Explainable Artificial Intelligence Techniques for the Analysis of Reinforcement Learning in Non-Linear Flight Regimes

de Haro Pizarroso, Gabriel; van Kampen, E.

DOI
10.2514/6.2023-2534

Publication date
2023

Document Version
Final published version

Published in
AIAA SciTech Forum 2023

AIAA SciTech Forum
23-27 January 2023, National Harbor, MD & Online
AIAA SCITECH 2023 Forum

10.2514/6.2023-2534

Explainable Artificial Intelligence Techniques for the Analysis of Reinforcement Learning in Non-Linear Flight Regimes

G. de Haro Pizarroso* and E. van Kampen †, *supervisor*
Delft University of Technology, P.O. Box 5058, 2600GB Delft, The Netherlands

Reinforcement Learning is being increasingly applied to flight control tasks, with the objective of developing truly autonomous flying vehicles able to traverse highly variable environments and adapt to unknown situations or possible failures. However, the development of these increasingly complex models and algorithms further reduces our understanding of their inner workings. This can affect the safety and reliability of the algorithms, as it is difficult or even impossible to determine which are their failure characteristics and how they will react in situations never tested before. It is possible to remedy this lack of understading through the development of eXplainable Artifial Intelligence and eXplainable Reinforcement Learning methods like SHapley Additive Explanations. This tool is used to analyze the strategy learnt by an Actor-Critic Incremental Dual Heuristic Programming controller architecture when presented with a pitch rate or roll rate tracking task in non-linear flying conditions, such as at high angles of attack and large sideslip angles. This same controller architecture has been previously explored with the same analysis tool but limited to the nominal linear flight regime, and it was observed that the controller learnt linear control laws, even though its Artificial Neural Networks should be able to approximate any function. Interestingly, it was discovered in this research paper that even in the non-linear flight regime it is still more optimal for this controller architecture to learn quasi-linear control laws, although it seems to continuously modify the linear slope as if it was an extreme case of the gain scheduling technique.

I. Introduction

Interest in the development of civil aerospace autonomous systems has continued to rise in the last few years. Whereas these systems were originally restricted to consumer recreation, modern Remotely Piloted Aircraft Systems (RPAS) have been found to be very reliable in a variety of industries. Some modern applications include the agricultural sector where RPAS are used for various tasks like monitoring crop health [1] and spraying pesticide [2, 3]; as well as different industrial sectors where RPAS can be used to help infrastructre development and maintenance [4, 5]. Moreover, these types of systems are also being considered as the future of urban transportation [6].

Combined with the large selection of possible business applications for RPAS, in the last few years regulators in Europe, the US, and the rest of the world have started to consider seriously the existence of these systems. Thanks to that, regulation is being developed that will propel the development and advancement of these technologies, as regulatory frameworks across the world are created and homogenized and these types of vehicles are included in non-segregated airspaces [7, 8].

However, to enable these new business applications and technologies, it is necessary to have fully autonomous Ground, Navigation and Control (GNC) systems. Otherwise, it is not possible to ensure the required levels of safety and to achieve economic feasibility. An important aspect of any of these GNC autonomous systems would be to be able to adapt to a changing environment [9] and to be sufficiently fault tolerant. In addition, if these systems are also model independent, it could significantly help in their development [10].

A possible set of technologies that could fill the need for fully autonomous GNC systems comes from the domain of Artificial Intelligence (AI). AI has been defined as the ability of an autonomous agent to interpret and learn from external data in an independent manner, in order to achieve a specific outcome through flexible adaptation [11]. Just like RPAS, AI applications have multiplied exponentially in the last few years. Nowadays it is used virtually in every industry, ranging from manufacturing [12], finance [13], social media [14], and even political campaigning [15]. Moreover, as computers continue to advance, more data becomes available thanks to the modern internet, and new algorithms are developed; AI will continue to expand its influence across our modern world. A subset of AI, known as Machine

*Graduate Student, Faculty of Aerospace Engineering, Control and Simulation Department, Delft University of Technology
†Assistant Professor, Faculty of Aerospace Engineering, Control and Simulation Department, Delft University of Technology

Learning (ML) has been one of the most influential fields of study of the last decade, affecting almost every aspect of our society and industries [16–20]. This field can be defined as the programming of machines to learn from external data or experiences, without being explicitly programmed for that task [21]. Usually, ML can be further divided into three main areas: Supervised Learning (SL), Unsupervised Learning (USL) and Reinforcement Learning (RL) [21]. RL is substantially different from SL and USL, as it seeks to replicate a biological learning process based on trial and error [22]. In RL, the learning algorithm is referred to as the agent. This agent is allowed to explore its environment and take decisions along the way. Based on the agent's decisions and the environment's response, a reward is given to the agent. The objective of the RL algorithm is to maximize the agent's reward over time. Through trial and error, the agent should learn which actions to take depending on its state and the environment. Thanks to this learning philosophy, RL is specially suited for real-time applications and decision-making processes such as the control of autonomous vehicles.

Traditional RL methods are only able to handle very reduced action spaces and discrete states. Whereas this can be enough for some applications, the use of RL in autonomous vehicles requires the use of continuous time and highly dimensional action spaces. Normally, this would lead to an exponential growth in computational cost known as the curse of dimensionality [23]. To avoid this problem, it is possible to introduce function approximators into RL. Some of the most powerful function approximators available today are Artificial Neural Networks (ANNs) [24]. Algorithms that combine ANNs with RL are said to belong to the Deep Reinforcement Learning (DRL) category. This type of algorithms have been very active in research in the last years [25, 26]. Initially, they were used to beat humans at practically any game such as Go [27] or even incredibly complex games such as Starcraft II [28]. With time, companies like DeepMind are starting to showcase the real potential of ANNs and RL, as well as many other ML techniques, thanks to developments like AlphaFold, a deep learning system that is able to accurately predict the folding of proteins and could revolutionize the way new medicines are developed [29].

Current flight control systems used in certified vehicles are based on the use of a series of linear controllers. Gain scheduling techniques are then used to switch between these controllers, depending on the operating conditions [30]. These different linear controllers are developed in advanced with the use of a model of the system. However, the lack of adaptability makes these techniques unsuitable for the future of autonomous vehicles. Whereas non-linear control strategies exist, like Nonlinear Dynamic Inversion (NDI) [31–33]; they are highly dependant on the availability of an accurate system model. This can be partially alleviated through the use of incremental models like Incremental Nonlinear Dynamic Inversion (INDI), but at the expense of needing high sample rate measurements [34–38].

On the other hand, DRL systems are able to solve many of these problems and more. In the first place, they are able to provide the required level of adaptability and fault-tolerance to allow navigation and control in complex and continuously changing environments [39]. Moreover, using DRL systems allows to avoid the time-consuming process of engineering the gain scheduling system. Some of these systems are also able to be trained completely online. Thanks to all these characteristics, DRL systems are being increasingly applied for the development of autonomous flight control systems [40–43].

However, all these great advances in DRL techniques come at the expense of a very significant increase in the complexity of the underlying algorithms. This increasing complexity of DRL techniques, or even ML algorithms in general, has a major drawback related with the lack of understanding of the models. Whereas traditional ML methods were sufficiently simple to be able to understand what the algorithm had learnt to do without external aid, modern versions of these algorithms are becoming increasingly complex [44]. For this reason, current ML models are often referred to as black boxes, in the sense that their inner workings are not easily understood [45]. As ML techniques keep on advancing, this problem will grow even worse. In general, this lack of transparency can slow down the progress of ML applications, due to the possibility of unknown interactions in the data that could lead to a variety of problems. However, this is even more acute in the case of flight control, as it would be unwise to certify a vehicle that flies with a control algorithm whose understanding is limited.

For all these reasons, this paper intends to continue previous efforts [46] to develop tools able to increase the transparency of RL techniques in the context of flight control. Thanks to these advances, RL flight control algorithms can be improved in terms of performance, reliability, and safety. Moreover, their teachings can also be used to improve other control techniques as they might shine a light on aspects previously ignored of the flight control process. On top of all this, it is possible to reduce the development time thanks to the better understanding of the models and being able to debug more easily the increasingly complex DRL models.

The remainder of this paper is structured in the following way. Section II will introduce the basis of RL applied to Adaptive Flight Control, the main characteristics of the Incremental Dual Heuristic Programming method selected as algorithm for this research and the eXplainable Artificial Intelligence (XAI) technique that will be applied to study the RL controller. Section III will introduce the flight simulation environment used as training for the RL framework.

Section IV will present the controller training in the non-linear flight regimes of the considered simulation environment, together with the generated explanations with the help of the selected XAI technique. Finally, the conclusions are included in Section V

II. Foundations

This section intends to present the foundational knowledge of the different aspects of this paper, including the application of RL for flight control and the selected XAI technique.

A. Reinforcement Learning for Adaptive Flight Control

Reinforcement Learning seems to present an adequate way to develop many of the necessary adaptive intelligent systems needed for the integration of autonomous flight vehicles. Adaptive control usually deals with what is considered low level tasks, such as following a designated trajectory and providing robustness or adaptability in the presence of failures or disturbances. Adaptive Critic Design (ACD) methods can be used to solve the adaptive control problem. They are based on the actor-critic framework. This allows them to provide the necessary level of adaptability, as well as the possibility to handle non-linear and continuous system dynamics. Moreover, these methods are said to belong to the Approximate Dynamic Programming family, as they use approximators like Artificial Neural Networks (ANNs) to compute the policy (the actor) and the value function (the critic).

There are several possible ACD frameworks, depending on the output of the critic and whether a system model is needed to perform the actor and critic updates. Examples of ACD methods include the Heuristic Dynamic Programming (HDP) and Dual Heuristic Programming (DHP), which have been used for the design of an auto-lander, including the effect of wind shear and stochastic wind gusts [47]. DHP has also been successfully applied to the full flight control problem of a business jet [48]. However, these types of ACD frameworks usually require a priori knowledge of the systems dynamics, either to perform the updates of the critic or the actor, or because the algorithms require a preliminary offline training phase.

Since a priori model information may not be available or it may have very large uncertainties, it is beneficial to avoid this dependence. For this reason, incremental model architectures based on classical adaptive control design were developed. These new methods were Incremental Heuristic Dynamic Programming (IHDP) [49] and Incremental Dual Heuristic Programming (IDHP) [50]. Thanks to these methods it is possible to eliminate the need for a preliminary offline training phase, as well as improve the efficiency of the online learning phase. This paper uses IDHP as the base RL algorithm to which the selected XAI technique will be applied.

