Online reinforcement learning for fixed-wing aircraft longitudinal control

Lee, J.H.; van Kampen, E.

DOI
10.2514/6.2021-0392

Publication date
2021

Document Version
Final published version

Published in
AIAA Scitech 2021 Forum

AIAA SciTech Forum
11–15 & 19–21 January 2021, VIRTUAL EVENT
AIAA Scitech 2021 Forum

10.2514/6.2021-0392

Online reinforcement learning for fixed-wing aircraft longitudinal control

Jun Hyeon Lee*, Erik-Jan van Kampen†
Delft University of Technology, P.O. Box 5058, 2600 GB Delft, The Netherlands

Reinforcement learning is used as a type of adaptive flight control. Adaptive Critic Design (ACD) is a popular approach for online reinforcement learning control due to its explicit generalization of the policy evaluation and the policy improvement elements. A variant of ACD, Incremental Dual Heuristic Programming (IDHP) has previously been developed that allows fully online adaptive control by online identification of system and control matrices. Previous implementation attempts to a high fidelity Cessna Citation model have shown accurate simultaneous altitude and roll angle reference tracking results with outer loop PID and inner loop IDHP rate controllers after an online training phase. This paper presents an implementation attempt to achieve full IDHP altitude control under the influence of measurement noise and atmospheric gusts. Two IDHP controller designs are proposed with and without the cascaded actor structure. Simulation results with measurement noise indicate that the IDHP controller design without the cascaded actor structure can achieve high success ratios. It is demonstrated that IDHP altitude control under measurement noise and atmospheric gusts are achievable under four flight conditions.

Nomenclature

n, m	=	number of states, number of actions
$\mathbf{x}, \mathbf{x}^r, \mathbf{u}$	=	aircraft states, reference aircraft states, aircraft inputs
$\mathbf{s}, \mathbf{a}$	=	reinforcement learning states, reinforcement learning actions
p, q, r	=	aircraft body rates: roll rate, pitch rate, yaw rate
V_{TAS}	=	aircraft true airspeed
α, β	=	angle of attack, sideslip angle
ϕ, θ, ψ	=	aircraft attitude angles: roll angle, pitch angle, yaw angle
x_E, y_E, h	=	aircraft earth reference frame positions
PLA_1, PLA_2	=	engine throttle settings
$\delta_e, \delta_a, \delta_r$	=	aircraft control surface deflection angles: elevator, aileron, rudder
r, c	=	reward, cost
κ	=	reward/cost normalizing term
$J\ \lambda$	=	cost-to-go function, cost-to-go function gradient
$\hat{J}, \hat{\lambda}$	=	cost-to-go function estimate, cost-to-go function gradient estimate
γ	=	discount factor
η_a	=	actor learning rate for baseline controller
η_{a1}, η_{a2}	=	actor 1 learning rate, actor 2 learning rate for cascaded controller
η_c	=	critic learning rate
A_{in}, C_{in}	=	actor input, critic input
γ_{RLS}	=	recursive least squares forgetting factor
$\mathbf{\Theta}, \mathbf{P}$	=	parameter matrix, covariance matrix
K , ϵ	=	recursive least squares kalman gain, innovation error
$\mathbf{F}, \mathbf{G}$	=	system matrix, control matrix
$\hat{\mathbf{F}}, \hat{\mathbf{G}}$	=	system matrix estimate, control matrix estimate

*MSc. Student, Control & Simulation Division, Faculty of Aerospace Engineering, Lee.Junhyeon@gmail.com
†Assisstant Professor, Control & Simulation Division, Faculty of Aerospace Engineering, E.vanKampen@tudelft.nl, AIAA member

I. Introduction

Within, and not limited to, the aerospace industry, there has been a clear trend towards increased automation where higher level tasks are relieved from human intervention. This trend is more apparent within the recreational sector of the industry where Unmanned Aerial Vehicles (UAVs) have gained popularity. A prime example of such increased level of automation is the "follow me mode" feature found in both professional and recreational UAVs demonstrating vision based control [1].

Although the level of autonomy of aircraft systems has been constantly increasing, the underlying attitude and heading flight controller design remains relatively unchanged. Most aircraft in operation to date rely on gain scheduled PID feedback controllers. System identification is essential at multiple operating points within the flight envelope at which controller gains must be tuned through an optimization process. Not only is the development costly, the designed controller show limitations in coping with unanticipated changes in system dynamics or large external disturbances. Nonlinear Dynamic Inversion(NDI) controllers found within some high performance aircraft in operation eliminate the gain scheduling process, but is yet dependent on identified model fidelity and online state measurements [2, 3].

When automated aircraft, such as UAVs, are readily available to the public and become more embedded in daily lives, the need for adaptive controllers is emphasized. Also from the manufacturer's perspective, the benefits of adaptive controllers are clear. Ability to maintain control throughout change in system dynamics due to system failure or large external disturbances such as gust are attractive features for an aircraft. Therefore it can be stated that there is an industry-wide demand for an adaptive flight controller.

Among nonlinear adaptive controller designs, Adaptive Dynamic Inversion (ADI) has been developed with the use of neural networks for online identification of plant inversion error [4] and online identification of control matrix triggered by fault detection [5]. Conceived by [6], Incremental Nonlinear Dynamic Inversion (INDI) based on angular acceleration feedback has shown successful simulation results for a T-tailed UAV model [7], helicopter model [8], and later successfully demonstrated for a flying wing UAV and a passenger aircraft [9, 10]. Applying similar measurement based controller design strategy seen in INDI, Incremental Backstepping (IBS) control method has also been developed [11].

As an alternative approach to adaptive flight control, Reinforcement Learning (RL) can be considered. Within the scope of control, RL can be defined as a process of deriving optimal control policy based on observations made from the environment [12]. Adaptive Critic Designs (ACDs) is a class of Dynamic Programming (DP) RL algorithms where the backwards recursive calculation of Bellman equation is redefined as an algorithm that steps forward in time [13]. ACD maintains separate parametrized actor and critic elements which respectively handles policy improvement and policy evaluation and can be categorized by the critic estimate: Heuristic Dynamic Programming (HDP), Dual Heuristic Dynamic Programming (DHP), and Globalized DHP (GDHP) [14–16]. The separate policy evaluation and improvement structure allows the ACD method to combine the benefits of critic-only and actor-only methods allowing facilitated online implementation [17]. For flight control, several implementation cases can be found. One of the first implementation cases for linear aircraft control simulation can be found in [18]. For nonlinear flight control, successful implementation cases have been reported for a full scale model of: missile [19, 20], fixed-wing aircraft [21–26], and helicopter [27].

Through online estimation of system and control matrices based on incremental measurements, Incremental HDP (IHDP) [24] and Incremental DHP (IDHP) [25] have demonstrated successful control of a nonlinear system without the need for an offline trained model. IDHP was then implemented to the full scale model of the Cessna Citation aircraft model for IDHP angular rate control under perfect sensor assumption [28]. The next step towards implementation in an actual Citation aircraft consists of removing the perfect sensor assumption and establishing longitudinal RL control for altitude tracking.

The contributions of this paper are: 1) Comparison of IDHP controller designs with and without cascaded actor network, 2) Measurement noise impact analysis to IDHP controller designs, 3) IDHP altitude control feasibility demonstration under measurement noise and atmospheric disturbances. Results from the implementation process of extending RL longitudinal control to altitude tracking while adding measurement noise and atmospheric disturbances are provided. The implementation process is divided into three stages. The first stage consists of batch simulation of the two designed IDHP controllers for longitudinal altitude tracking task of the full scale Cessna Citation model under perfect sensor assumption. In the second stage, measurement noise is added to the simulation environment and analysis of its impact is carried out with batch simulation results. The first and second stages combined can be understood as a process to determine a more suitable controller design for altitude tracking task. Finally, the third stage performs empirical analysis on the chosen controller design under light and moderate atmospheric gust scenarios for four flight conditions.

Section II defines the flight control problem as a Markov Decision Process (MDP) and presents the structure and the

adaptation process of the IDHP controller. Section III describes the simulation environment and the design process of IDHP controllers are presented. Section IV presents the batch simulation results and analysis of designed controllers for the longitudinal altitude tracking task. Finally, V concludes the paper and provides recommendations for further analysis.

II. Reinforcement Learning Framework

In this section the RL framework is presented. First, continuous reference tracking task is redefined within the Reinforcement Learning (RL) Adaptive Critic Designs (ACDs) framework. Second, Incremental Dual Heuristic Programming(IDHP) is described regarding its algorithm procedure modification and overall structure.

A. Adaptive Critic Designs Problem

Flight control can be described as a process to achieve the primary goal of error minimization between the desired and the actual states of an aircraft, x_t^r and x_t. Reformulated as a MDP, the process can be represented by an agent occupying state $s_t \in \mathbb{R}^n$ at decision epoch t choosing action $a_t \in \mathbb{R}^m$ according to policy π to receive scalar reward $r_{t+1} \in \mathbb{R}$ and next state $s_{t+1} \in \mathbb{R}^m$ signal from the environment.

Let us define the reward to be negatively proportional to be the sum of squared error at decision epoch t. The reward function is shown in Eq. 1 where l is the number of tracked states and κ_j is a normalizing term for each tracked state s_j. As an alternative view used in this paper, the goal can be represented in terms of minimizing cost where cost is defined as negative reward shown in Eq. 2. Through minimizing the cumulative sum of cost the agent receives, the primary goal of flight control can be achieved.

$$\mathbf{r}_t = r(\mathbf{s}_t) = -\sum_{j=1}^{l} \kappa_j (\mathbf{s}_{t,j}^r - \mathbf{s}_{t,j})^2 \tag{1}$$

$$\mathbf{c}_t = -\mathbf{r}_t = \sum_{j=1}^{l} \kappa_j (\mathbf{s}_{t,j}^r - \mathbf{s}_{t,j})^2 \tag{2}$$

Within the ACD framework, cumulative cost minimization is achieved through the forward step calculation of the Bellman equation [13]. The non-negative cost-to-go function $J(s_t)$ is defined in the form of infinite horizon discounted model as shown in Eq. 3 where $\gamma \in [0, 1]$ represents the discount factor. The flight control problem thus becomes a problem of parametrizing a policy such that the cost-to-go function is minimized.

$$J(\mathbf{s}_t) = \sum_{i=t}^{\infty} \gamma^{i-t} \mathbf{c}_i \tag{3}$$

B. IDHP Agent Overview

The overall structure of the agent follows from [25] as initially conceived with minor changes to the algorithm procedure. Figure 1 shows a schematic of the IDHP agent interacting with an environment. The shaded blocks represent elements that belong to the agent and the white blocks represent elements within the environment. An overview of each element within the agent is given in this subsection.

