Risk-sensitive Distributional Reinforcement Learning for Flight Control

Seres, Peter; Liu, Cheng; van Kampen, Erik Jan

DOI
10.1016/j.ifacol.2023.10.1097

Publication date
2023

Document Version
Final published version

Published in
IFAC-PapersOnLine

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 2013–2018

Risk-sensitive Distributional Reinforcement Learning for Flight Control

Peter Seres * **Cheng Liu** * **Erik-Jan van Kampen** *

* *Aerospace Engineering, Delft University of Technology, 2629HS Delft*
peter.seres.ae@gmail.com, c.liu-10@tudelft.nl, e.vankampen@tudelft.nl

Abstract: Recent aerospace systems increasingly demand model-free controller synthesis, and autonomous operations require adaptability to uncertainties in partially observable environments. This paper applies distributional reinforcement learning to synthesize risk-sensitive, robust model-free policies for aerospace control. We investigate the use of distributional soft actor-critic (DSAC) agents for flight control and compare their learning characteristics and tracking performance with the soft actor-critic (SAC) algorithm. The results show that (1) the addition of distributional critics significantly improves learning consistency, (2) risk-averse agents increase flight safety by avoiding uncertainties in the environment.

Keywords: Guidance, navigation and control of vehicles, Reinforcement learning control, Distributional reinforcement learning, Risk-sensitive learning

1. INTRODUCTION

In recent years, technological advancements have resulted in increased complexity in the dynamics of aerospace systems. Such complex control systems have to maintain safety and performance in challenging, partially observable environments with unforeseen circumstances. These factors drive the need for increased levels of intelligence and autonomy in the control of aerospace systems.

Traditional approaches to flight control synthesis rely on the costly identification of high-fidelity models and predefined operating conditions, and therefore reduce robustness and adaptability. Advanced control methods, such as incremental non-linear dynamic inversion (INDI) reduce modelling requirements, and have shown fault-tolerant capability, but introduce challenges with sensor synchronization (Pollack and Van Kampen, 2022).

Deep reinforcement learning (DRL) methods have shown capability to solve large-scale real-world problems in decision making and control. Reinforcement learning (RL) is a goal-oriented model-free approach to synthesize policies for complex tasks. DRL algorithms, such as the soft actor-critic (SAC) (Haarnoja et al., 2019) have been shown to achieve fault-tolerant flight control, while maintaining robustness to varying flight conditions and sensor noise (Dally and Van Kampen, 2022).

Even though such algorithms show great control performance and generalization power, their learning behaviour is inconsistent and sensitive to hyperparameters. In order to facilitate the application of DRL algorithms on safety-critical systems, it is desirable to improve the reliability of these approaches. In order to reduce model dependence, risk-sensitive policies are needed to handle the uncertainty in the environment.

Unlike traditional RL methods, distributional RL algorithms (Bellemare et al., 2017; Dabney et al., 2018) represent the full probability distribution of the reward and achieve improved learning characteristics as a result. They also enable the synthesis of risk-sensitive control laws. Liu et al. (2022) have shown that the use of risk-sensitive distributional RL agents improves the safety of drone navigation in uncertain environments.

We implement the distributional soft actor-critic (DSAC) (Ma et al., 2020) to solve an attitude control task using a validated model of a research aircraft. A population of agents is trained to investigate the learning and tracking performance of agents trained using distributional critics.

The contribution of this paper is two-fold. Firstly, we demonstrate that using DSAC for flight control significantly improves learning consistency, while achieving similar tracking performance to SAC controllers. Secondly, we show that risk-averse policies achieve safer flight control by sacrificing rewards to avoid uncertainty in the environment.

The structure of the paper is as follows. Section 2 provides background on RL-based flight control. Then, Section 3 discusses the methodology used to train the agents. Section 4 presents the results followed by concluding remarks in Section 5.