As already indicated, the RL framework used in this thesis is based on an actor-critic architecture. On the one hand, the actor represents the policy $\pi(\mathbf{s}, \mathbf{s}^R)$ that is used to select the next action vector $\mathbf{a}_t \in \Re^{n\times 1}$ where n is the number of actions, based on the current state vector $\mathbf{s}_t \in \Re^{m\times 1}$ where m is the number of states, and the reference state vector $\mathbf{s}_t^R \in \Re^{p\times 1}$ where p is the number of reference states; as indicated by Eq. 1. As a function of the current state and the selected action, the system will evolve based on a non-linear discrete transition as indicated by f in Eq. 2. Finally, the agent will receive a reward r_t depending on its current state and the intended reference state. This reward is used to construct the total return at time step t, defined by the state-value function $v(\mathbf{s}, \mathbf{s}^R)$, as indicated in Eq. 3, where γ is the discount factor used to reduce the weight of future rewards and control how far-sighted the actor is.

$$\mathbf{a}_t = \pi(\mathbf{s}_t, \mathbf{s}_t^R) \tag{1}$$

$$\mathbf{s}_{t+1} = f(\mathbf{s}_t, \mathbf{a}_t) \tag{2}$$

$$G_t = v(\mathbf{s}_t, \mathbf{s}_t^R) = \sum_{j=t}^{\infty} \gamma^{j-t} r_j \tag{3}$$

1. Incremental Dual Heuristic Programming

The IDHP framework presented here and used for the development of this paper is based on [51]. The reward that is produced by the environment every time step is given by Eq. 4. It is defined as the negative weighted difference between $\mathbf{s}_t^R$ and $\mathbf{s}_{t+1}$ in order to penalize the system if the achieved state is different from the desired one. The boolean matrix $\mathbf{P} \in \Re^{p\times m}$ is used to select the controlled states from $\mathbf{s}_{t+1}$, whereas the weight matrix $\mathbf{Q} \in \Re^{p\times p}$ is used

to set the relative cost between the different controlled states. Just like DHP, IDHP does not use the scalar reward itself for updating the ANNs weights, it employs the derivative of the reward function with respect to the state vector $\frac{\delta r_{t+1}}{\delta \mathbf{s}_{t+1}} \in \Re^{1\times m}$, as given in Eq. 5.

$$r_{t+1} = r(\mathbf{s}_t^R, \mathbf{s}_{t+1}) = -[\mathbf{P}\cdot\mathbf{s}_{t+1} - \mathbf{s}_t^R]^T\mathbf{Q}[\mathbf{P}\cdot\mathbf{s}_{t+1} - \mathbf{s}_t^R] \tag{4}$$

$$\frac{\delta r_{t+1}}{\delta \mathbf{s}_{t+1}} = -2[\mathbf{P}\cdot\mathbf{s}_{t+1} - \mathbf{s}_t^R]^T\mathbf{QP} \tag{5}$$

A schematic representation of the feed-forward signal flow of the IDHP framework is included in Fig. 1. In this figure it is possible to observe the three main modules: the critic which outputs the derivative of the state-value function with respect to the state $\hat{\lambda}_t = \frac{\delta v(\mathbf{s}_t, \mathbf{s}_t^R)}{\delta \mathbf{s}_t}$, the actor which outputs the action $\mathbf{a}_t$ as indicated by the policy $\hat{\pi}(\mathbf{s}, \mathbf{s}^R)$, and the incremental model which outputs the estimations of the system matrix $\hat{\mathbf{F}}_{t-1}$ and of the control effectiveness matrix $\hat{\mathbf{G}}_{t-1}$ which are defined in Section II.B.2. The incremental model uses Recursive Least Squares to approximate the unknown dynamics of the system, whereas the actor and the critic use ANNs to approximate the optimal control policy and the derivative of the value function with respect to the state, respectively. The $\hat{}$ is used to indicate that the outputs of these models are estimates or parametric approximations. Just this combination of modules is enough to obscure the inner workings of the algorithms and complicate the understanding of the controller. However, the selected XAI technique will only be applied to the actor module in order to extract information equivalent to the control laws by analyzing the policy $\pi(\mathbf{s}, \mathbf{s}^R)$. Note that the feed-forward diagram also includes a target critic module with an output denoted by an apostrophe λ', however this capability was not used for this research paper and the mixing factor τ was always left equal to 1, such that the target critic is always equal to the critic.

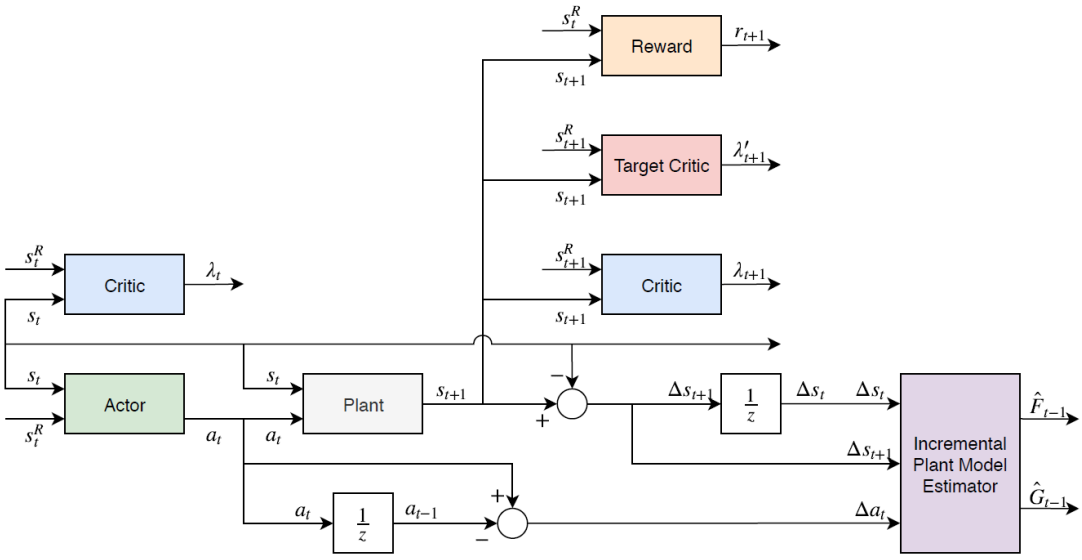


Fig. 1 Schematic of the feed-forward signal of the learning framework, extracted from [51]. The $1/z$ blocks represent one time step delays

B. Learning Framework

1. Critic and Actor Neural Networks

Both the actor and the critic use a similar ANN. They both use a single hidden layer fully connected Multilayer Perceptron. They both have the same input, which includes the current state vector and the reference state tracking error. Moreover, both have their single hidden layer made up of 10 fully connected neurons with a hyperbolic tangent activation function. The main difference is in the output layer, as the actor uses a scaled hyperbolic tangent function

whereas the critic utilizes a linear output. The actor's output is scaled according to the saturation limit of the control surfaces of the aircraft model. If the limits are asymmetric, then the largest absolute limit is used as scaling constant. Finally, the ANNs do not use neuron bias as the trim values of the aircraft are known to the controller.

2. Incremental Model

The use of an incremental model is the main advantage of IDHP as compared with other ACD frameworks. As long as measurements have a sufficiently high frequency, and the state evolves relatively slow, the incremental model should be able to approximate the non-linear unknown system dynamics [35, 39]. The main equation for the incremental model can be derived from the original non-linear dynamics

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

Finally, the previous equation can be written in incremental form

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

where $\Delta\mathbf{s}_{t+1} = \mathbf{s}_{t+1} - \mathbf{s}_t$ is the state increment, $\mathbf{F}_{t-1} = \mathbf{F}(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})$ is the system matrix, $\mathbf{G}_{t-1} = \mathbf{G}(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})$ is the control effectiveness matrix, and $\Delta\mathbf{a}_t = \mathbf{a}_t - \mathbf{a}_{t-1}$ is the control increment. The update rules for the incremental model are derived in [52].

C. Explainable Artificial Intelligence

Artificial Intelligence and Machine Learning algorithms have exploded in terms of new applications in the last decades. However, this has been at the expense of increasing the complexity of these models. Whereas the capabilities of the new much more complex algorithms are necessary to achieve accomplishments such as the control of fully autonomous flying vehicles, tools must be developed to compensate for the increased complexity and empower the different audiences to understand and interpret what these black-box algorithms are doing.

The most important terms that must be defined for this purpose are [53]:

- Interpretability: It is defined as the ability to explain or to provide meaning in understandable terms to a human
- Explainability: Explainability is associated with the notion of explanation as an interface between humans and a decision maker that is, at the same time, both an accurate proxy of the decision maker and comprehensible to humans
- Transparency: A model is considered to be transparent if by itself it is understandable.

Moreover, it is necessary since the beginning to define which target audience are the explanations intended for, as this will determine what is understandable or not [53]. For reference, this paper will produce explanations intended for RL researchers and control design engineers.

The XAI techniques can be grouped in two categories: transparent models and post-hoc explainability [53]. Transparent models are not necessarily the same as an interpretable AI algorithm. An interpretable AI algorithm is inherently understandable, such as a linear regression model or a decision tree algorithm. On the other hand, transparent models are all those that include some method of interpretability in their design such as reward decomposition [54]. Reward decomposition separates the single numerical reward signal into different components, and computes the value function for each of these reward components. Through an analysis of the components of the value function and reward signal it is possible to gather knowledge about the algorithm's inner workings. Finally, post-hoc techniques include methods like text explanations and visualizations, together with feature relevance [53] which is the method of interest for this paper. The main advantage of post-hoc techniques is that they do not need to modify the ML algorithm. As such, they can be applied to most models directly.

1. SHapley Additive exPlanations (SHAP)

The chosen XAI technique for this research is SHAP analysis [55]. It is a post-hoc technique based on the Additive Feature Attribution Method which generates a simple surrogate model to explain the workings of the desired algorithm. As it is a post-hoc technique, it does not require any modifications of the RL algorithm and it can be applied directly to the trained model. Compared with other feature relevance techniques such as LIME [56], SHAP is able to provide more accurate explanations, although at the expense of a higher computational cost. In spite of this, since the target audience for these explanations are RL researches and control engineers, the increased computational cost is less important than the improved accuracy.

SHAP analysis is inspired by Shapley values from cooperative game theory and it employs a simple surrogate model approximation

$$f(x) \approx g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x_i' \tag{8}$$

where $f(x)$ is the original model, $g(x')$ is the approximate explanation model, and x_i' is the simplified binary input found using the mapping $x = h_x(x')$. The variable ϕ_0 is known as the base value and it represents the mean output value, as this would be the output of the model without any knowledge of the features. Finally, the variables ϕ_i are known as SHAP values and they represent the marginal contributions of each feature to the overall model output.

To apply SHAP analysis to the actor's ANN in this paper, the algorithm DeepSHAP [57] is used, which is based on a combination of SHAP and DeepLIFT [58]. In this algorithm, the mapping $x = h(x')$ simply represents whether a feature is included in the data set. As this research assumes that the states are constantly observable, all features will be assumed to be known and hence $x_i' = 1\ \forall i$.

III. Flight Simulation

This research paper uses a flight control framework adapted from [51]. This framework utilizes a high-fidelity Cessna Citation II aircraft simulation model as the environment, that is controlled by an IDHP model as already indicated.