1. IDHP Algorithm Procedure Modification

The IDHP agent presented in [25] shows that the critic is trained backwards in time. The actor is trained forwards in time, but requires the next time step critic estimations based on next time step states predicted by the incremental model. The advantage of such an adaptation scheme is that additional actor-critic adaptation cycles are possible during one decision epoch. The disadvantage is the reliance on the incremental model to predict the next time step states. This disadvantage is more pronounced in the presence of noise and disturbances that further decreases the reliability of the incremental model. Therefore a measurement based adaptation scheme as seen in [29] and previously implemented in [28] is employed.

Figure 1 shows the agent adaptation occurring over two measurement time steps where the forward pass through the critic is done once in each time step providing the current and next time step cost-to-go function gradients. The

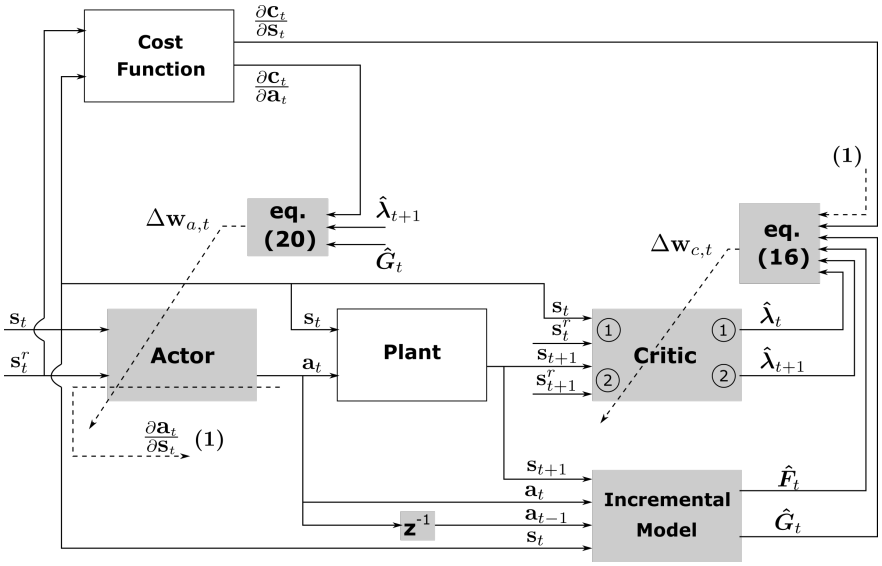


Fig. 1 IDHP schematic diagram with measurement based adaptation paths

motivation for employing such an adaptation scheme is based on the assumption of small time steps and relatively slow changing environment dynamics which forms the fundamental basis for the use of incremental model in IDHP. If the decision epoch frequency is sufficiently high, then the advantage of reduced reliance on the incremental model outweighs the benefit of multiple adaptations during one epoch.

2. Incremental Model

IDHP utilizes an instantaneous linear system model identified through a first order Taylor expansion. Given sufficient sampling rate and relatively slow changing system dynamics the identified linear model at each time instance can be said to be time varying and adequately representative for use in ACDs [24, 25].

Consider a nonlinear discrete system where its time varying states are defined by Eq. 4. The first order Taylor expansion around t_0 gives Eq. 5. Setting $t_0 = t - 1$, Eq. 5 becomes Eq. 6 where $\mathbf{F}_{t-1} = \frac{\partial f(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})}{\partial \mathbf{s}_{t-1}} \in \mathbb{R}^{n \times n}$ is the system matrix and $\mathbf{G}_{t-1} = \frac{\partial f(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})}{\partial \mathbf{a}_{t-1}} \in \mathbb{R}^{n \times m}$ is the control matrix. The incremental form of the Taylor expansion equation is shown in Eq. 7.

$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t) \in \mathbb{R}^n \tag{4}$$

$$\mathbf{s}_{t+1} \approx f(\mathbf{s}_{t_0}, \mathbf{a}_{t_0}) + \frac{\partial f(\mathbf{s}, \mathbf{a})}{\partial \mathbf{s}}|_{\mathbf{s}_{t_0}, \mathbf{a}_{t_0}} (\mathbf{s}_t - \mathbf{s}_{t_0}) + \frac{\partial f(\mathbf{s}, \mathbf{a})}{\partial \mathbf{a}}|_{\mathbf{s}_{t_0}, \mathbf{a}_{t_0}} (\mathbf{a}_t - \mathbf{a}_{t_0}) \tag{5}$$

$$\mathbf{s}_{t+1} \approx \mathbf{s}_t + \mathbf{F}_{t-1}(\mathbf{s}_t - \mathbf{s}_{t-1}) + \mathbf{G}_{t-1}(\mathbf{a}_t - \mathbf{a}_{t-1}) \tag{6}$$

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

It has been shown that given the assumptions of high sampling frequency and relatively slow system dynamics, the change in system states can be approximated by previous incremental state and action measurements with $\mathbf{F}_{t-1}$ and $\mathbf{G}_{t-1}$ approximated by the incremental model. Due to algorithm procedure modification, the adaptation occurs after state measurements based on agent action output and does not rely on incremental state prediction. Therefore, $\mathbf{F}_t$ and $\mathbf{G}_t$ estimated by the incremental model with newly measured state information are utilized for the critic and the actor adaptation procedure. The structure of the system matrix $\mathbf{F}_t$ and control matrix $\mathbf{G}_t$ are given in Eq. 8 and Eq. 9.

$$\mathbf{F}_t = \begin{bmatrix} \frac{\partial s_{1,t}}{\partial s_{1,t}} & \frac{\partial s_{1,t}}{\partial s_{2,t}} & \cdots \\ \vdots & \ddots & \\ \frac{\partial s_{n,t}}{\partial s_{1,t}} & & \frac{\partial s_{n,t}}{\partial s_{n,t}} \end{bmatrix} \tag{8}$$

$$\mathbf{G}_t = \begin{bmatrix} \frac{\partial s_{1,t}}{\partial a_{1,t}} & \frac{\partial s_{1,t}}{\partial a_{2,t}} & \cdots \\ \vdots & \ddots & \\ \frac{\partial s_{n,t}}{\partial a_{1,t}} & & \frac{\partial s_{n,t}}{\partial a_{m,t}} \end{bmatrix} \tag{9}$$

3. Critic

The critic in an ACD structure is responsible for policy evaluation of the separated policy evaluation and improvement structure. The critic block within IDHP estimates the partial derivative of the cost-to-go function with respect to the states, $\lambda(\mathbf{s}_t) = \frac{\partial J(\mathbf{s}_t)}{\partial \mathbf{s}_t}$, and must be continuously differentiable. Therefore the critic can be defined as a differentiable $\lambda(\mathbf{s}, \mathbf{w}_c)$ parametrization.

The adaptation rule of the critic follows the one step temporal difference TD(0) gradient descent method. The TD error is defined in Eq. 10 where $\hat{\lambda}(\mathbf{s})$ represents the estimated partial derivative of the cost-to-go function under the current policy. The adaptation process aims to minimize the error function given in Eq. 11.

$$\mathbf{e}_{c,t} = \frac{\partial\left(\hat{J}(\mathbf{s}_t) - c_t - \gamma\hat{J}(\mathbf{s}_{t+1})\right)}{\partial \mathbf{s}_t} = \hat{\lambda}(\mathbf{s}_t) - \frac{\partial c_t}{\partial \mathbf{s}_t} - \gamma\hat{\lambda}(\mathbf{s}_{t+1})\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{s}_t} \tag{10}$$

$$E_{c,t} = \frac{1}{2}\mathbf{e}_{c,t}^T\mathbf{e}_{c,t} \tag{11}$$

The critic weight update process can thus be defined in Eq. 12 where $\Delta\mathbf{w}_{c,t}$ is given in Eq. 13 with η_c as learning rate for the critic.

$$\mathbf{w}_{c,t+1} = \mathbf{w}_{c,t} + \Delta\mathbf{w}_{c,t} \tag{12}$$

$$\Delta\mathbf{w}_{c,t} = -\eta_c\frac{\partial E_{c,t}}{\partial\hat{\lambda}(\mathbf{s}_t)}\frac{\partial\hat{\lambda}(\mathbf{s}_t)}{\partial\mathbf{w}_{c,t}} = -\eta_c\left(\hat{\lambda}(\mathbf{s}_t) - \frac{\partial c_t}{\partial \mathbf{s}_t} - \gamma\hat{\lambda}(\mathbf{s}_{t+1})\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{s}_t}\right)\frac{\partial\hat{\lambda}(\mathbf{s}_t)}{\partial\mathbf{w}_{c,t}} \tag{13}$$

Both $\hat{\lambda}(\mathbf{s}_t)$ and $\hat{\lambda}(\mathbf{s}_{t+1})$ are directly available from the critic through two separate runs with current and next time step critic inputs. The cost gradient vector $\frac{\partial c_t}{\partial \mathbf{s}_t}$ is obtained from the predefined cost function. The remaining term $\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{s}_t}$ is calculated by Eq. 14 acknowledging the influences of both the current state and the current action to the next state.

$$\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{s}_t} = \frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{s}_t} + \frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{a}_t}\frac{\partial \mathbf{a}_t}{\partial \mathbf{s}_t} \tag{14}$$

The term $\frac{\partial \mathbf{a}_t}{\partial \mathbf{s}_t}$ can be retrieved from the actor gradients. The partial derivatives $\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{s}_t}$ and $\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{a}_t}$ can be calculated using $\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t)$. Recalling the definition of system and control matrices, $\mathbf{F}_t = \frac{\partial f(\mathbf{s}_t, \mathbf{a}_t)}{\partial \mathbf{s}_t}$ and $\mathbf{G}_t = \frac{\partial f(\mathbf{s}_t, \mathbf{a}_t)}{\partial \mathbf{a}_t}$ can be defined resulting in $\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{s}_t} \approx \hat{\mathbf{F}}_t$ and $\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{a}_t} \approx \hat{\mathbf{G}}_t$. Therefore Eq. 14 can be reformulated into Eq. 15. The final critic weight update process is defined in Eq. 16

$$\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{s}_t} \approx \hat{\mathbf{F}}_t + \hat{\mathbf{G}}_t\frac{\partial \mathbf{a}_t}{\partial \mathbf{s}_t} \tag{15}$$

$$\mathbf{w}_{c,t+1} = \mathbf{w}_{c,t} - \eta_c\left[\hat{\lambda}(\mathbf{s}_t) - \frac{\partial c_t}{\partial \mathbf{s}_t} - \gamma\hat{\lambda}(\mathbf{s}_{t+1})\left(\hat{\mathbf{F}}_t + \hat{\mathbf{G}}_t\frac{\partial \mathbf{a}_t}{\partial \mathbf{s}_t}\right)\right]\frac{\partial\hat{\lambda}(\mathbf{s})_t}{\partial\mathbf{w}_{c,t}} \tag{16}$$

4. Actor

While the critic is responsible for policy evaluation, the actor handles policy improvement completing the policy iteration procedure within ACDs. Given states s_t, the actor outputs action a_t according to the parametrized policy $\pi(\mathbf{s}, \mathbf{w}_a)$. The actor function approximator must also be differentiable.