2. BACKGROUND

2.1 Reinforcement Learning

We consider a sequential decision making task formulated as a Markov Decision Process (MDP) described by the structured set $\mathcal{M} \sim \langle \mathcal{S}, \mathcal{A}, R, \mathcal{P}, \gamma \rangle$, with state-space $\mathcal{S} \subset \mathbb{R}^n$, action-space $\mathcal{A} \subset \mathbb{R}^m$, reward function $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, stochastic state transition $\mathcal{P} : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, \infty)$ and discount factor $\gamma \in [0, 1)$. The decision making agent chooses action $a_t \in \mathcal{A}$ according to policy $a_t \sim \pi(a_t|s_t)$ at time-step t, and observes the transition tuple $\mathbb{T}_t = \langle s, a, r, s' \rangle$, where r is the immediate reward and s' is the next-state $s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$. The traditional RL

10.1016/j.ifacol.2023.10.1097

task is to find the optimal policy π^* that maximizes the expected return, i.e. the expected cumulative rewards of this sequential decision making task.

The action-value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the expected return of the agent choosing action a in state s, and following policy π thereafter, as given by (1):

$$Q^\pi(s,a) \coloneqq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \tag{1}$$

In order to find the optimal policy π^*, RL algorithms repeatedly apply the contractive Bellman operator $\mathcal{T}^\pi$:

$$\mathcal{T}^\pi Q(s,a) \coloneqq \mathbb{E}\left[R(s,a)\right] + \gamma \mathbb{E}_{\mathcal{P},\pi}\left[Q(s',a')\right] \tag{2}$$

In order to solve complex tasks, the field of DRL introduces deep neural networks (DNNs) as function approximators to parameterize the action-value function $Q_k(s,a) \approx Q(s,a)$ (value-based methods), the policy directly $\pi_w(s) \approx \pi^*(s)$ (policy-based methods) or both (actor-critic methods).

Actor-critic methods combine the advantages of both value-based and policy-based approaches: the direct parameterization of the policy enables the use of continuous actions, and the critic provides a biased estimate of the return, greatly improving the learning behaviour. State-of-the-art algorithms, such as DDPG (Lillicrap et al., 2015), TD3 (Fujimoto et al., 2018), and SAC (Haarnoja et al., 2019) are capable of tackling high-dimensional control tasks.

2.2 Soft actor-critic

The SAC algorithm (Haarnoja et al., 2019) makes use of *maximum entropy RL*, which extends the objective function to maximize not only the return, but an additional entropy term $\mathcal{H}_\pi$, shown in (3). Maximizing the entropy of the stochastic policy results in more efficient exploration, encouraging the diversity of actions. The resulting soft Bellman operator $\mathcal{T}_S^\pi$ is shown in (4):

$$\mathcal{H}(\pi_w(\cdot|s)) = \mathbb{E}_{a\sim\pi_w}\left[-\log\, \pi_w(a|s)\right] \tag{3}$$

$$\mathcal{T}_S^\pi Q(s,a) \coloneqq \mathbb{E}\left[R(s,a)\right] + \gamma \mathbb{E}_{\mathcal{P},\pi}\big[Q(s',a') - \eta\, \log\, \pi(a'|s')\big], \tag{4}$$

where η is the temperature parameter balancing the prioritization of rewards and the entropy of the policy.

The approach by Haarnoja et al. (2019) utilizes double-critics to prevent the overestimation of the value function (Fujimoto et al., 2018). Two soft Q-functions are trained in parallel $Q_{k_{1,2}}(s,a)$ and the minimum is taken to determine the temporal-difference (TD) error. The use of fixed Q-networks $Q_{\bar{k}}$ is adopted to stabilize learning, and the parameters of the target-networks are interpolated with step-size ζ towards the local networks.

The critic loss function $\mathcal{L}_Q$ minimizes the mean-squared TD-error δ_l given in (5) for both Q-networks $l = 1, 2$:

$$\delta_l = r + \gamma\Big(\min_{l=1,2} Q_{\bar{k}_l}(s',a') - \eta\, \log\, \pi_w(a'|s')\Big) - Q_{k_l}(s,a)$$

$$\mathcal{L}_Q^{\mathcal{B}}(k_l) = \mathbb{E}_{\mathcal{B}}\left[\delta_l^2\right], \tag{5}$$

where $\mathcal{B}$ is a mini-batch of transitions sampled from an experience replay buffer $\mathcal{B} \sim \mathcal{D} = \{\mathbb{T}_0, \mathbb{T}_1, ...\}$, $Q_{k_l}(s,a)$ is the local soft action-value estimate and $Q_{\bar{k}_l}(s',a')$ is the one-step ahead prediction.