A. Cessna Citation II Simulation Model

The environment for the flight control framework is composed of a high-fidelity, six degrees-of-freedom, non-linear model of the Cessna Citation 500, developed by Delft University of Technology. The model includes engine dynamics, as well as potentially actuator dynamics and sensor models. The state space and action space are included in Eqs. 9 and 10, respectively.

$$\mathbf{s}^T = [p,\ q,\ r,\ V_{TAS},\ \alpha,\ \beta,\ \phi,\ \theta,\ H] \tag{9}$$

$$\mathbf{a}^T = [\delta_e,\ \delta_a,\ \delta_r] \tag{10}$$

The elements of the state space include: p, q, and r as the roll rate, the pitch rate, and the yaw rate, respectively; V_{TAS} as the true airspeed; α, and β as the angle of attack, and sideslip angle; θ, and ϕ as the pitch angle, and the roll angle; and finally H as the aircraft's altitude. While the reference signals are not part of the simulation model, they follow a similar notation being q^R, p^R, and β^R the reference pitch rate, roll rate, and sideslip angle, respectively.The elements of the action space are the elevator deflection δ_e, the aileron deflection δ_a, and the rudder deflection δ_r.

Together with the IDHP flight controller architecture, the simulation model includes a built-in auto-throttle to control the airspeed. The model also included an additional yaw-damper, but it was disabled to let the IDHP controller have full control of all the aerodynamic surfaces. Moreover, the states are assumed to be perfectly observable and all the sensor models were deactivated. The control surface deflection limits are included in Table 1, and these are enforced inside the simulation model by trimming the controller inputs if necessary.

Table 1 Control surface deflection limits

Control surface	Saturation limits
Elevator (δ_e)	[-20.05, 14.90] deg
Aileron (δ_a)	[-37.24, 37.24] deg
Rudder (δ_r)	[-21.77, 21.77] deg

B. Flight Controller

The flight controller adapted from [51] is included in Fig. 2. This architecture controls the longitudinal and lateral degrees of freedom separately. A series of PID controllers are used to produce the desired reference angular rates. These are then used as input for the actual actor-critic IDHP longitudinal and lateral controllers. The selection matrices

from Eq. 4 are defined to select the relevant controlled states for each sub-controller. Additionally, the weight matrix $\mathbf{Q}$ for the longitudinal controller is simply 1, as it only considers the pitch rate error in the reward; whereas the weight matrix for the lateral controller is $\mathbf{Q} = diag(1, 100)$ to compensate for the difference in scale between the roll rate error $p - p^R$ and the sideslip error $\beta - \beta^R$, being the latter usually smaller.

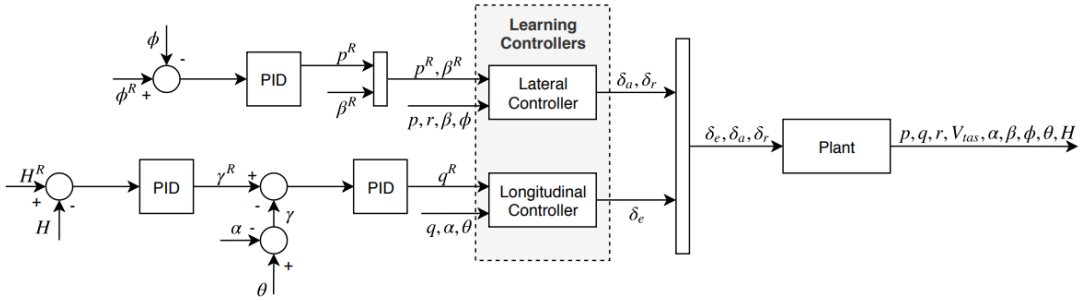


Fig. 2 Flight controller architecture, extracted from [51]

As observed, this controller architecture assumes no significant coupling between the longitudinal and lateral controller. This approximation may not hold in the non-linear flight regimes which are to be explored in this research paper. This suggests that further investigation could try to analyze fully coupled models, like the one based in the Soft Actor Critic (SAC) algorithm developed in [59]. Regardless, whereas longitudinal-lateral coupling effects will not be investigated, many other non-linear maneuvers are available for study.

Finally, the hyperparameters of the lateral and longitudinal controllers can be adapted to help the agents train in the different maneuvers. Since the IDHP controller will be tested in non-linear flying conditions, the learning rates will have to be adjusted to achieve training without divergence. Some parameters that were maintained constant for all maneuvers were the discount factor $\gamma = 0.8$, and the initialization parameters of the actor's and critic's ANNs, which both used a normal distribution $N(\mu = 0, \sigma = 0.01)$. Finally, the maneuvers were always carried out with the same time step, $\Delta t = 0.02$ s.

IV. Results and Discussion

This section presents the main results of this research. Firstly, a series of maneuvers are selected with the objective of studying the performance of the IDHP controller and the analysis capability of the SHAP technique in a non-linear flying regime. Afterwards, the controllers are trained for these maneuvers and a segment of time is chosen to be used in the SHAP analysis. This is important as SHAP assumes the weights of the neural network are fixed, however the IDHP controller is made to continuously adapt. For this reason, it is necessary to select a segment of time with converged, or at least slowly changing, weights to be suitable for the SHAP analysis. Finally, the results are verified by reconstructing the control laws predicted by the SHAP analysis. This final step also helps in analysing the strengths and weaknesses of this XAI technique.

A. Maneuver selection

In previous research [46], it was already shown that the IDHP controller learns linear control laws when trained to fly in a linear flight regime. However, this does not mean that an IDHP controller can be directly substituted by a gain scheduled family of linear controllers. It could simply showcase that the IDHP controller was able to identify the linearity of the dynamics in the considered flight regime. Therefore, it is of interest to study the performance of this architecture in a non-linear flight condition.

Previous analysis of this controller architecture [46, 51] had considered sinusoidal pitch rate and roll rate tracking tasks to initially train the IDHP. This paper considers the same simple tasks to train the controller, although the trim conditions are modified to force the airplane into a non-linear regime. For the longitudinal maneuvers, from the simulator aerodynamic models it was identified that the static stall angle of attack was between 10 and 11 degrees. At the selected flying altitude of 2000 m, this corresponded to a stall velocity close to 55 m/s. However, since the longitudinal maneuvers use a pitch rate tracking task to train the agent, a slightly higher velocity between 56 and 58 m/s was selected. On the other hand, the lateral roll rate tracking task was modified by trimming the aircraft with a large

reference sideslip angle. This can induce a non-linear coupling of the lateral and directional dynamics. All in all, the three considered maneuvers are summarised in Table 2.

Table 2 Selected non-linear maneuvers, all performed at a trim altitude of 2000 m

Maneuver	Amplitude [deg/s]	Frequency [Hz]	Trim velocity [m/s]	Reference sideslip angle [deg]
Pitch rate tracking	5	0.2	58	0
Pitch rate tracking	5	0.05	56	0
Roll rate tracking	5	0.2	90	13

B. Training and segment selection

Similar to [51], the longitudinal and lateral controllers are trained separately to allow the weights and the incremental model to converge properly. Given the complexity of the dynamics, whereas it took more or less 60 seconds to train the controllers in the linear flight regime [51], in this case it required between 500 and 1000 seconds of simulation. Moreover, the hyperparameters had to be significantly modified to allow the model to train reliably. The learning rates of the critic η_C had to be reduced for all the maneuvers, from an original value of 10 down to just 1 or 2. The learning rates of the actor η_A also had to be reduced in general down to 1 or 2, except for the case of the pitch rate tracking at 0.05 Hz in which the learning rate of the actor could be left at its original value of 5. Finally, a damped sinusoidal excitation signal was applied to the elevator in the longitudinal maneuvers and to the ailerons in the lateral ones to promote exploration, just like it was done in the original controller implementation [51]. However, the excitation amplitude A_0 was doubled for the longitudinal maneuvers to account for the larger range of required elevator movement. The hyperparameters are summarized in Table 3, noting that the mixing factor τ of the target critic is always equal to 1 as this feature of the architecture was not used for this research.

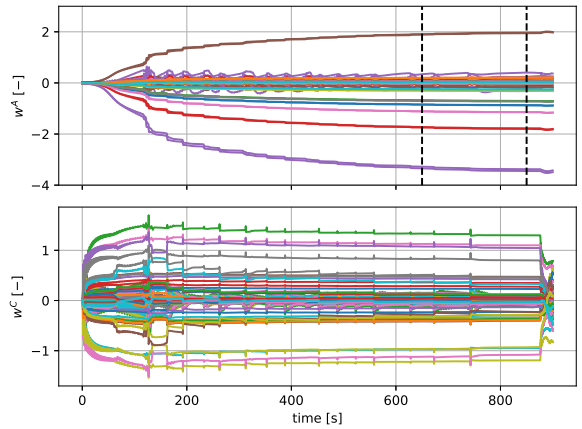
In general, the maneuvers could be trained reliably, although they failed in some cases. Moreover, the pitch rate tracking at 0.05 Hz proved to be the maneuver most difficult to train, with a failure rate of approximately 50%. Whereas it may seem like decreasing the actor's learning rate could have improved this failure rate, this was tested and no significant reliability improvement was observed. Figure 3 shows the convergence of the weights for each of the three maneuvers. Even though it takes a significant amount of time, the weights of the actor's neural network do seem to converge eventually. In spite of this, a small periodic variation of some weights can be observed in some cases as in Fig. 3b.

Table 3 Training hyperparameters for the two longitudinal and single lateral maneuvers

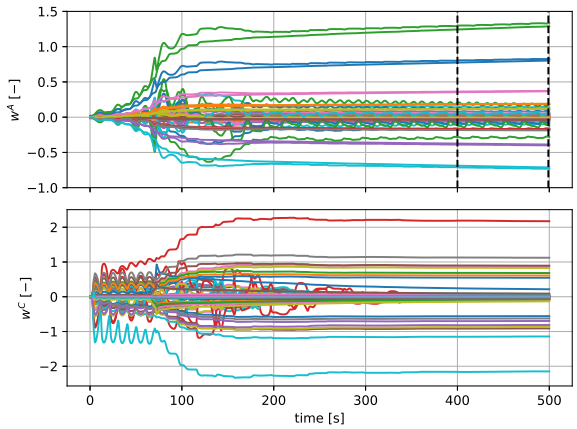
Maneuver	η_A	η_C	A_0 [deg]	τ
Pitch rate tracking at 0.2 Hz	2	2	1	1
Pitch rate tracking at 0.05 Hz	5	1.5	1	1
Roll rate tracking at 0.2 Hz	1	1	0.5	1

Based on Fig. 3, it is possible to select a segment of time for each of the three maneuvers to analyse with the SHAP analysis. Since the weights should ideally be constant, or at least slowly varying, the segments were selected ensuring that the variation in the weights, defined as the difference between the maximum and minimum of each weight in the segment, was less than 10% of the maximum absolute weight magnitude in that same segment. While this selection criteria is rather arbitrary, the results seemed to be satisfactory. With this weight variation criteria, the time segments delimited in Fig. 3 were chosen*. Moreover, whereas the actor's weights are mostly converged in all the maneuvers, the critic's weights of Fig. 3a have a series of spikes. These are related with the stall and recovery events that are observed in the evolution of the states in Fig. 4a, and showcase that in those moments the IDHP architecture is changing its value function quite suddenly. Additionally, the last seconds of Fig. 3a show a very significant change in the critic's weights which is caused by the divergence of the controller.

