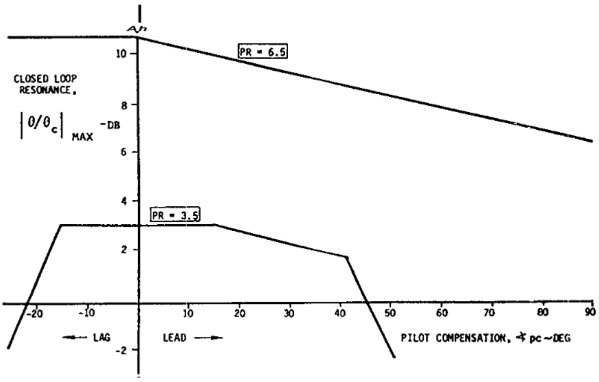


Figure 4.7: Neal-Smith criterion showing the pilot compensation versus the closed loop resonance [23].

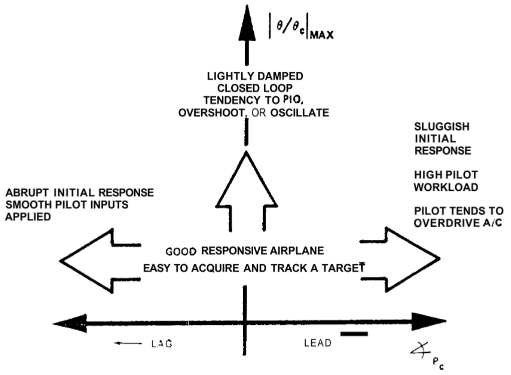

$|\theta/\theta_c|_{MAX}$
LIGHTLY DAMPED CLOSED LOOP TENDENCY TO PIO, OVERSHOOT, OR OSCILLATE
SLUGGISH INITIAL RESPONSE
HIGH PILOT WORKLOAD
PILOT TENDS TO OVERDRIVE A/C
ABRUPT INITIAL RESPONSE SMOOTH PILOT INPUTS APPLIED
GOOD RESPONSIVE AIRPLANE
EASY TO ACQUIRE AND TRACK A TARGET
LAG
LEAD
$\measuredangle_{P_c}$

Figure 4.8: Neal-Smith criterion with pilot comments [23].

4.1.6. Gibson Criterion

The Gibson criterion was developed for highly augmented aircraft and consists of two individual components; the pitch attitude dropback and the phase rate [25]. The former is based on an step input response where the following parameters are used:

- q_{max}, the maximum pitch rate
- q_{ss}, the steady state pitch rate
- DB, the dropback parameter, which is the peak value of the pitch attitude θ_{peak} minus the steady state pitch angle θ_{ss} when the step input is taken away

The effect of the abovementioned parameters on the experience of the pilot when flying the aircraft are qualitatively discribed in Figure 4.9.

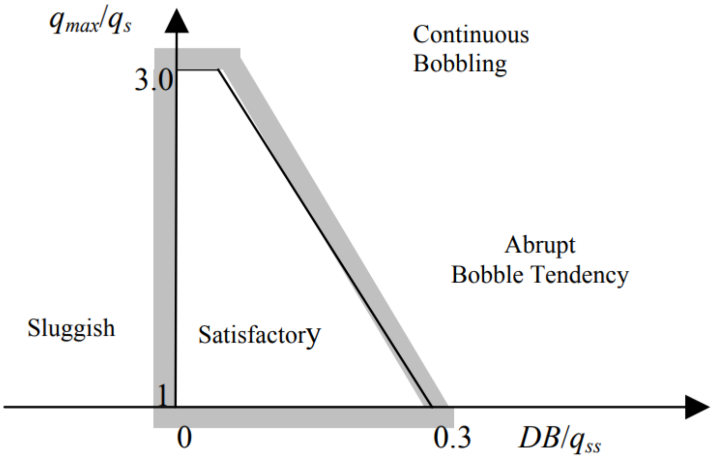


Figure 4.9: Bounds of the Gibson dropback criterion [26].

The other component of the Gibson criterion, the phase rate, is comparable to the phase delay defined in Section 4.1.4. The average phase rate, is a function of the frequency where the phase is -180° and the phase at twice that frequency as defined in Equation 4.9. The relation to the HQ ratings is portrayed in Figure 4.10, where the frequencies are shown in Hz instead of rad/s.

$$\text{Average phase rate} = \frac{-\phi_{2\omega_{-180^\circ}} + 180^\circ}{\omega_{-180^\circ}} \qquad (4.9)$$

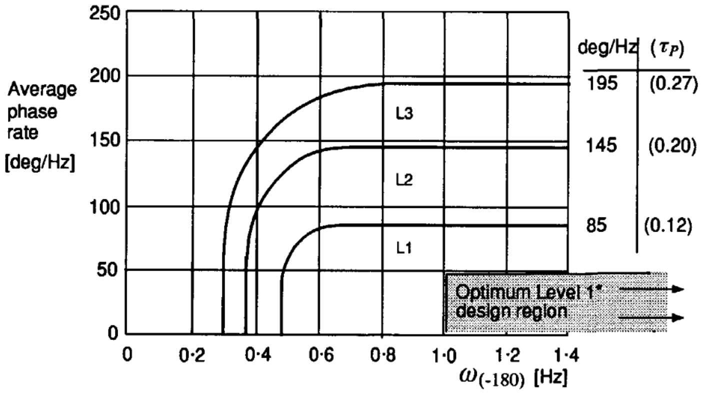


Figure 4.10: Pilot HQ ratings for the Gibson phase rate criterion [25].

4.2. Analysis of HQ&S for nonlinear Flight Control Systems

Most of the previously mentioned HQ&S guidelines have been widely applied to linear flight control systems [6]. When nonlinearities are present in the control system however, for instance due to actuator limiters, the analysis of HQ&S as discussed in the preceding section is not entirely applicable any more. For that reason, this section will cover methods found in literature that demonstrate how to evaluate HQ&S for nonlinear flight control systems.

4.2.1. Time Domain Analysis

Simulation in the time domain is one of the methods to extract HQ&S from an intelligent or nonlinear flight control system. Several of the guidelines mentioned in Section 4.1 can be directly retrieved from step response simulation and therefore it is not considered complicated. Next to that, time domain simulation with the full flight control system has the benefit that none of the higher order or nonlinear dynamics are lost, since no linearization or order reduction is required. Hence, it captures the full properties of the system. The downside however, is that a full step response simulation requires additional computational power, when it is desired to analyse the HQ&S during the training of an RL agent at every time step for example.

Recently, a research was performed where the objective was to learn autopilot gains with a Deep Deterministic Policy Gradient (DDPG) algorithm [27]. An input gain k and time delay Δt were placed in the control block diagram according to Figure 4.11 and Figure 4.12 respectively. Numerical time domain simulation was performed to examine at which values the of the gain and time delay the control system would get unstable. Together with the crossover frequency f, which can be obtained from the time response, the values at the point of instability were taken to compute the Gain Margin (GM) and Phase Margin (PM) according to following equations:

$$\text{GM} = 20log(k) \tag{4.10}$$

$$\text{PM} = 360f\Delta t \tag{4.11}$$

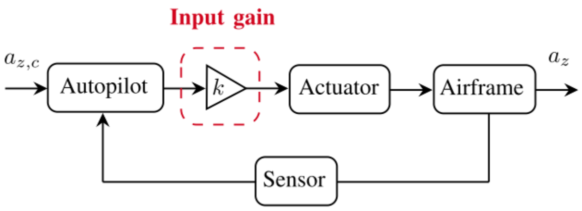


Figure 4.11: Control diagram with an input gain for determining the GM [27].

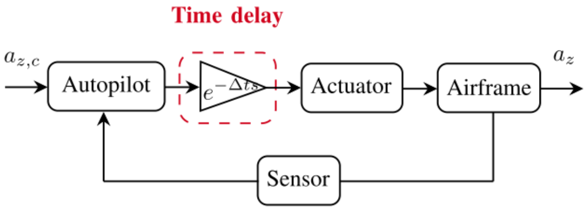


Figure 4.12: Control diagram with a time delay for determining the PM [27].

Although some of the nonlinear dynamics get lost with linearization, it is still a feasible option for examining HQ&S in the time domain. For instance, a research was performed on the Flying-V aircraft where the nonlinear equations of motion obtained from wind tunnel data were linearized. Subsequently, the eigenvalues of the linear system were evaluated and translated into damping ratio's and natural frequencies of the response modes [28]. The model was reduced to second order by isolating the longitudinal states that affect the short period response the most, with the purpose of evaluating the CAP. One of the findings was that the CAP was lower for an aft center of gravity location compared to a more forward location, in line with the theory discussed by Roskam [22]. Another paper by the same author on the Flying-V demonstrated that the HQ&S could be steered towards more desired locations with an Incremental nonlinear Dynamic Inversion (INDI) flight control system. The second order model used for HQ&S evaluation was, nevertheless, not the most accurate representation of the higher order linear model and it was recommend to develop a Low Order Equivalent System [29].

4.2.2. Low Order Equivalent Systems

The development of augmented flight control systems in the 1960's lead to higher order flight control systems due to additional sensor, actuator and control dynamics. It was found that a higher order linear control system for the longitudinal short period motion could be represented reasonably well by a second order model with an equivalent time delay that captures the higher order dynamics [30]. The so-called Low Order Equivalent System (LOES), in the form of Equation 4.2, can be found by matching the gain and phase at a range of frequencies [31]. It is recommended to match at frequencies between 0.1 rad/s and 10 rad/s, because this the range where pilots are most sensitive to feel the dynamics [6]. Equation 4.12 contains the cost function used for matching the gain G and phase ϕ of the Higher Order System (HOS) and the LOES, with a scaling factor κ for the phase (usually 0.02 [6]), at a number of logarithmically spaced frequencies N_ω The scaling factor $\frac{20}{N_\omega}$ is incorporated for the sake of comparison with other studies, as typically 20 frequency points are used [31]. The objective is to minimize cost function J to achieve the most optimal LOES fit.

$$J = \frac{20}{N_\omega} \sum_{k=1}^{N_\omega} \left[(G(\omega_k)_{HOS} - G(\omega_k)_{LOES})^2 + \kappa(\phi(\omega_k)_{HOS} - \phi(\omega_k)_{LOES})^2 \right] \tag{4.12}$$

An alternative, though similar fitting method, was developed where the cost function was weighted with metrics that contemplate to what extent dynamics could be added to the flight control system without the pilot noticing it. These metrics are better known as the Maximum Unnoticable Added Dynamics (MUAD), which put emphasis on the frequencies where the pilot is the most sensitive [32]. Figure 4.13 shows the bounds for both the gain and phase plots in purple, where it can be observed that the frequencies in between 1 and 4 rad/s are the most significant. In order to have a fit that is satisfactory, the error between the HOS and the LOES at the entire frequency range should fall within these bounds. Additionally, verification of the MUAD concept was performed through human-in-the-loop experiments, where the results were consistent with the developed MUAD bounds [33].

One of the primary topics of discussion concerning the LOES fit optimization in literature is the ”galloping time constant”. When the second order model parameter T_{θ_2} is left free during optimization, it might occur that the parameter takes unrealistic values. One approach is to keep the constant at a fixed value, but this might lead to suboptimal fits. Currently, the best method for fixing this issue is by performing a simultaneous load factor fit in addition to the pitch rate transfer function fit. This takes away a degree of freedom and therefore keeps the time constant within realistic bounds [34].

When the aircraft model is (partially) unknown the LOES approach can be used on flight test data to identify and validate the aircraft's full closed loop flight control system. Morelli showed this for the NASA's F-18 research aircraft [35]. Determining the LOES from flight test data requires a slightly different approach, as not the entire bandwith for frequency matching is available. In the research, the Fourier transform was used to convert the flight test data from the time domain to the frequency domain, after which the LOES was fit to the data with output and equation errors [35].

Application to nonlinear flight control systems

The LOES method can not be directly applied to nonlinear flight control systems, as the principle solely holds for linear control theory. A recent study provides a clear overview on how to convert a nonlinear

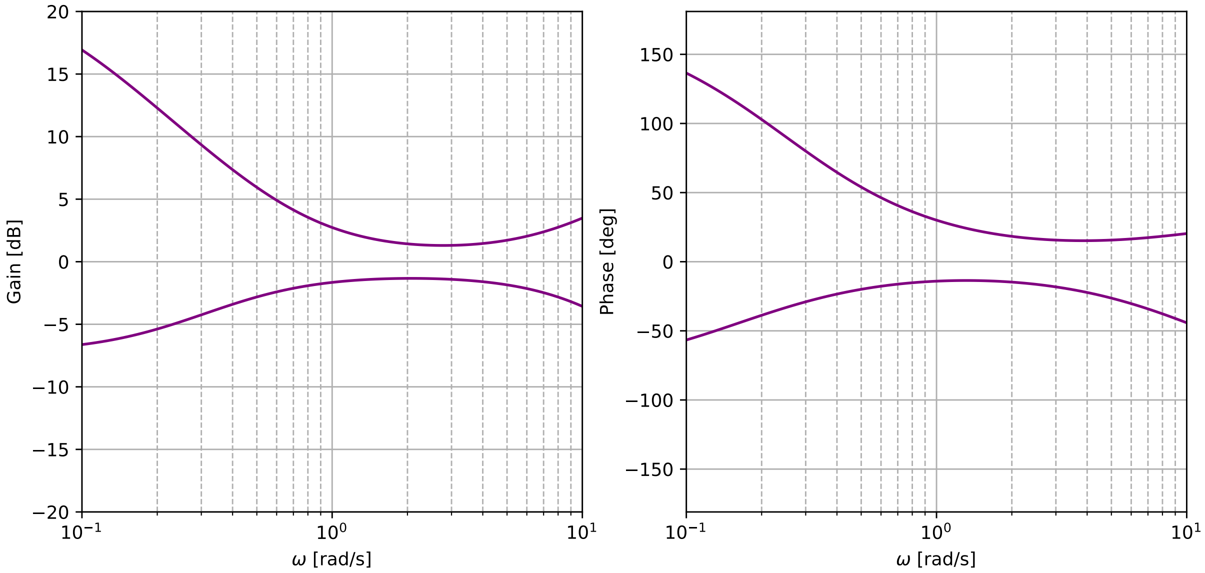


Figure 4.13: MUAD bounds for gain and phase [32].

longitudinal flight control system to a second order short period model [36]. In this research, incremental backstepping is applied to a F/A-18 aircraft model to aid in the design of nonlinear control laws, whilst keeping track of HQ&S. Figure 4.14 shows the general procedure for the reduction of a nonlinear control system, where the complete system is linearized to a HOS and subsequently fitted with a LOES. One of the main findings of the research was how the incremental backstepping parameters affect the CAP [36].

Several other studies have been performed where the HQ&S are evaluated for nonlinear flight control systems. For example, the application of a nonlinear adaptive backstepping controller on a high-fidelity F-16 model, with online aircraft model identification [37]. In this study command filters were used to smoothen the pilot input and steer the aircraft to the desired behaviour. Afterwards, HQ&S were evaluated for verification through the use of simulation data. Time domain data was converted into frequency domain data and then fitted with a LOES. This approach is more similar to the one posed by Morelli, as it uses time domain data as a basis [35]. Furthermore, LOES fits were applied to the lateral modes and dynamics of the aircraft as well [37].

Additionally, Smit et al. investigated the effect of the control effectiveness and center of gravity shifts on HQ&S for an INDI controller. A second order command filter was applied and when it was accurately followed, the control system satisfied the desired HQ&S [38]. In the research, a software package named CONDUIT® was used for the optimization of flight control design with HQ&S compliance. The optimization software for all relevant HQ&S evolves around the same principles as LOES fits [39]. After optimization a decrease in HQ&S variation due to changes in control effectiveness and center of gravity shifts was observed [38].

As a final example, a research was performed wherein a model reference nonlinear dynamic inversion controller was developed and applied to Nasa's F/A-18 testbed aircraft[40]. The study showed a comparison based on the HQ&S for simulation and flight data. A LOES fit with MUAD bounds was applied to the flight data and as it gave smooth fits that represented the dynamics very well, the fits were used for the Neal-Smith criterion instead of raw flight data. Pilot's comments were used to verify the HQ&S analysis.

To conclude, HQ&S evaluation and analysis for nonlinear flight control systems is common and typically done by linearization and model order reduction. However, as far as known, the methodology described in this section applied to RL flight control systems remains absent.

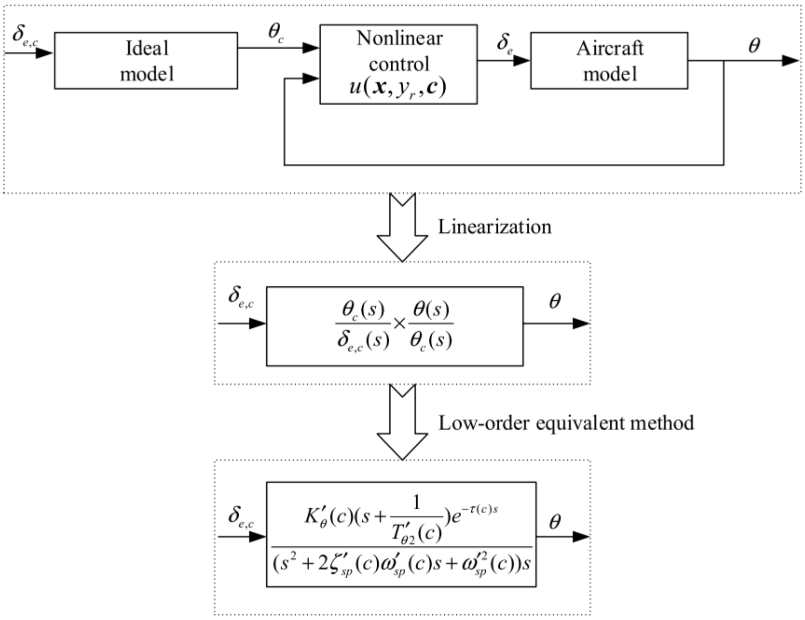

$\delta_{e,c}$
Ideal model
θ_c
Nonlinear control $u(\boldsymbol{x}, y_r, \boldsymbol{c})$
δ_e
Aircraft model
θ
Linearization
$\delta_{e,c}$
$\frac{\theta_c(s)}{\delta_{e,c}(s)} \times \frac{\theta(s)}{\theta_c(s)}$
θ
Low-order equivalent method
$\delta_{e,c}$
$\frac{K'_\theta(c)(s+\frac{1}{T'_{\theta 2}(c)})e^{-\tau(c)s}}{(s^2+2\zeta'_{sp}(c)\omega'_{sp}(c)s+\omega'^2_{sp}(c))s}$
θ

Figure 4.14: Linearization of a nonlinear flight control system and LOES fitting procedure [36].

4.3. Adaptation to HQ&S for nonlinear Flight Control Systems

Now that methods for the evaluation of HQ&S for nonlinear flight control methods have been discussed, the question is whether it is possible to steer the HQ&S in the desired direction. This section will cover two potential approaches for the adaptation of RL flight control systems to the preferred HQ&S. First, the concept of reference model following will be explained, which has been applied to several nonlinear flight control systems and might thus be feasible to implement to an RL agent as well. Secondly, two studies with RL agents that learn from reference models are discussed. When the reference model contains the preferred behaviour, the RL aims to learn the similar properties. Note that this is different from reference model following, as now it directly learns and tries to mimic the behaviour of the reference model instead of following the commands.

4.3.1. Reference Model Following

The basic principle of reference model following in flight control is that when a reference model is placed between the pilot and the control system and he is able to follow the reference model commands closely, the full flight control system feels like flying the reference model. An example of reference model following is posed by Rysdyk et al [41]. In the research, a tiltrotor aircraft with an adaptive nonlinear dynamic inversion controller is commanded to follow a reference model. It shows that when the aircraft is able to follow commands up to a certain frequency bandwidth, it has L1 handling qualities [41].

Another field of study, that is often interconnected with reference model following, is the design of Variable Stability (VS) flight control systems, where the dynamics of the aircraft, or simulator, can be adapted for a variable flying experience. Two studies performed on the Cessna Citation II with VS flight control systems use methodologies that might be used for HQ&S adaptation in RL flight control. The first study incorporates a VS system through response feedback, where the gains are tuned manually for the desired HQ&S [42]. The research shows that it is not entirely possible to exactly place the HQ&S on the desired location, but compliance with Level 1 pilot work load ratings is possible. The second study uses a form of reference model following [43]. The parameters of the second order reference model are tuned such that it has the desired CAP and damping ratio of the short period motion. An INDI controller is then used to follow the commands of the reference model closely. The basic principle is similar to the control block diagram shown in Figure 4.14.

This methodology might be useful for steering the HQ&S properties of an RL agent towards the desired goal. Further research on RL flight control with a reference model used as a command filter will be performed by means of preliminary experiments in Chapter 5.

4.3.2. Reference Model RL

Instead of using the reference model as a command filter, it can be used as a model where a RL agent learns from. A recent study shows how a SAC agent runs in parallel with a baseline controller . The latter is used for basic control and ensuring stability, whilst the SAC agent learns from a nominal model which contains the desired performance. A schematic of the control system is given in Figure 4.15. The results of the study show that the control system with a nominal model is superior over the control system without the model (only SAC) in terms of stability and tracking performance [44].