SAC uses a stochastic actor to ensure improved exploration: a multivariate Gaussian distribution with a diagonal covariance matrix. The mean vector $\mu_w \in \mathbb{R}^m$ and the covariance diagonal $\sigma_w \in \mathbb{R}^m$ are estimated by a DNN. The actions sampled from the distribution are passed through a *tanh* squashing function to ensure they are defined on a finite bound (Haarnoja et al., 2019):

$$a_w(s) = \tanh(\tilde{a}_w), \text{ with } \tilde{a}_w \sim \mathcal{N}(\mu_w(s), \sigma_w(s)) \tag{6}$$

The loss function $\mathcal{L}_\pi$ maximizes both return and entropy:

$$\mathcal{L}_\pi^{\mathcal{B}}(w) = \mathbb{E}_{\mathcal{B}}\left[\eta\, \log\, \pi_w(a_w|s) - \min_{l=1,2} Q_{k_l}(s, a_w(s))\right] \tag{7}$$

The temperature η is dynamically optimized to achieve a target entropy $\bar{\mathcal{H}}$ (Haarnoja et al., 2019), in order to improve exploration, via the loss function:

$$\mathcal{L}^{\mathcal{B}}(\eta) = \mathbb{E}_{\mathcal{B}}\left[\eta\bar{\mathcal{H}} - \eta\, \log\, \pi_w(a|s)\right] \tag{8}$$

2.3 Distributional RL

While traditional RL maximizes the expected cumulative rewards, distributional RL gets rid of the expectation and estimates the entire probability distribution of returns, i.e. the *return distribution function*.

The action-value function defined in (1) is the first moment of the return distribution $Q^\pi(s,a) \coloneqq \mathbb{E}\left[Z^\pi(s,a)\right]$, and the random variable Z is the discounted cumulative reward $Z^\pi(s,a) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)$. As shown by Bellemare et al. (2017), the distributional Bellman operator $\mathcal{T}_D^\pi$ can be formulated as given in (9):

$$\mathcal{T}_D^\pi Z(s,a) \overset{\mathcal{D}}{\coloneqq} R(s,a) + \gamma Z(s',a'), \tag{9}$$

where $\overset{\mathcal{D}}{=}$ denotes equality by distribution, i.e. the notion that two random variables are equal when their distributions are equal.

A distributional RL method is primarily defined by two attributes: the probability metric used to measure distances between distributions and the parameterization of the approximate return distribution.

Bellemare et al. (2017) showed that the distributional Bellman operator is a contraction under the p-Wasserstein metric defined in terms of the inverse cumulative distribution function (c.d.f.) of the random return. Given random variable Z, the c.d.f. is defined as $F_Z(z) \coloneqq \mathbb{P}\left[Z < z\right]$ and the quantile function is $F_Z^{-1}(\tau) \coloneqq \inf\left\{z \in \mathbb{R} : \tau \leq F_Z(z)\right\}$, where τ is the quantile fraction. Hereinafter, the notation $Z_k^\tau(s,a)$ is used for the approximate quantile function.

In order to parameterize the return distribution, C51 (Bellemare et al., 2017) uses discrete atoms, whereas implicit quantile networks (IQN) (Dabney et al., 2018) use quantile regression to approximate the continuous quantile function implicitly.

2.4 Flight Control as an RL Task

To represent aircraft dynamics, we consider the non-linear, non-affine, stationary system $\dot{x} = f(x,u,t) \approx f(x,u)$, where f is the state transition function, $x \in \mathbb{R}^{n'}$ is the dynamic state vector, $u \in \mathbb{R}^{m'}$ is the control input vector.

A *tracking control task* is to minimize the tracking error between references and the controlled states. To represent the control task as an MDP, the observation space of the RL agent must include either the reference y_r, or the tracking error $e = y_r - x_c$, where $x_c \subset x$ is the vector of controlled states. The reward function is often defined as a penalty proportional to the tracking error $R \propto \|e\|_1$. Flight control tasks often contain intrinsic uncertainty, either due to stochastic processes, such as turbulence and sensor noise, or due to the influence of unobservable states on the dynamics of the aircraft.