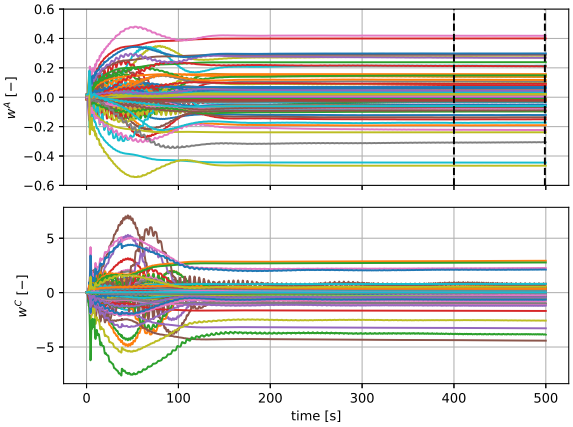
*The weights' variation were 7.8% for the pitch rate tracking at 0.2 Hz, 7.2% for the pitch rate tracking at 0.05 Hz, and 1.9% for the roll rate tracking wih β^R = 13 deg



(a) Longitudinal pitch rate tracking at 0.2 Hz near the stall regime



(b) Longitudinal pitch rate tracking at 0.05 Hz near the stall regime

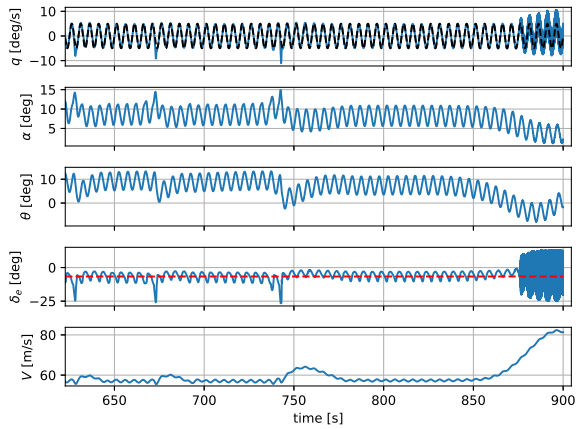


(c) Lateral roll rate tracking with β^R = 13 deg

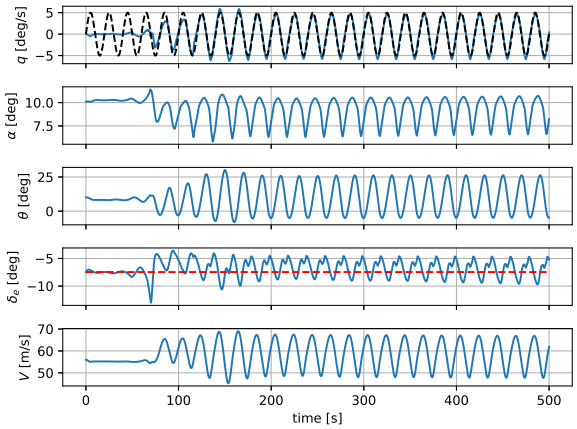
Fig. 3 Evolution of the weights for the three selected maneuvers where $\mathbf{w}^A$ and $\mathbf{w}^C$ are the actor's and the critic's weights, respectively

Some interesting characteristics can be observed from the evolution of the states during training in Fig. 4. Firstly, only a section of the full training in shown for the pitch rate tracking at 0.2 Hz since due to the large frequency and long training times, it was not possible to appreciate the results. Secondly, Fig. 4a shows that at some points during the training, the aircraft actually entered into a stall but it was able to recover. Whereas it is possible the behaviour could be improved thanks to the IDHP controller, it was observed that a simple PD controller on the pitch rate was able to achieve a very similar recovery in case of stall. Hence, these recovery events are probably thanks to the natural stability of the aircraft rather than any controller. Additionally, the last 25 seconds show a significant worsening in the behaviour of the controller and the appearance of chatter in the elevator deflection. This is caused by the divergence of the controller, as pointed out in the discussion of the critic's weights in Fig. 3a. This divergence was simply avoided when selecting the time segment for analysis, but it could probably be eliminated if instead of using constant learning rates, these were reduced dynamically whenever the controller starts to converge on a working flight control policy. In Fig. 4b, it can be observed that the elevator deflection seems to have a double frequency component, indicating the possible existence of a non-linear control law. Finally, the sideslip maneuver in Fig. 4c shows a very good performance and it does not seem to present nothing too dissimilar to what was observed in previous studies.

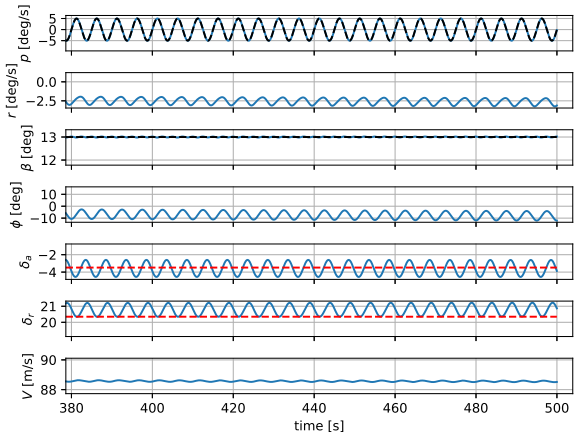
It was attempted to increase the sideslip angle of the roll rate tracking maneuver further. However, it was found that this maneuver at a larger sideslip angle required a larger rudder deflection than the maximum limit of $\delta_{r,max}$ = 21.77 deg. The maneuver was tested at β^R = 14 deg, which has a trim rudder deflection of 21.89 deg. This value is barely



(a) Longitudinal pitch rate tracking at 0.2 Hz near the stall regime



(b) Longitudinal pitch rate tracking at 0.05 Hz near the stall regime



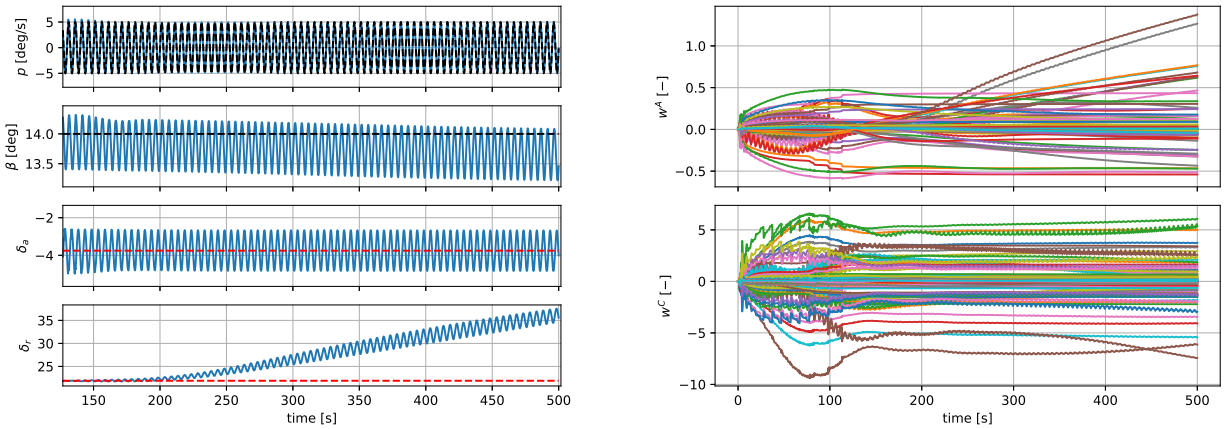
(c) Lateral roll rate tracking with β^R = 13 deg

Fig. 4 Evolution of the main state and control variables for the three selected maneuvers. The black dashed line indicates the reference signals, whereas the red dashed line indicates the trim value for the different control surfaces

over the limit of the available rudder deflection. In spite of this, the controller was trained for this reference sideslip angle producing the results shown in Fig. 5. Whereas the roll rate tracking performance is decent, the sideslip angle error never decreases and stays oscillatory for the entire duration of the simulation. This occurs because of the inability of the controller to command a larger rudder deflection. In fact, the attempts by the controller to correct the observed sideslip error are apparent, as the rudder deflection continuously increases in magnitude, even though the command is then cut-off at the deflection limit by the environment model. In Fig. 5b, it can be observed how a few of the weights also tend to continuously increase. This indicates that these weights are probably directly correlated with the rudder deflection magnitude. Since the controller was not able to achieve a decent level of performance, as it failed to control the sideslip angle, it was decided to stay at $\beta^R = 13$ deg. This reference sideslip angle was deemed high enough to observe any existing non-linear coupling effects, while ensuring a minimum level of performance.

C. SHAP analysis

After training the agents and selecting the time segments for analysis, it is possible to evaluate the learnt control laws by applying the SHAP technique to the actor's neural network. This will directly return the SHAP values of each input feature, representing the marginal contribution of each of them to the overall control surface deflection. In addition, since for this research $x_i' = 1 \; \forall i$, the SHAP values ϕ_i as a function of the input features can be considered directly as the

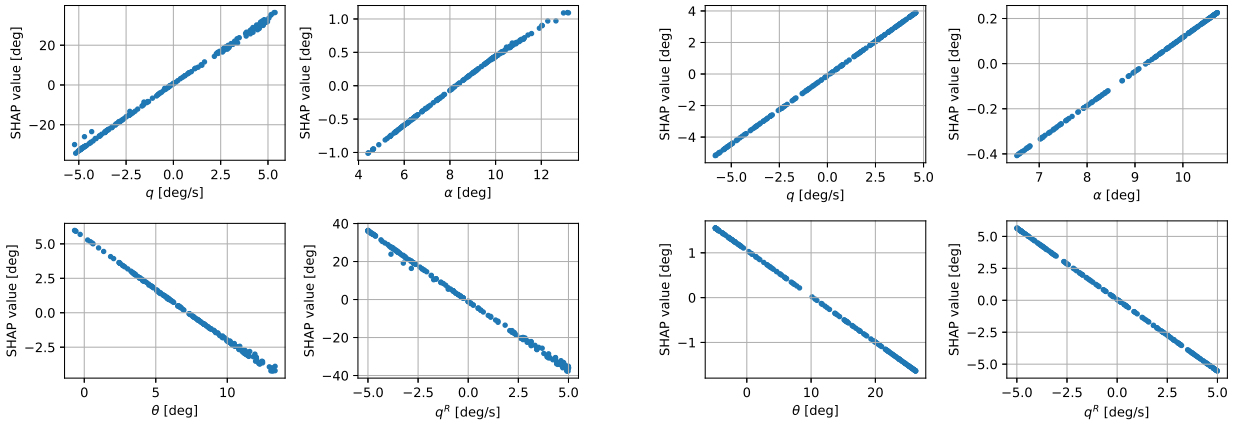


(a) Evolution of some selected lateral states and controls

(b) Evolution of the controller's weights

Fig. 5 Lateral roll rate tracking maneuver with β^R = 14 deg. The black dashed line indicates the reference signals, whereas the red dashed line indicates the trim value for the different control surfaces

equivalent control law of the IDHP architecture. There is no general consensus on the necessary number of background samples to achieve sufficient global accuracy, but based on previous research [46], it was considered that 200 random samples for each segment of time would provide enough resolution. Moreover, the weights at the mid point of the segment would be considered as representative and used for the SHAP analysis. With all these details in mind, the final results are shown in Figs. 6 and 7.