The goal of the actor is to find a policy such that $J(\mathbf{s}_t)\,\forall\, t$ is minimized as shown in Eq. 17. This can then be reformulated into actor weight update process shown in Eq. 18 and Eq. 19.

$$arg\min_{\mathbf{a}_t} \frac{\partial J(\mathbf{s}_t)}{\partial \mathbf{a}_t} = arg\min_{\mathbf{a}_t} \frac{\partial \big(c_t + \gamma J(\mathbf{s}_{t+1})\big)}{\partial \mathbf{a}_t} \tag{17}$$

$$\mathbf{w}_{a,t+1} = \mathbf{w}_{a,t} + \Delta \mathbf{w}_{a,t} \tag{18}$$

$$\Delta \mathbf{w}_{a,t} = -\eta_a \left[\frac{\partial c_t}{\partial \mathbf{a}_t} + \gamma \frac{\partial J(\mathbf{s}_{t+1})}{\partial \mathbf{a}_t}\right] \frac{\partial \mathbf{a}_t}{\partial \mathbf{w}_{a,t}} = -\eta_a \left[\frac{\partial c_t}{\partial \mathbf{a}_t} + \gamma \hat{\lambda}(\mathbf{s})_{t+1} \frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{a}_t}\right] \frac{\partial \mathbf{a}_t}{\partial \mathbf{w}_{a,t}} \tag{19}$$

The $\frac{\partial c_t}{\partial \mathbf{a}_t}$ term can be obtained from the cost function if set to depend on action. $\hat{\lambda}(\mathbf{s})_{t+1}$ is estimated by the critic and $\frac{\partial \mathbf{s}_{t+1}}{\partial \mathbf{a}_t}$ is provided by incremental model output $\hat{\mathbf{G}}_t$. The actor weight update equation can therefore be finalized in Eq. 20.

$$\mathbf{w}_{a,t+1} = \mathbf{w}_{a,t} - \eta_a \left[\frac{\partial c_t}{\partial \mathbf{a}_t} - \gamma \hat{\lambda}(\mathbf{s})_{t+1} \hat{\mathbf{G}}_t\right] \frac{\partial \mathbf{a}_t}{\partial \mathbf{w}_{a,t}} \tag{20}$$

III. Environment Setup and Flight Controller Designs

In this section the simulation setup for the Cessna Citation aircraft longitudinal altitude reference tracking task is presented. First the environment setup including the Cessna Citation I simulation model, gust model, and the cost are given. Then, the structure of the incremental model, critic, and actor as implemented in the simulation are illustrated. Note that that two IDHP based controllers have been designed. One featuring cascaded actor network and one without.

A. Environment

The environment embodies all elements external to the decision making IDHP agent within the simulation. Software package developed within TU Delft combined with the gust model have been utilized to simulate Cessna Citation aircraft dynamics with external atmospheric disturbances as a black box model of the plant to be controlled. A cost function defining the objective of the IDHP controller is designed.

1. Cessna Citation Dynamics

The DASMAT software package, developed within TU Delft for high fidelity aircraft simulation, has been configured with Cessna Citation I aircraft specific parameters. In such configuration settings, DASMAT software package represents the Cessna Citation aircraft model including aerodynamic, propulsion, engine, and control actuator models and will henceforth be referred as the Citation model. The Citation model provides next time step aircraft states $\mathbf{x}_{t+1}$ given actuator command $\mathbf{u}_t$, throttle command $\mathbf{u}_{eng,t}$, and gust terms $\mathbf{u}_{g,t}$ where the current aircraft states $\mathbf{x}_t$ are maintained within the model. The model is simulated at 100 Hz assuming synchronous aircraft state $\mathbf{x}$ measurements. Figure 2 provides an overview of the input-output relationship of the model where the input and output terms are given in Eq. 21 to Eq. 24.

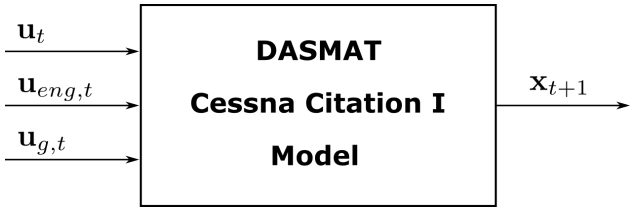


Fig. 2 DASMAT Cessna Citation model showing input and output variables

$$\mathbf{u} = [\delta_e,\ \delta_a,\ \delta_r] \tag{21}$$

$$\mathbf{u}_{eng} = [PLA_1,\ PLA_2] \tag{22}$$

$$\mathbf{u}_g = \left[\hat{u}_g,\ \alpha_g,\ \beta_g,\ \dot{\hat{u}}_g\bar{c}/V,\ \dot{\alpha}_g\bar{c}/V,\ \dot{\beta}_g,\ \hat{u}_{ga},\ \alpha_{ga}\right] \tag{23}$$

$$\mathbf{x} = [p,\ q,\ r,\ V_{TAS},\ \alpha,\ \beta,\ \phi,\ \theta,\ \psi,\ h,\ x_E,\ y_E] \tag{24}$$

The Citation model is initialized at steady flight trimmed states before controlled flight simulation. For full RL control during operation, it is clear that such trimmed initial condition needs to be removed. In [27] a separate neural network has been utilized to determine the initial trim commands. However, both controller designs have demonstrated its ability to control the system within operating regions considered in this research without the need to re-trim. Therefore, online trimming has not been considered in this research.

To analyze the longitudinal motion characteristic of the Citation model during altitude tracking, states q, α, θ, and h are treated as observable variables controlled with actuator command δ_e. Therefore the reinforcement learning states for altitude tracking can be summarized as shown in Eq. 25.

$$\mathbf{s} = [q,\ \alpha,\ \theta,\ h] \quad \mathbf{a} = [\delta_e] \tag{25}$$

The Citation model is provided including simple proportional feedback controllers for yaw damping and maintaining constant airspeed. For the purpose of this research, which focuses on the longitudinal motion control without airspeed control, both controllers provided within the model are utilized to minimize the influence of uncontrolled states within experiments.

To analyze controller design under the presence of measurement noise, the noise terms have been implemented in the form of zero mean Gaussian noise. Cessna Citation aircraft specific noise terms for states q, V_{TAS} and θ have been set following [30]. Because α is a relatively difficult state to measure as the measurement process involves a mechanical wind vane when compared to body rate and attitude angle states, a larger noise standard deviation value of 0.5° has been given. Perhaps the most important noise parameter for this research is the altitude noise standard deviation as it directly affects both critic and actor as input. No clear measurement noise statistics are available for the Cessna Citation and a standard deviation of 0.5 meters is assumed. The measurement noise parameters can be summarized in Table 1. It is to be noted that no error is given to the control state δ_e as measured δ_e is not utilized during IDHP control.

Table 1 Gaussian noise parameters

States	q **[rad/s]**	α **[rad]**	V_{TAS} **[m/s]**	θ **[rad]**	h **[m]**
Noise standard deviations	$6.325 \cdot 10^{-4}$	$8.73 \cdot 10^{-3}$	0.029	$6.325 \cdot 10^{-4}$	0.5

2. Gust model

As it can be seen in Figure 2, the Citation model allows gust terms as input. The gust terms are processed within the aerodynamic sub model of the Citation model. The aerodynamic force and moment coefficients due to turbulence are calculated and incorporated into calculating total aerodynamic forces and moments.

Only symmetrical gust terms are considered as any asymmetrical gust terms may interfere with longitudinal motion assessment and may even lead to failure when lateral motion is not controlled. The longitudinal gust terms provided to the Citation model can be summarized as: $\hat{u}_g$, α_g, $\dot{\hat{u}}_g\hat{c}/V$, and $\dot{\alpha}_g\hat{c}/V$.

The longitudinal Dryden gust model used in this paper is defined by the transfer functions shown in Eq. 26 and Eq. 27 to calculate the aforementioned symmetric gust terms. Above 609.6 m, Dryden gust model assumes isotropic gust and is completely defined by three variables: intensity σ_g $[m^2/s^2]$, scale length L_g $[m]$, and airspeed V_{TAS} $[m/s]$. σ_g needs to be continuously referenced from a lookup table based on gust intensity probability of exceedance. For the purpose of controller performance analysis to different gust intensities, σ_g will instead be set at a constant value representative of the probability of exceedance magnitudes corresponding to light and moderate gust scenarios.

$$H_{\hat{u}_g w}(s) = \frac{\sigma_g}{V_{TAS}} \sqrt{\frac{2L_g}{V_{TAS}}} \frac{1}{1 + \frac{L_g}{V_{TAS}} s} \tag{26}$$

$$H_{\hat{\alpha}_g w}(s) = \frac{\sigma_g}{V_{TAS}} \sqrt{\frac{L_g}{V_{TAS}}} \frac{1 + \sqrt{3}\frac{L_g}{V_{TAS}} s}{1 + \frac{L_g}{V_{TAS}} s} \tag{27}$$

Two different settings for the gust model are used within this paper, summarized in a list shown below. The parameters σ_g and L_g have been referenced from MIL-HDBK-1797 [31]. Scale length has been set to $L_g = 533.4m$ using the Dryden model for high altitude (above $609.6m$). σ_g is set at $\sigma_g = 0.836\,m^2/s^2$ and $\sigma_g = 5.946\,m^2/s^2$ each representing the probability of exceedance $P_{exc} = 10^{-1} \in [0, 1]$ "light gust" and $P_{exc} = 10^{-2} \in [0, 1]$ "moderate gust" for all simulated flight condition altitudes.

1) **No gust**: Zero gust input vector to the Citation model
2) **Light gust**: $\sigma_g = 0.84\,m^2/s^2$ and $L_g = 533.4\,m$
3) **Moderate gust**: $\sigma_g = 5.95\,m^2/s^2$ and $L_g = 533.4\,m$.