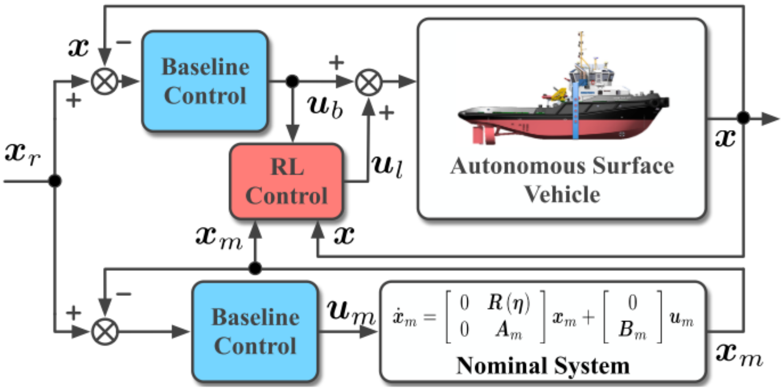

x
$-$
$+$
Baseline Control
$+$
u_b
$+$
x_r
RL Control
u_l
Autonomous Surface Vehicle
x
x_m
x
$+$
$-$
Baseline Control
u_m
$\dot{x}_m = \begin{bmatrix} 0 & R(\eta) \\ 0 & A_m \end{bmatrix} x_m + \begin{bmatrix} 0 \\ B_m \end{bmatrix} u_m$
Nominal System
x_m

Figure 4.15: Control diagram for reference model RL applied to an autonomous surface vehicle [44].

A similar study assesses how Model Reference Adapative Control (MRAC) can be use to provide stability for DRL algorithms. An overview of the control system is illustrated in Figure 4.16, where DRL is used to cope with the uncertainties of the plant. A reference model is implemented in parallel and the goal of the controller is to follow the desired response posed by the reference model. The controller itself consists of a linear feedback and feedforward term, together with an adaptive term resulting from the uncertainty approximation by the DRL agent. Simulation are performed with a 6 degree-of-freedom quadrotor and results show that the combined system shows better performance the traditional MRAC, as DRL has more generalization power [45].

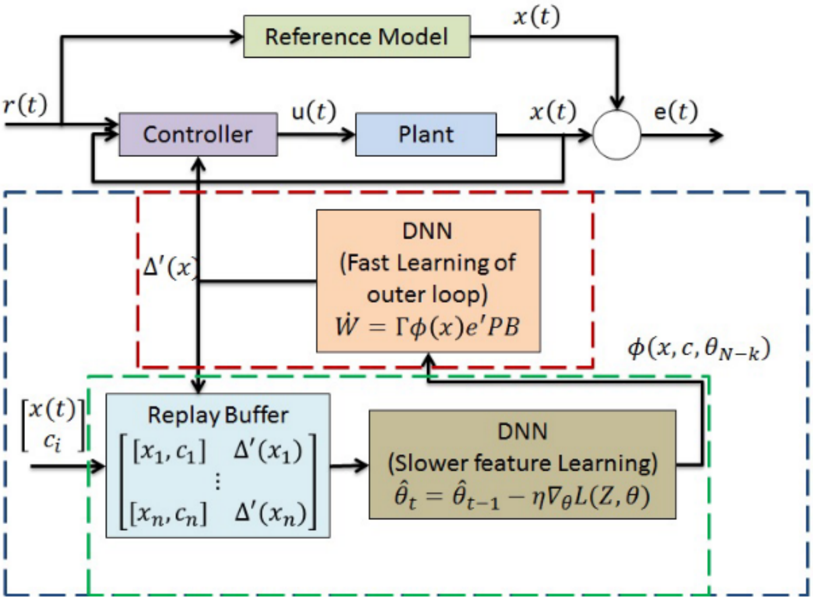

Reference Model
$x(t)$
$r(t)$
Controller
$u(t)$
Plant
$x(t)$
$e(t)$
$\Delta'(x)$
DNN
(Fast Learning of outer loop)
$\dot{W} = \Gamma\phi(x)e'PB$
$\phi(x, c, \theta_{N-k})$
$\begin{bmatrix} x(t) \\ c_i \end{bmatrix}$
Replay Buffer
$\begin{bmatrix} [x_1, c_1] & \Delta'(x_1) \\ \vdots & \\ [x_n, c_n] & \Delta'(x_n) \end{bmatrix}$
DNN
(Slower feature Learning)
$\hat{\theta}_t = \hat{\theta}_{t-1} - \eta\nabla_\theta L(Z, \theta)$

Figure 4.16: Control diagram of DRL MRAC [45].

The general concept of using a reference model parallel to the RL agent to prescribe the desired behaviour might be suitable for the adaptation to HQ&S. A more elaborate analysis of this principle will be performed in Chapter 5.

4.4. Synopsis

This chapter provided an overview of relevant HQ&S for flight control, specifically applied to the longitudinal motion of the aircraft. In particular, quantitative HQ&S were addressed, as piloted experiments with RL agents and thus qualitative assessment is not feasible (yet). The three different levels of HQ&S applied the various flight phases and aircraft categories were explained, all together providing the groundwork for the analysis of HQ&S for RL flight control. According to the military standards, the CAP and short period parameters have priority in the design of longitudinal flight control systems and will be therefore considered first in further analysis, whereas the additional HQ&S outlined in this chapter could be assessed and integrated in a later stage of this research, when time allows [6]. Thereby, **RQ 2.1** is answered.

Besides that, it was shown that the HQ&S can be extracted from nonlinear flight control systems by either time domain or frequency domain analysis. The former requires step response simulations for determining the CAP, which increases the computational load. Furthermore, a method has been shown where the GM and PM could be determined by numerically increasing the gain and delay until instability was observed in the time response. For frequency domain analysis it was found that for nonlinear flight control systems linearization is required, after which order reduction should be applied. The LOES fit approach was found to be a widely applied method for reducing a HOS to an equivalent second order model from which HQ&S could be assessed directly. The overview of time and frequency domain methods for evaluating HQ&S provide a starting point for their application to RL flight control, which gives an answer to **RQ 2.2**.

Finally, two potential methods for the adaptation of RL flight control systems to the desired HQ&S were presented. One of the approaches is to use a second order reference model as a command filter with pre-selected HQ&S. In this framework, the goal of the RL is to follow the commands of the reference model as closely as possible, such that the full control system have similar HQ&S as the reference model. The alternative option is to use a reference model in parallel, inspired partially on MRAC. In this case, the RL agent aims to mimic the reference model's behaviour, with the goal of reaching comparable HQ&S. These approaches already partially answer **RQ 2.3** and **RQ 3.3**, but in order to investigate whether these methods are feasible for shaping the RL agent to the preferred behaviour, further analysis is provided in Chapter 5.

5 Preliminary Analysis

This chapter will provide an overview of the preliminary experiments that were performed to get an initial insight in the feasibility of evaluating HQ&S and in RL flight control and steering them towards the desired values. The code is based on the code for a SAC controller developed by Dally et al. and will be referred to throughout this chapter [4]. It should be noted that the preliminary experiments are a way of showing the proof-of-concept and therefore only longitudinal dynamics and the most important HQ&S, according to the military standards [6], are considered. First, the environment in which the SAC agent acts, the Cessna Citation II, is briefly explained in Section 5.1. The SAC agent that is used will be outlined in Section 5.2, where the hyperparameters of the agent are presented. The two ways of implementing the reference model for HQ&S adaptation, as explained in the previous chapter, will be visualized in Section 5.3. Subsequently, the tracking tasks are provided in Section 5.4. The results are presented in Section 5.5 and the preliminary analysis will be concluded in Section 5.6.

5.1. Environment

The original research of the SAC controller was performed with a nonlinear model of the Cessna Citation II [46, 4]. For the scope of the preliminary analysis, the same aircraft, but some simplifications are applied. The envionment that will be used throughout this chapter is a linearized reduced longitudinal short period model of the Cessna Citation II, as developed in [18]. The model is represented by a state space system:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u} \tag{5.1}$$

Or, in discretized form:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + (A\mathbf{x}_t + B\mathbf{u}_t)\Delta t \tag{5.2}$$

The state vector **x** consists of the angle of attack α and pitch rate q:

$$\mathbf{x} = \begin{bmatrix} \alpha \\ q \end{bmatrix} \tag{5.3}$$

The only control input **u** is the elevator deflection δ_e:

$$\mathbf{u} = \begin{bmatrix} \delta_e \end{bmatrix} \tag{5.4}$$

The system matrix A and input matrix B are defined by Equation 5.5 and Equation 5.6 respectively. The values of the geometric properties, stability derivatives and the velocity V are presented in Table 5.1.

$$A = \begin{bmatrix} \frac{V}{\bar{c}} \frac{C_{Z_\alpha}}{2\mu_c - C_{Z_{\dot{\alpha}}}} & \frac{2\mu_c + C_{Z_q}}{2\mu c - C_{Z_{\dot{\alpha}}}} \\ \frac{V^2}{\bar{c}^2} \frac{C_{m_\alpha} + C_{Z_\alpha} \frac{C_{m_{\dot{\alpha}}}}{2\mu_c - C_{Z_{\dot{\alpha}}}}}{2\mu_c K_Y^2} & \frac{V}{\bar{c}} \frac{C_{m_q} + C_{m_{\dot{\alpha}}} \frac{2\mu_c + C_{Z_q}}{2\mu_c - C_{Z_{\dot{\alpha}}}}}{2\mu_c K_Y^2} \end{bmatrix} \tag{5.5}$$

$$B = \begin{bmatrix} \frac{V}{\bar{c}} \frac{C_{Z_{\delta_e}}}{2\mu_c - C_{Z_{\dot{\alpha}}}} \\ \frac{V^2}{\bar{c}^2} \frac{C_{m_{\delta_e}} + C_{Z_{\delta_e}} \frac{C_{m_{\dot{\alpha}}}}{2\mu_c - C_{Z_{\dot{\alpha}}}}}{2\mu_c K_Y^2} \end{bmatrix} \tag{5.6}$$

Table 5.1: Simplified longitudinal short period model parameters of the Cessna Citation II [18].

V	=	59.9 m/s	C_{Z_α}	=	-5.16	C_{m_α}	=	-0.43
$\bar{c}$	=	2.022 m	$C_{Z_{\dot{\alpha}}}$	=	-1.43	$C_{m_{\dot{\alpha}}}$	=	-3.7
μ_c	=	102.7	C_{Z_q}	=	-3.86	C_{m_q}	=	-7.04
K_Y^2	=	0.98	$C_{Z_{\delta_e}}$	=	-0.6238	$C_{m_{\delta_e}}$	=	-1.553

5.2. Agent

The SAC agent used in this analysis is directly taken from [4]. The schematic of the actor-critic structure is shown in Figure 5.1. Furthermore, the hyperparamters are presented in Table 5.2. These parameters were not adapted, as they are already tuned. Besides that, it is not the goal of the project to optimize hyperparameters.

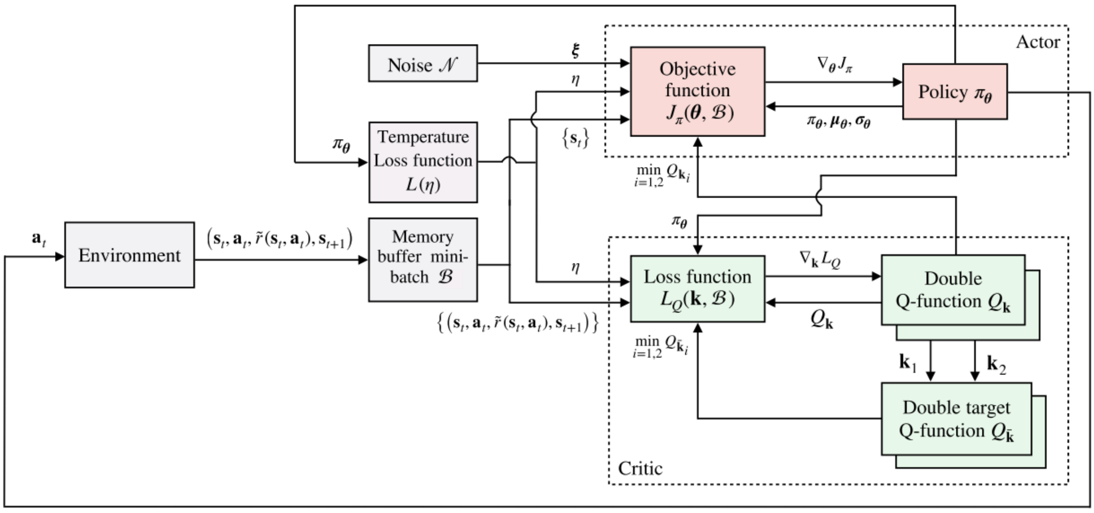

Actor
Noise $\mathcal{N}$
ξ
η
Objective function $J_\pi(\theta, \mathcal{B})$
$\nabla_\theta J_\pi$
Policy π_θ
$\pi_\theta, \mu_\theta, \sigma_\theta$
π_θ
Temperature Loss function $L(\eta)$
$\{s_t\}$
$\min_{i=1,2} Q_{k_i}$
π_θ
a_t
Environment
$(s_t, a_t, \tilde{r}(s_t, a_t), s_{t+1})$
Memory buffer mini-batch $\mathcal{B}$
η
Loss function $L_Q(k, \mathcal{B})$
$\nabla_k L_Q$
Double Q-function Q_k
Q_k
$\{(s_t, a_t, \tilde{r}(s_t, a_t), s_{t+1})\}$
$\min_{i=1,2} Q_{\bar{k}_i}$
k_1
k_2
Double target Q-function $Q_{\bar{k}}$
Critic

Figure 5.1: Detailed schematic of the SAC framework, adapted from [4].

Table 5.2: Hyperparameters of the SAC agent, adapted from [4]

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	η	9.4e-4
Replay buffer batch size	$\|\mathcal{B}\|$	256
Replay buffer maximum size	$\|\mathcal{D}\|$	50000
Initial entropy coefficient	η_0	1.0
Number of episodes	N_e	100

5.3. Reference Model Placement

Now that the agent and environment are developed, the methodology for HQ&S evaluation and adaptation, as explained in Chapter 4, can be applied. It was found that a reference model is necessary for steering the HQ&S towards the desired values. The reference model used for the preliminary analysis is formulated as follows:

$$\frac{q_{r,m}(s)}{q_r(s)} = \frac{K_\theta\left(s + 1/T_{\theta_2}\right)}{s^2 + 2\zeta_{sp}\omega_{sp} + \omega_{sp}^2} \tag{5.7}$$

Note that in this case, the pilot is directly controlling reference pitch rate q_r and the output of the reference model is $q_{r,m}$. The parameters of the reference model are presented in Table 5.3. These values were selected such that the model has a damping ratio of 0.707 and CAP of 1.0, which are in the center of the L1 region for the CAP criterion Figure 4.3.

Table 5.3: Parameters of the second order reference model.

Parameter	Value	Unit
K_θ	6.1	-
$T_{\theta 2}$	0.5	s
ζ_{sp}	0.707	-
ω_{sp}	3.49	rad/s

This section will present two approaches of placing the reference model in the SAC flight control system: reference model following and reference model reward modification.

5.3.1. Reference Model Following

With Reference Model Following (RMF), the reference model is used as a command filter and placed in front of the SAC controller Figure 5.2. Here, the goal of the SAC controller is to track the reference model pitch rate signal $q_{r,m}$ as closesly as possible in order to approach the desired CAP. The observation that the agent uses consists of the reference model pitch rate tracking error $(q - q_{r,m})$ and the reward signal is the negative squared pitch rate tracking error.

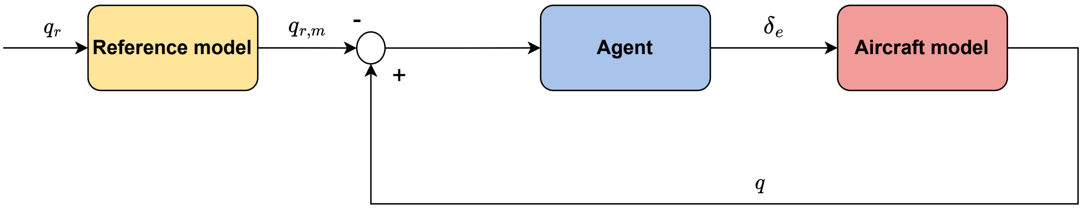


Figure 5.2: Block diagram of the SAC controller for RMF.

5.3.2. Reference Model Reward Modification

The second approach is to use the reference model as a way of modifying the reward signal. A schematic overview of this method, for now named as Reference Model Reward Modification (RMR), is illustrated in Figure 5.3. In this case, the pilot command q_r is directly send to the agent and the reference model pitch rate $q_{r,m}$ is used to give the agent feedback of its action by incorporating it in the reward signal. The agent therefore tries to mimic the reference model behaviour, without receiving reference model commands (only feedback through the reward afterwards). It turned out this is not a straightforward process and in order to make it work additional information should be provided to the agent.

The observation of the agent consists of the reference pitch rate tracking error $(q - q_r)$, the pitch rate q, the pitch acceleration $\dot{q}$ and the elevator deflection δ_e. The reason that the pitch acceleration is added is that the agent needs information on how fast the pitch rate is changing and results in more smooth step responses. Furthermore, the agent does not directly control the elevator deflection δ_e, but it uses an incremental approach where the agent controls the elevator deflection derivative $\dot{delta}_e$. Hence, the elevator deflection is added to the observation of the agent, such that it knows the current deflection. A rate limit of 100 deg/s was set on the incremental, to prevent aggressive behaviour of the agent.

The reward of the agent is defined by Equation 5.8. Note that instead of taking the square, the absolute values are taken. The reason for this is that it showed better results compared to the squared error. Furthermore, the pitch acceleration error is added to the reward signal, to aid the agent in approaching the reference model HQ&S. A scaling factor of $\frac{1}{10}$ is multiplied with the pitch acceleration error, as this value led to the best results.

One final addition to the RMR approach was made to reach a more smooth behaviour, namely condition for action policy smoothness [47]. This methods adds two loss terms related to temporal smoothness, meaning that the action at time $t+1$ should be similar to the action at time t, and a spatial smoothness term ensuring that actions should be the same for similar states.

All these adaptation were made to the RMR SAC controller, because without it, the controller would only receive the reference pitch rate q_r. When that is the case it would try to track the reference signal as closely as possible, whereas this is not desired. For instance, when a step input is commanded, the HQ&S guidelines prescribe a smooth response with a short rise time and small overshoot. To accurately follow this behaviour, derivative terms are needed. This will be further demonstrated in Section 5.5.

$$r = -|q - q_{r,m}| - \frac{|\dot{q} - \dot{q}_{r,m}|}{10} \tag{5.8}$$

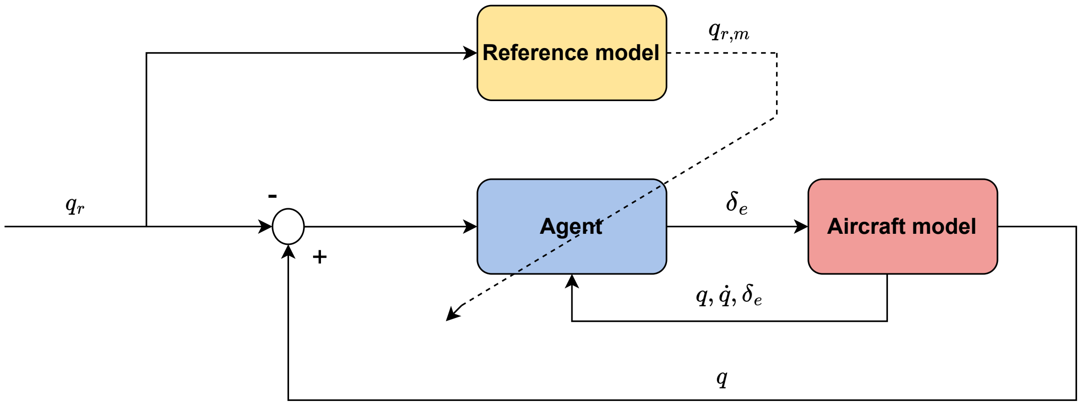


Figure 5.3: Block diagram of the SAC controller for RMR.

5.4. Task

The training process for the RMF approach is done without the command filter, as it is not part of the closed loop. A signal consisting of several superimposed sinusoids was therefore developed as a reference model pitch rate tracking task q_r, shown in Figure 5.4. For the RMR approach, on the other hand, the reference model is part of the closed loop system as it it is used in the reward signal. Hence, it is necessary to use step inputs as tracking tasks, as the step response is crucial for the correct adaptation to the HQ&S guidelines. A sequence of step inputs with random magnitudes between -4 and 4 degrees was generated and a realization of such a training task is illustrated in Figure 5.5.

For the evaluation of the two SAC controller, a similar task is considered to be useful for the sake of comparison. A 3-2-1-1 step input sequences with magnitudes of -3 and 3 degrees for the reference pitch rate is used, to assess the controller's performance. An illustration of the evaluation task is given in Figure 5.6. The 3-2-1-1 signal is chosen to investigate whether the duration of the step response has a significant influence on the behaviour of the controllers or not.

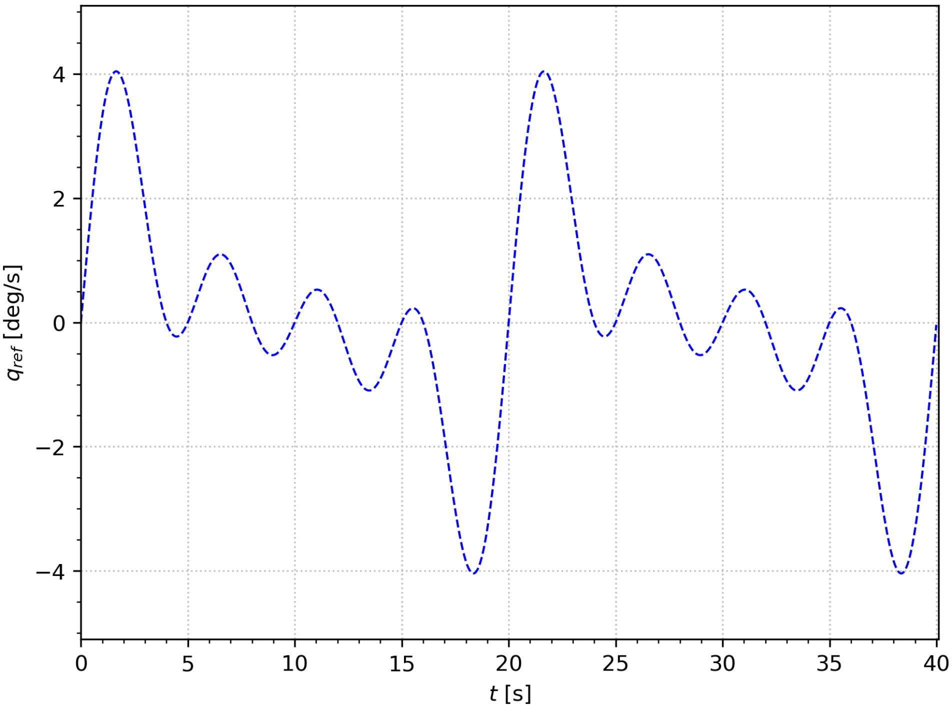


Figure 5.4: Training task for the SAC controller with RMF.