(a) Longitudinal pitch rate tracking at 0.2 Hz near the stall regime

(b) Longitudinal pitch rate tracking at 0.05 Hz near the stall regime

Fig. 6 SHAP analysis of the longitudinal pitch rate tracking maneuvers with a single time segment. Note that the SHAP values axes of each feature have a different scale

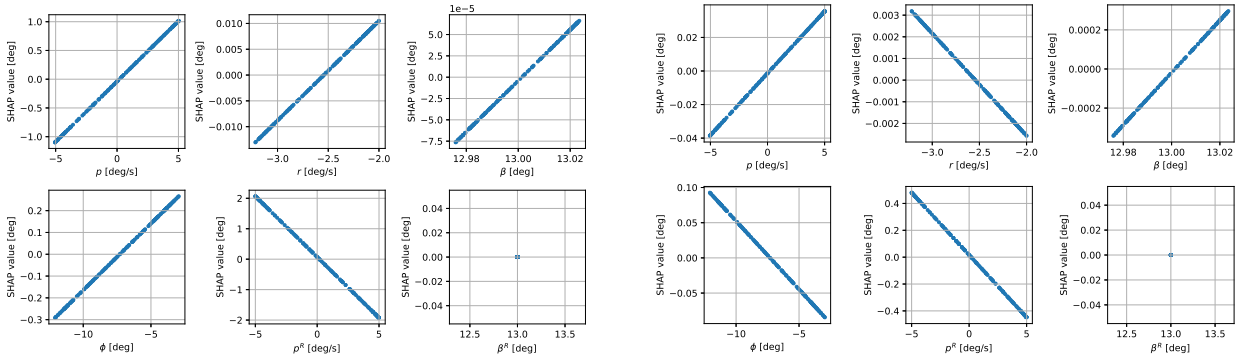
According to the SHAP analysis in Fig. 6, the following qualitative relations exist between the longitudinal features and the final elevator deflection for the longitudinal maneuvers:

- The pitch rate and the elevator deflection are related such that a positive pitch rate induces a positive elevator deflection and viceversa. This behaviour implies that for more positive pitch rates, the controller wants to pitch down the aircraft, and it does so by using more positive elevator deflections. This behaviour is stabilizing, and it makes sense as the controller is trying to anticipate the fact that the pitch rate should not increase beyond the necessary value.
- The relation between the angle of attack and the elevator deflection is similar to the pitch rate, in the sense that a

more positive angle of attack leads to a more positive elevator deflection. In this case, the controller is again showcasing a stabilizing behaviour as it is trying to reduce the angle of attack by pitching down the aircraft when it is close to the maximum.

- The elevator deflection and the pitch angle are related in an opposite way to the pitch rate and the angle of attack. For more positive pitch angles, the controller tends to decrease the elevator deflection and pitch up the aircraft. This behaviour seems to be de-stabilizing, but it might be the result of the phase lag between the pitch rate and the pitch angle or because the controller is using the pitch angle to try to compensate for the natural aerodynamic damping of the aircraft.
- Finally, the reference pitch rate has a relation similar to the pitch angle. The more positive reference pitch rate, the more negative the elevator deflection. This apparently unstable behaviour seems to try to reduce the error between the reference signal and the actual pitch rate. As the reference signal increases in magnitude, the elevator deflection decreases to make the aircraft follow the reference signal, while it uses the actual pitch rate to damp the motion and know when to stop deflecting the elevator upwards.

The same qualitative behaviour is observed in both longitudinal pitch rate tracking maneuvers. Even though the behaviour is qualitatively the same, it does not mean that the actual control law is equal in both cases. In fact, it can be observed in Fig. 6 that the SHAP values of each maneuver are very different from each other. This suggests that the underlying dynamics of both maneuvers are similar in nature, even if the final required deflection is very different in each of them. This was to be expected as both maneuvers are pitch rate tracking tasks with the same amplitude and at a similar trim velocity, although with a different frequency.



(a) Aileron input-output mapping **(b) Rudder input-output mapping**

Fig. 7 SHAP analysis of the lateral roll rate tracking maneuver with β^R = 13 deg with a single time segment. Note that the SHAP values axes of each feature have a different scale

According to the SHAP analysis of the lateral roll rate tracking maneuver in Fig. 7, the predicted qualitative relation between the lateral features and the aileron deflection includes:

- Just like in the case of the pitch rate for the longitudinal maneuvers, a more positive roll rate is associated with a more positive aileron deflection. This behaviour is stabilizing and allows the controller to damp the motion when close to the desired maximum.
- The relation between the yaw rate and the aileron deflection is also the same as the roll rate. The more positive the yaw rate, the more positive the aileron deflection. This effect is more complex as the yaw rate and the aileron deflection are connected through the increased drag of the semi-wing with the positive aileron deflection. A more positive aileron deflection implies a more positive yaw rate, as the right semi-wing tends to generate more lift and thus more drag, which in turn leads to a positive yaw rate. Therefore, it seems like this relation between the yaw rate and aileron deflection is a de-stabilizing one. However, the effect of the increased aerodynamic velocity in the opposite semi-wing due to the yaw rate, could alter this relation. The yaw rate may be used by the controller to counteract some damping effect or to control the dutch roll coupling. Regardless, the SHAP values are very small, which suggests that the effect of the yaw rate in the aileron deflection is small.
- The effect of the sideslip angle is to increase the aileron deflection for more positive angles. This relation seems to be stabilizing since a more positive aileron deflection would generate a positive yaw rate and thus correct the sideslip angle error. However, like in the case of the yaw rate, the SHAP values are orders of magnitude smaller

than the roll rate, which imply the sideslip angle barely influences the aileron deflection.

- The relation of the roll angle with the aileron, is just like the roll rate. The more positive the angle, the more positive the deflection. As in the case of the roll rate, this should be a stabilizing influence. This contrasts with the longitudinal maneuver, where it was observed that the pitch angle had a de-stabilizing relation with the elevator.
- The reference roll rate is related with the aileron such that a more positive roll rate induces a more negative aileron deflection. As in the case of the reference pitch rate in the longitudinal maneuvers, this effect is de-stabilizing. Once more, it seems like it helps the controller track the reference signal as when it increases in magnitude the controller will tend to deflect the ailerons to follow the signal. Then, the relation with the roll rate is used to damp the motion when closed to the maximum.
- Finally, the reference sideslip angle does not seem to affect the final aileron deflection. This reference signal has a constant value of 13 deg, and the SHAP value is always equal to 0. It seems like the controller is learning to maintain the required sideslip angle just considering the sideslip angle itself, without relying on the reference signal. This behaviour is observed at least when the weights have converged, however it is possible that during the initial training period, the reference sideslip angle does have an influence.

Finally, for the lateral roll rate tracking maneuver in Fig. 7, the predicted qualitative relation between the lateral features and the rudder deflection includes:

- The roll rate and the rudder deflection are related exactly like they were for the aileron. This relation is de-stabilizing as, due to the dutch roll coupling and the increased induced drag, a positive roll rate is associated with a negative yaw rate. This effect might mean that the controller is trying to use the rudder to help with the roll rate tracking.
- The yaw rate and the rudder are related in an opposite way to the aileron, such that a more positive yaw rate induces a more negative rudder deflection. This effect is also de-stabilizing. It seems like the controller is fully focused just on achieving the required roll rate tracking performance, without caring much about the effects on the yaw rate. This would make sense, as the reward of the agent does not include the yaw rate in any form, only the error with respect to the reference roll rate and sideslip angle.
- A more positive sideslip angle seems to produce a more positive rudder deflection. This behaviour is de-estabilizing as a more positive rudder deflection would produce a negative yaw rate and thus increase the sideslip angle. The controller may be using this to counteract the natural reaction of the aircraft to fly at zero sideslip, that is the weathercock stability. Since the aircraft has to fly with $\beta^R = 13$ deg, it needs to overpower its natural static stability.
- Opposite to the aileron, a more positive roll angle produces a more negative rudder deflection. A more negative rudder input leads to a positive yaw rate, and the positive yaw rate may lead, at least initially, to a more positive roll angle due to the increase in velocity on the left semi-wing. However, the increase in lift also increases the induced drag and leads to a negative yaw rate. Therefore, while this seems like a de-stabilizing effect at first, depending on the relative strength of each of these lateral-directional couplings, this may change.
- The reference roll rate has the same relation with the rudder as with the aileron. Considering the discussion of the effect of the roll rate on the rudder deflection, this could be a stabilizing influence. However, this depends on the relative strength between the different competing effects and thus depends on the particular characteristics and dynamic derivatives of this aircraft's dutch roll dynamics.
- Finally, the reference sideslip has no effect on the rudder deflection, at least after the weights have converged. Its SHAP values are always 0.

The dynamics of the aileron and the rudder deflection are much more complex to analyse due to the intricate coupling between the lateral and directional motions. Moreover, this coupling depends on the characteristics of each aircraft, as there are a series of competing effects that depend on the actual geometry and aerodynamic qualities of each vehicle. However, it seems like the controller is focusing solely on achieving the roll rate tracking task, at the expense of a possibly unstable yaw rate. In fact, in Fig. 4c it can be observed that the yaw rate is slightly increasing in magnitude over time. However, since the reward of the agent only considers the error with respect to the reference roll rate and sideslip angle signals, it makes sense that it is ignoring the effects on the yaw rate. Whereas this may not be a problem when the target is to fly with a null reference sideslip angle $\beta^R = 0$, for non-zero sideslip angle maneuvers, this may eventually lead to problems due to slow but unrestrained yaw rate increase.

Having analysed the qualitative relations between the input features and the control surface deflections, it is possible to focus on the actual function shape that relates these two magnitudes. Quite surprisingly, it can be observed that all the analysed maneuvers present a linear relation between the features and the final control deflection. Even the stall at 0.2 Hz which presents the stall and recovery events showcases an almost perfect linear relationship. Moreover, the stall at 0.05 Hz presents this linear relation even though the elevator deflection showcased at least a double frequency

component that is not present in any of the input features. This unexpected result can be the product of several different reasons. The first and least likely, is that the IDHP architecture is only able to learn linear control relations. Since the architecture uses a fully connected neural network with a non-linear activation function, which could in theory approximate functions of any characteristics, it is very improbable that the IDHP is limited to just linear control laws. Secondly, it is possible that the considered IDHP architecture with its separated longitudinal and lateral-directional dynamics is forcing the agent to learn linear control laws. However, the selected maneuvers were focused only on the longitudinal or the lateral-directional motions to work within the limitations of the selected architecture; hence this option is mostly discarded as well. A third possibility is that, thanks to its online learning capabilities, the IDHP controller is able to adapt fast enough to changes in the dynamics, that it is simply more optimal to learn continuously changing linear control laws. This option is further explored in Section IV.D. Finally, it could occur that the selected maneuvers are not sufficiently non-linear to force the controller to learn non-linear control laws. However, it can be observed how the aircraft is already stalling and the expected stall angle of attack is reached, or the largest sideslip angle that the aircraft can sustain with the allowed rudder deflection is used. Moreover, it is doubtful the controller could be properly trained in even more adverse conditions or even that the simulation model has an accurate representation of the most non-linear regimes. For this reason, it is assumed that the presented maneuvers are enough to excite the non-linear dynamics of the aircraft.