3. Cost function

For the task of altitude control, the only predetermined reference provided by the environment is h^r. Following the definition of cumulative cost previously stated in Eq. 2, the cost provided by the environment to the agent can be defined as a scalar term defined by Eq. 28 where κ_h is the scaling term. The scaling term $\kappa_h = 0.0001$ has been chosen to keep the magnitude of cost scalar comparable to cost-to-go function gradient estimates during controller adaptation.

$$c_t = \kappa_h (h_t^r - h_t)^2 \tag{28}$$

For the cascaded model, an additional cost signal based on θ is available due to the reference generated by the outer layer actor network in a cascaded actor network design. With $\kappa_\theta = 1$, Eq. 29 defines the cost function for the cascaded IDHP controller design.

$$c_t = \kappa_h (h_t^r - h_t)^2 + \kappa_\theta (\theta_t^r - \theta_t)^2 \tag{29}$$

B. Altitude Control with IDHP

The agent setup including parameter estimation methods and input/output states are discussed in this subsection. The IDHP agent design largely follows from [25] where the IDHP agent has been introduced. For the purpose of altitude tracking task, the state vector used within IDHP is appended with user generated altitude reference. Therefore the IDHP states and action vectors are redefined in Eq. 30:

$$\mathbf{s} = [q,\ \alpha,\ \theta,\ h,\ h^r]\,, \qquad \mathbf{a} = [\delta_e] \tag{30}$$

1. Incremental Model

The incremental model is identified through Recursive Least Squares (RLS) parameter estimation method. RLS method has been chosen due to its incremental update rule beneficial for online parameter estimation reducing computational cost and alleviating issues arising from matrix inversion. Given states and actions in Eq. 25 the RLS update method can be defined.

Incremental state and action vectors are defined by $\Delta\mathbf{s}_t = \mathbf{s}_t - \mathbf{s}_{t-1}$ and $\Delta\mathbf{a}_t = \mathbf{a}_t - \mathbf{a}_{t-1}$. The goal is to estimate $\mathbf{F}_t$ and $\mathbf{G}_t$ after next time step states have been measured from the plant. The algorithm requires initial covariance matrix $\mathbf{P}_0 \in \mathbb{R}^{n+m \times n+m}$ and $\mathbf{\Theta}_0 \in \mathbb{R}^{n+m \times n}$ to be known. At each RLS algorithm epoch k, the regression vector $\mathbf{r}_k$ and measurement vector $\mathbf{y}_k$ are defined as shown in Eq. 31. The RLS Kalman gain can then be calculated using Eq. 32 with RLS forgetting factor followed by the Kalman error ϵ_k calculation in Eq. 33. Finally, $\hat{\mathbf{P}}_k$ and $\hat{\mathbf{\Theta}}_k$ can be calculated using Eq. 34 and Eq. 35. From $\hat{\mathbf{\Theta}}_k$, both $\hat{\mathbf{F}}_t$ and $\hat{\mathbf{G}}_t$ can be found according to Eq. 36.

$$\mathbf{r}_k = [\Delta\mathbf{s}_t,\ \Delta\mathbf{a}_t]\,, \qquad \mathbf{y}_k = [\Delta\mathbf{s}_{t+1}] \tag{31}$$

$$K_k = \mathbf{P}_{k-1} \cdot \mathbf{r}_k^T \left[\mathbf{r}_k \cdot \mathbf{P}_{k-1} \cdot \mathbf{r}_k^T + \gamma_{RLS}\right]^{-1} \tag{32}$$

$$\boldsymbol{\epsilon}_k = \mathbf{y}_k - \mathbf{r}_k \cdot \hat{\boldsymbol{\Theta}}_{k-1} \tag{33}$$

$$\hat{\boldsymbol{\Theta}}_k = \hat{\boldsymbol{\Theta}}_{k-1} + K_k \cdot \hat{\boldsymbol{\epsilon}}_k \tag{34}$$

$$\hat{\mathbf{P}}_k = \mathbf{P}_{k-1} - K_k \cdot \mathbf{r}_k \cdot \mathbf{P}_{k-1} \tag{35}$$

$$\hat{\boldsymbol{\Theta}}_k = \begin{bmatrix} \hat{\mathbf{F}}_t^T \\ \hat{\mathbf{G}}_t^T \end{bmatrix} \tag{36}$$

The RLS estimation method shown above requires an initial $\mathbf{P}_0$ and $\boldsymbol{\Theta}_0$. The matrices are initialized as shown in Eq. 37. For $\mathbf{F}_0^T$ in $\boldsymbol{\Theta}_0$, the off diagonal elements are set as small non zero values to avoid negligible one step ahead $\boldsymbol{\lambda}_{t+1}$ contribution to the critic update process during early adaptation phase. The initial $\mathbf{G}_0$ found in $\boldsymbol{\Theta}_0$ have been given negative values with knowledge that positive elevator deflection results in negative change in chosen aircraft states. The final goal of this research is to contribute to developing an online flight controller where minimal knowledge of the controlled system is assumed. To avoid providing too much information to the initial parameter matrix, equal values have been chosen. As for the initial covariance matrix $\mathbf{P}_0$, a simple identity matrix has been given as no prior knowledge of parameter covariances is assumed.

The only tunable parameter in the RLS parameter estimation method is the forgetting factor $\gamma_{RLS} \in [0, 1]$ where higher γ_{RLS} leads to a more conservative update process retaining previous information. The forgetting factor is set to be 0.9 for all simulations conducted in this paper. This decision has been made based on the fact that a relatively stable $\mathbf{F}_t$ and $\mathbf{G}_t$ estimations are desirable for a stable cost-to-go function derivative estimation by the critic.

$$\boldsymbol{\Theta}_0 = \begin{bmatrix} 1 & 0.01 & 0.01 & 0.01 \\ 0.01 & 1 & 0.01 & 0.01 \\ 0.01 & 0.01 & 1 & 0.01 \\ 0.01 & 0.01 & 0.01 & 1 \\ -0.1 & -0.1 & -0.1 & -0.1 \end{bmatrix} \qquad \mathbf{P}_0 = \mathbf{I}_5 \tag{37}$$

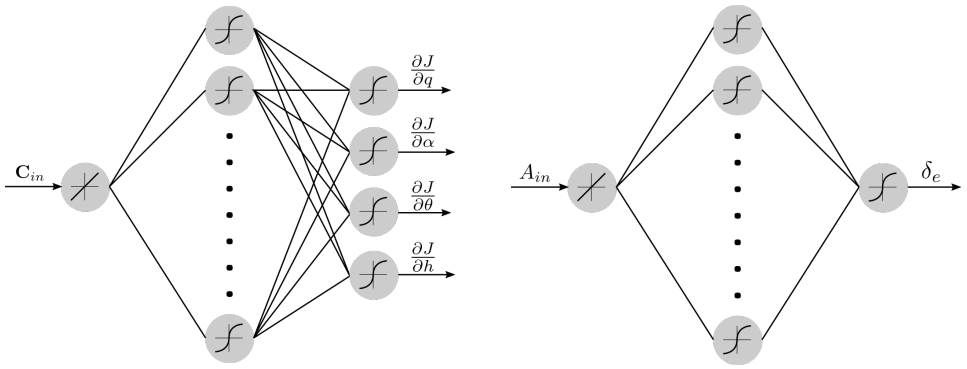
2. Critic and Actor Design

The critic and the actor within an IDHP agent are represented by fully-connected single layer neural networks and their weights are updated by means of back propagation. Figures 3b and 3a show the structure of the neural networks. The critic estimates the partial derivative of the cost-to-go function $\mathbf{J}$ with respect to states while the actor directly outputs the longitudinal control action δ_e.

Linear activation functions have been used for both critic and actor input layers. Hyperbolic tangent, or tanh, activation functions have been used for the hidden layer of both actor and critic neural networks. For the output layer, tanh activation function is chosen again for the critic while a scaled tanh activation function is chosen for the actor.

Hyperbolic tangent activation functions used for the hidden layers and the output layers show desirable qualities when compared to other activation functions. It is a differentiable nonlinear activation function and thus qualifies for use in both actor and critic neural networks. The activation function behaves similarly to the linear activation function until it reaches activation limits, to which it smoothly converges. This bounded property of the hyperbolic tangent activation function is especially useful as an actor output layer activation function. The output activation function dictates the overall shape of the control policy maintained by the actor. And the extent of its influence over the overall control policy is greater when the number actor network input are relatively low. Therefore, by using a scaled tanh activation function, a natural control policy boundary can be set within the physical limitations of the Cessna Citation aircraft.

For both critic and actor input C_{in} and A_{in}, scaled altitude tracking error $h_{scale} \cdot h_e = \kappa_h(h^r - h)$ is used where $\kappa_h = 0.0001$ is a normalizing term to keep the order of magnitude among states similar. For the altitude tracking task,



(a) Critic neural network **(b) Actor neural network**

Fig. 3 Critic and actor structure

altitude error is the most important and relevant information. By only providing error term h_e the cost-to-go function derivative dimensions are reduced compared to providing all longitudinal states and error information to the critic while minimizing the loss of meaningful information necessary for obtaining a good cost-to-go function derivative estimate. A similar argument can be made about only providing h_e term to the actor. Although it is true that extra states may help the actor to form a more refined policy, the increased dimensionality will hinder learning speed and success ratio. This is commonly known as the curse of dimensionality [29].

The temporal difference forgetting factor $\gamma \in [0, 1]$ directly affects the learning process of the IDHP agent. If set too low, the agent becomes myopic and greedy [29]. Because only the altitude error information is used to estimate the cost-to-go function derivative, a relatively high forgetting factor of 0.9 is set. The number of neurons within the hidden layer is another important factor in ACD agent design. If set too low, the parametrization will be too coarse while a large number of neurons will increase computational complexity. Because of the relative simplicity of the actor and critic neural input, preliminary analysis has shown that the 10 hidden neurons are adequate for generalization.

3. Cascaded Actor Structure

A conventional cascaded PID controller structure utilizes an outer loop controller generating a reference signal for the successive inner loop controller to track. This form of controller structure is typically employed when a controlled state of the inner loop has a faster response in relation to the control input [32]. For fixed-wing aircraft systems, the cascaded controller architecture is implemented in the form of inner loop rate Stability Augmentation System (SAS) and outer loop Control Augmentation System (CAS) enclosed by an autopilot [33].

A cascaded actor network design method has been used for ACDs in several previous studies to explicitly structure the actor network using expert knowledge of the system [23, 25, 27]. Under the primary goal of establishing RL Citation altitude control, the cascaded actor network is implemented for IDHP controller design and compared to the design without. The schematic of the designed cascaded actor network is shown in Figure 4. As an intermediary state, θ has been chosen based on its relative ease of measurement when compared to α and lower signal to noise ratio when compared to q.