3. METHODOLOGY

3.1 Distributional Soft Actor-Critic

The distributional RL approach is adopted to the continuous control task by using distributional critics in an actor-critic architecture. The DSAC method by Ma et al. (2020) combines maximum entropy RL and distributional critics to enable the training of risk-sensitive actors.

Figure (1) shows the DSAC architecture where the critic networks are distributional Z-function approximators and a risk-distortion step is introduced in the policy loss function.

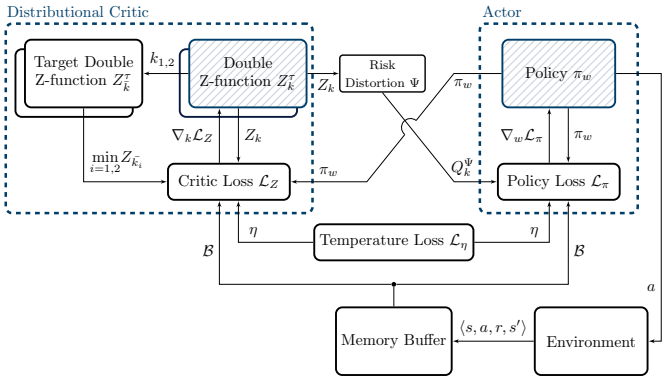

Fig. 1. DSAC architecture with risk-sensitive learning.

DSAC uses the quantile Huber loss as a substitute for the Wasserstein metric. The pairwise TD-error between two quantile fractions τ_i and τ_j is given by (10):

$$\delta^l_{ij} = r + \gamma \left(\min_{l=1,2} Z^{\tau_i}_{k_l}(s', a') - \eta \log \pi_w(a'|s') \right) - Z^{\tau_j}_{k_l}(s, a) \tag{10}$$

where the quantile fractions are sampled independently $\tau_i, \tau_j \sim U([0,1])$ and $a' \sim \pi_w(\cdot|s')$. The Huber loss for quantile fraction τ is given by (11):

$$\rho^{\kappa}_{\tau}(\delta) = |\tau - \mathbb{I}\{\delta < 0\}| \cdot \mathcal{L}^{\kappa}(\delta), \text{with}$$
$$\mathcal{L}^{\kappa}(\delta) = \begin{cases} \frac{1}{2}\delta^2 & \text{for } |\delta| \le \kappa \\ \kappa\left(|\delta| - \frac{1}{2}\kappa\right) & \text{otherwise} \end{cases}, \tag{11}$$

where $\mathbb{I}$ is the indicator function, and κ is the Huber-loss threshold (commonly $\kappa = 1$). The approximate quantile loss is estimated using a set of N independent quantiles sampled for both target and local networks:

$$\mathcal{L}^{\mathcal{B}}_Z(k_l) = \mathbb{E}_{\mathcal{B}} \left[\frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \rho^{\kappa}_{\tau_j}(\delta^l_{ij}) \right] \tag{12}$$

Risk-sensitive learning can be achieved by maximizing a distorted expectation of the soft action-value distribution (Dabney et al., 2018). Let $\Psi : [0,1] \to [0,1]$ be a continuous monotonic function, which acts as a *distortion risk measure*:

$$Q^{\Psi}_k(s,a) = \mathbb{E}_{\tau} \left[Z^{\Psi(\tau)}_k(s, \pi_w(\cdot|s)) \right], \tag{13}$$

The risk-distortion is parameterized towards risk-averse or risk-seeking learning by using the Wang risk-distortion function (Wang, 2000):

$$\text{Wang}(\tau; \xi) = \Phi(\Phi^{-1}(\tau) + \xi), \tag{14}$$

where Φ is the c.d.f. of the normal distribution and ξ is the distortion parameter for risk-averse $\xi < 0$ and risk-seeking $\xi > 0$ learning. Figure (2) shows the difference in the risk-distorted expectation of two random variables with different variances. Under risk-averse distortion, a return distribution with higher uncertainty results in a lower expected reward.