D. Verification of the results

In the previous section, it was observed that the IDHP controller had learnt linear control laws, even in the presence of non-linear dynamics. However, it was still able to achieve good tracking performance and complete the desired task while avoiding loss of control. To verify whether these results are correct, it is possible to reconstruct the predicted controllers based on the SHAP analysis. This can be done by fitting the straight line to the predicted SHAP values for each input feature for both the longitudinal and lateral controllers. Afterwards, this controller can be applied to the recorded state history of the system to compare the control surface deflections commanded by the reconstructed controller and the original actions taken by the actor. Whereas it could be possible to directly use the reconstructed controllers to repeat the simulations and compare the state evolution, the comparison of the control surface deflection for the same states should provide a more direct contrast.

From the SHAP model, if the SHAP values ϕ_i are assumed to be a linear function of the features x_i, the following controller can be obtained

$$f(x) \approx \phi_0 + \sum_{i=1}^{M} \phi_i x_i' = \phi_0 + \sum_{i=1}^{M} (c_i x_i + d_i) = \left(\phi_0 + \sum_{i=1}^{M} d_i\right) + \sum_{i=1}^{M} c_i x_i = D + \sum_{i=1}^{M} c_i x_i \tag{11}$$

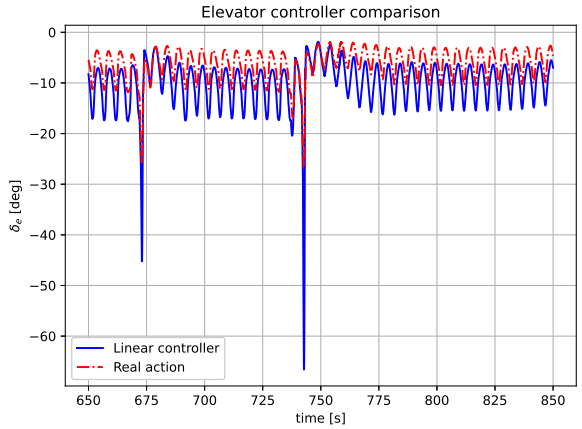
where it has been applied that $x_i' = 1$ for all the features. Additionally, as the actor ANN does not have any bias, then $\Delta\delta_e = 0$ when $\mathbf{s} = \mathbf{x} = 0$. If this is applied to the linear controller model, then $D = 0$. This indicates that the y-intercept of the straight line should not matter for the final controller model. Hence, only changes in the slope actually modify the approximated linear controller as long as it is true that the SHAP values can be approximated as a straight line. Based on all this, the longitudinal and lateral linear controllers can be written as

$$\delta_e = \delta_{trim,e} + c_{q,e} q + c_{\alpha,e} \alpha + c_{\theta,e} \theta + c_{q^R,e} q^R \tag{12}$$

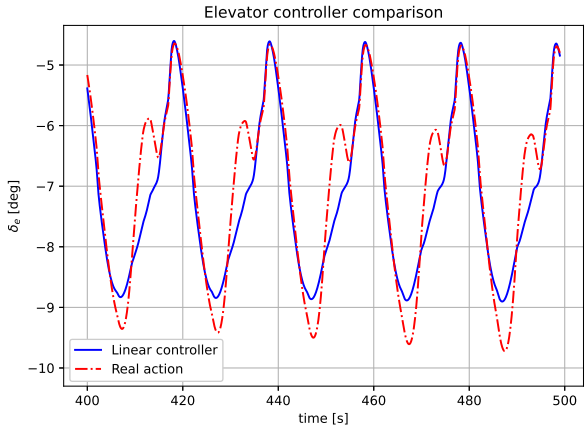
$$\delta_a = \delta_{trim,a} + c_{p,a} p + c_{r,a} r + c_{\beta,a} \beta + c_{\phi,a} \phi + c_{p^R,a} p^R + c_{\beta^R,a} \beta^R \tag{13}$$

$$\delta_r = \delta_{trim,r} + c_{p,r} p + c_{r,r} r + c_{\beta,r} \beta + c_{\phi,r} \phi + c_{p^R,r} p^R + c_{\beta^R,r} \beta^R \tag{14}$$

where the trim values have been added to the predicted incremental action of the control surfaces. It is possible to compare these reconstructed controllers with the real actions and verify the predicted SHAP model, as shown in Figs. 8 and 9. Whereas the lateral controller comparison proves that the linear approximation is very close to the real ANN controller, the longitudinal maneuvers are less well approximated. In the case of the pitch rate tracking at 0.2 Hz, the shape of the deflection function is very close to the real actor, but the values are not the expected ones except close to the stall recoveries. In the case of the pitch rate tracking at 0.05 Hz, the linear controller is even worse as it is not able to reproduce the double frequency component of the original elevator deflection. This seems to show that even though the SHAP analysis had predicted a linear controller, the IDHP actor follows a different behaviour and the method is not able to properly predict the control laws.

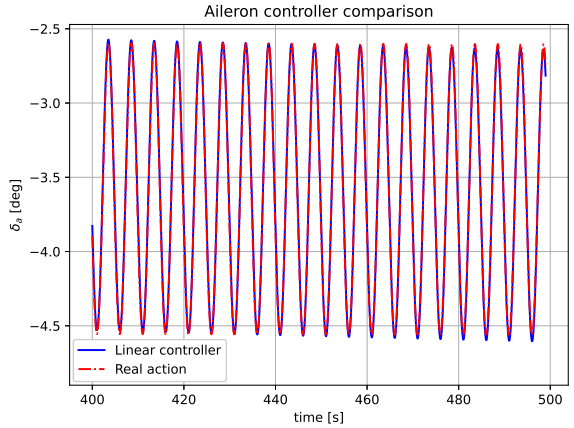


(a) Longitudinal pitch rate tracking at 0.2 Hz near the stall regime

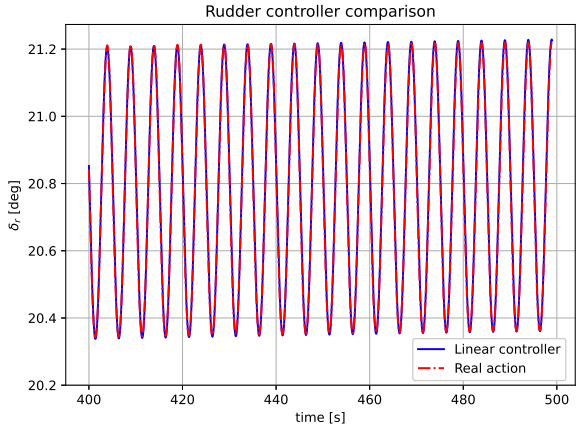


(b) Longitudinal pitch rate tracking at 0.05 Hz near the stall regime

Fig. 8 Linear controller comparison with the real action taken by the IDHP controller in the longitudinal pitch rate tracking maneuvers, using a single time segment for the SHAP analysis



(a) Aileron deflection comparison



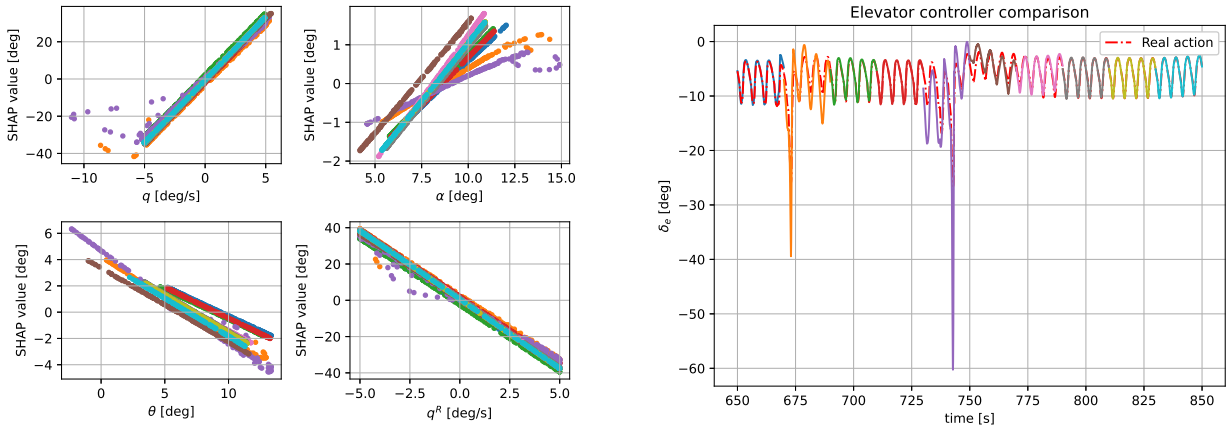
(b) Rudder deflection comparison

Fig. 9 Linear controller comparison with the real action taken by the IDHP controller in the lateral roll rate tracking maneuver with β^R = 13 deg, using a single time segment for the SHAP analysis

However, this lack of explainability is due to the SHAP assumption of constant weights. Even though the weight variation in the selected time segments is very small compared to the weights' magnitude, even small changes in some weights can significantly modify the final controller behaviour, especially at large pitch angles and angles of attack. It must be noted that since the weights are taken in the middle of the time segment, the regions with weights similar to those are the ones that are best approximated by the linear controller. This is especially apparent in Fig. 8a, as the stall recovery sections are approximated much better than the rest of the segment. It is possible to investigate whether the change in weights is the real reason behind the differences, by further dividing the single time segment into smaller sub-segments.

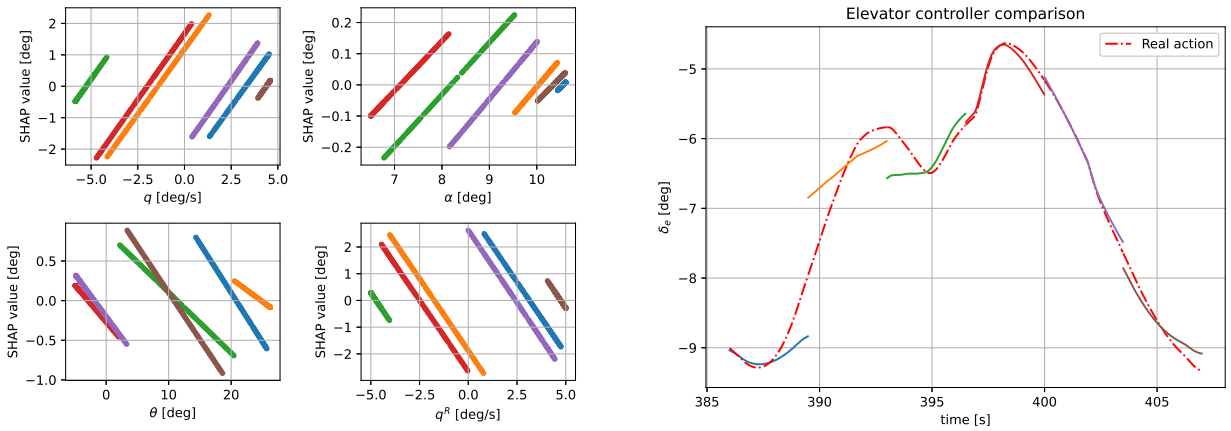
Firstly, the pitch rate tracking at 0.2 Hz was investigated by dividing the previously considered time segment into 10 pieces, to repeat the SHAP analysis. The results in Fig. 10 showcase two very interesting findings. In the first place, it can be observed how the different linear segments have a changing slope, especially in the angle of attack and pitch angle. This supports the previous conclusion that the IDHP was indeed modifying its control law even if the weight variation was minimal. Moreover, the IDHP was changing between linear controllers, as if it was applying a continuous gain scheduling method. It seems like thanks to the great online learning capabilities of the IDHP architecture, it is more

optimal to learn a continuously changing simple linear control law, rather than a complex non-linear one. However, the second finding is that in the stall recovery segments, the controller is actually learning slightly non-linear laws. The main non-linearities are present for very negative pitch rates and for very large angles of attack, which are the states found during the stall and recovery. It seems like for the very quick and very non-linear ($\alpha > 12.5$ deg) stall and recovery events, the IDHP is not able to adapt its linear law and needs to rely on a non-linear one. In spite of this, it could be argued that, except for the pitch rate, the non-linearities can actually be neglected and approximated with a straight line with reasonable precision. This can be checked in Fig. 10b, as the linear controller is still able to perform a good approximation in those segments.