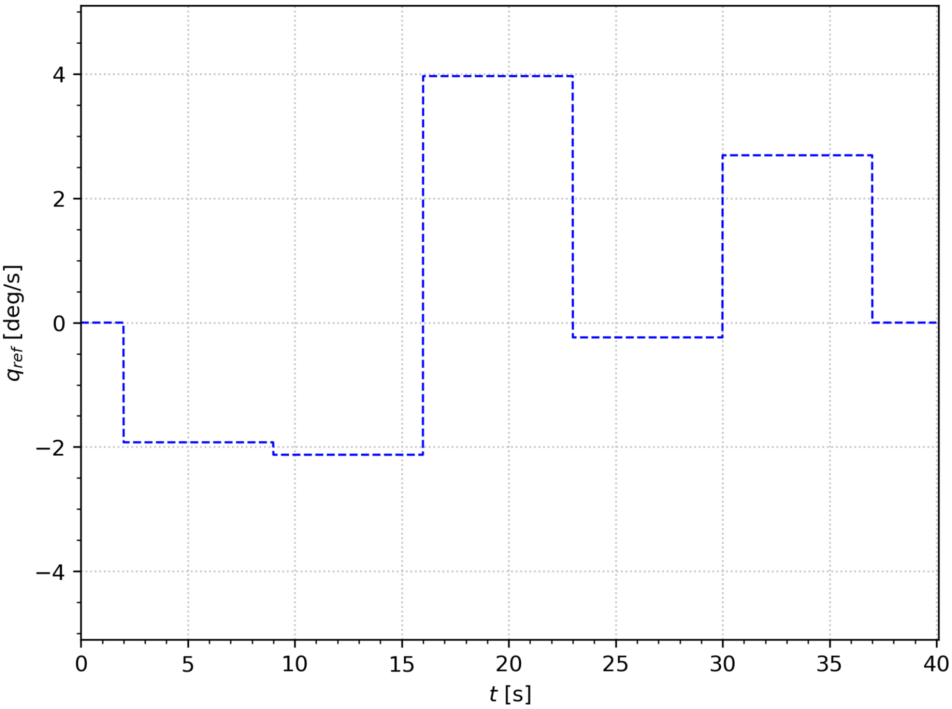


Figure 5.5: Training task for the SAC controller with RMR.

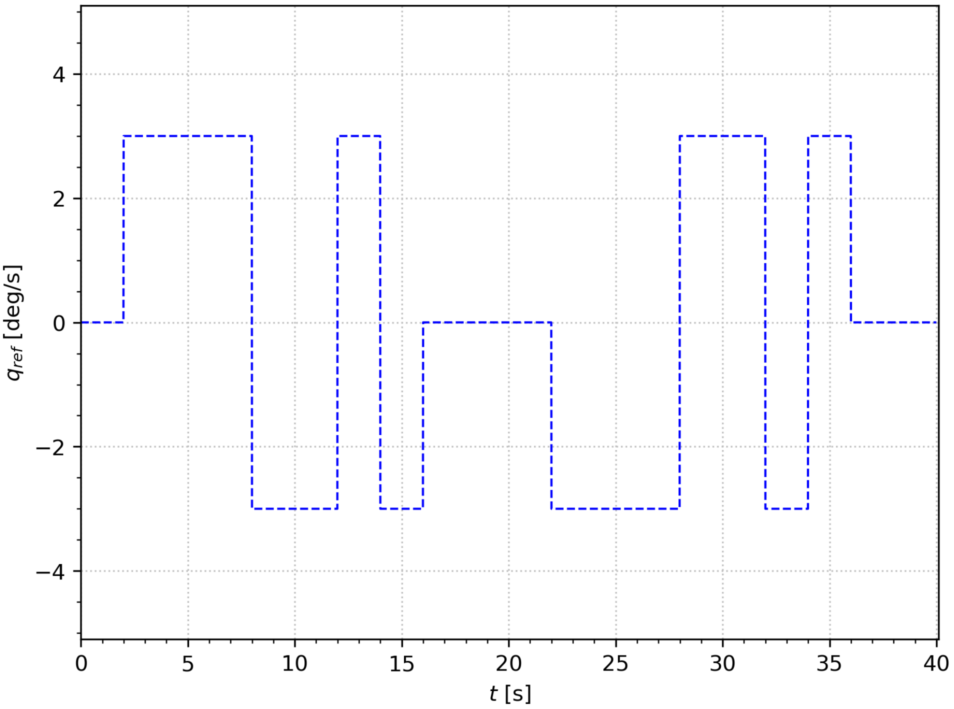


Figure 5.6: Evaluation task for the SAC agent for both reference model following and reward modification.

5.5. Results

This section will present the results of the developed SAC controller with the RMF and RMR approach. First, the training and evaluation results are shown, after which the HQ&S evaluation and adaptation is highlighted. Furthermore, a sensitivity analysis on the effect of measurement noise for both controller will be presented.

5.5.1. Training

The training processes of both controllers have a duration of 100 episodes. The training curves with the total return for each episode is visualized in Figure 5.7. It should be noted that the results of both controllers can not be compared accurately, because the reward signals are different. Since the reward signal of the RMF is squared, it results in smaller values of the return, but does not mean that it has better performance. The RMF approach shows convergence and the RMR seems to converge too, but has more variation in its total return value.

The final episode of training is shown in Figure 5.8 and Figure 5.9 for RMF and RMR respectively. It can be observed that the elevator deflection of both controllers shows oscillatory behaviour for exploration, but the magnitude of the oscillations is significantly higher for RMF. This could be due to the fact that the RMR controller contains various measures for smoothening the policy.

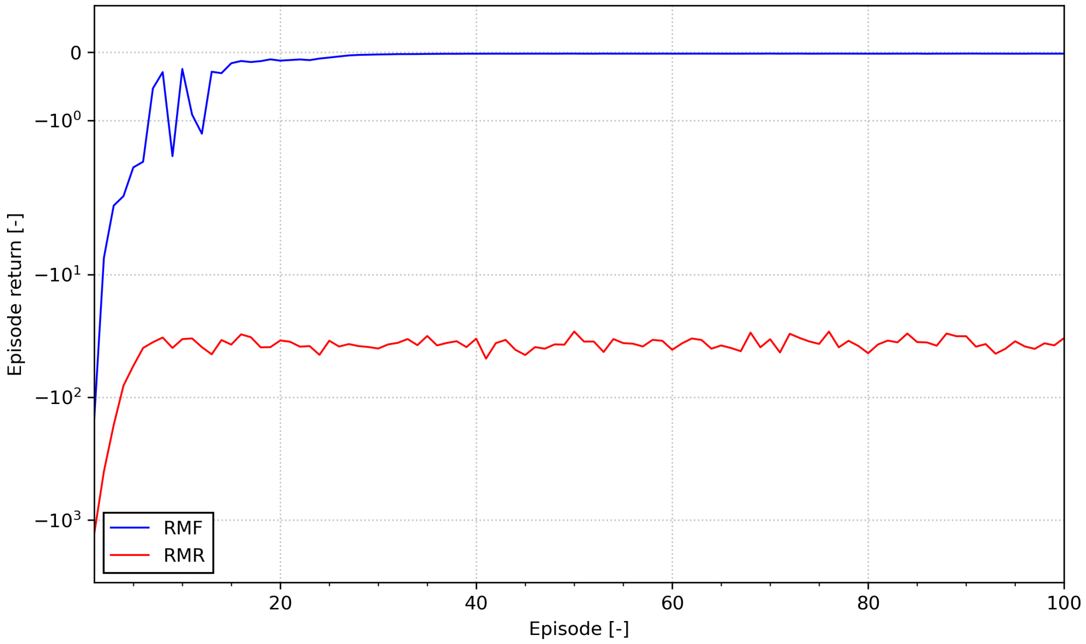


Figure 5.7: Training curves with average return for the SAC controllers with RMF and RMR.

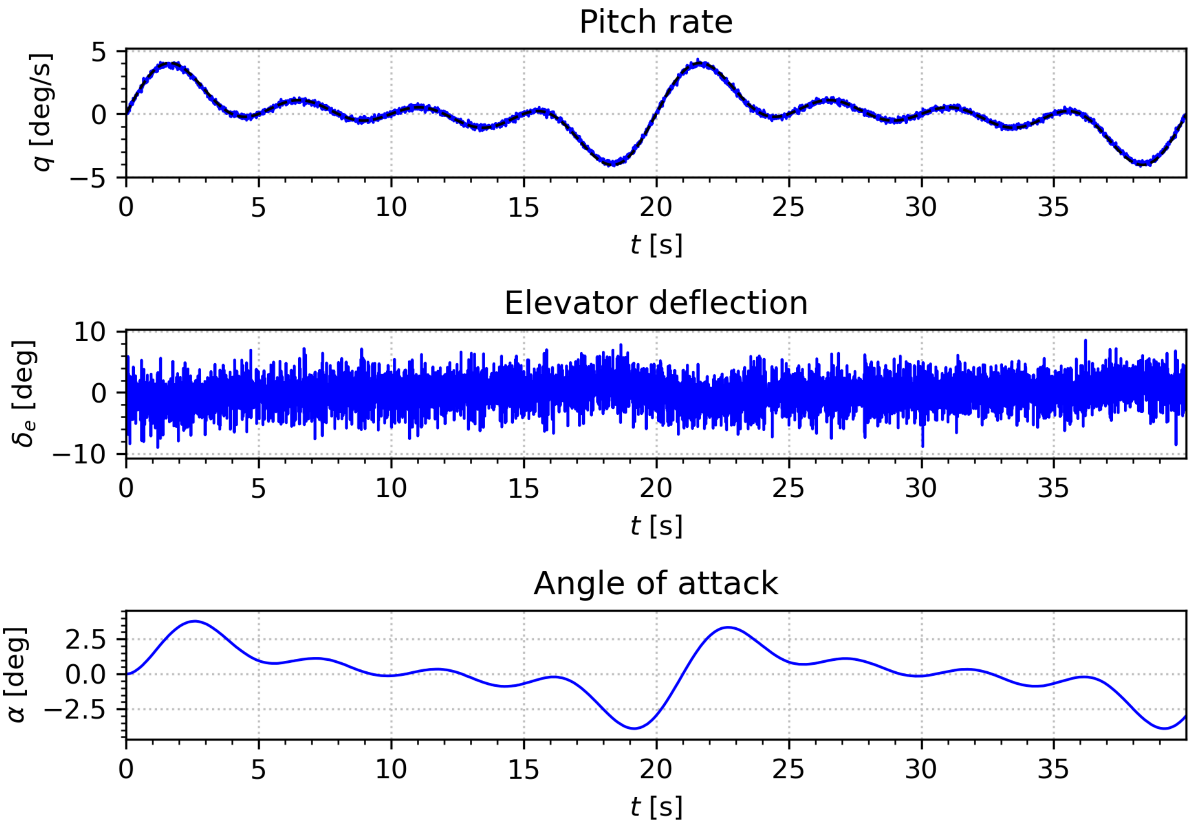


Figure 5.8: Training of the SAC controller during the final episode for RMF.

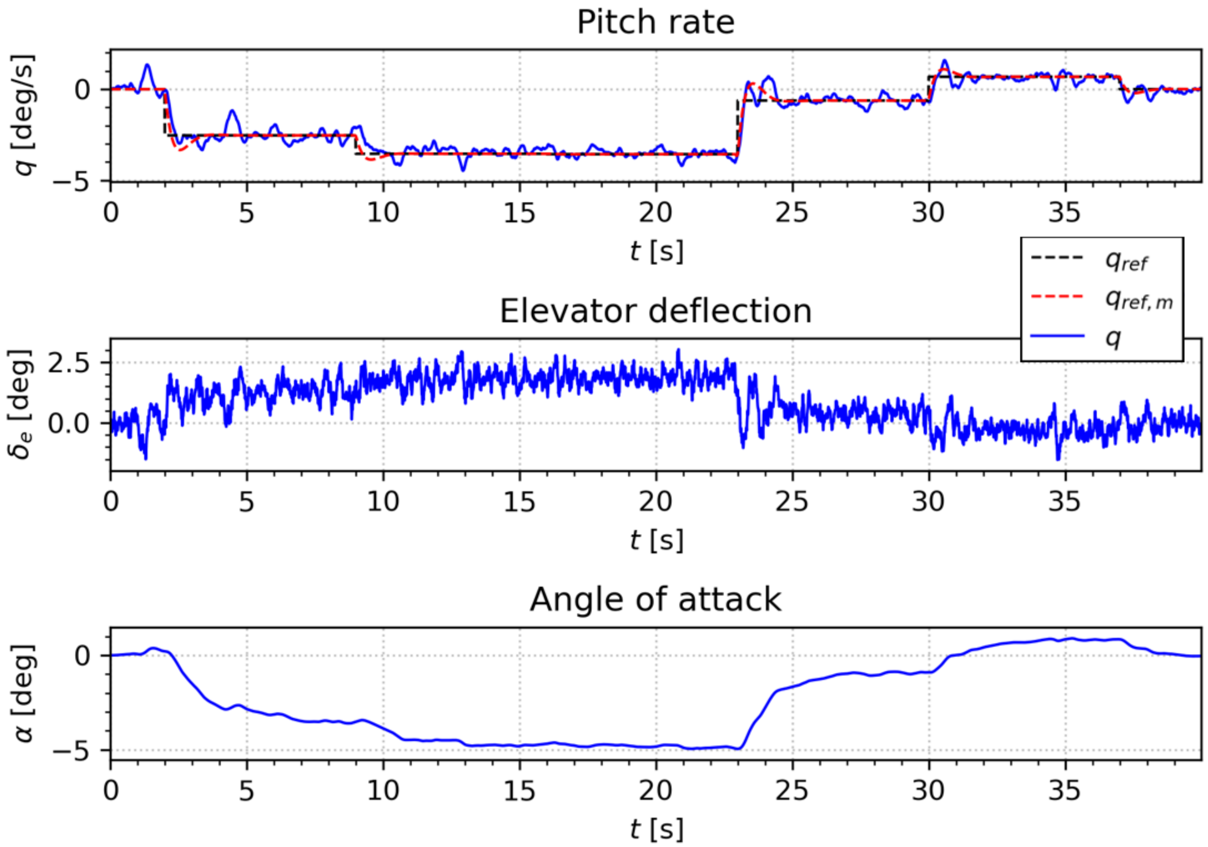


Figure 5.9: Training of the SAC controller during the final episode for RMR.

5.5.2. Evaluation

The evaluation of the fully trained SAC controllers for RMF and RMR is shown Figure 5.10 and Figure 5.11 respectively. It can be observed that the RMF SAC controller is able to track the reference model pitch rate $q_{r,m}$ very well. This could possibly be explained by the fact that the pilot reference pitch rate q_r is command filtered to a signal that is more easy to be tracked $q_{r,m}$ as it does not include the sudden changes of the step input, but merely the desired response.

When looking at the evaluation of the task for RMR controller it can be concluded that it is worse in approximating the reference model pitch rate $q_{r,m}$ than the RMF approach. This is obviously due to the fact that the RMR controller does not directly receive the reference model pitch rate, but only knows the pitch rate applied by the pilot q_r and the other observations as mentioned in Section 5.3. It is observable, however, that the agent is able to adapt to the general shape of the reference model behaviour, which is one of the main goals. Further analysis of the HQ&S will be provided in the next section.

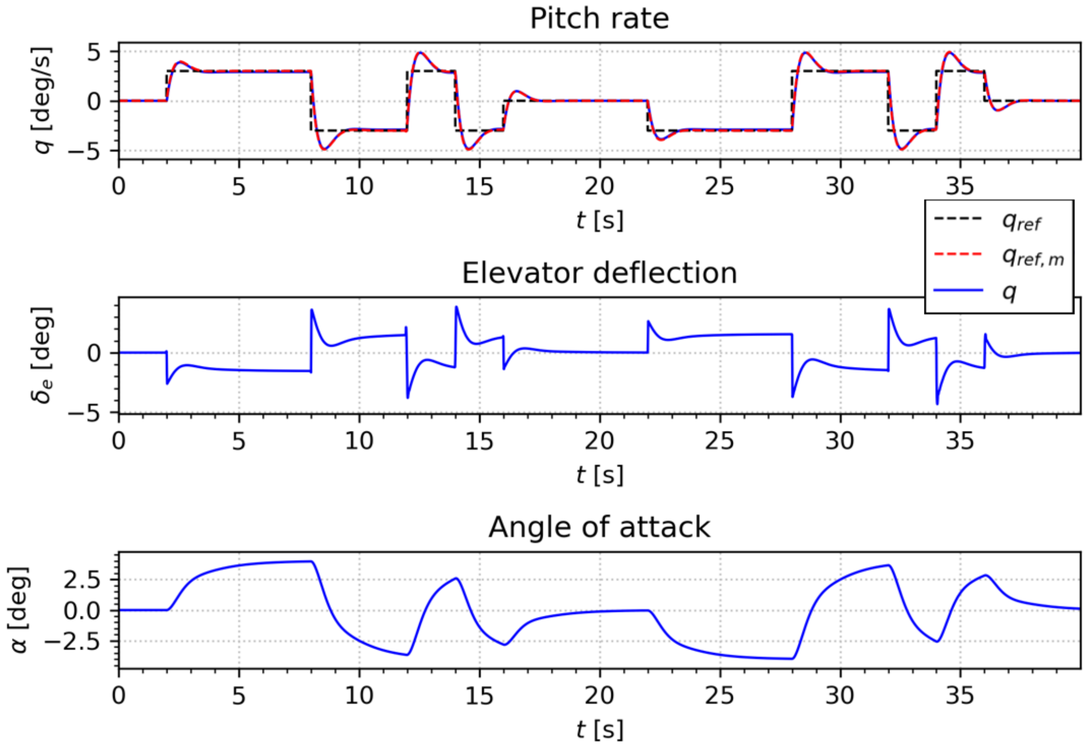


Figure 5.10: Evaluation of the SAC controller after the final episode for RMF.

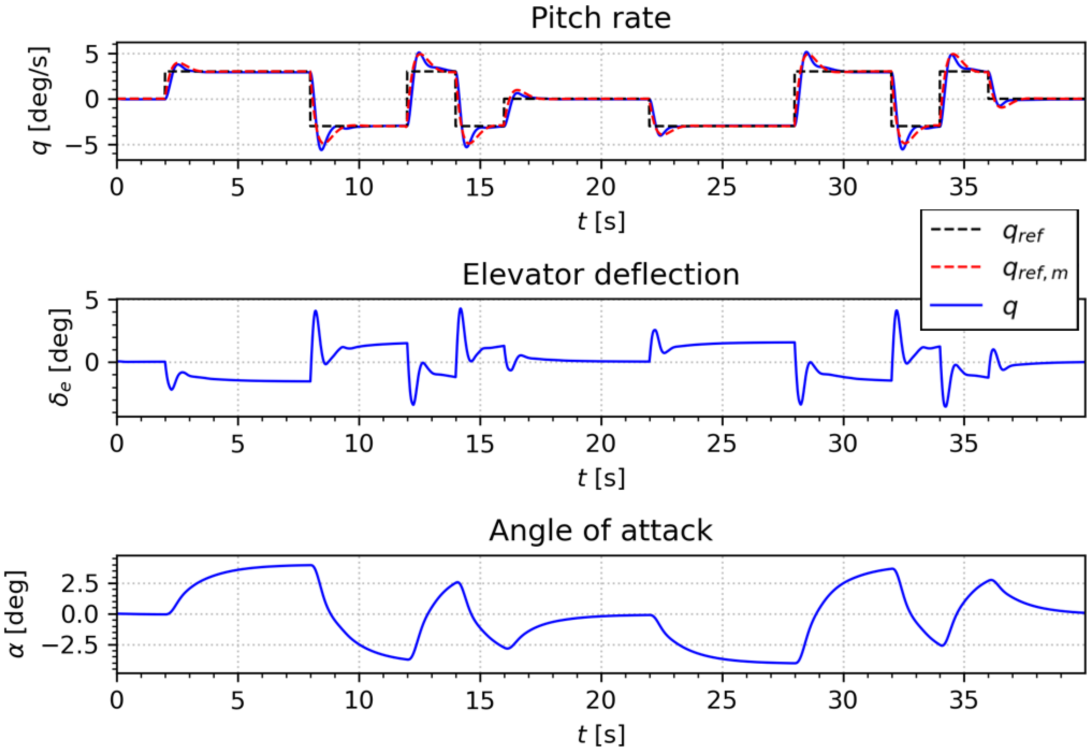


Figure 5.11: Evaluation of the SAC controller after the final episode for RMR.

5.5.3. HQ&S Analysis

For the analysis of the HQ&S, the procedure developed in Chapter 4 was used. Linearization of the flight control system and a subsequent Low Order Equivalent System (LOES) fit, was applied after every episode for both controllers. Figure 5.12 shows the LOES fitting cost of the RMF and RMR approach during training. It can be concluded that the LOES fits are better for the RMF SAC control system, which can be also observed from the LOES fits at after training shown in Figure 5.13 and Figure 5.14 respectively. The RMF controller has an almost perfect fit, while the fit of the RMR controller is slightly off. It should be noted that the fit of the RMR controller remains within the bounds of the Maximum Unnoticable Added Dynamics (MUAD) and is therefore considered still to be an accurate enough fit.