The use of cascaded actor network allows an additional θ_e signal to be used as critic input. However, this is not done for two reasons. First, preliminary analysis has shown that additional critic input term negatively affects the overall success ratio of the controller. Second, a more direct comparison with the baseline controller can be made when the information provided to the critic are the same among both designs.

4. Adaptive Learning Rates

The last tunable parameters for the IDHP agent are the critic and actor learning rates: η_a and η_c. These non-negative parameters govern the adaptation speed of the critic and the actor. As a relatively large learning rate after convergence to target may lead to divergence due to unnecessary adaptation, the learning rate are typically set to decrease with time [16, 27].

An alternative approach is the use of adaptive learning rates where the learning rates are tuned based on error measurements instead of decaying over time. Successful ACD controller implementation results have been reported in

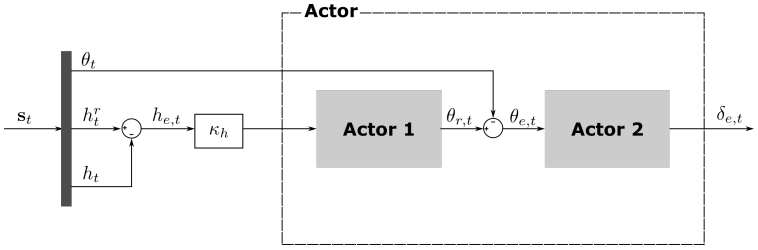


Fig. 4 Schematic of the cascaded actor for altitude tracking task

[34] and [25] using adaptive learning rates. The main advantage of such error based adaptive learning rates compared to time decaying learning rates is that the agent can re-adapt to changed environment with sufficient learning rate regardless of the amount of time that had passed.

Experiments conducted in this paper use error based adaptive learning rates. The learning rates are adjusted based on the Root Mean Squared Error (RMSE) of altitude to reference over the past 100 measurements which corresponds to 1 second in this simulation setting. If the RMSE of altitude is lower than h_{thresh}, learning rate decreases to η_{low}. Otherwise the learning rate is maintained at initial value η_{high}. This "on-off" strategy based on error threshold is less refined when compared to gradual adjustments seen in [34] and [25] but allows faster and stronger adaptation when environment changes are sudden. The parameters for the adaptive learning rate strategy used in the simulation are given in Eq. 38 for the baseline controller design and in Eq. 39 for the cascaded controller design. For the cascaded controller design, an additional θ_{thresh} is set for the past 10 measurements to adjust the learning rate of the second actor network.

$$\eta_{a,high} = [5,\ 10,\ 25], \quad \eta_{c,high} = [2,\ 5,\ 10], \quad \eta_{a,low} = 0.2, \quad \eta_{c,low} = 0.2, \quad h_{thresh} = 20m \tag{38}$$

$$\eta_{a1,high} = 25, \quad \eta_{a2,high} = 1, \quad \eta_{c,high} = 10, \quad \eta_{a,low} = 0.2, \quad \eta_{c,low} = 0.2, \quad h_{thresh} = 20m, \quad \theta_{thresh} = 2^\circ \tag{39}$$

IV. Results and Discussions

This section presents and discusses simulation results of 3 experiments. The first experiment is set up to compare the non-cascaded IDHP controller design (baseline controller) and the cascaded actor IDHP controller design (cascaded controller) in a perfect scenario where both measurement noise and disturbances are not present. The second experiment is designed to determine a controller design favorable for online longitudinal control of the Citation model in the presence of measurement noise. The third experiment is aimed to assess the performance of the chosen controller to external disturbances as well as measurement noise.

The altitude reference signal for experiments 1 and 2 is generated by a sinusoidal function with amplitude 250 meters and frequency 0.005 Hz to represent repeated climb and descent at approximately $1500 ft/min$. For experiment 3, a reference altitude consisting of climb, short cruise, descent, and cruise has been set to simulate a realistic scenario of cruise altitude increase shortly followed by cruise altitude correction.

Each controller design has been simulated for 300 independent runs at 100 Hz with no prior training. Through multiple batch simulations, it has been found that 300 independent runs are adequately representative of the controller performance for the chosen neural network weight initialization.

For experiments 1 and 2, the success conditions are defined as follows. The first sinusoidal period is named the transient period where the IDHP controller adapts its policy for reference tracking. The second sinusoidal period is defined as the steady phase where the controller should have converged to a stable near optimal policy tracking the reference altitude. Therefore the following success condition can be defined in Eq. 40 based on RMS error calculated in Eq. 41 where the number of time steps within steady phase is given by N_{steady}. Success condition threshold S_{thresh} has been set at 20 meters for tight success, 40 meters for loose success, and 100 meters for converging runs. This success condition structuring is a modified version of the tight and loose success condition structuring given in [15] for ACD comparison that allows controller comparison in terms of accuracy and convergence.

$$S_{thresh} > RMSE_{steady} \tag{40}$$

$$RMSE_{steady} = \sqrt{\frac{\sum_{i=0}^{N_{steady}} (h_i^r - h_i)^2}{N_{steady}}} \tag{41}$$

For experiment 3, the success condition has been defined based on the RMSE during final 30 seconds of the cruise phase. The run is considered successful if the calculated RMSE is within 40 meters of the reference altitude.

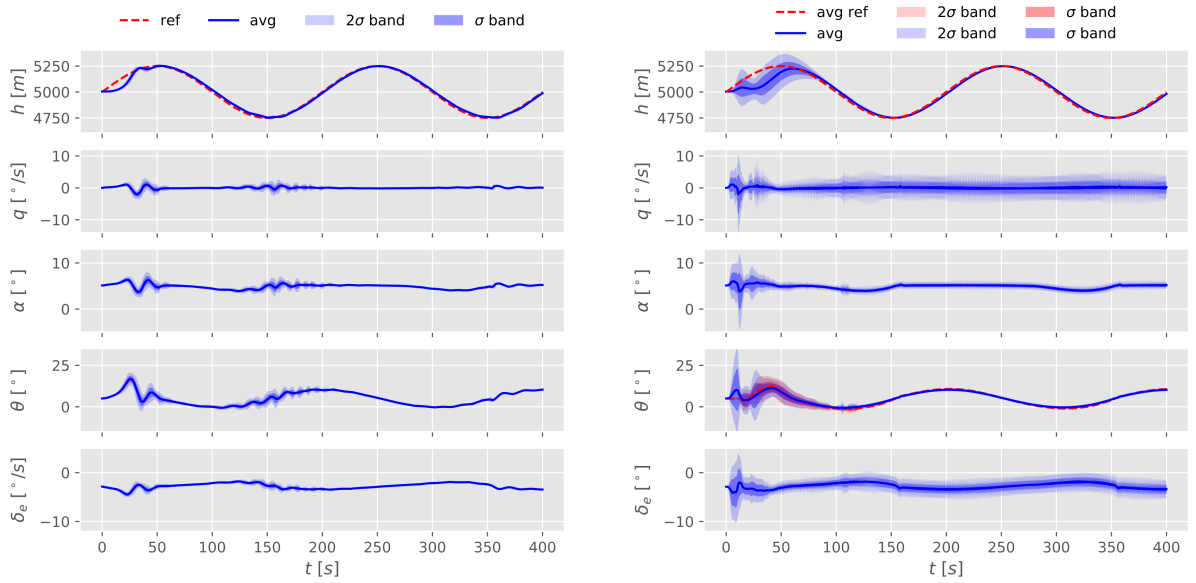
For online application of the IDHP controller, convergence speed to reference is also an important aspect of controller performance to consider. The measure of convergence speed is quantified in the form of rise time and is defined as the elapsed time before simulated Cessna Citation altitude is within 20 meters of the reference altitude.

A. Experiment 1: Comparison of Baseline and Cascaded Controller Performance Under Perfect Conditions

Experiment 1 is set up to compare the baseline and cascaded IDHP agent performance for the altitude tracking task. Tunable parameters have been kept constant among both controller designs. However, due to the fundamentally different actor structure, an additional actor learning rate $\eta_{a,2}$ needs to be defined for the cascaded controller.

The cascaded actor design utilizes an additional state θ for the second actor network. From preliminary analysis of baseline controller tracking task for θ, it has been found that the actor is able to quickly converge to the reference due to faster system dynamic relationship of state θ to δ_e. Additionally, it has been found that a lower learning rate for the inner actor is required to ensure stable learning process as the second actor network governs the the first actor network adaptation. Therefore the learning rates for the cascaded controller are set at: $\eta_{a,1} = 25$, $\eta_{a,2} = 1$, and $\eta_c = 10$. These learning rates have been found through empirical analysis around the reference learning rates obtained from [25]. Learning rates $\eta_a = 25$ and $\eta_c = 10$ have been chosen for the baseline controller in an effort to minimize learning rate difference between the baseline and the cascaded controllers.

Figures 5a and 5b compare the runs from batch simulation with σ and 2σ confidence bands assuming normally distributed states and input over independent runs. The number of runs satisfying converged success condition for each designs is also shown in figure caption.



(a) Baseline controller simulation results, 17/300 runs **(b) Cascaded controller simulation results, 228/300 runs**

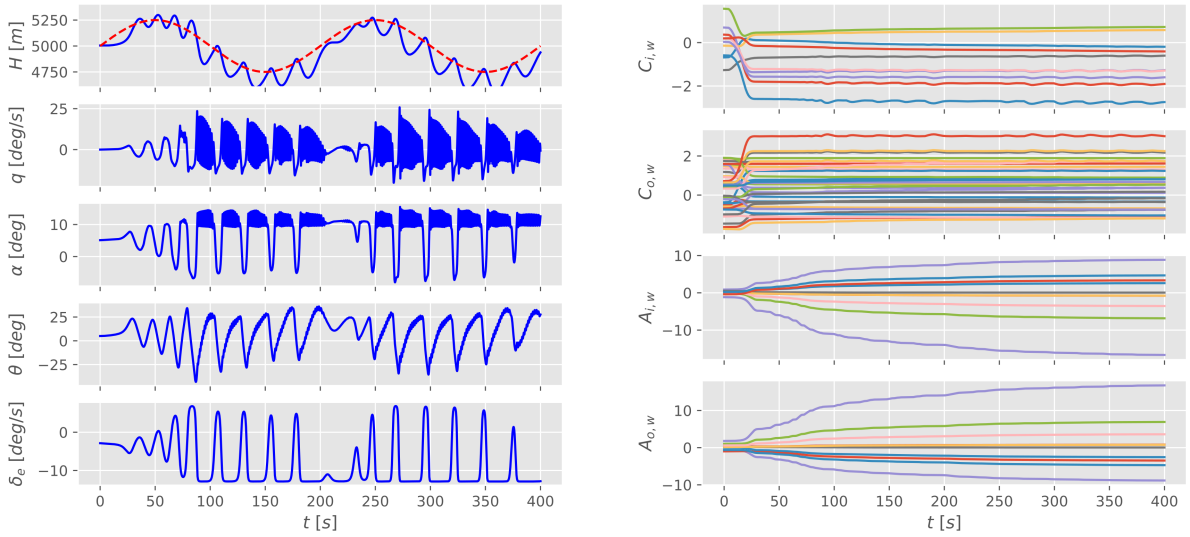
Fig. 5 Comparison of converged runs under perfect conditions

In Figure 5, both controllers show good tracking results with steady phase average RMSE of 10.9 meters for the baseline controller and 12.4 meters for the cascaded controller. One apparent difference between the controllers seen in Figure 5 is that the cascaded controller is slower to converge to the reference. This is due to the existence of a second

actor network in the cascaded controller design. Intermediary pitch angle reference parametrization by the first actor and the tracking there of by the second actor requires additional time.