(a) Input-output mapping analysis **(b) Linear controller comparison**

Fig. 10 SHAP analysis and linear controller comparison with the real action for the longitudinal pitch rate tracking maneuver at 0.2 Hz near the stall regime, using 10 equally-spaced time segments. The colour of each sub-segment is the same in both sub-figures. Note that the SHAP values axes of each feature have a different scale



(a) Input-output mapping analysis **(b) Linear controller comparison**

Fig. 11 SHAP analysis and linear controller comparison with the real action for the longitudinal pitch rate tracking maneuver at 0.05 Hz near the stall regime, considering a single elevator deflection cycle sub-divided in 6 equally-spaced time segments. The colour of each sub-segment is the same in both sub-figures. Note that the SHAP values axes of each feature have a different scale

To further investigate the pitch rate tracking at 0.05 Hz, it was decided to study only a single period of the elevator deflection, to achieve a better detail of the double frequency component. A total of 6 time segments were used, by reducing the number of samples from 200 to 150. This should still provide a decent level of local and global accuracy

while allowing for the desired 6 segments. The results in Fig. 11 show very similar characteristics to the previous analysis. Once again, the different segments have a slowly changing slope which leads to the same gain scheduling phenomena as in the case of the pitch rate tracking at 0.2 Hz. The most significant slope changes seem to be found in the pitch angle, which also played a noticeable role in the pitch rate tracking at 0.2 Hz. However, since no major non-linear and very quick events like the stall recovery are present here, the IDHP controller has enough time to modify its linear law and no non-linearities are found. As observed in Fig. 11b, the second frequency component is now better predicted, although the division into smaller sub-segments has penalized the global accuracy of the linear controller model.

V. Conclusion

This paper has shown the capabilities of the Shapley Additive exPlanations (SHAP) analysis to capture the inner workings of a complex and opaque Reinforcement Learning (RL) framework like Incremental Dual Heuristic Programming (IDHP). Thanks to its use, it has been possible to understand how the IDHP architecture was able to adapt in a complex environment, like flying very close to the stall limit and experiencing a stall and recovery event. It was observed, that even though the environment was non-linear, it was more optimal in general for the IDHP controller to learn linear control laws. It used its online learning capabilities and high sample efficiency to modify this linear control law continuously through weight changes smaller than 10% of the largest weight magnitude, instead of having to learn a more complex non-linear relation between the input features and the control surface deflection. Only when strictly necessary and forced by the suddenness of the environment, like in the stall and recovery, did the IDHP showcase a non-linear behaviour. However, even in that case, the SHAP analysis was able to provide a useful representation of the input-output mapping of the controller.

Even though this analysis has been focused on a particular RL framework, the use of SHAP could very easily be applied to more complex controller architectures in order to help understand them better. This showcases the real power of eXplainable Reinforcement Learning (XRL) and eXplainable Artificial Intelligence (XAI) techniques at least when applied to flight control, as techniques like SHAP can help break open the black box to look inside and discover the real behaviour of the controllers in a manner understandable to the target audience. In the case of this research, these results are oriented for RL researchers and control experts, but it should be relatively simple to use already existing tools and techniques from linear and non-linear classic control theory to present these findings to a much wider audience. Moreover, control experts can use the knowledge gained from studying these complex RL frameworks to improve currently applied control techniques.

Future research on this topic could try to analyse more complex, possibly offline, RL frameworks like those based on Soft Actor Critic (SAC) or Twin-Delayed Deep Deterministic Policy Gradient (TD3). Preliminary results indicate that SAC controllers learn much more complex and non-linear control laws, which allow them to provide robustness against many different failure situations, but more extensive analysis is necessary. Additionally, it could be of interest to analyse the critic input-output mapping as this could provide information about what the controller valued the most during training which lead to the observed linear laws. Once more, these results could possibly improve our current understanding of control theory and improve the current flight controllers. Finally, it could be possible to apply SHAP analysis to study how some RL frameworks are able to provide fault-tolerance and robustness to flight control applications.

References

[1] de Castro, A. I., Shi, Y., Maja, J. M., and Peña, J. M., "UAVs for Vegetation Monitoring: Overview and Recent Scientific Contributions," *Remote Sensing*, Vol. 13, No. 11, 2021. https://doi.org/10.3390/rs13112139, URL https://www.mdpi.com/2072-4292/13/11/2139.

[2] Mogili, U. R., and Deepak, B. B. V. L., "Review on Application of Drone Systems in Precision Agriculture," *Procedia Computer Science*, Vol. 133, 2018, pp. 502–509. https://doi.org/https://doi.org/10.1016/j.procs.2018.07.063, URL https://www.sciencedirect.com/science/article/pii/S1877050918310081, international Conference on Robotics and Smart Manufacturing (RoSMa2018).

[3] Singh, P. K., and Sharma, A., "An intelligent WSN-UAV-based IoT framework for precision agriculture application," *Computers and Electrical Engineering*, Vol. 100, 2022, p. 107912. https://doi.org/https://doi.org/10.1016/j.compeleceng.2022.107912, URL https://www.sciencedirect.com/science/article/pii/S0045790622001951.

[4] Zhou, S., and Gheisari, M., "Unmanned aerial system applications in construction: a systematic review," *Construction Innovation*, Vol. 18, No. 4, 2018, pp. 453–468. https://doi.org/10.1108/CI-02-2018-0010, URL https://doi.org/10.1108/CI-02-2018-0010.

[5] Schnebele, E., Tanyu, B. F., Cervone, G., and Waters, N., "Review of remote sensing methodologies for pavement management and assessment," *European Transport Research Review*, Vol. 7, No. 2, 2015, p. 7. https://doi.org/10.1007/s12544-015-0156-6, URL https://doi.org/10.1007/s12544-015-0156-6.

[6] Zaß, S., Windisch, E., Seer, S., Scheibenreif, M., Schechtner, K., Richter, A., Ponto, L., Pak, H., Ørving, T., Metz, I., Malaud, F., Kleczatsky, A., Kim, Y., Kägi, M., Irvine, P., Han, J., Habán, T., Haas, L., Geister, D., and Straubinger, A., *Ready for Take-Off? Integrating Drones into the Transport System*, OECD/ITF, 2021.

[7] Bauranov, A., and Rakas, J., "Designing airspace for urban air mobility: A review of concepts and approaches," *Progress in Aerospace Sciences*, Vol. 125, 2021, p. 100726. https://doi.org/https://doi.org/10.1016/j.paerosci.2021.100726, URL https://www.sciencedirect.com/science/article/pii/S0376042121000312.

[8] Pérez-Castán, J., Gómez Comendador, F., Rodríguez-Sanz, A., Arnaldo Valdés, R., Águeda, G., Zambrano, S., and Torrecilla, J., "Decision framework for the integration of RPAS in non-segregated airspace," *Safety Science*, Vol. 130, 2020, p. 104860. https://doi.org/https://doi.org/10.1016/j.ssci.2020.104860, URL https://www.sciencedirect.com/science/article/pii/S0925753520302575.

[9] Woods, D., "The Risks of Autonomy: Doyles Catch," *Journal of Cognitive Engineering and Decision Making*, Vol. 10, No. 2, 2016, pp. 131–133. https://doi.org/10.1177/1555343416653562.

[10] Harel, D., Marron, A., and Sifakis, J., "Creating a Foundation for Next-Generation Autonomous Systems," *IEEE Design Test*, Vol. 39, No. 1, 2022, pp. 49–56. https://doi.org/10.1109/MDAT.2021.3069959.

[11] Kaplan, A., and Haenlein, M., "Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence," *Business Horizons*, Vol. 62, No. 1, 2019, pp. 15–25. https://doi.org/https://doi.org/10.1016/j.bushor.2018.08.004, URL https://www.sciencedirect.com/science/article/pii/S0007681318301393.

[12] Li, B.-h., Hou, B.-c., Yu, W.-t., Lu, X.-b., and Yang, C.-w., "Applications of artificial intelligence in intelligent manufacturing: a review," *Frontiers of Information Technology & Electronic Engineering*, Vol. 18, No. 1, 2017, pp. 86–96. https://doi.org/10.1631/FITEE.1601885, URL https://doi.org/10.1631/FITEE.1601885.

[13] Cao, L., "AI in Finance: A Review," *SSRN Electronic Journal*, 2020. https://doi.org/10.2139/ssrn.3647625.

[14] Zeng, D., Chen, H., Lusch, R., and Li, S.-H., "Social Media Analytics and Intelligence," *IEEE Intelligent Systems*, Vol. 25, No. 6, 2010, pp. 13–16. https://doi.org/10.1109/MIS.2010.151.

[15] Bartlett, J., Smith, J. T., and Acton, R. A., *The Future of Political Campaigning*, Demos, London, UK, 2018.

[16] Pallathadka, H., Mustafa, M., Sanchez, D. T., Sekhar Sajja, G., Gour, S., and Naved, M., "IMPACT OF MACHINE learning ON Management, healthcare AND AGRICULTURE," *Materials Today: Proceedings*, 2021. https://doi.org/https://doi.org/10.1016/j.matpr.2021.07.042, URL https://www.sciencedirect.com/science/article/pii/S221478532104894X.

[17] Choy, G., Khalilzadeh, O., Michalski, M., Do, S., Samir, A. E., Pianykh, O. S., Geis, J. R., Pandharipande, P. V., Brink, J. A., and Dreyer, K. J., "Current Applications and Future Impact of Machine Learning in Radiology," *Radiology*, Vol. 288, No. 2, 2018, pp. 318–328. https://doi.org/10.1148/radiol.2018171820, URL https://doi.org/10.1148/radiol.2018171820, pMID: 29944078.