The mismatch of the RMR controller is also visible when zooming in on the individual LOES model parameters shown in Figure 5.15 to Figure 5.19, where the reference model values are indicated with the black dotted line. The RMF controller approaches the reference model parameters very closely, already suggesting that the CAP might be accurate as well. The reason for the mismatch of the RMR controller could be the linearization procedure. Linearization is performed at the trimming point of the aircraft by using small perturbations. The smoothing factor and derivative terms for RMR, however, could potentially manipulate the linearization as might result in delay effects. This is an issue that should be further analyzed in the next stage of the research.

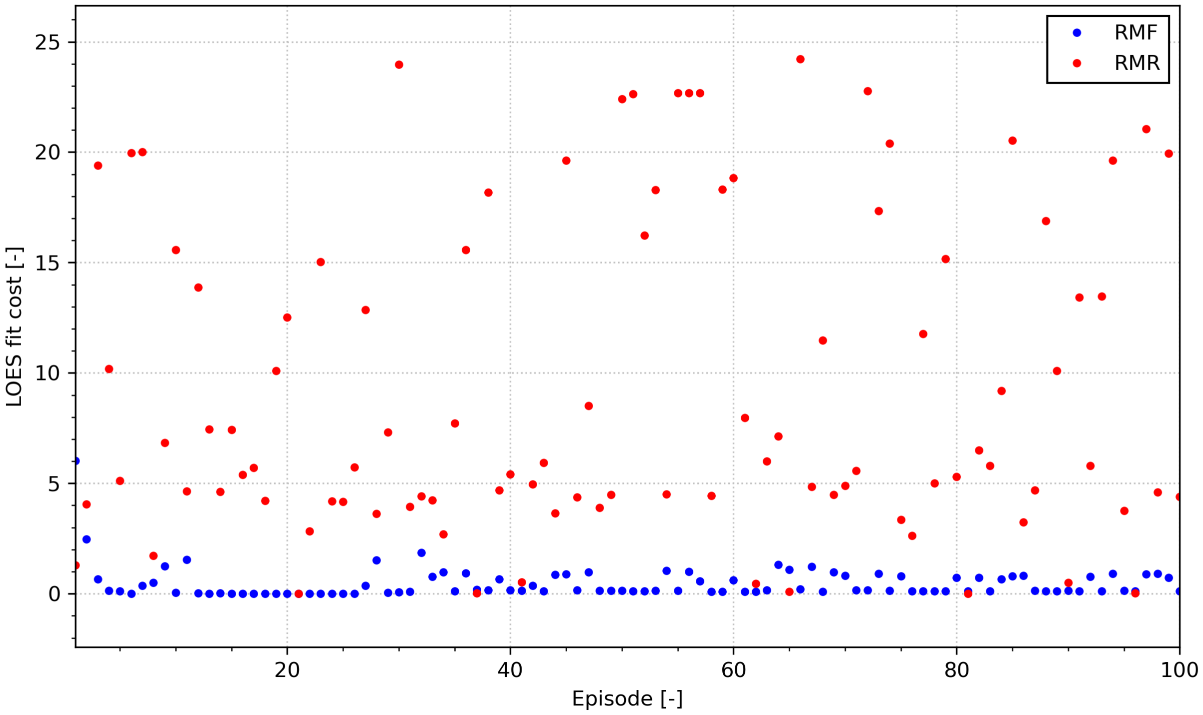


Figure 5.12: LOES fiting cost during training for the SAC controllers with RMF and RMR.

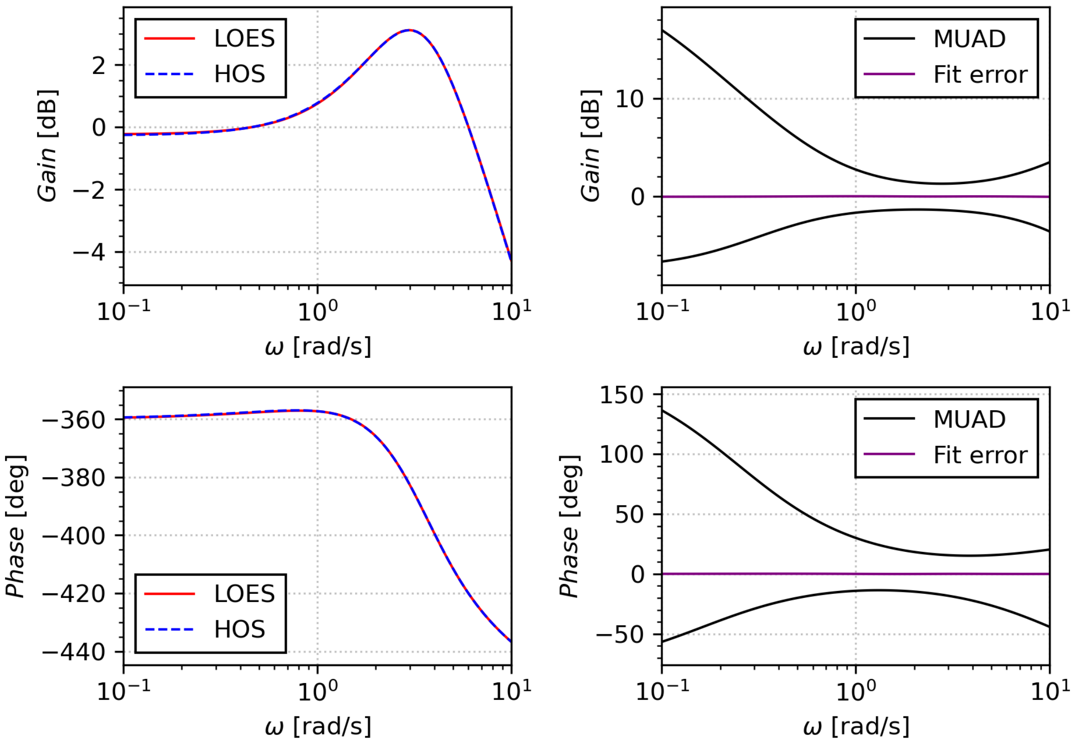


Figure 5.13: LOES fits and MUAD bounds for the SAC controller with RMF after the final episode.

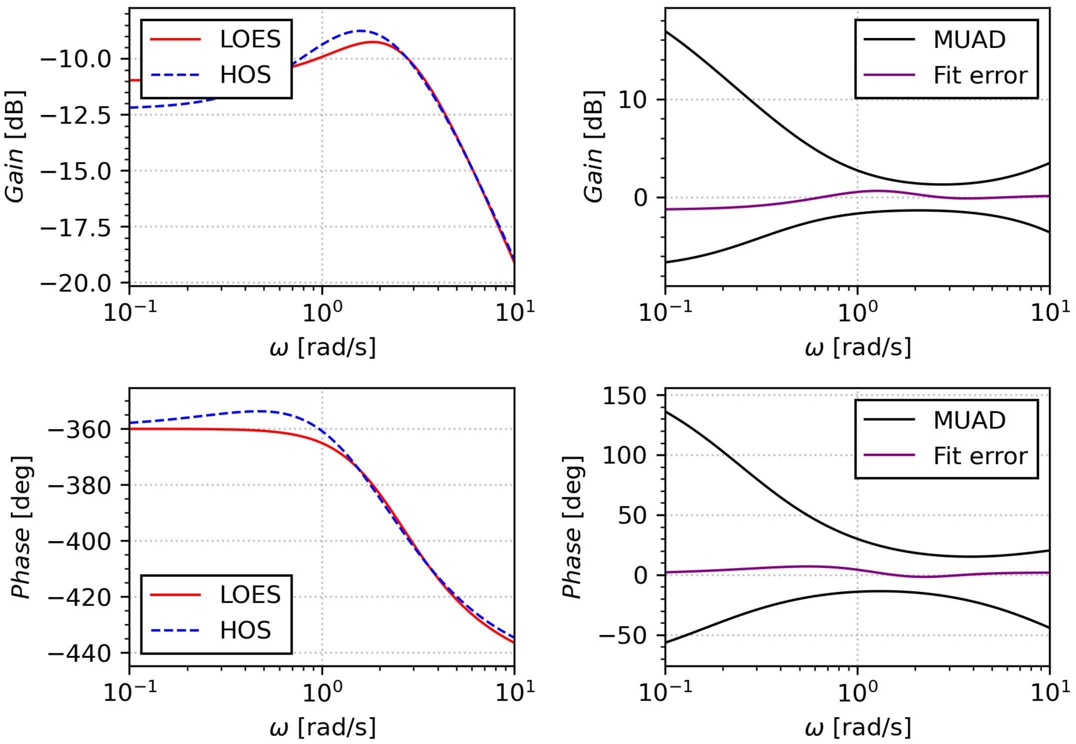


Figure 5.14: LOES fits and MUAD bounds for the SAC controller with RMR after the final episode.

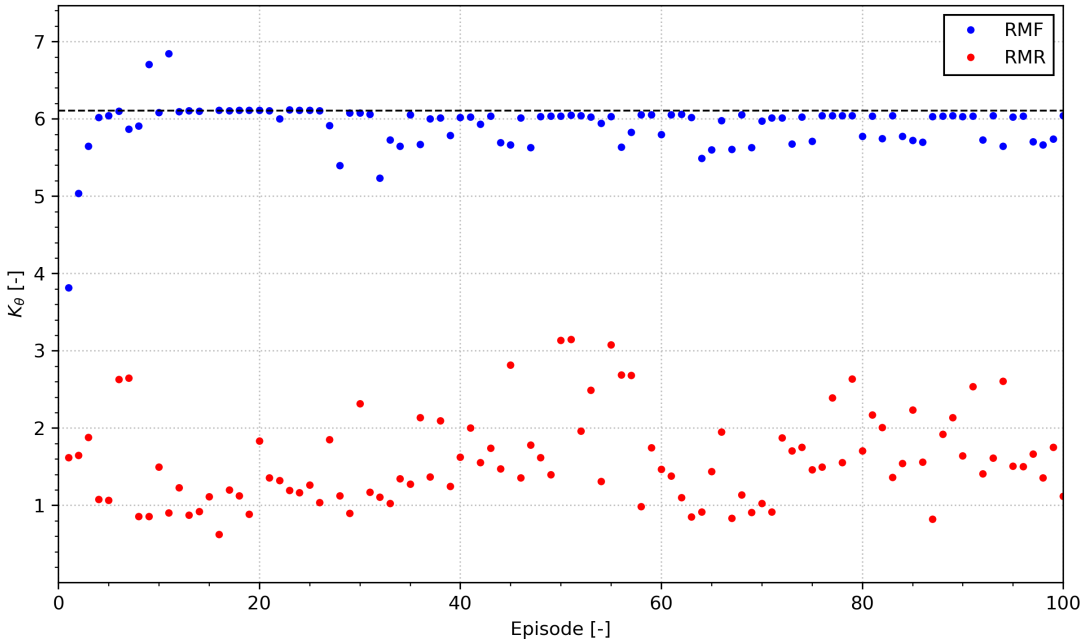


Figure 5.15: LOES gain comparison of the SAC controllers with RMF and RMR.

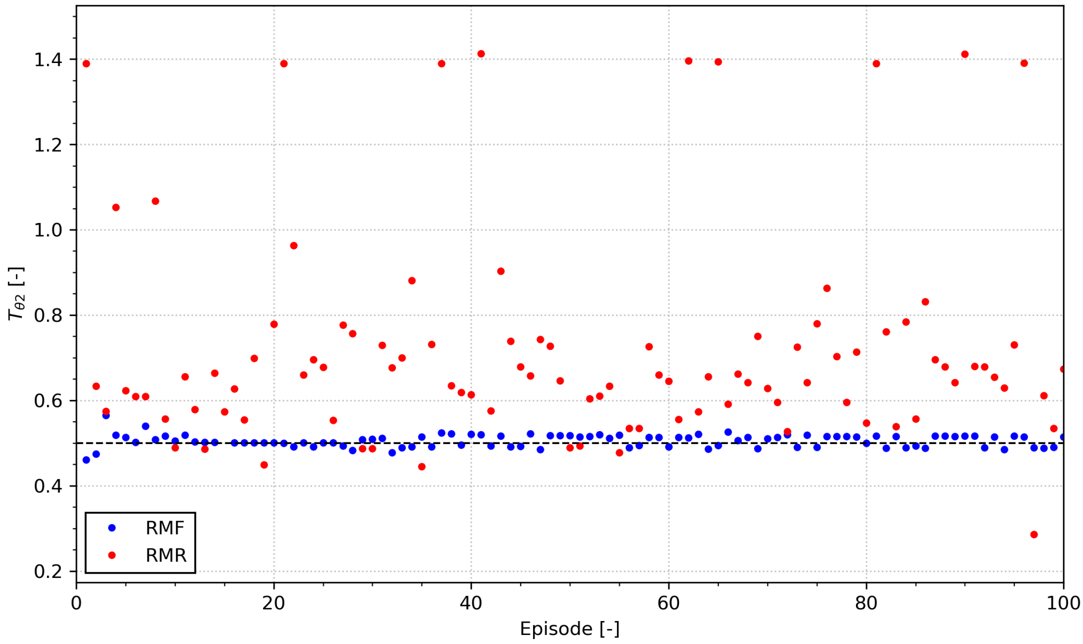


Figure 5.16: LOES time constant comparison of the SAC controllers with RMF and RMR.

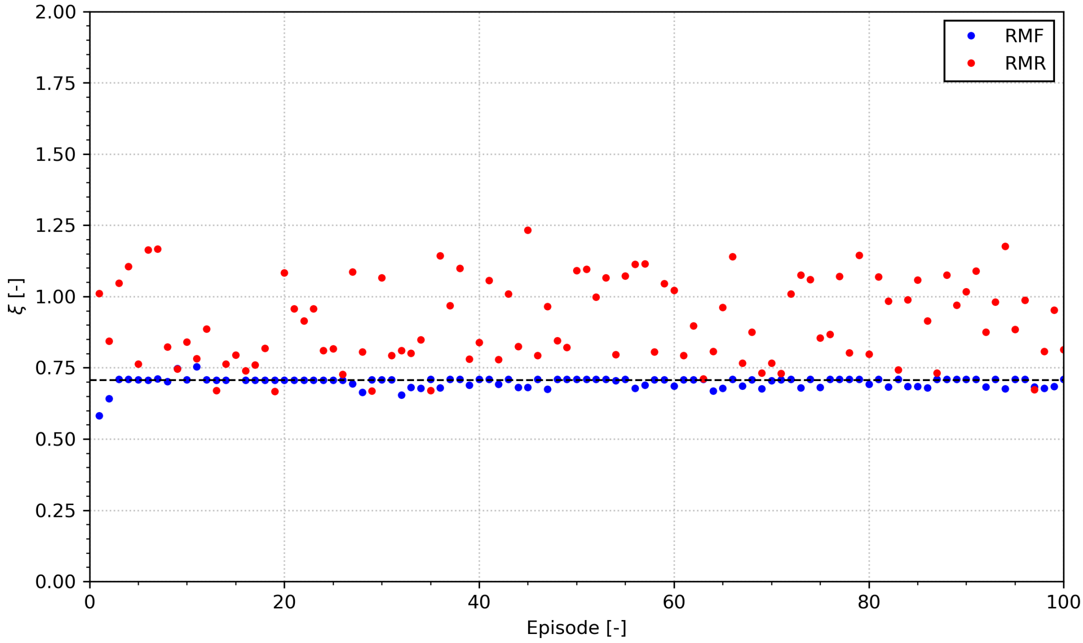


Figure 5.17: LOES damping ratio comparison of the SAC controllers with RMF and RMR.

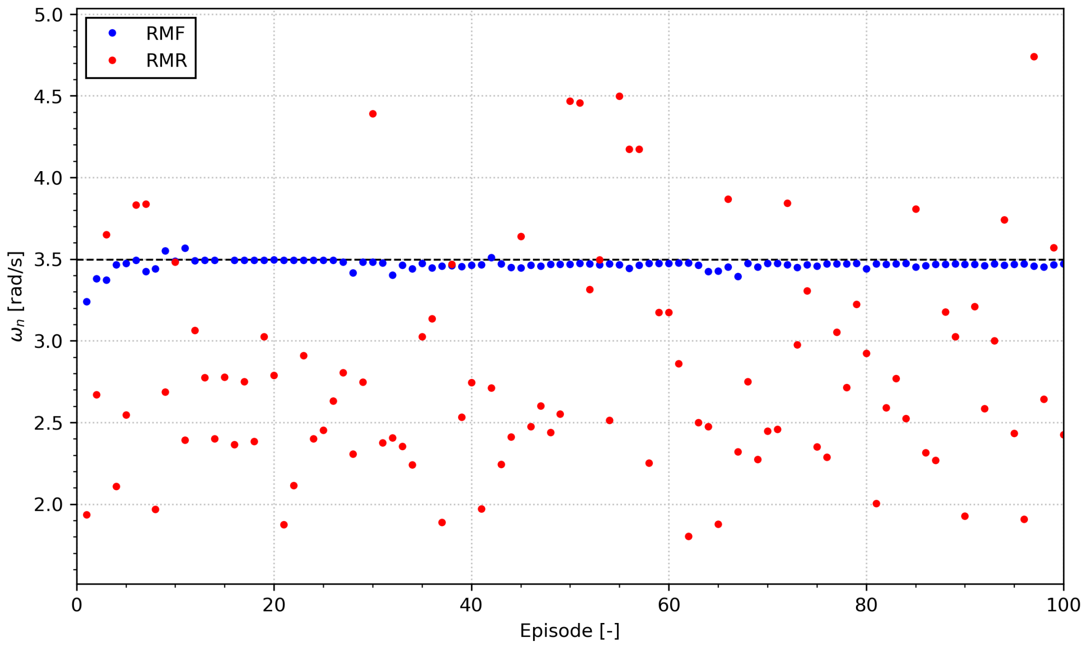


Figure 5.18: LOES natural frequency comparison of the SAC controllers with RMF and RMR.

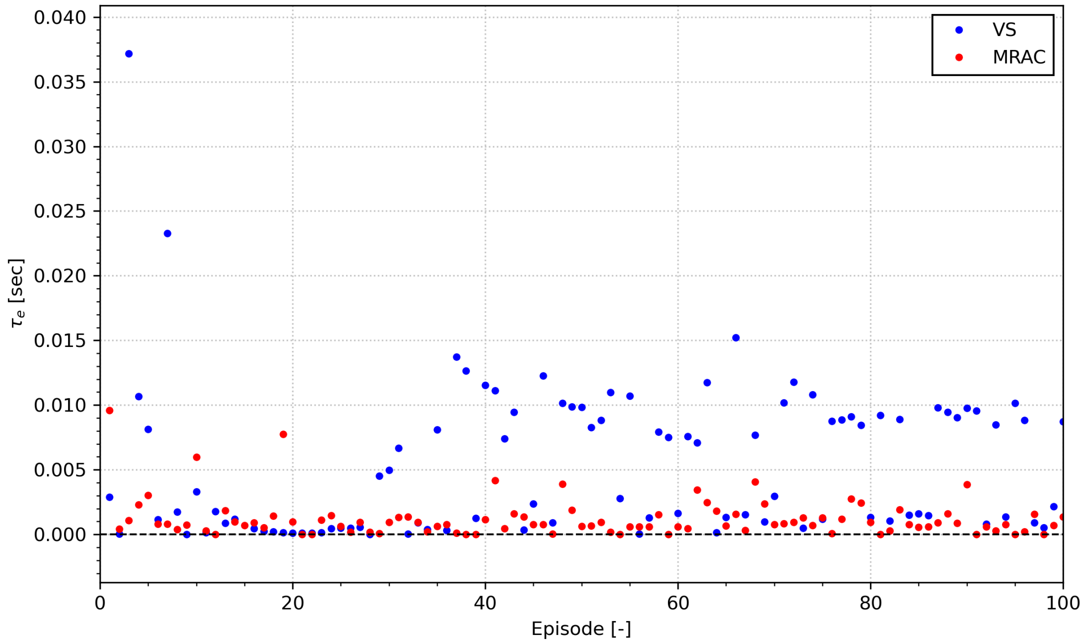


Figure 5.19: LOES time delay comparison of the SAC controllers with RMF and RMR.

Further anlaysis was performed by zooming in on a single step response. As explained in Chapter 4, the CAP can be determined from a step response in the time domain, by using the maximum pitch accelration and steady state pitch rate. These can be taken directly from the step responses shown in Figure 5.20 and Figure 5.21 for RMF and RMR respectively. Again, from the step responses it can be concluded that the RMF approach reaches better tracking performance.

The time domain analysis of the CAP was performed at each episode for both controllers. The CAP obtained from the frequency response using the LOES parameters is also determined after each episode. The results and are shown in Figure 5.22 and Figure 5.23 respectively. One of the main conclusions that can be drawn from these figures is that both controllers show a L1 rating for the CAP at the end of training. Furthermore, the time domain CAP of the RMR approach seems to be more stable over time. This could be due to the fact that the RMF is very sensitive to minor offsets, for example due to trimming. Hence, a sensitivity study of measurement noise was performed, to further assess this phenomenon.