From Figure 5b it can be seen that the θ reference generated by the outer actor is almost instantaneously tracked by the inner actor network despite having a relatively low learning rate of 1. This can be explained by the fact that a faster angular dynamics response exist between the pitch angle and the elevator deflection when compared to altitude governed by slower translational dynamics. Therefore it can be concluded that among the successful runs, the cascaded controller was able to correctly exploit the expert knowledge provided by the cascaded actor design.

The baseline controller outperforms the cascaded controller in terms of rise time and overall accuracy. However, a significantly lower success ratio suggests that the baseline controller is unable to converge to a near-optimal policy. To investigate the cause of such low success ratio, a representative run is chosen from a set of failed runs and its critic and actor weight changes over time are shown in Figure 6.



(a) Simulation results of a representative failed run

(b) Critic and actor weights of a representative failed run

Fig. 6 Representative failed run of a baseline controller under perfect conditions

From Figure 6a it can be seen that although aggressive, the controller is tracking the overall trend of the reference altitude. Due to the aggressive elevator command generated by the actor, the aircraft reaches high α. At this high angle of attack, stall buffeting can be observed in the form of poorly damped oscillation in pitch rate that in turn manifests itself in α and θ as well. Although undesirable, the pitch rate oscillation is not the cause of the unsuccessful tracking as the increase in control policy aggressiveness can be observed before pitch rate oscillation occurs.

From critic weights progression seen in Figure 6b, no significant changes to λ parametrization have been made after initial adaptations between 0 to approximately 25 seconds. First meaningful actor weight adaptations follow subsequently to the critic weight gradient descent. It is evident that based on the large initial λ estimation provided by the critic and the large magnitude of cost during critic weights adaptation when the reference is yet to be tracked, the actor quickly adapts to an aggressive policy. The resultant overshoot is then aggressively corrected by the error-symmetric policy maintained due to the hyperbolic tangent output activation function of the actor. As it is clear that the problem originates from the fast initial policy iteration, it can be alleviated by slowing down the learning process by decreasing both η_a and η_c.

To demonstrate the effect of decreasing learning rates, two additional 300 independent batch simulations of the baseline controller with $\eta_a = 10$, $\eta_c = 5$ and $\eta_a = 5$, $\eta_c = 2$ have been made. The results obtained are summarized in Table 2 along with previously generated results. It can be seen that, as a direct result of lower learning rates, the steady phase RMSE value increase and rise time increase can be seen across all success conditions. However, an overall increase in success ratio by decreasing learning rates can be seen with the exception of the tight success condition for the

lowest learning rate settings. This is mainly due to the time frame in which the success condition is defined in. Many runs with the lowest learning rate settings have shown continuous adaptation beyond the first reference period resulting in such result. Considering that the success ratio is the most important parameter determining controller reliability, an overall beneficiary effect can be seen by lowering the learning rates.

Two conclusions can be drawn through comparison of the baseline controller design to the cascaded controller design under perfect conditions. The first conclusion is that the cascaded controller design is able to accurately track the altitude reference signal when compared to the baseline controller. The second conclusion is that a longer convergence time can be expected for the cascaded actor design due to an additional state tracking found within the actor network.

Table 2 Experiment 1 batch simulation results under perfect conditions

		Baseline			**Cascaded**
		$\eta_a = 25\ \eta_c = 10$	$\eta_a = 10\ \eta_c = 5$	$\eta_a = 5\ \eta_c = 2$	$\eta_a = 25,\ 1\ \eta_c = 10$
Tight Success	**Success Ratio**	0.06	0.35	0.25	0.76
	Rise Time s	30.1	36.6	44.0	57.7
	S.P. Avg. RMSE m	10.9	14.9	18.0	12.4
Loose Success	**Success Ratio**	0.06	0.35	0.57	0.76
	Rise Time s	30.1	36.6	46.0	57.7
	S.P. Avg. RMSE m	10.9	14.9	19.9	12.4
Converged	**Success Ratio**	0.06	0.41	0.82	0.76
	Rise Time s	30.1	36.2	44.9	57.7
	S.P. Avg. RMSE m	10.9	26.8	42.3	12.4

B. Experiment 2: Controller Selection

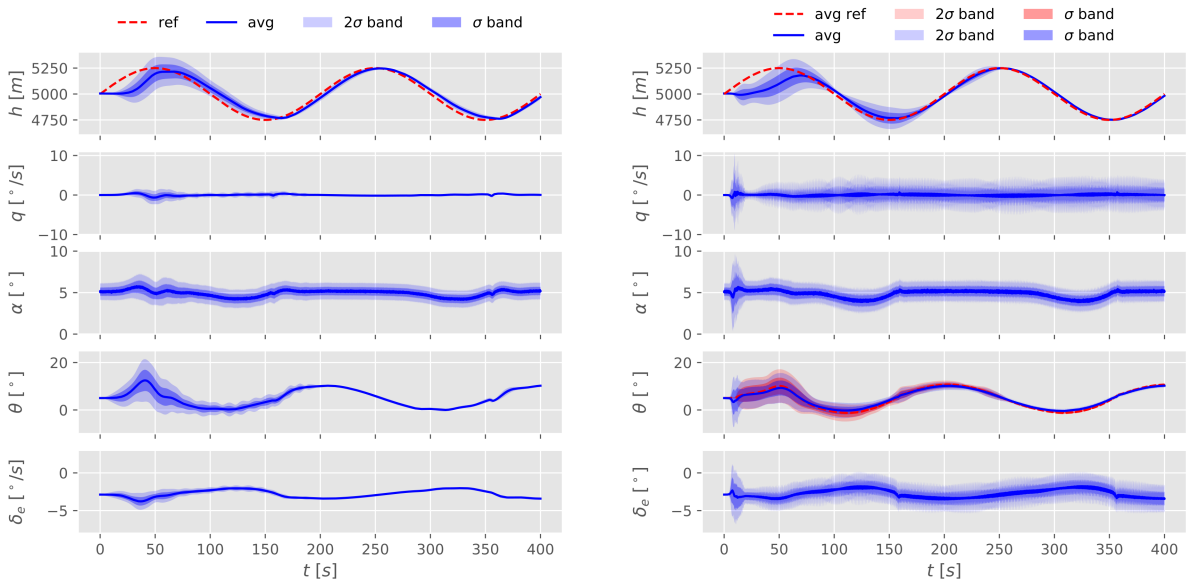
Two main characteristic differences have been observed for the baseline and the cascaded controller designs in experiment 1. As a step towards reducing the gap between simulation and reality, perfect sensor assumption is removed. This experiment has been set up to analyze characteristic differences between the two controller designs under measurement noise.

The baseline controller and the cascaded controller designs are simulated for 300 independent runs with Gaussian measurement noise. Simulation results with confidence bands σ and 2σ are shown for runs satisfying the converged success condition in Figure 7.

For the baseline controller design, comparing Figure 5a from previous experiment to Figure 7a, it can be seen that the addition of noise has a noticeable impact on the steady phase RMSE and rise time for either controller designs. Comparing Figure 5b to 7b for the cascaded controller design, steady phase RMSE is relatively unaffected while the rise time increases with the addition of measurement noise. Therefore it can be concluded that measurement noise negatively affects the overall learning speed of either controller designs but the cascaded controller design shows relative advantage in its ability to approximate a near optimal control policy.

Table 3 is provided to give an overview of the effect of measurement noise to both controller designs. An interesting result is the increase in success ratio seen for the baseline controller. An explanation for this phenomenon can be given by understanding the addition of measurement noise as a cause of slower learning by directly introducing uncertainty to critic, actor, and incremental model input and adaptation terms. From experiment 1, the simulation results of baseline controller with varying learning rates have shown that decreasing learning rates can increase success ratio at the cost of rise time and steady phase average RMSE. Similar effects are observed for the baseline controller by introducing measurement noise with the difference of greater success ratio increase. This greater success ratio increase can be explained by the fact that decreasing learning rate only affects the critic and actor update path while measurement noise affects the critic and actor input as well. The critic is discouraged from quickly converging to λ estimations that may lead to aggressive policy generation due to an overall slower learning caused by state uncertainty. Therefore it can be summarized that including measurement noise at given magnitudes to the simulation had an overall beneficial impact on the baseline controller by decreasing the overall learning process prohibiting aggressive policy generation.

While the introduction of measurement noise has a beneficial impact on the baseline controller success ratio at given learning rates, an overall decrease in success ratio for the cascaded controller design is observed. This is due



(a) Baseline controller simulation results 279/300 runs **(b) Cascaded controller simulation results 153/300 runs**

Fig. 7 Comparison of converged runs under measurement noise

to the relatively complex structure of the actor network seen in cascaded controller design. Not only is an additional state information provided to the actor network with added uncertainty, the overall control policy relies on both actor networks successfully generalizing dynamics of given states and control command. Additionally, aggressive policy generation, which can be alleviated by slower learning, is not an existing failure mode for the cascaded controller design. Therefore the addition of measurement noise negatively impacts the cascaded controller design to a greater degree when compared to the baseline controller design.