[18] Dimiduk, D. M., Holm, E. A., and Niezgoda, S. R., "Perspectives on the Impact of Machine Learning, Deep Learning, and Artificial Intelligence on Materials, Processes, and Structures Engineering," *Integrating Materials and Manufacturing Innovation*, Vol. 7, No. 3, 2018, pp. 157–172. https://doi.org/10.1007/s40192-018-0117-8, URL https://doi.org/10.1007/s40192-018-0117-8.

[19] Weber, F., and Schütte, R., "A Domain-Oriented Analysis of the Impact of Machine Learning—The Case of Retailing," *Big Data and Cognitive Computing*, Vol. 3, No. 1, 2019. https://doi.org/10.3390/bdcc3010011, URL https://www.mdpi.com/2504-2289/3/1/11.

[20] Ben-Israel, D., Jacobs, W. B., Casha, S., Lang, S., Ryu, W. H. A., de Lotbiniere-Bassett, M., and Cadotte, D. W., "The impact of machine learning on patient care: A systematic review," *Artificial Intelligence in Medicine*, Vol. 103, 2020. https://doi.org/https://doi.org/10.1016/j.artmed.2019.101785, URL https://www.sciencedirect.com/science/article/pii/S0933365719303951.

[21] Géron, A., *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed., O'Reilly Media, Inc., Sebastopol, CA, USA, 2019.

[22] Schultz, W., Dayan, P., and Montague, P. R., "A Neural Substrate of Prediction and Reward," *Science*, Vol. 275, No. 5306, 1997, pp. 1593–1599. https://doi.org/10.1126/science.275.5306.1593, URL https://www.science.org/doi/abs/10.1126/science.275.5306.1593.

[23] Powell, W. B., *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, John Wiley & Sons, Hoboken, NJ, USA, 2011. https://doi.org/10.1002/9781118029176.

[24] Schäfer, A. M., and Zimmermann, H. G., "Recurrent Neural Networks Are Universal Approximators," *Artificial Neural Networks – ICANN 2006*, edited by S. D. Kollias, A. Stafylopatis, W. Duch, and E. Oja, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 632–640.

[25] Mousavi, S. S., Schukat, M., and Howley, E., "Deep Reinforcement Learning: An Overview," *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*, edited by Y. Bi, S. Kapoor, and R. Bhatia, Springer International Publishing, Cham, 2018, pp. 426–440.

[26] Wang, H.-n., Liu, N., Zhang, Y.-y., Feng, D.-w., Huang, F., Li, D.-s., and Zhang, Y., "Deep reinforcement learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, Vol. 21, 2020. https://doi.org/10.1631/FITEE.1900533.

[27] Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D., "Mastering the game of Go with deep neural networks and tree search," *Nature*, Vol. 529, 2016, pp. 484–489. https://doi.org/10.1038/nature16961.

[28] Vinyals, O., Babuschkin, I., Czarnecki, W., Mathieu, M., Dudzik, A., Chung, J., Choi, D., Powell, R., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J., Jaderberg, M., and Silver, D., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, Vol. 575, 2019. https://doi.org/10.1038/s41586-019-1724-z.

[29] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., and Hassabis, D., "Highly accurate protein structure prediction with AlphaFold," *Nature*, Vol. 596, 2021, pp. 1–11. https://doi.org/10.1038/s41586-021-03819-2.

[30] Balas, G. J., "Flight Control Law Design: An Industry Perspective," *European Journal of Control*, Vol. 9, No. 2, 2003, pp. 207–226. https://doi.org/https://doi.org/10.3166/ejc.9.207-226, URL https://www.sciencedirect.com/science/article/pii/S0947358003702763.

[31] Horn, J. F., "Non-Linear Dynamic Inversion Control Design for Rotorcraft," *Aerospace*, Vol. 6, No. 3, 2019. https://doi.org/10.3390/aerospace6030038, URL https://www.mdpi.com/2226-4310/6/3/38.

[32] Smith, P., and Berry, A., "Flight test experience of a non-linear dynamic inversion control law on the VAAC Harrier," *Atmospheric flight mechanics conference*, 2000. https://doi.org/10.2514/6.2000-3914, URL https://arc.aiaa.org/doi/abs/10.2514/6.2000-3914.

[33] Juliana, S., Chu, Q., Mulder, J., and van Baten, T., "Flight control of atmospheric re-entry vehicle with non-linear dynamic inversion," *AIAA guidance, navigation, and control conference and exhibit*, 2004. https://doi.org/10.2514/6.2004-5330, URL https://arc.aiaa.org/doi/abs/10.2514/6.2004-5330.

[34] Lu, P., van Kampen, E.-J., de Visser, C., and Chu, Q., "Aircraft fault-tolerant trajectory control using Incremental Nonlinear Dynamic Inversion," *Control Engineering Practice*, Vol. 57, 2016, pp. 126–141. https://doi.org/https://doi.org/10.1016/j.conengprac.2016.09.010, URL https://www.sciencedirect.com/science/article/pii/S0967066116302118.

[35] 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, URL https://doi.org/10.2514/1.49978.

[36] Simplício, P., Pavel, M., van Kampen, E., and Chu, Q., "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/https://doi.org/10.1016/j.conengprac.2013.03.009, URL https://www.sciencedirect.com/science/article/pii/S0967066113000634.

[37] Smeur, E. J. J., Chu, Q., 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, URL https://doi.org/10.2514/1.G001490.

[38] Acquatella, P., Falkena, W., van Kampen, E.-J., and Chu, Q. P., "Robust Nonlinear Spacecraft Attitude Control using Incremental Nonlinear Dynamic Inversion." *AIAA Guidance, Navigation, and Control Conference*, 2012. https://doi.org/10.2514/6.2012-4623, URL https://arc.aiaa.org/doi/abs/10.2514/6.2012-4623.

[39] 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/https://doi.org/10.1016/j.conengprac.2017.12.011, URL https://www.sciencedirect.com/science/article/pii/S096706611730285X.

[40] d'Apolito, F., and Sulzbachner, C., "Flight Control of a Multicopter using Reinforcement Learning," *IFAC-PapersOnLine*, Vol. 54, No. 13, 2021, pp. 251–255. https://doi.org/https://doi.org/10.1016/j.ifacol.2021.10.454, URL https://www.sciencedirect.com/science/article/pii/S2405896321018917, 20th IFAC Conference on Technology, Culture, and International Stability TECIS 2021.

[41] Koch, W., Mancuso, R., West, R., and Bestavros, A., "Reinforcement Learning for UAV Attitude Control," *ACM Transactions on Cyber-Physical Systems*, Vol. 3, 2018. https://doi.org/10.1145/3301273.

[42] Clarke, S. G., and Hwang, I., "Deep Reinforcement Learning Control for Aerobatic Maneuvering of Agile Fixed-Wing Aircraft," *AIAA SCITECH 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-0136, URL https://arc.aiaa.org/doi/abs/10.2514/6.2020-0136.

[43] Azar, A. T., Koubaa, A., Ali Mohamed, N., Ibrahim, H. A., Ibrahim, Z. F., Kazim, M., Ammar, A., Benjdira, B., Khamis, A. M., Hameed, I. A., and Casalino, G., "Drone Deep Reinforcement Learning: A Review," *Electronics*, Vol. 10, No. 9, 2021. https://doi.org/10.3390/electronics10090999, URL https://www.mdpi.com/2079-9292/10/9/999.

[44] Choraś, M., Pawlicki, M., Puchalski, D., and Kozik, R., "Machine Learning – The Results Are Not the only Thing that Matters! What About Security, Explainability and Fairness?" *Computational Science – ICCS 2020*, edited by V. V. Krzhizhanovskaya, G. Závodszky, M. H. Lees, J. J. Dongarra, P. M. A. Sloot, S. Brissos, and J. Teixeira, Springer International Publishing, Cham, 2020, pp. 615–628. https://doi.org/10.1007/978-3-030-50423-6_46.

[45] Guidotti, R., Monreale, A., Turini, F., Pedreschi, D., and Giannotti, F., "A Survey of Methods for Explaining Black Box Models," *ACM Computing Surveys*, Vol. 51, 2018. https://doi.org/10.1145/3236009.

[46] van Zijl, J., "Using Explainable Artificial Intelligence to Improve Transparency of Reinforcement Learning for Online Adaptive Flight Control: Breaking Open the Black Box," Master's thesis, TU Delft Aerospace Engineering, February 2022. URL http://resolver.tudelft.nl/uuid:66a5fdba-6508-4b6f-b20f-c1766ec4fc7f.

[47] 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/https://doi.org/10.1016/0893-6080(95)00042-9, URL https://www.sciencedirect.com/science/article/pii/0893608095000429.

[48] 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, URL https://doi.org/10.2514/1.12597.

[49] Zhou, Y., Van Kampen, E.-J., and Chu, Q., "Launch Vehicle Adaptive Flight Control with Incremental Model Based Heuristic Dynamic Programming," *68th International Astronautical Congress (IAC)*, 2017.

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

[51] Heyer, S., Kroezen, D., and Kampen, E.-J. V., "Online Adaptive Incremental Reinforcement Learning Flight Control for a CS-25 Class Aircraft," *AIAA SCITECH 2020 Forum*, 2020. https://doi.org/10.2514/6.2020-1844, URL https://arc.aiaa.org/doi/abs/10.2514/6.2020-1844.

[52] Haykin, S., *Adaptive Filter Theory*, 5th ed., Pearson Education, Harlow, Essex, UK, 2013.

[53] Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F., "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, Vol. 58, 2020, pp. 82–115. https://doi.org/https://doi.org/10.1016/j.inffus.2019.12.012, URL https://www.sciencedirect.com/science/article/pii/S1566253519308103.

[54] Juozapaitis, Z., Koul, A., Fern, A., Erwig, M., and Doshi-Velez, F., "Explainable Reinforcement Learning via Reward Decomposition," *IJCAI/ECAI Workshop on explainable artificial intelligence*, 2019.

[55] Lundberg, S. M., and Lee, S.-I., "A Unified Approach to Interpreting Model Predictions," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Curran Associates Inc., Red Hook, NY, USA, 2017, p. 4768–4777.

[56] Ribeiro, M. T., Singh, S., and Guestrin, C., ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Association for Computing Machinery, New York, NY, USA, 2016, p. 1135–1144. https://doi.org/10.1145/2939672.2939778, URL https://doi.org/10.1145/2939672.2939778.

[57] Chen, H., Lundberg, S., and Lee, S.-I., "Explaining Models by Propagating Shapley Values of Local Components," *Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability*, Studies in Computational Intelligence, Vol. 914, edited by A. Shaban-Nejad, M. Michalowski, and D. L. Buckeridge, Springer International Publishing, Cham, 2021, pp. 261–270. https://doi.org/10.1007/978-3-030-53352-6_24, URL https://doi.org/10.1007/978-3-030-53352-6_24.

[58] Shrikumar, A., Greenside, P., and Kundaje, A., "Learning Important Features through Propagating Activation Differences," *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, JMLR.org, 2017, p. 3145–3153.

[59] Dally, K., and Kampen, E.-J. V., "Soft Actor-Critic Deep Reinforcement Learning for Fault Tolerant Flight Control," *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics, 2022. https://doi.org/10.2514/6.2022-2078, URL https://doi.org/10.2514%2F6.2022-2078.