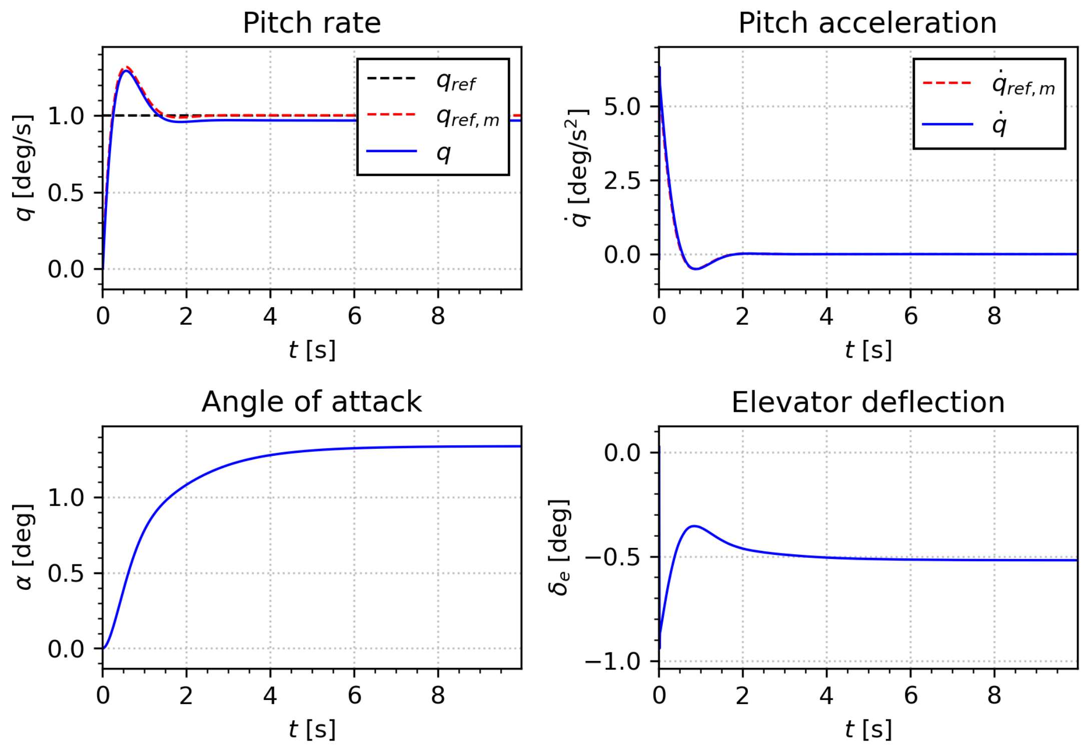


Figure 5.20: Step response of the SAC controller with RMF.

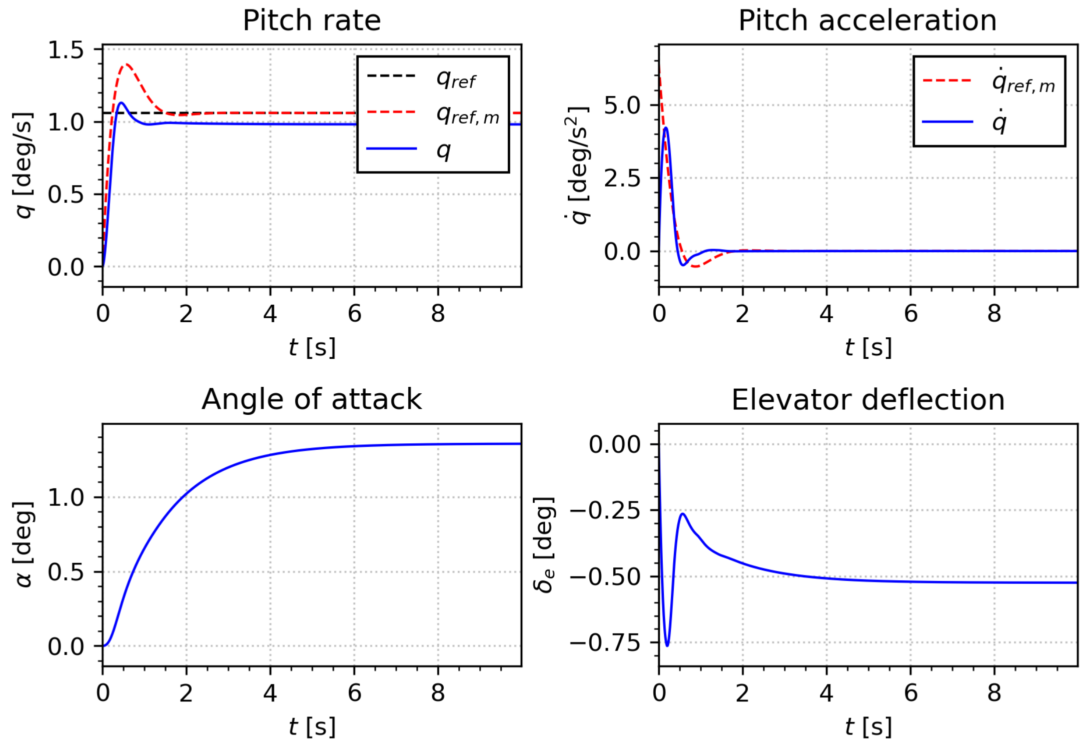


Figure 5.21: Step response of the SAC controller with RMR.

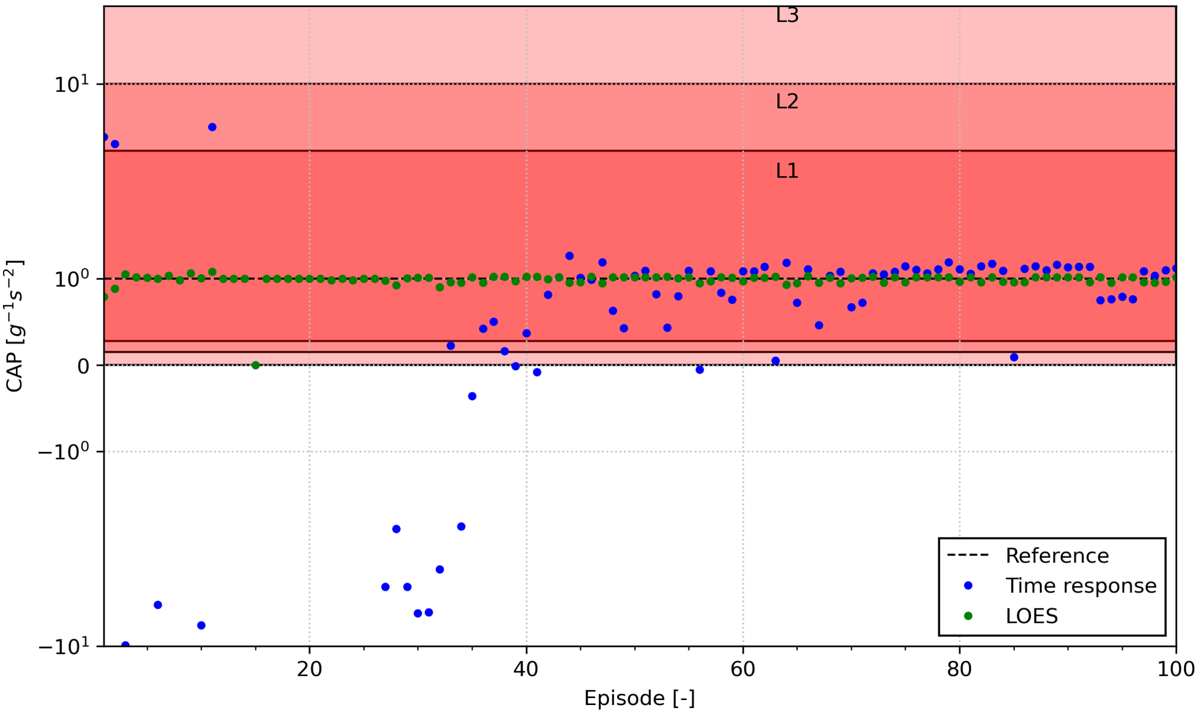


Figure 5.22: CAP during training, obtained from the LOES and time domain response for the SAC controller with RMF.

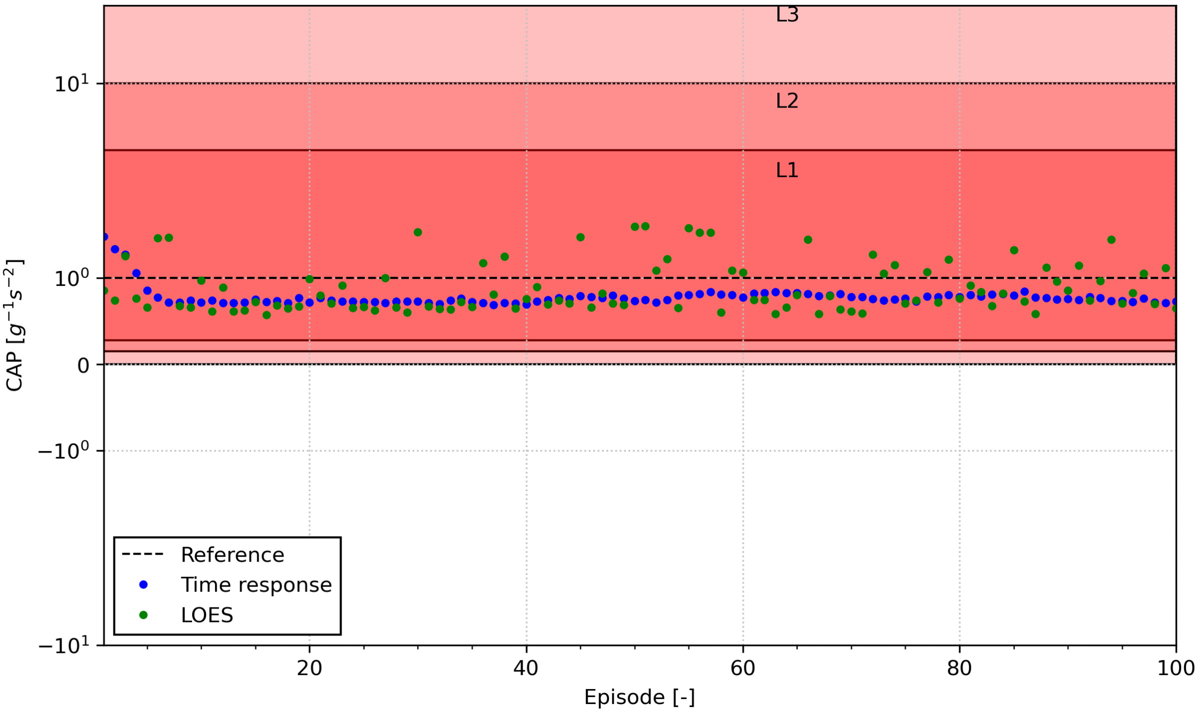


Figure 5.23: CAP during training, obtained from the LOES and time domain response for the SAC controller with RMR.

5.5.4. Measurement Noise Sensitivity

To assess the sensitivity of both controller to measurement noise, a zero-mean gaussian noise signal was added to the pitch rate measurements. A standard deviation of 0.01 deg/s and 0.1 deg/s was applied and the results in the form of step response are shown in Figure 5.24 to Figure 5.27. From close inspection of the step responses, it can be seen that the RMF approach is much more sensitive to noise than the RMR approach. The most probable explanation is that the RMF approximates a very high gain controller, while the RMR does not. The RMR controller learns to mimic the behaviour of the reference model and is therefore less sensitive too small offsets, while the RMF controller is tracking the reference model and corrects small offsets with high deflections. This is an interesting finding and will be further anlysed in the next phase of the research.

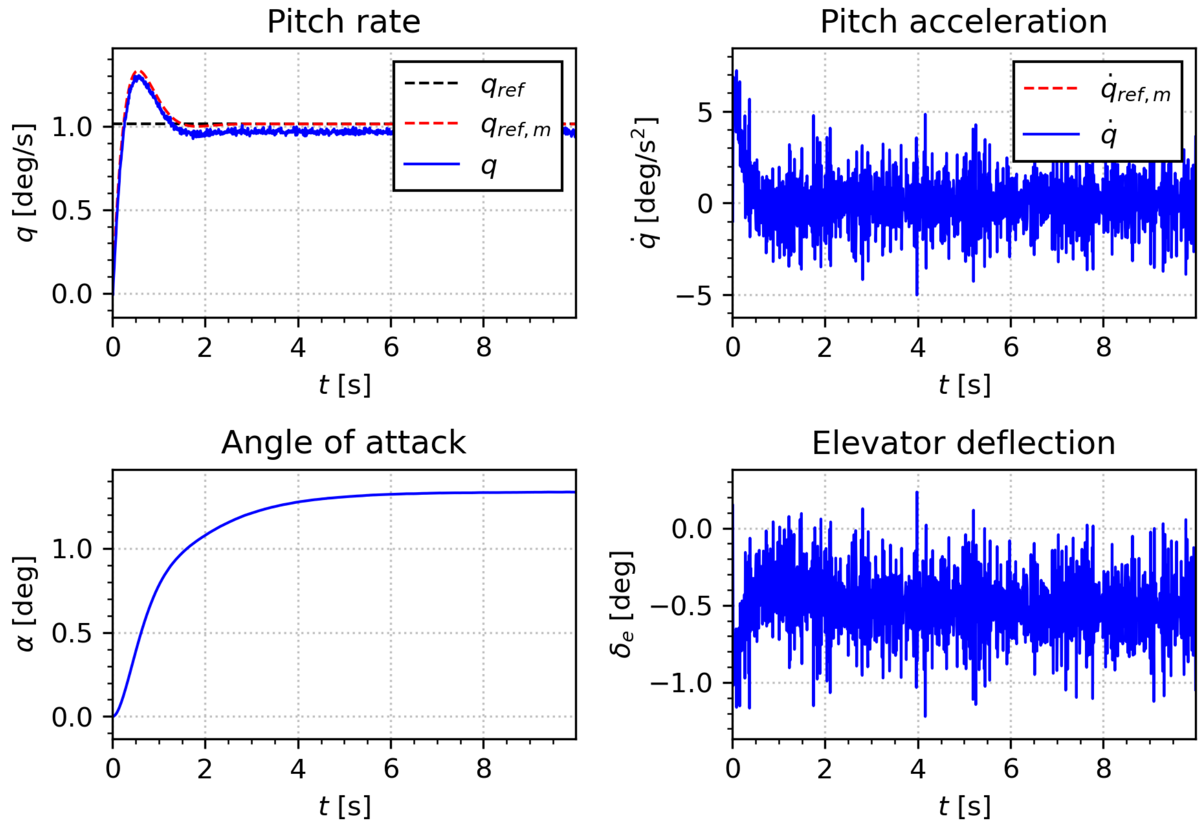


Figure 5.24: Step response of the SAC controller with RMF with a measurement noise standard deviation on the pitch rate of 0.01 deg/s.

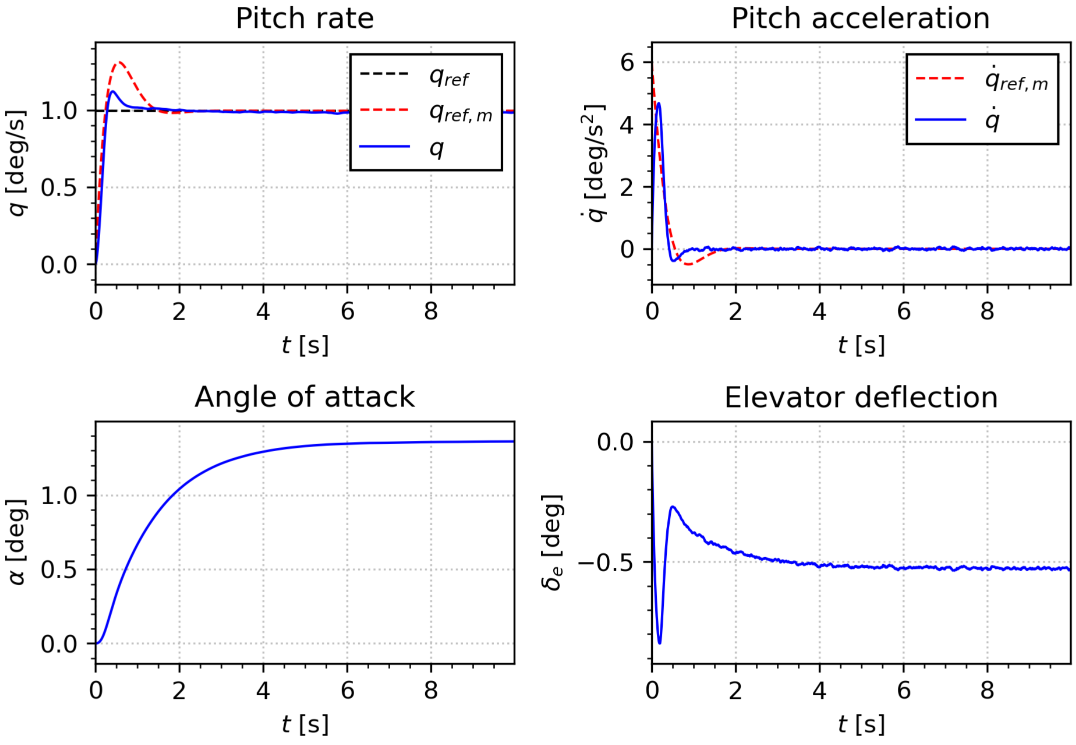


Figure 5.25: Step response of the SAC controller with RMR with a measurement noise standard deviation on the pitch rate of 0.01 deg/s.

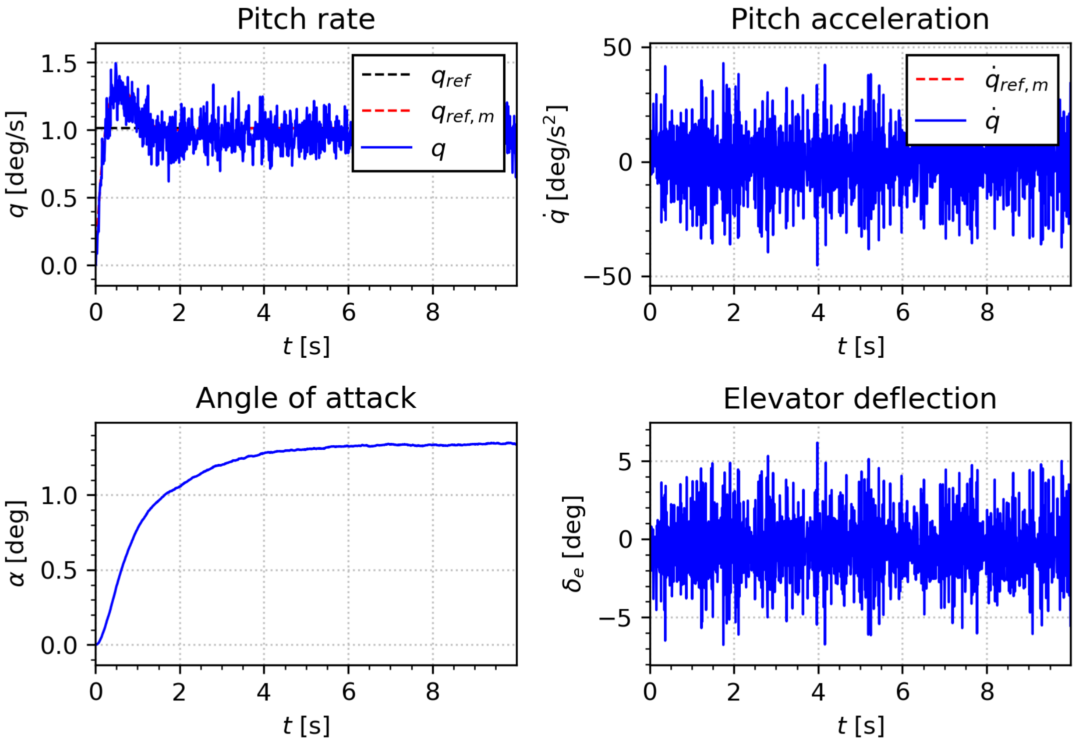


Figure 5.26: Step response of the SAC controller with RMF with a measurement noise standard deviation on the pitch rate of 0.1 deg/s.

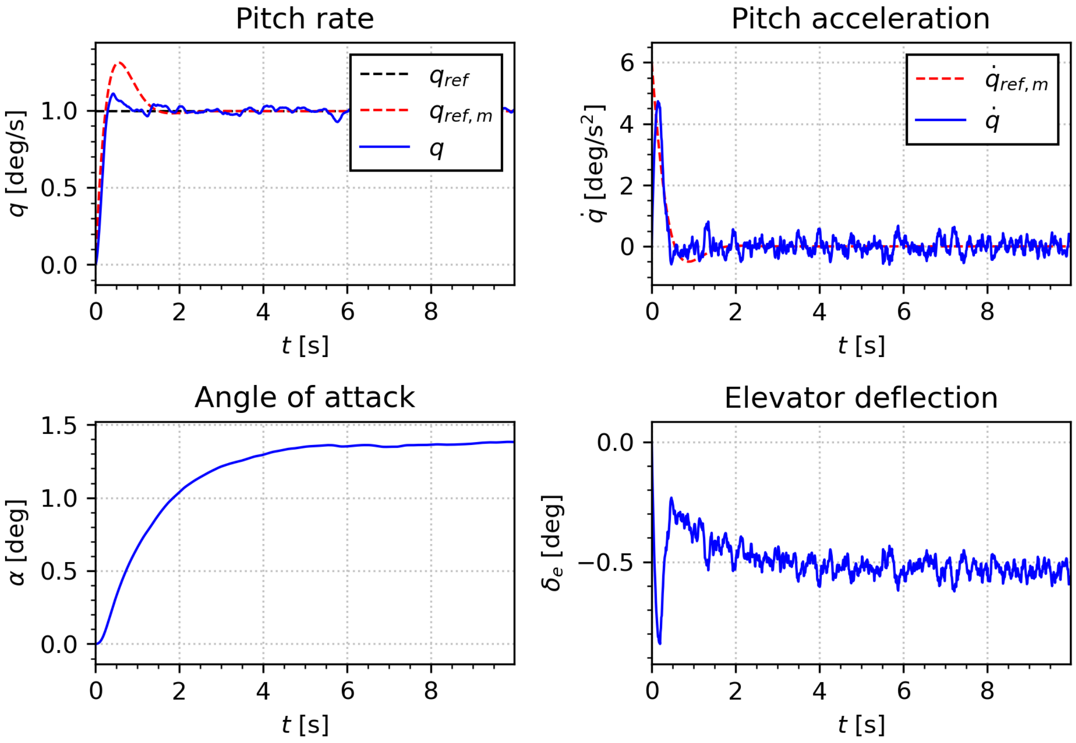


Figure 5.27: Step response of the SAC controller with RMR with a measurement noise standard deviation on the pitch rate of 0.1 deg/s.