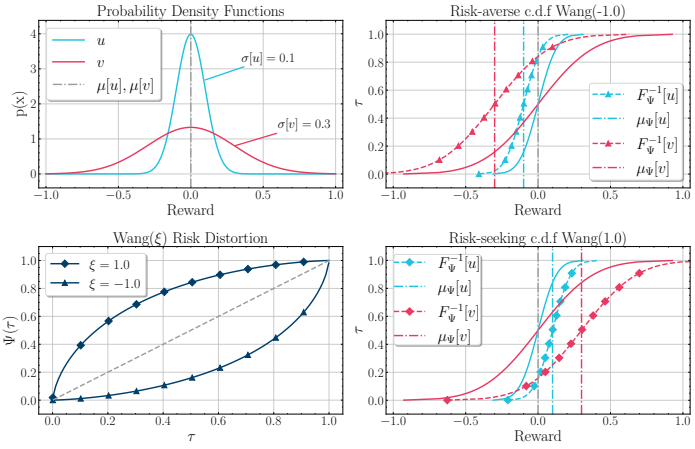

Fig. 2. Risk-averse and risk-seeking distortions.

3.2 Policy regularization

A phenomenon of converged DRL control laws is the lack of smoothness, which reduces the practical utility, causing degraded tracking performance, high power usage, and actuator failure. The approach of Conditioning for Action Policy Smoothness (CAPS) by Mysore et al. (2021) adds the regularizing term $\mathcal{L}^{\text{C}}_{\pi}$ to the policy loss to encourage spatial and temporal smoothness:

$$\mathcal{L}_S = \|\pi_w(s) - \pi_w(\tilde{s}\ )\|_2, \quad \tilde{s} \sim \mathcal{N}(s, \tilde{\sigma}) \tag{15}$$
$$\mathcal{L}_T = \|\pi_w(s) - \pi_w(s')\|_2 \tag{16}$$
$$\mathcal{L}^{\text{C}}_{\pi} = \lambda_S \mathcal{L}_S + \lambda_T \mathcal{L}_T, \tag{17}$$

where $\tilde{s}$ are the proximal states and λ_S, λ_T tune the prevalence of smoothness regularization.

Thus, the final objective function of the policy maximizes entropy to facilitate diverse actions, maximizes the risk-distorted soft action-value Q^{Ψ}_k to facilitate risk-sensitive policies, and encourages smooth control laws using the regularization term $\mathcal{L}^{\text{C}}_{\pi}$. The combined policy objective function is formulated as a loss in (18):

$$\mathcal{L}^{\mathcal{B}}_{\pi}(w) = \mathbb{E}_{\mathcal{B}} \left[\eta \log \pi_w(a_w(s)|s) - Q^{\Psi}_k(s,a) + \mathcal{L}^{\text{C}}_{\pi} \right] \tag{18}$$

3.3 Attitude Control

The tracking task investigated in this paper is the attitude control of a validated high-fidelity model of a Cessna Citation II research aircraft, with fully-coupled non-linear dynamics (Van den Hoek et al., 2018). The task is to track pitch θ_r and roll ϕ_r, and regulate the sideslip to $\beta_r = 0$.

Aircraft Model The model has 10 dynamic states: altitude h, true airspeed V, angle of attack α, angle of sideslip β, angular velocities p, q and r and Euler-angles ϕ, θ and ψ for roll, pitch and yaw respectively.

Similarly to the methodology of Dally and Van Kampen (2022), the thrust control is delegated to an inner loop controller that regulates velocity to the trim-condition. The available control surfaces are the elevator δ_e, aileron δ_a and rudder δ_r deflections. The resulting dynamic state vector $x \in \mathbb{R}^{10}$ and control input vector $u \in \mathbb{R}^3$ are:

$$x = [p, q, r, V, \alpha, \beta, \phi, \theta, \psi, h]^T \tag{19}$$

$$u = [\delta_e, \delta_a, \delta_r]^T \tag{20}$$

The aircraft model is initialized from a trimmed condition at $h = 2,000$ (m) and $V = 90$ (m/s). The refresh rate of the simulation is 100 (Hz) with ideal sensors. The actuators are modeled as low-pass filters with fixed deflection saturation which results in the action space:

$$\mathcal{A} = \{[-17°, 15°] \times [-19°, 15°] \times [-22°, 22°]\} \subset \mathbb{R}^3 \tag{21}$$