Table 3 Experiment 2 batch simulation results under measurement noise

		Baseline $\eta_a = 25, \eta_c = 10$		**Cascaded** $\eta_a = 25, 1, \eta_c = 10$	
		No Noise	**With Noise**	**No Noise**	**With Noise**
Tight Success	**Success Ratio**	0.06	0.00	0.76	0.51
	Rise Time s	30.1	42.3	57.7	69.4
	S.P. Avg. RMSE m	10.9	20.0	12.4	13.1
Loose Success	**Success Ratio**	0.06	0.81	0.76	0.52
	Rise Time s	30.1	56.8	57.7	69.7
	S.P. Avg. RMSE m	10.9	28.5	12.4	13.1
Converged	**Success Ratio**	0.06	0.93	0.76	0.53
	Rise Time s	30.1	60.5	57.7	70.7
	S.P. Avg. RMSE m	10.9	31.7	12.4	14.6

The results of this experiment show that both the baseline controller design and the cascaded controller design are feasible candidates to achieve near optimal control policy for altitude tracking task with varying success ratios. Through experiment 1 and experiment 2, it has been established that the baseline controller is able to achieve faster learning speed while the cascaded controller shows advantage in overall accuracy in the presence of measurement noise. Although the cascaded controller design may achieve higher success ratios through learning rate tuning, this is not

pursued given the high success ratio already achieved by the baseline controller. Additionally, increased susceptibility to measurement noise and unique failure modes of incorrect θ reference generation adds to the disadvantage of the cascaded controller design. For the altitude control task, the baseline controller is a favorable controller design and will thus be chosen for further analysis under atmospheric gusts in experiment 3.

C. Experiment 3: IDHP Controller Performance Under Gust

In addition to measurement noise, atmospheric gusts are also expected during aircraft operation. To demonstrate the baseline controller performance under atmospheric disturbances, batch simulations of 300 independent runs at 4 different flight conditions have been performed under "light" and "moderate" gust scenarios.

The outline of the 4 flight conditions is presented in Table 4. The flight conditions have been chosen in the order of increasing dynamic pressure where aerodynamic damping effects and elevator effectiveness are expected to increase.

Table 4 Description of Flight Conditions used in Experiment 3

Flight Condition	Initial Altitude $[m]$	Initial Airspeed $[m/s]$
FC0	5000	90
FC1	2000	90
FC2	5000	140
FC3	2000	140

Figures 8 and 9 show the average timescale plots of all runs satisfying the success condition of $RMSE < 40m$ for the last 30 seconds of the simulation. The shaded regions represent the σ and 2σ bands among independent runs.

In both Figures 8 and 9, an overall decrease in success ratio can be observed when increasing gust intensities. This is a direct consequence of increased disturbance affecting the learning stability of the designed controller.

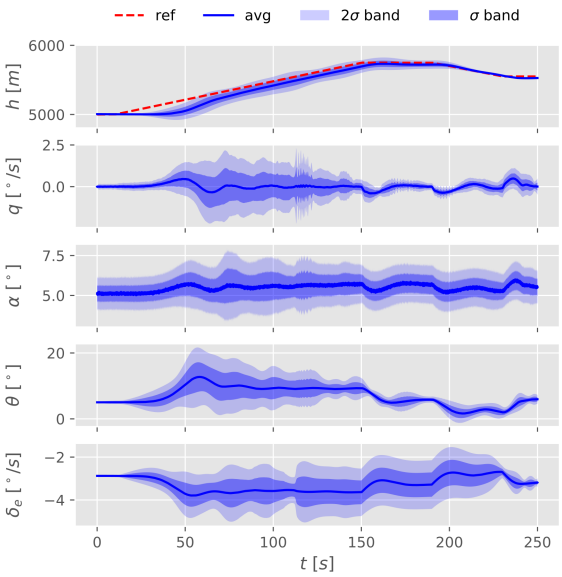
Another point of consideration is that the success ratio for FC2 and FC3 are typically lower than FC0 and FC1. This can be attributed to higher aircraft V_{TAS} for FC2 and FC3. Dynamic pressure is proportional to the square of airspeed. As elevator control effectiveness is proportional to dynamic pressure, elevator control effectiveness is in turn proportional to the square of airspeed. It has been previously established that aggressive control policy is the main failure mode for the baseline controller design. Increased control effectiveness amplifies such problem leading to the overall lower success ratio.

It is demonstrated that a relatively simple altitude error based IDHP design is capable of near optimal control policy generalization in the presence of measurement noise and atmospheric gusts for altitude tracking task. Considering that no prior online training was performed, further success ratio increase can be expected through online training phase design accompanied with learning rate tuning.

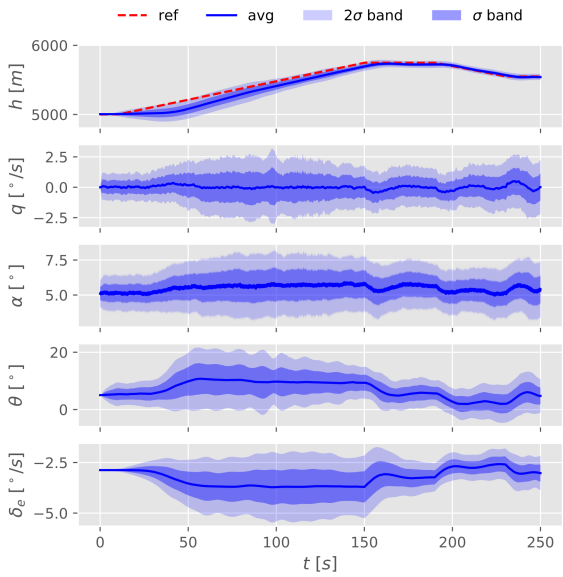
V. Conclusion and Recommendations

Two IDHP based controllers have been designed for the task of altitude reference tracking. The results have shown that the cascaded controller can characteristically generalize a more optimal control policy at the cost of slower learning speeds. Simulation results from experiments 1 and 2 showed the effects of adding measurement noise. The negative effects can be summarized by decreased cost-to-go function estimation performance and slower learning process due to state uncertainty. Such negative effects have impacted the cascaded controller design to a greater extent due to additional state information utilized by the cascaded controller. A beneficial effect of measurement noise on the learning process has also been observed. At certain learning rates, the baseline controller design saw an increase in the overall learning stability as state uncertainty repressed aggressive policy generation. Finally, experiment 3 has shown that the baseline controller is able to learn a near optimal altitude control policy in the presence of atmospheric disturbances and measurement noise. Overall, the findings within this research indicate that a relatively simple IDHP controller provided with only altitude tracking error to the actor and critic network can be sufficient for Cessna Citation aircraft altitude tracking task.

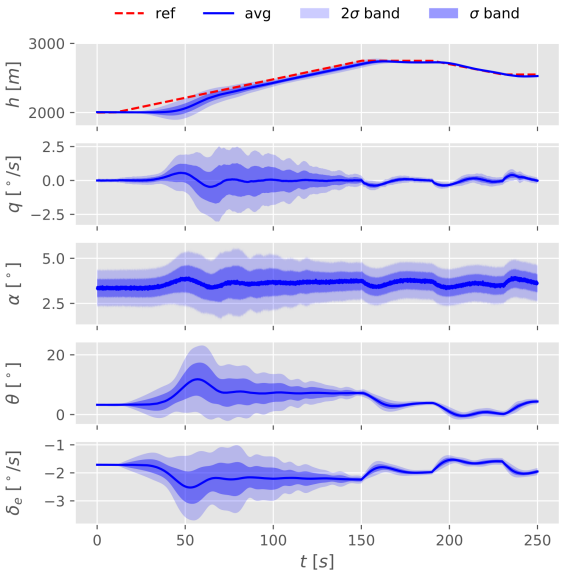
To increase the success ratio of the presented IDHP controller design, further research is needed in terms of in-depth measurement noise sensitivity analysis, critic estimation performance increase, and policy shaping. The first point relates to the fixed measurement noise parameters used for the simulation. Although the causality of success ratio increase has been identified, the exact relationship between success ratio and measurement noise amplitude remains