5.6. Conclusion

This chapter showed the first experiments performed on the evaluation of HQ&S of an RL flight control system. The SAC framework was used on a simplified longitudinal short period model of the Cessna Citation II. The relation between the Cessna Citation II flight control framework and HQ&S was therefore determined to be the short period motion and its corresponding HQ&S like the CAP and short period damping. For now, this gives an answer to **RQ 3.2**. Two approaches were used for the implementation of a second order reference model. The first approach, RMF, is a command filter approach, where the reference model is placed between the pilot and the closed-loop SAC controller to shape the pilot's input according to desired HQ&S. The second method, RMR, is based on imitating the behaviour of the reference model. The agent receives the pilot pitch rate tracking error, the elevator deflection and the pitch rate and acceleration of the aircraft. The reference model error is not fed to the agent as an observation, but as part of the reward signal, aiming to steer the agent in the direction with the desired HQ&S. This provides an answer to **RQ 3.1**.

Both approaches were applied to the SAC controller before being trained and evaluated. From the results it can be concluded that the RFM approach shows superior performance over the RMR approach in terms of tracking. Furthermore, the LOES fit was better and the HQ&S from the frequency domain analysis showed comparable values to the ones of the reference model. The RMR approach had more mismatch in the LOES fits, which might be due to inaccurate linearization. In the time domain analysis, however, it was shown that the CAP is more stable for the RMR approach. The RMF turned out to be very sensitive to small disturbances. The consequences of this property should be further research and if possible avoided. All in all, both approaches led to a L1 HQ rating for the CAP and therefore the adaptation to HQ&S properties is considered to be performed successfully, thereby answering **RQ 3.3**.

Part III

Additional Results

6
Robustness Analysis

This chapter shows the time response simulation for the online evaluation signal for the flight conditions and CG shifts specified in Part I. The responses for one selected successful run for each SAC controller will be presented and discussed to get more insight in the effect of altering flight conditions and CG shifts. For the selected successful runs the nMAEs are 0.95% and 3.53% and the values for the elevator activity are 0.37 deg/s and 0.39 deg/s for the SAC baseline controller and SAC controller with CAPS respectively, as outlined in Part I.

6.1. H = 2000 m, V = 140 m/s

For the flight condition with H = 2000 m and V = 140 m/s, the results for both SAC controllers are presented in Figure 6.1. This specific successful training run of the SAC baseline controller has a nMAE of 2.35% and an elevator activity of 0.37 deg/s for the given flight condition. The SAC controller with CAPS results in an nMAE of 3.82% and an elevator activity of 0.28 deg/s.

The most significant difference with the nominal flight condition is that the Power Lever Angle (PLA) gets saturated from around $t = 9$ seconds. For both controllers, the velocity gradually decreases after the saturation occurs, but other than that there are no major tracking issues. The nMAEs of both controllers are slightly worse than the nominal flight condition for this reason.

6.2. H = 5000 m, V = 90 m/s

For the flight condition with H = 5000 m and V = 90 m/s, the results for both SAC controllers are presented in Figure 6.2. This specific successful training run of the SAC baseline controller has a nMAE of 3.01% and an elevator activity of 0.54 deg/s for the given flight condition. The SAC controller with CAPS results in an nMAE of 5.19 % and an elevator activity of 0.55 deg/s.

This flight condition is the condition with the lowest dynamic pressure, which decreases the elevator effectiveness. It results in reduced tracking performance and the SAC controller with CAPS does not satisfy the bound of the nMAE anymore. Both controllers are still stable and keep following the reference signal.

6.3. H = 5000 m, V = 140 m/s

For the flight condition with H = 5000 m and V = 140 m/s, the results for both SAC controllers are presented in Figure 6.3. This specific successful training run of the SAC baseline controller has a nMAE of 1.71% and an elevator activity of 0.42 deg/s for the given flight condition. The SAC controller with CAPS results in an nMAE of 4.69% and an elevator activity of 0.41 deg/s.

For this specific flight condition, the PLA gets saturated again due to the increased reference velocity. Therefore the autothrottle can not keep up with the reference velocity.

6.4. Aft CG

For the nominal flight condition (H = 2000 m and V = 90 m/s) with an aft CG shift, the results for both SAC controllers are presented in Figure 6.4. This specific successful training run of the SAC baseline controller has a nMAE of 3.12% and an elevator activity of 0.44 deg/s for the given flight condition. The SAC controller with CAPS results in an nMAE of 4.23% and an elevator activity of 0.47 deg/s.

6.5. Forward CG

For the nominal flight condition (H = 2000 m and V = 90 m/s) with a forward CG shift, the results for both SAC controllers are presented in Figure 6.5. This specific successful training run of the SAC baseline controller has a nMAE of 6.90% and an elevator activity of 0.38 deg/s for the given flight condition. The SAC controller with CAPS results in an nMAE of 5.52% and an elevator activity of 0.47 deg/s.

For the forward CG shift, both controllers have difficulty with removing the steady-state error. There is an offset in the elevator trim angle, which is caused by this CG shift. Therefore, the nMAEs of both controllers are slightly above the threshold of 5%.

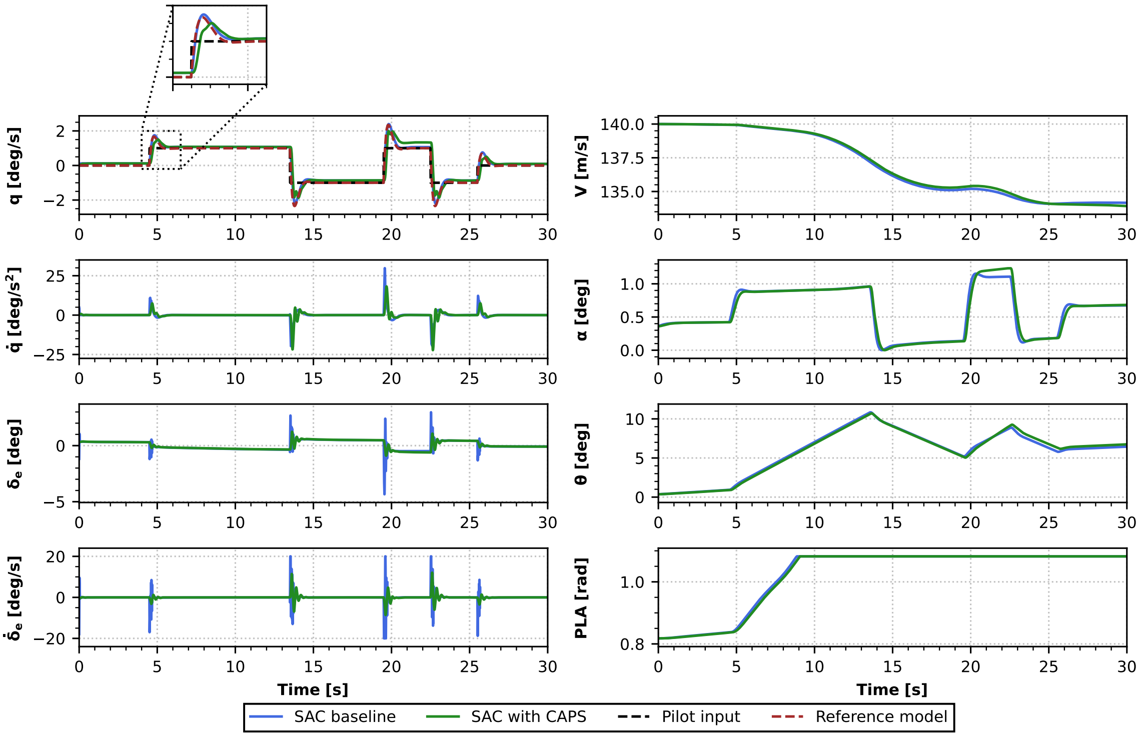


Figure 6.1: Time response of the SAC baseline controller and SAC controller with CAPS for the 3-2-1-1 evaluation signal, for H = 2000 m and V = 140 m/s

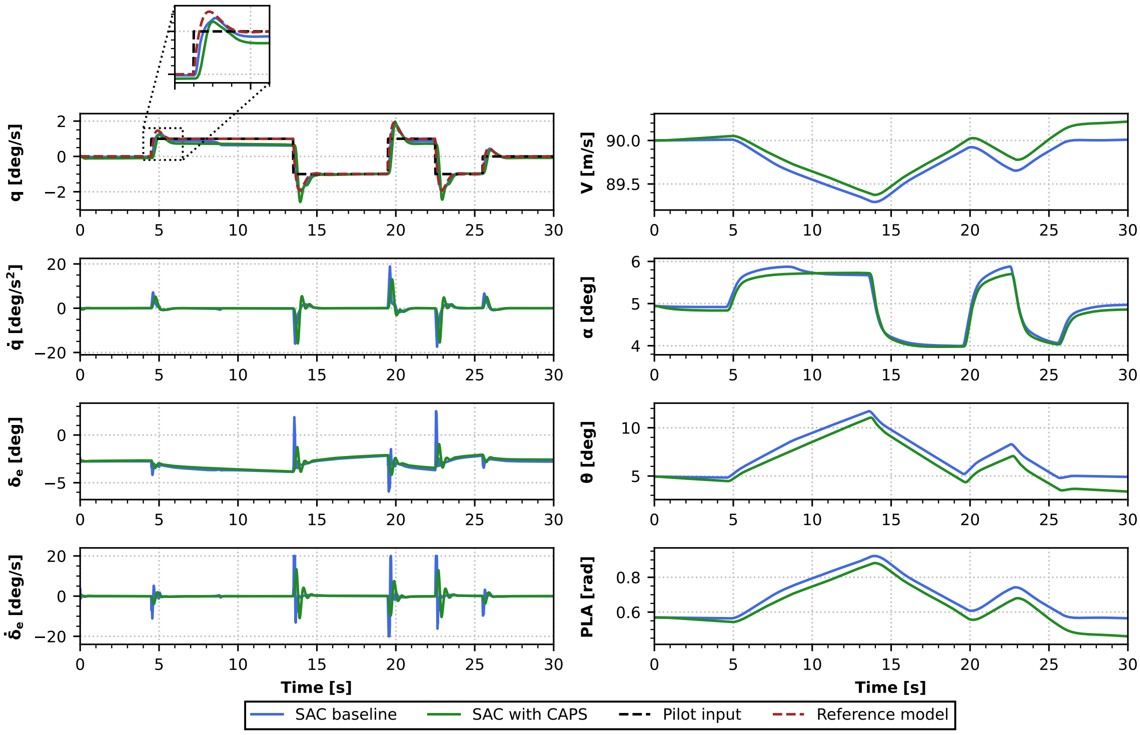


Figure 6.2: Time response of the SAC baseline controller and SAC controller with CAPS for the 3-2-1-1 evaluation signal, for H = 5000 m and V = 90 m/s.

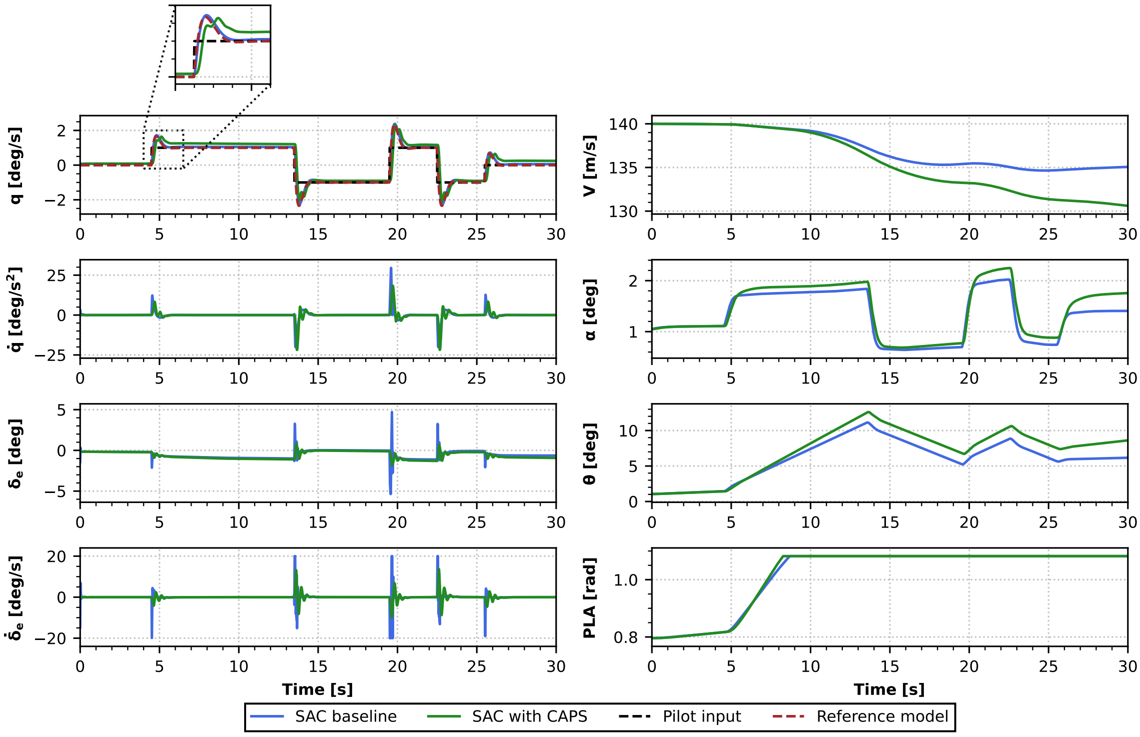


Figure 6.3: Time response of the SAC baseline controller and SAC controller with CAPS for the 3-2-1-1 evaluation signal, for H = 5000 m and V = 140 m/s.

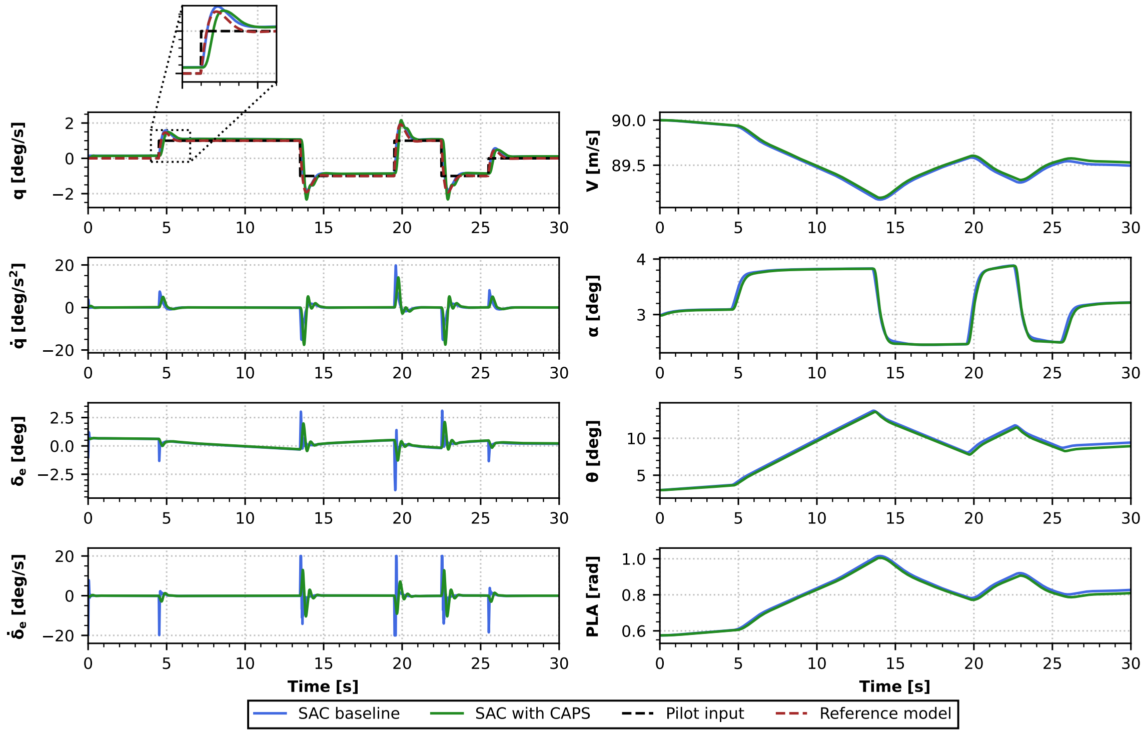


Figure 6.4: Time response of the SAC baseline controller and SAC controller with CAPS for the 3-2-1-1 evaluation signal, for nominal flight conditions and aft CG shift.

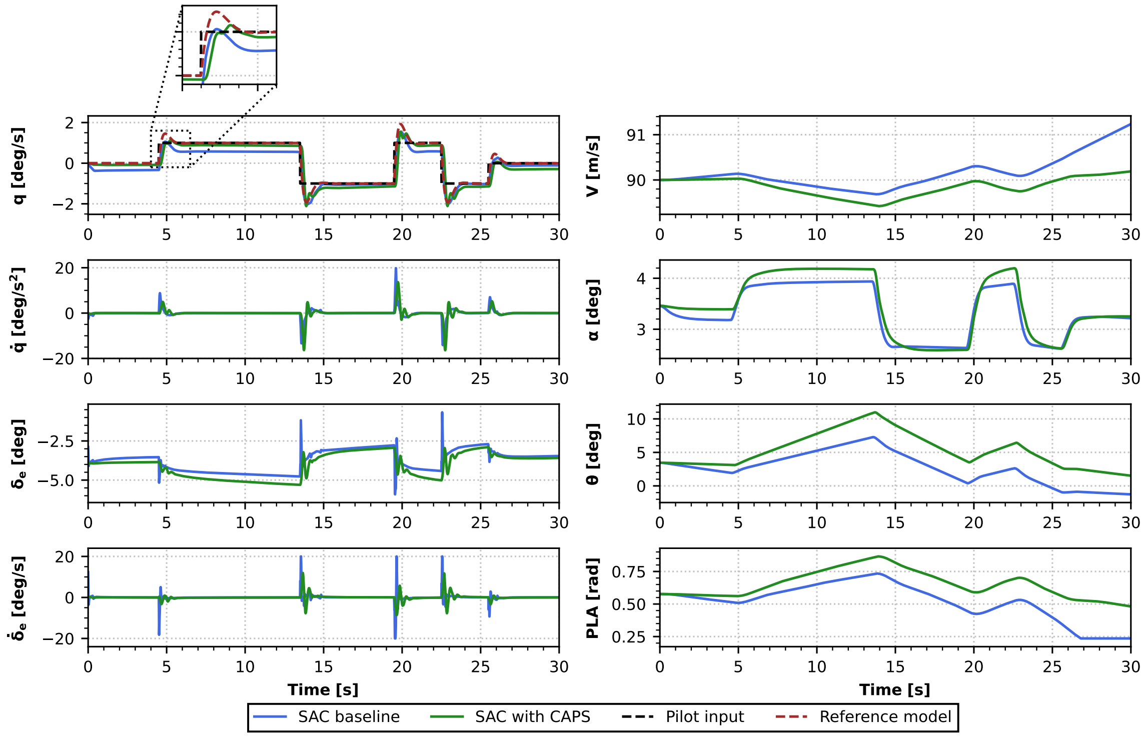


Figure 6.5: Time response of the SAC baseline controller and SAC controller with CAPS for the 3-2-1-1 evaluation signal, for nominal flight conditions and forward CG shift.

6.6. Additional Remarks

In general, it can be concluded that the PLA saturation limits are reached when the reference velocity is increased and velocity thereby decreases. This is also due to the design of the tracking task, which takes too long resulting in relatively high pitch angles, which in turn lead to a higher thrust demand. Furthermore, it can be noticed that the SAC baseline controller reaches the elevator rate saturation limits for all of the simulations, indicating its aggressive policy.

In Part I it was shown that the SAC controller with CAPS is generally more robust to altering flight conditions, which is not directly visible from the results presented in this chapter. The reason for this is that the results presented in Part I are the average of all successful runs, whereas the results shown in this chapter are only single selected successful runs. This means that of all successful runs, there are several runs of the SAC baseline controller that lead to very bad results and therefore the average tracking performance becomes worse as well. This is not the case for the SAC controller with CAPS and therefore can be said to be more robust against off-nominal flight conditions.

7

Threshold Sensitivity Analysis

In Part I the nMAE and elevator activity, $\delta_{e,act}$, bounds were set to 5% and 0.5 deg/s respectively. These were used to determine whether a training run was successful and used for further analysis. The values of the nMAE and elevator activity were calculated for the online evaluation of the 3-2-1-1 step input signal. This chapter shows the impact of relaxing the thresholds for successful runs.

7.1. Elevator Activity Relaxation

In the first sensitivity analysis the nMAE bound is kept at 5% and the elevator activity bound is relaxed to a value of 1 deg/s. Figure 7.1 and Figure 7.2 show the training curves and the progression of the equivalent CAP$_e$ during training. The percentage of successful runs increased significantly for the SAC baseline controller to a value of 75%, whereas the success rate of the SAC controller with CAPS increases only slightly to 57%. The increased successful runs percentage of the SAC baseline controller is clearly visible in the figures, as the shaded blue areas are even more widely spread than they were in the results shown in the article in Part I. This also implies that the SAC baseline controller is mainly limited by the elevator activity bound, which could be explained by the fact that the policy is more aggressive and therefore exerts a higher elevator activity.

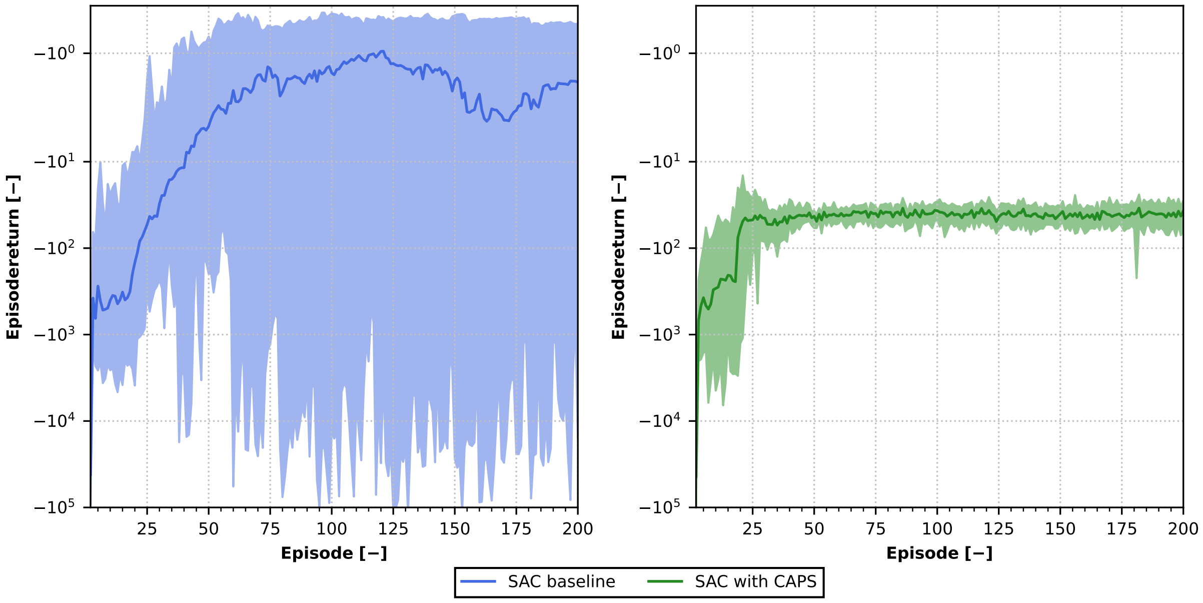


Figure 7.1: Training curves with elevator activity relaxation, showing the episode return for the SAC baseline controller and SAC controller with CAPS during offline learning. Solid blue and green lines present the mean and shaded regions in blue and green show all successful runs.

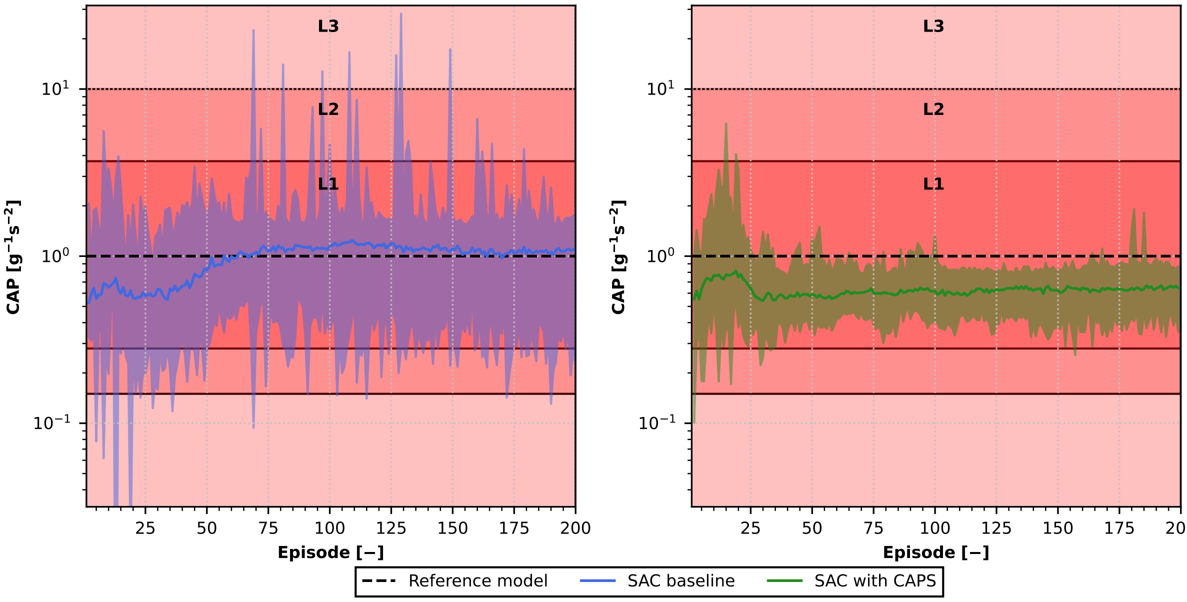


Figure 7.2: The development of the equivalent CAP$_e$ for the SAC baseline controller and SAC controller with CAPS during offline learning with elevator activity and elevator activity relaxation. Solid blue and green lines present the mean and shaded regions in blue and green show all successful runs. The levels of pilot workload ratings are indicated with the red shaded areas.

7.2. Normalized Mean Absolute Error Relaxation

A similar analysis is performed for the nMAE, where the value of the bound is relaxed from 5% to 10%, while the elevator activity bound is kept at 0.5 deg/s. The results of the learning curves and CAP$_e$ during training are shown in Figure 7.3 and Figure 7.4. In this case the SAC baseline success rate stays the same as for the nominal bound, at 26%. The SAC controller with CAPS on the other hand sees a significantly increased number of successful runs with a success rate of 79%. The effects of the additional successful runs for the SAC controller are less visible in the figures, especially the training curve remains fairly constant from 50 episodes onwards. The results indicate that the SAC controller with CAPS is mainly limited due to the nMAE bound, which is in contrast with the SAC baseline controller. This could be explained by the effect that the smooth policy makes the controller slightly more sluggish, reducing the tracking performance and thus the nMAE (compared to the SAC baseline controller).

7.3. Elevator Activity and Normalized Mean Absolute Error Relaxation

When both the elevator activity and nMAE are relaxed to values of 1 deg/s and 10% respectively, the highest number of successful runs is realized for both SAC controller. The SAC baseline controller has a success rate 75% for this scenario and the SAC controller with CAPS reaches 86%. Results are shown in Figure 7.5 and Figure 7.6 for the training curves and equivalent CAP$_e$ development respectively.

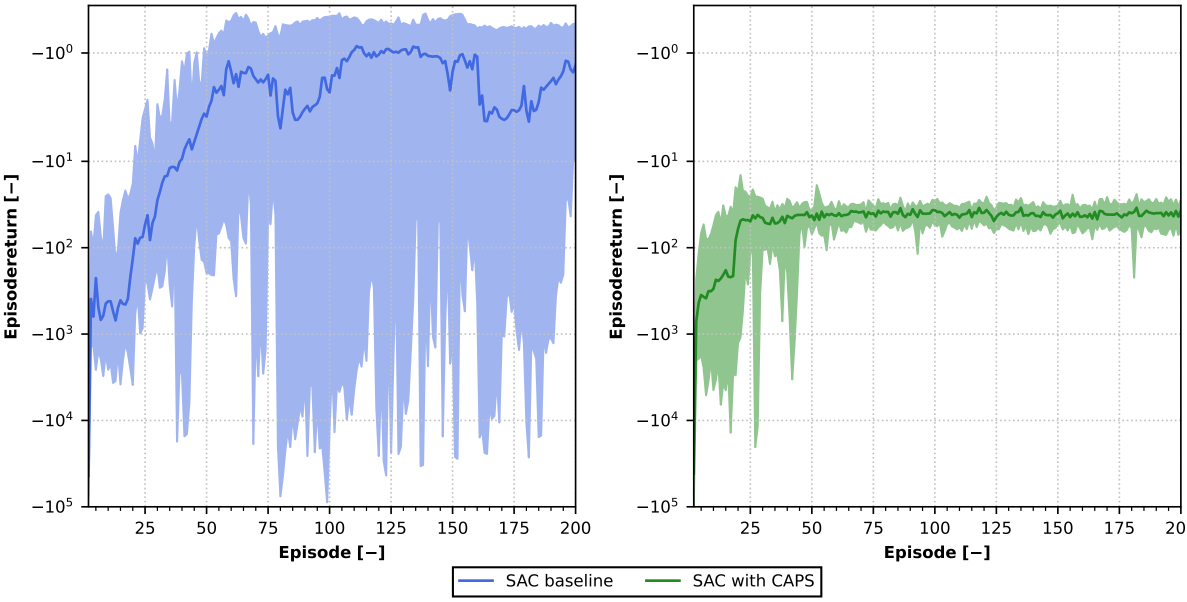


Figure 7.3: Training curves with nMAE relaxation, showing the episode return for the SAC baseline controller and SAC controller with CAPS during offline learning. Solid blue and green lines present the mean and shaded regions in blue and green show all successful runs.

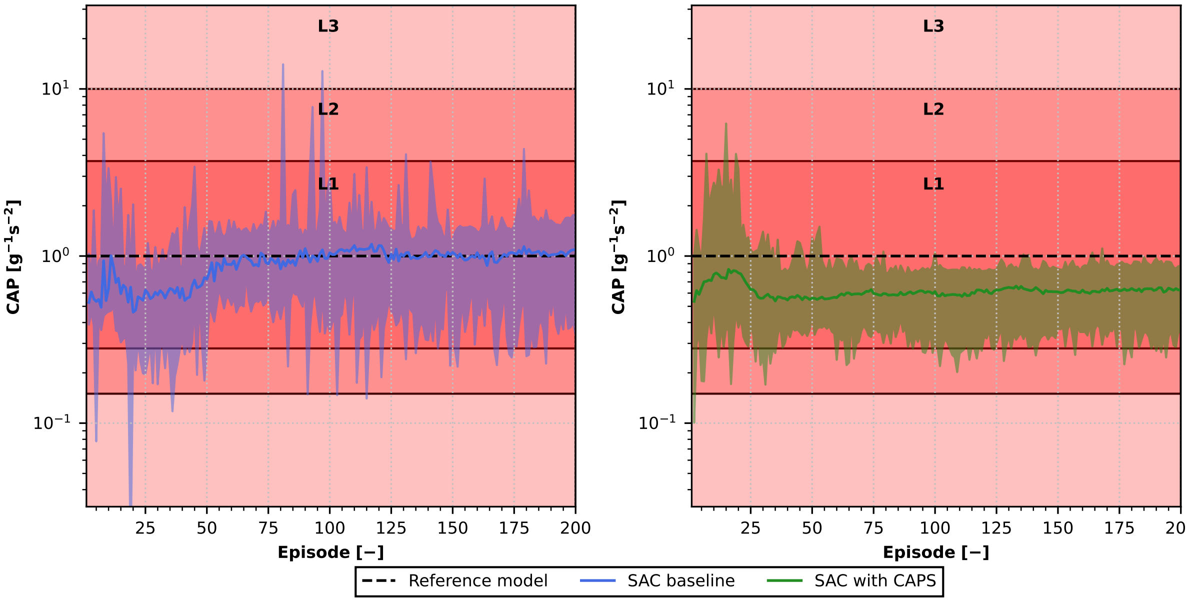


Figure 7.4: The development of the equivalent CAP$_e$ for the SAC baseline controller and SAC controller with CAPS during offline learning with nMAE relaxation. Solid blue and green lines present the mean and shaded regions in blue and green show all successful runs. The levels of pilot workload ratings are indicated with the red shaded areas.

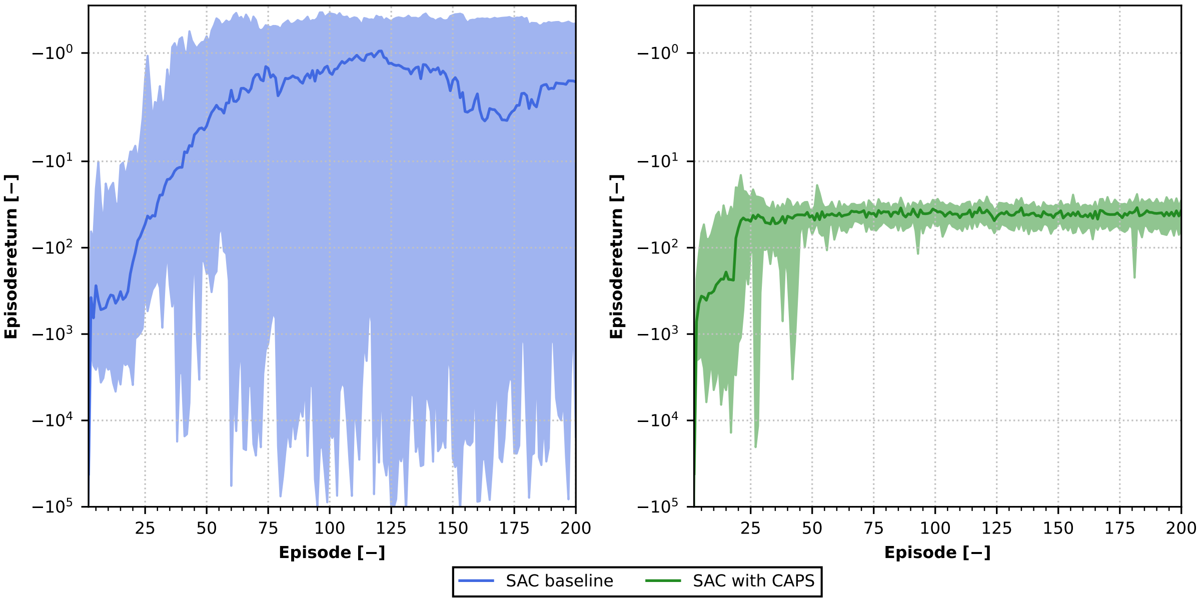


Figure 7.5: Training curves with elevator activity and nMAE relaxation, showing the episode return for the SAC baseline controller and SAC controller with CAPS during offline learning. Solid blue and green lines present the mean and shaded regions in blue and green show all successful runs.

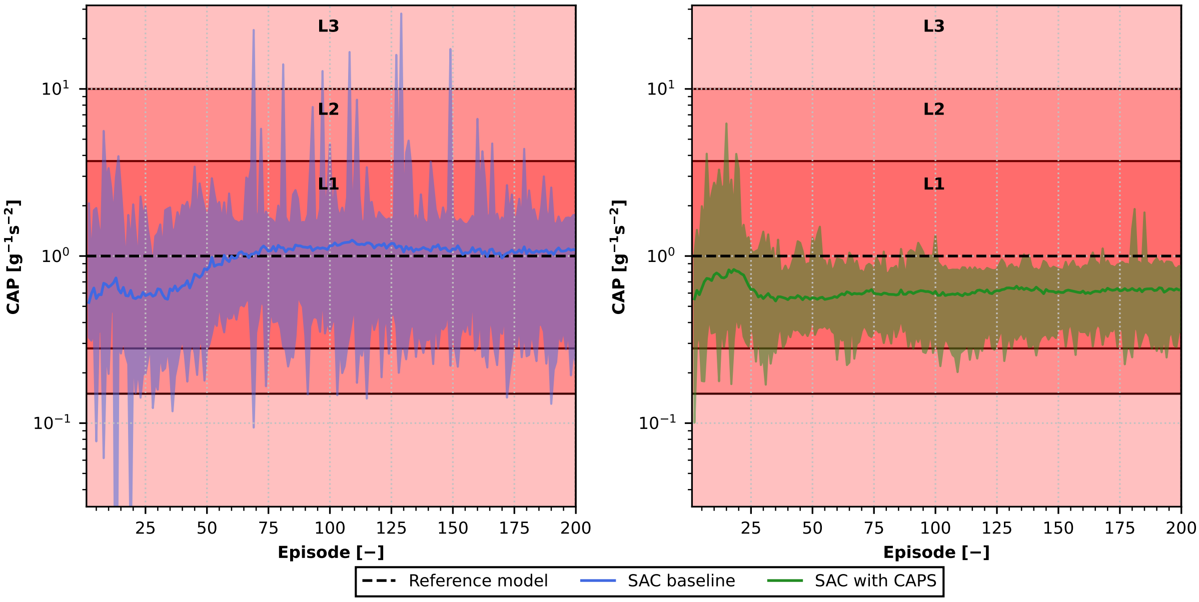


Figure 7.6: The development of the equivalent CAP$_e$ for the SAC baseline controller and SAC controller with CAPS during offline learning with elevator activity and nMAE relaxation. Solid blue and green lines present the mean and shaded regions in blue and green show all successful runs. The levels of pilot workload ratings are indicated with the red shaded areas.

8

Verification and Validation

To ensure that the SAC controller and the tools that are used throughout the research project are implemented correctly, this chapter is devoted to verification and validation. It is required to perform verification and validation of the work to allow for proper further research as well.

8.1. High-fidelity Simulation Model

The nonlinear high-fidelity simulation of the Cessna Citation 500 developed with the Delft University of Technology Aircraft Simulation Model and Analysis Tool (DASMAT) [46] is only available in Matlab and therefore it has to be compiled with C code to use it in Python. It should be verified whether this process is implemented correctly. Figure 8.1 shows the time response of the nonlinear simulation model to a double step input on the elevator deflection angle δ_e for both Matlab and Python. From visual inspection, it can already be observed that there is almost no distinction between the simulation outputs of Matlab and Python. This is further supported by the fact that the average nMAE of all the states is equal to 0.045 %. Furthermore, the nonlinear was found to be a valid representation of the Cessna Citation II PH-LAB [48].

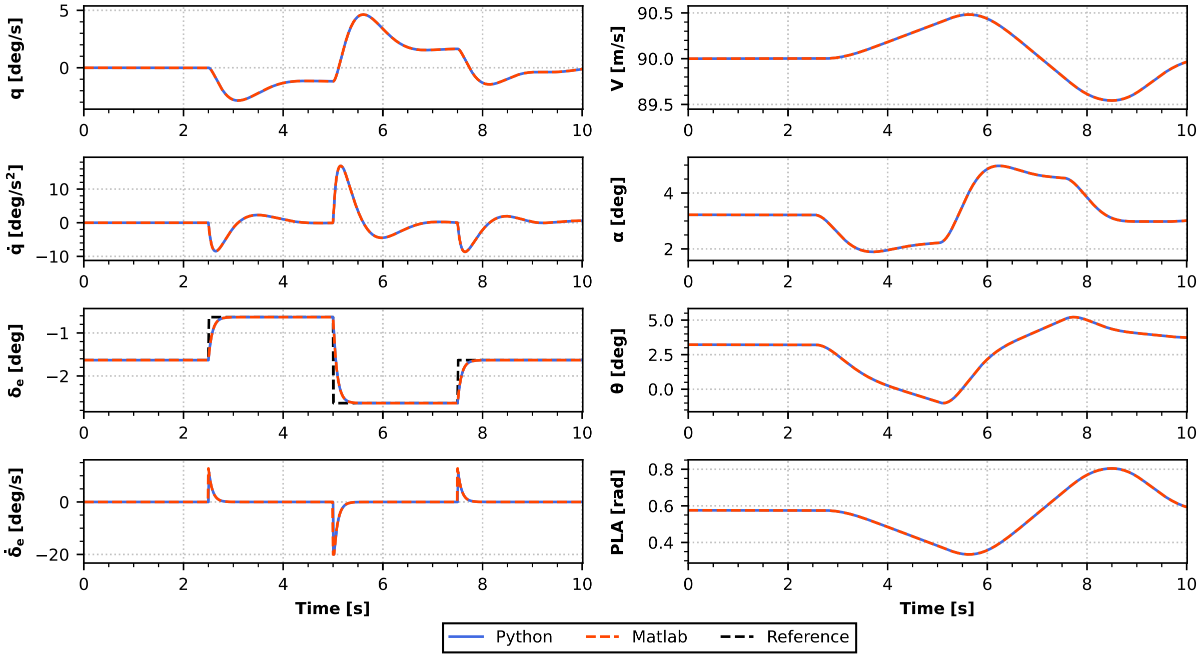


Figure 8.1: Time response of Matlab and Python nonlinear aircraft models to a double step input on the elevator.

8.2. SAC Controller

To ensure that the SAC controllers are implemented correctly, the positive training curves that are presented throughout this report can be used as a reference. These indicate that the SAC controllers have learnt from the environment. The tracking performance of the SAC controller with CAPS in the presence of biased sensor noise as presented in the article in Part I, shows that the controller can be implemented in reality. Furthermore, the implementation of the code is partially based on earlier work [4].

8.3. Linearization

During training a linearized model of the high-fidelity simulation model is used. The linearization is performed with state and input perturbations. To verify whether the linearization of the aircraft model is implemented correctly, the results of a successfully trained SAC controller applied to both the linear and nonlinear aircraft model, subject to the 3-2-1-1 step input evaluation signal are shown in Figure 8.2. In the figure, it can be observed that there are minor differences in the elevator deflection δ_e, velocity V and PLA. These could be the effect of saturation limits and other nonlinearities. However, for the evaluation of the SAC controllers the nonlinear aircraft model is alway used throughout this project, so the nonlinearities will be present there.

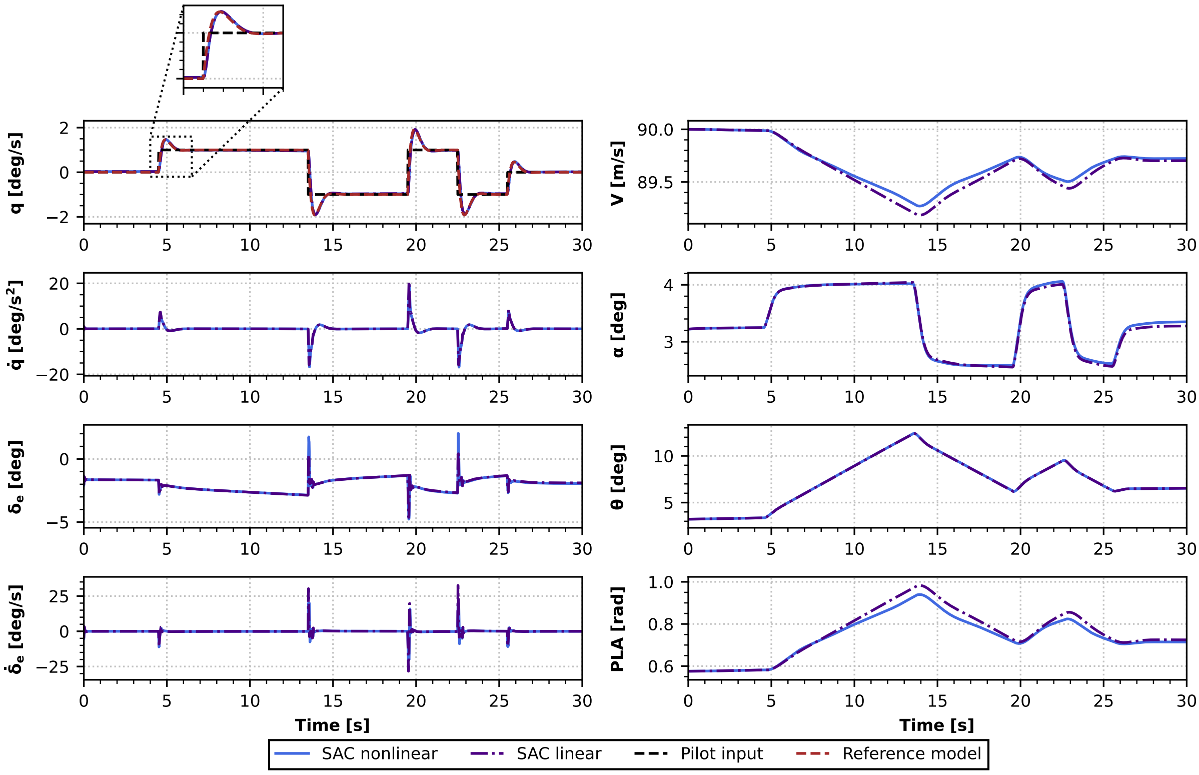


Figure 8.2: Time response for the online 3-2-1-1 evaluation signal, showing the SAC with the linear and nonlinear aircraft model.