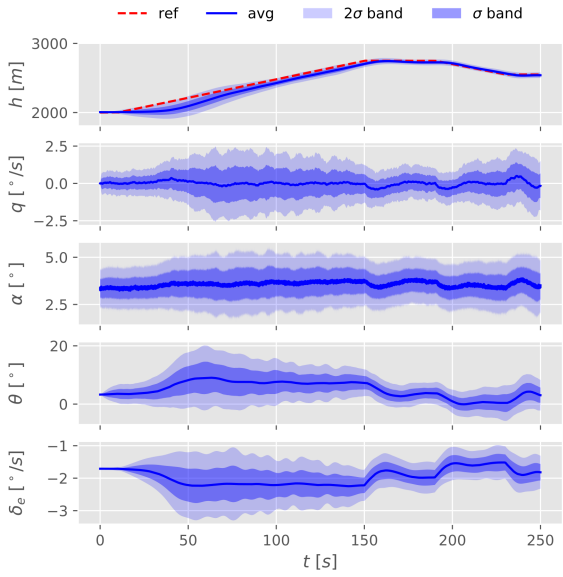
(a) FC0 light gust 283/300 runs



(b) FC0 moderate gust 248/300 runs

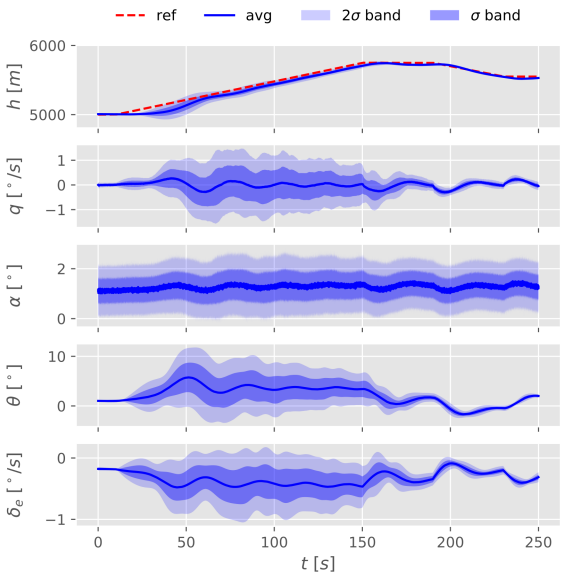


(c) FC1 light gust 270/300 runs

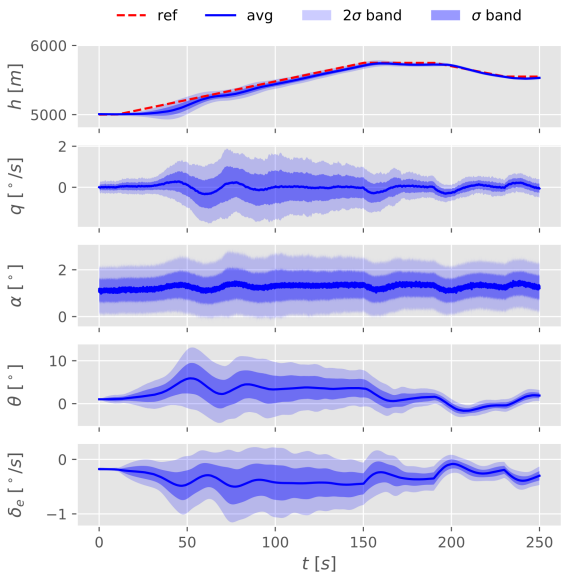


(d) FC1 moderate gust 257/300 runs

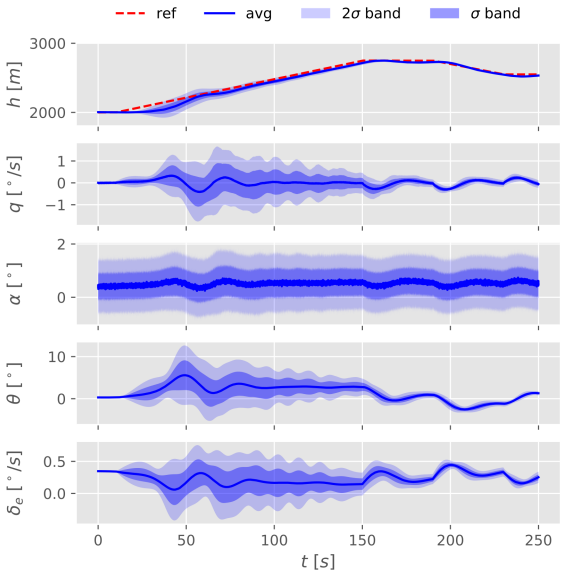
Fig. 8 Simulation results obtained with measurement noise and varying gust conditions for FC0 and FC1



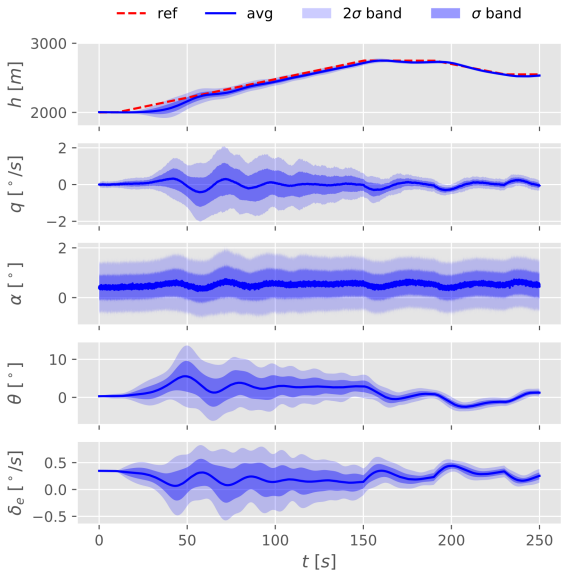
(a) FC2 light gust 226/300 runs



(b) FC2 moderate gust 206/300 runs



(c) FC3 light gust 222/300 runs



(d) FC3 moderate gust 202/300 runs

Fig. 9 Simulation results obtained with measurement noise and varying gust conditions for FC2 and FC3

unknown. Noise sensitivity analysis can help determine to which degree the measurement noise is beneficial to the learning process. The second point of further research is derived from the fact that some failed runs can be attributed to poor critic performance. To help stabilize critic estimation performance, a target critic previously utilized in [28, 35] can be considered. Although it is true that the overall convergence success ratio is largely dependent on critic estimation performance, the overall shape of the policy maintained by the actor ultimately leads to aggressive control policy from which destabilization occurs. This effect is more pronounced when a relatively simple policy is maintained. To alleviate aggressive policy generation, choosing a different activation function with a smoother gradient for the output layer can be considered. For future research, a sigmoidal actor output activation function offset by the trim point is suggested.

References

[1] Teuliere, C., Eck, L., and Marchand, E., "Chasing a moving target from a flying UAV," *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2011, pp. 4929–4934. https://doi.org/10.1109/IROS.2011.6048050.

[2] Enns, D., Bugajski, D., Hendrick, R., and Stein, G., "Dynamic inversion: an evolving methodology for flight control design," *International Journal of Control*, Vol. 59, No. 1, 1994, pp. 71–91. https://doi.org/10.1080/00207179408923070.

[3] Durham, W., Bordignon, K. A., and Beck, R., *Aircraft Control Allocation*, John Wiley & Sons, Ltd, Chichester, UK, 2016. https://doi.org/10.1002/9781118827789.

[4] Brinker, J., and Wise, K., "Nonlinear simulation analysis of a tailless advanced fighter aircraft reconfigurable flight control law," *Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, Virigina, 1999. https://doi.org/10.2514/6.1999-4040.

[5] Doman, D. B., and Ngo, A. D., "Dynamic Inversion-Based Adaptive/Reconfigurable Control of the X-33 on Ascent," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, 2002, pp. 275–284. https://doi.org/10.2514/2.4879.

[6] Bacon, B., and Ostroff, A., "Reconfigurable flight control using nonlinear dynamic inversion with a special accelerometer implementation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, Virigina, 2000. https://doi.org/10.2514/6.2000-4565.

[7] Sieberling, S., Chu, Q. P., and Mulder, J. A., "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 6, 2010, pp. 1732–1742. https://doi.org/10.2514/1.49978.

[8] Simplício, P., Pavel, M. D., van Kampen, E., and Chu, Q. P., "An acceleration measurements-based approach for helicopter nonlinear flight control using incremental nonlinear dynamic inversion," *Control Engineering Practice*, Vol. 21, No. 8, 2013, pp. 1065–1077. https://doi.org/10.1016/j.conengprac.2013.03.009.

[9] Smeur, E. J. J., Chu, Q. P., and de Croon, G. C. H. E., "Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2016, pp. 450–461. https://doi.org/10.2514/1.G001490.

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

[11] Acquatella, P., van Kampen, E. J., and Chu, Q. P., "Incremental backstepping for robust nonlinear flight control," *Proceedings of the EuroGNC 2013, 2nd CEAS Specialist Conference on Guidance, Navigation and Control*, 2013, pp. 1444–1463.

[12] Lewis, F. L., and Vrabie, D., "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Systems Magazine*, Vol. 9, No. 3, 2009, pp. 32–50. https://doi.org/10.1109/MCAS.2009.933854.

[13] Powell, W. B., *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd ed., John Wiley & Sons, Inc., Princeton, 2011.

[14] Werbos, P. J., "Advanced forecasting methods for global crisis warning and models of intelligence," *General Systems Yearbook*, Vol. 22, 1977, pp. 25–38.

[15] Prokhorov, D. V., Santiago, R. A., and Wunsch, D. C., "Adaptive critic designs: A case study for neurocontrol," *Neural Networks*, Vol. 8, No. 9, 1995, pp. 1367–1372. https://doi.org/10.1016/0893-6080(95)00042-9.

[16] Si, J., and Wang, Y. T., "On-line learning control by association and reinforcement," *IEEE Transactions on Neural Networks*, Vol. 12, No. 2, 2001, pp. 264–276. https://doi.org/10.1109/72.914523.

[17] Grondman, I., Busoniu, L., Lopes, G. A. D., and Babuska, R., "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 42, No. 6, 2012, pp. 1291–1307. https://doi.org/10.1109/TSMCC.2012.2218595.

[18] Balakrishnan, S. N., and Biega, V., "Adaptive-critic-based neural networks for aircraft optimal control," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, 1996, pp. 893–898. https://doi.org/10.2514/3.21715.

[19] Han, D., and Balakrishnan, S. N., "State-constrained agile missile control with adaptive-critic-based neural networks," *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 4, 2002, pp. 481–489. https://doi.org/10.1109/TCST.2002.1014669.

[20] Lin, C. K., "Adaptive critic autopilot design of bank-to-turn missiles using fuzzy basis function networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 35, No. 2, 2005, pp. 197–207. https://doi.org/10.1109/TSMCB.2004.842246.

[21] Ferrari, S., and Stengel, R. F., "An adaptive critic global controller," *Proceedings of the American Control Conference*, Vol. 4, 2002, pp. 2665–2670. https://doi.org/10.1109/ACC.2002.1025189.

[22] Ferrari, S., and Stengel, R. F., "Online Adaptive Critic Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 5, 2004, pp. 777–786. https://doi.org/10.2514/1.12597.

[23] van Kampen, E. J., Chu, Q. P., and Mulder, J. A., "Continuous Adaptive Critic Flight Control Aided with Approximated Plant Dynamics," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, Virigina, 2006. https://doi.org/10.2514/6.2006-6429.

[24] Zhou, Y., Van Kampen, E. J., and Chu, Q. P., "Incremental Model Based Heuristic Dynamic Programming for Nonlinear Adaptive Flight Control," *Proceedings of the International Micro Air Vehicles Conference and Competition 2016*, Beijing, China, 2016.

[25] Zhou, Y., van Kampen, E. J., and Chu, Q. P., "Incremental model based online dual heuristic programming for nonlinear adaptive control," *Control Engineering Practice*, Vol. 73, 2018, pp. 13–25. https://doi.org/10.1016/j.conengprac.2017.12.011.

[26] Heyer, S., Kroezen, D., and Van Kampen, E. J., "Online Adaptive Incremental Reinforcement Learning Flight Control for a CS-25 Class Aircraft," *AIAA Scitech 2020 Forum*, American Institute of Aeronautics and Astronautics, 2020.

[27] Enns, R., and Si, J., "Helicopter trimming and tracking control using direct neural dynamic programming," *IEEE transactions on neural networks*, Vol. 14, No. 4, 2003, pp. 929–39. https://doi.org/10.1109/TNN.2003.813839.

[28] Heyer, S., "Reinforcement Learning for Flight Control," Master thesis, Delft University of Technology, 2019.

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

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

[31] *MIL-HDBK- 1797: Flying Qulities of Piloted Aircraft*, U.S. Department of Defense, 1997.

[32] King, M., *Process Control*, John Wiley & Sons, Ltd, Chichester, UK, 2016. https://doi.org/10.1002/9781119157779, URL http://doi.wiley.com/10.1002/9781119157779.

[33] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., John Wiley & Sons, Inc, Hoboken, NJ, USA, 2015. https://doi.org/10.1002/9781119174882.

[34] Ni, Z., He, H., and Wen, J., "Adaptive learning in tracking control based on the dual critic network design," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 24, No. 6, 2013, pp. 913–928. https://doi.org/10.1109/TNNLS.2013.2247627.

[35] Kroezen, D., "Online Reinforcement Learning for Flight Control," Master thesis, Delft University of Technology, 2019.