8.4. LOES fit

To ensure that the LOES fit algorithm, used throughout the project, is implemented correctly the time response of a step input on the elevator deflection is shown for a HOS (derived from a linearized succesfully trained SAC controller) and the corresponding LOES is plotted in Figure 8.3. It can be seen that the transient response is nearly identical and the steady state pitch rates are slightly different for the LOES and HOS, but sufficiently accurate to determine the short period HQ.

This is further supported by Figure 8.4, where the frequency domain plots of the LOES and HOS are shown. The fit error remains well within the MUAD bounds and therefore the fit is considered succesful. It shows that the LOES fit algorithm is able to fit a LOES to a HOS accurately and correctly.

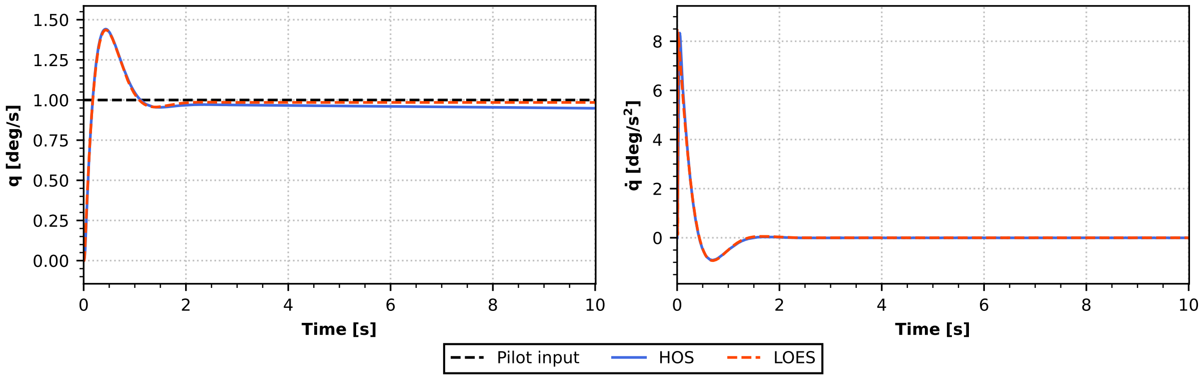


Figure 8.3: Pitch rate and acceleration response to a step input on the elevator for the HOS and LOES.

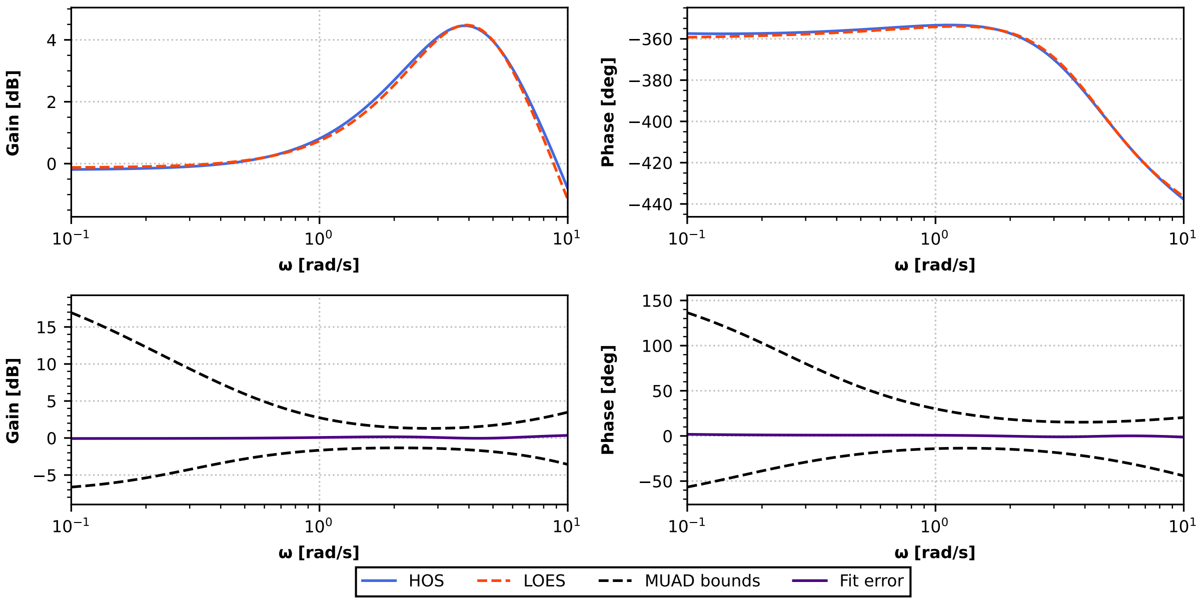


Figure 8.4: LOES fits in the frequency domain showing the fit error and MUAD bounds.

Part IV

Closure

9 Conclusion

The aim of this research is to contribute to the development of RL flight control systems by assessing and integrating HQ&S requirements in the control loop. This report presented the development of two SAC controllers, which both comply with Level 1 short period HQ as outlined in the scientific article in Part I. Furthermore, the report contains a literature survey, in which relevant studies have been highlighted and used as a foundation of this project. Preliminary experiments were performed to get an initial insight in the feasibility of the research. The research is considered to be a proof-of-concept, because to the best of the author's knowledge, no significant studies have been done on the evaluation of HQ&S for RL flight control systems. The research questions have been answered throughout the report and will be summarized here.

Research Question 1

RQ 1 Which RL framework is the most suitable for continuous flight control and the integration of handling qualities and stability properties?

- **RQ 1.1** What are the state-of-the-art RL frameworks for continuous flight control?
- **RQ 1.2** What flight control frameworks will be used for the analysis?
- **RQ 1.3** How will the RL framework be integrated with the flight control framework?

RQ 1 was answered fully in Chapter 3, where an analysis of the general principle behind RL was given and key characterizing concepts were presented to develop a basic understanding of RL. State-of-the-art RL frameworks were discussed and their flight control application was reviewed. The three potential candidates were the TD3 algorithm, ACDs and the SAC framework, which answers **RQ 1.1**. It was found that offline training is the most applicable to this research, as online training is most likely not feasible for a proof-of-concept, eliminating ACDs. The SAC was selected as the best candidate, as it has a better sample efficiency than TD3 methods. Furthermore, the code of a fully developed SAC framework applied to the Cessna Citation II was readily available for use. In the scientific article the Command and Stability Augmentation System (CSAS) was presented as the final flight control framework and the integration with the SAC controller was explained as well, and thus **RQ 1.2** and **RQ 1.3** were answered.

Research Question 2

RQ 2 How will the performance of the RL flight controller be assessed?

- **RQ 2.1** Which handling qualities and stability requirements will be considered as performance criteria?
- **RQ 2.2** How can the selected performance criteria be obtained from flight data?
- **RQ 2.3** How will the selected performance criteria be included in the optimization process?

Chapter 4 provided an overview of all relevant HQ&S guidelines. In the scientific article the short period parameters and CAP were used as HQ&S requirements, which answers **RQ 2.1**. Furthermore, two methodologies were presented for the extraction of HQ&S from nonlinear flight control systems. Time domain analysis is useful, as no linearization is required and higher order dynamics are incorporated, but has the downside that step response simulation is required which might add more computational load. An alternative approach is frequency domain analysis. The LOES concept was presented, which is a method that linearizes nonlinear systems and reduces it to a second order model for direct HQ&S analysis. A combination of both methods is used for the evaluation of the short period HQ in the final version of the SAC controllers, thereby answering **RQ 2.2**. Finally, two methods were suggested for the placement of HQ&S at the desired design point. These methods are based on second order reference models. In the end, the CSAS flight control framework was used to incorporate short period reference HQ, which is basically the same as the RMF method proposed in Chapter 5, which answers **RQ 2.3**. Alltogheter, **RQ 2** has been answered and the performance criteria were established.

Research Question 3

RQ 3 How can the performance criteria be integrated in the RL flight control loop?

RQ 3.1 How can the reward function be modified such that the stability and handling qualities requirements are complied with?

RQ 3.2 What is the relation between the flight control framework structure and the selected performance parameters?

RQ 3.3 How can the RL framework and flight control framework structures be adapted to reach the best integration of the performance parameters?

In the preliminary analysis, described in Chapter 5, two methods for placing HQ&S at the desired location were proposed. One of the methods uses the reference model as a command filter, and applies reference model following. The other approach is based on imitating the behaviour of the reference model, by using reference model feedback in the reward signal. These approaches were not directly used in the final results of the scientific article, but the CSAS structure was used. The reward function is not modified in this case, but the reference model with desired short period HQ is placed in the feedforward path of the pitch rate command system. Two SAC controllers were designed, trained and evaluated. The SAC controller with CAPS showed better tracking performance in realistic simulations with biased sensor noise and was therefore selected as the final controller. This controller was adapted to comply with Level 1 short period HQ. In conclusion, this answers **RQ 3.1**, **RQ 3.2** and **RQ 3.3** and so it provides the answer to **RQ 3**.

Research Question 4

RQ 4 What is the performance of the adapted RL flight controller?

RQ 4.1 How can the RL controller be verified and validated?

RQ 4.2 How does the adapted RL flight controller compare with the same RL flight controller that uses only the tracking error as performance parameter?

The SAC controllers, high-fidelity simulation model of the PH-LAB and LOES fit algorithm were all verified and validated in Chapter 8, which answers **RQ 4.1**. There is no significant distinction between a SAC controller that uses HQ&S as additional performance parameters and a controller that does not. The HQ&S do not change the controller itself but are rather a property of the controller. The reference HQ however do have influence on the HQ of the final SAC controller, and they are left to be selected by the designer of the flight controller in the future. This answer **RQ 4.2** and therefore **RQ 4** is answered completely.

Research Objective

The aim of this research is to contribute to the development of Reinforcement Learning for continuous flight control, by assessing handling qualities and stability properties and integrating them in the control loop.

To conclude, this research has contributed to the development of RL to continuous flight control and stimulates civil aviation to move towards the implementation in practice. The evaluation of longitudinal HQ has provided more insight in the unpredictable black box of the SAC framework, by translating the unknown into well known classical flight control terminology. In future work, it should be investigated whether the RL controller works well in a real-time environment. Next to that, tests of the developed RL controller with hardware should be performed. These two aspects are crucial for the real-world implementation and together with this research they can enable RL flight control for civil aviation.

10 Recommendations

The scope of this research was limited to developing a proof-of-concept of the evaluation of HQ for a state-of-the-art framework, hence it provides a foundation for future work. The main recommendation for future research are highlighted here:

- Most of the hyperparameters were adapted from earlier work on the development of a SAC controller for the PH-LAB [4]. The flight control framework in this research is, however, not exactly the same and therefore different hyperparameters could provide better results. Optimizing the hyperparameters is one of the key points that could improve this research, making the controller even more robust.
- Currently, the actor and critic use the same number of layers and similar initial learning rates. This is probably not ideal and therefore they should be treated as separate hyperparameters to yield better results.
- The different flight conditions used in this research provided an initial insight in the robustness of the developed SAC controllers. It would be interesting to find out how the controllers would react to more extreme flight conditions. Next to that, more failures should be incorporated, like reduced elevator effectiveness or even an inverted elevator.
- An integral term should be added to the state observation vector to remove the steady state error. For the pitch rate command system developed for this research, the integral term would be the pitch angle θ.
- The training could be done for a wider range of pitch rate commands. Different reference signals could be developed, as the more different state-transitions the SAC controller experiences, the better it could generalize for larger state and action spaces. This will increase training complexity and duration.
- More HQ could be added like stability margins and bandwidth criteria, as described in the literature survey of this research project. These could give more insight into the black box of the RL controllers.
- The HQ could be taken into account in the reward function, such that the controller not only learns with the tracking error. This is not straightforward, but could lead to further increased authority over the exact placement of the HQ.
- Research could be performed on the hardware implementation of the developed controller. The goal is to eventually do real flight tests, but in order to realize that, more research should be done on how this controller would operate in real-time.
- A similar approach, with the CSAS flight control framework, could be applied to online learning state-of-the-art RL frameworks such as IDHP. These frameworks have high adaptive capabilities and in combination with HQ evaluation and integration these frameworks could provide further improvements.

References

[1] G. J. Balas. “Flight Control Law Design: An Industry Perspective”. In: *European Journal of Control* 9 (2003), pp. 207–226. DOI: 10.3166/ejc.9.207-226.

[2] R. F. Stengel. “Toward intelligent flight control”. In: *IEEE Transactions on Systems, Man and Cybernetics* 23.6 (1993), pp. 1699–1717. DOI: 10.1109/21.257764.

[3] S. Heyer, D. Kroezen, and E. Van Kampen. “Online Adaptive Incremental Reinforcement Learning Flight Control for a CS-25 Class Aircraft”. In: *AIAA Scitech 2020 Forum*. January. Orlando, Florida, 2020. DOI: 10.2514/6.2020-1844.

[4] K. Dally and E. Van Kampen. “Soft Actor-Critic Deep Reinforcement Learning for Fault-Tolerant Flight Control”. In: *AIAA SciTech Forum*. San Diego, California, 2022. DOI: 10.2514/6.2022-2078.

[5] S. A. Jacklin. “Closing the Certification Gaps in Adaptive Flight Control Software”. In: *AIAA Guidance, Navigation and Control Conference and Exhibit*. Honolulu, Hawaii, 2008. DOI: 10.2514/6.2008-6988.

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

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

[8] H. Dong, Z. Ding, and S. Zhang. *Deep Reinforcement Learning*. Springer Nature Singapore Pte Ltd, 2020. DOI: 10.1007/978-981-15-4095-0.

[9] W. B. Powell, A. George, B. Bouzaiene-Ayari, and H.P. Simao. “Approximate dynamic programming for high dimensional resource allocation problems”. In: *IEEE International Joint Conference on Neural Networks*. 2005. DOI: 10.1109/IJCNN.2005.1556401.

[10] D. V. Prokhorov and D. C. Wunsch. “Adaptive critic designs”. In: *IEEE Transactions on Neural Networks* 8.5 (1997), pp. 997–1007. DOI: 10.1109/72.623201.

[11] B. Sun and E. Van Kampen. “Incremental model-based global dual heuristic programming with explicit analytical calculations applied to flight control”. In: *Engineering Applications of Artificial Intelligence* 89 (2020). DOI: 10.1016/j.engappai.2019.103425.

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

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

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

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

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

[17] G. E. Cooper and R. P. Harper Jr. *The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities*. Tech. rep. Neuilly sur Seine, France: Advisory Group for Aerospace Research and Development, 1969.

[18] J.A. Mulder, W.H.J.J. van Staveren, J.C. van der Vaart, E. de Weerdt, A.C. in ’t Veld, and E. Mooij. *Flight Dynamics*. Delft: Delft University of Technology, 2013.

[19] W. Bihrle. *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.

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

[21] D. A. DiFranco. *Flight Investigation of Longitudinal Short Period Frequency Requirements and PIO Tendencies*. Tech. rep. Air Force Flight Dynamics Laboratory, Research and Techology Division, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, 1967.

[22] J Roskam. *Airplane Flight Dynamics and Automatic Flight Controls Part I*. DARcorporation, 2001.

[23] AGARD. *AGARD 279 - Handling Qualities of Unstable Highly Augmented Aircraft*. Tech. rep. Neuilly sur Seine, France: Adisory Group for Aerospace Research and Development, 1991.

[24] T. P. Neal and R. E. Smith. *An In-Flight Investigation to Develop Control System Design Criteria for Fighter Airplanes*. Tech. rep. Air Force Flight Dynamics Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio, 1972.

[25] J. Gibson. *Development of a design methodology for handling qualities excellence in fly by wire aircraft*. Delft: Delft University Press, 1999.

[26] M. Hendarko. “Development of a Handling Qualities Evaluation Toolbox on the Basis of Gibson Criteria”. In: *International Council of the Aeronautical Sciences Congress*. 2002. URL: `https://api.semanticscholar.org/CorpusID:171085190`.

[27] H-S. Shin, S. He, and A. Tsourdos. *A Domain-Knowledge-Aided Deep Reinforcement Learning Approach for Flight Control Design*. 2019. DOI: `10.48550/arXiv.1908.06884`. URL: `http://arxiv.org/abs/1908.06884`.

[28] S. Van Overeem, X. Wang, and E. Van Kampen. “Modelling and Handling Quality Assessment of the Flying-V Aircraft”. In: *AIAA SciTech 2022 Forum*. DOI: `10.2514/6.2022-1429`.

[29] S. Van Overeem, X. Wang, and E. Van Kampen. “Handling Quality Improvements for the Flying-V Aircraft using Incremental Nonlinear Dynamic Inversion”. In: *AIAA SciTech 2023 Forum*. DOI: `10.2514/6.2023-0105`.

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

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

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

[33] I. Matamoros, T. Lu, M. M.van Paassen, and D. M. Pool. “A Cybernetic Analysis of Maximum Unnoticeable Added Dynamics for Different Baseline Controlled Systems”. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 15847–15852. DOI: `10.1016/j.ifacol.2017.08.2328`.

[34] J. Hodgkinson. “History of Low-Order Equivalent Systems”. In: *Journal of Guidance, Control, and Dynamics* 28.4 (2005). DOI: `10.2514/1.3787`.

[35] E. A. Morelli. *Identification of Low Order System Models From Flight Equivalent Test Data*. Tech. rep. Hampton, Virginia: Langley Research Center, 2000.

[36] L. Sun, L. Shi, W. Tan, and X. Liu. “Flying qualities evaluation based nonlinear flight control law design method for aircraft”. In: *Aerospace Science and Technology* 106 (Nov. 2020). DOI: `10.1016/J.AST.2020.106126`.

[37] L. Sonneveldt, E. R. Van Oort, Q. P. Chu, and J. A. Mulder. “Nonlinear adaptive flight control law design and handling qualities evaluation”. In: *Proceedings of the IEEE Conference on Decision and Control* (2009), pp. 7333–7338. DOI: 10.1109/CDC.2009.5400209.

[38] B. Smit, T. S.C. Pollack, and E. Kampen. “Adaptive Incremental Nonlinear Dynamic Inversion Flight Control for Consistent Handling Qualities”. In: *AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2022* (2022), pp. 1–20. DOI: 10.2514/6.2022-1394.

[39] M. B. Tischler, T. Berger, C. M. Ivler, M. H. Mansur, K. K. Cheung, and J. Y. Soong. *Practical Methods for Aircraft and Rotorcraft Flight Control Design: An Optimization-Based Approach*. American Institute of Aeronautics and Astronautics, 2017.

[40] C. J. Miller. “Nonlinear dynamic inversion baseline control law: Flight-test results for the full-scale advanced systems testbed F/A-18 airplane”. In: *AIAA Guidance, Navigation, and Control Conference 2011* August (2011). DOI: 10.2514/6.2011-6468.

[41] R. Rysdyk and A. J. Calise. “Robust nonlinear adaptive flight control for consistent handling qualities”. In: *IEEE Transactions on Control Systems Technology* 13.6 (2005), pp. 896–910. DOI: 10.1109/TCST.2005.854345.

[42] A. Mirza, M. M . Van Paassen, T. J. Mulder, and M. Mulder. “Simulator Evaluation of a Medium-Cost Variable Stability System for a Business Jet”. In: *AIAA Scitech 2019 Forum*. San Diego, California. DOI: 10.2514/6.2019-0370.

[43] P. A. Scholten, M. M. Van Paassen, Q. P. Chu, and M. Mulder. “A Variable Stability In-Flight Simulation System based on existing Autopilot Hardware”. In: *Guidance, Control and Dynamics* (2020). DOI: 10.2514/1.G005066.

[44] Q. Zhang, W. Pan, and V. Reppa. “Model-Reference Reinforcement Learning Control of Autonomous Surface Vehicles”. In: *59th IEEE Conference on Decision and Control*. 2020. DOI: 10.1109/CDC42340.2020.9304347.

[45] G. Joshi and G. Chowdhary. “Deep Model Reference Adaptive Control”. In: *IEEE Conference on Decision and Control* (2019). DOI: 10.48550/arXiv.1909.08602.

[46] C.A.A.M. Van der Linden. *DASMAT-Delft University Aircraft Simulation Model and Analysis Tool*. Tech. rep. Delft University of Technology, 1996.

[47] S. Mysore, B. Mabsout, R. Mancuso, and K. Saenko. “Regularizing Action Policies for Smooth Control with Reinforcement Learning”. In: *IEEE International Conference on Robotics and Automation*. 2021, pp. 1810–1816. DOI: 10.1109/ICRA48506.2021.9561138.

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