Controller Architecture We consider only the safety-critical inner loop control, as opposed to previous studies (Dally and Van Kampen, 2022; Teirlinck and Van Kampen, 2022), with cascaded control architectures. This isolates the effect of distributional RL without the complexity of multi-agent systems. We consider the architecture shown in Figure (3), with observation and action:

$$s = [\theta_e, \phi_e, \beta_e, p, q, r, \alpha]^T, \quad a = [\delta_e, \delta_a, \delta_r]^T \tag{22}$$

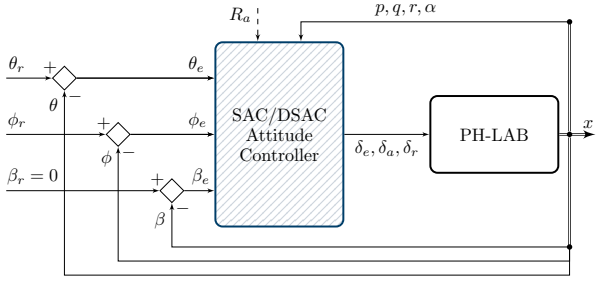

Fig. 3. SAC/DSAC attitude controller architecture.

The reward function found optimal by Dally and Van Kampen (2022) is adopted to penalize the tracking error:

$$R(s,a) = -\frac{1}{3}\left\| \text{clip}\left(\left[\frac{6}{\pi} c \odot e \right], -1, 0 \right) \right\|_1, \tag{23}$$

where $e \in \mathbb{R}^3$ is the tracking error $e = [\theta_r - \theta, \phi_r - \phi, \beta_r - \beta]^T$ and $c \in \mathbb{R}^3$ is the relative cost $[1, 1, 4]^T$. The agents are trained using 30 (s) episodes with randomly generated $\langle \theta_r, \phi_r \rangle$ reference signals, which are sequences of cosine-smoothed step inputs with uniformly sampled amplitudes.

3.4 Agent Training

Critic Network IQN estimators by Dabney et al. (2018) are used as distributional critics. The quantiles are passed through a cosine embedding layer $C_j(\tau) := \mathbb{F}\left(\sum_{i=1}^{N} cos(\pi i \tau) w_{ij} + b_j\right)$, where $\mathbb{F}$ is a sigmoid function, and w_{ij}, b_j are the individual weights and biases. Interaction between the state-action pair $[s, a]^T$ and the sample embedding is achieved using the Hadamard product, and layer normalization is used to ensure well-bounded quantiles. Figure (4) shows the architecture of the Z-function approximator networks.

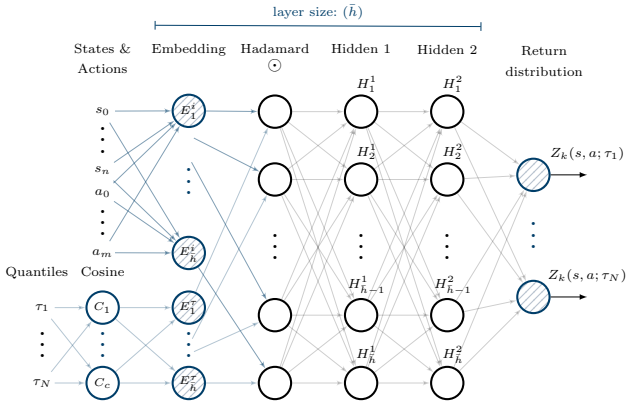

Fig. 4. IQN network based on Dabney et al. (2018)

Experiment Design To assess the effect of distributional RL, batches (10 each) of three agent variants are trained: baseline SAC, risk-neutral (R.N.) DSAC and risk-averse (R.A.) DSAC agents. The *learning performance* of the agents is assessed with respect to consistency, sample efficiency, and mean and variance of converged rewards. The *tracking performance* is assessed using a normalized mean absolute error (nMAE) metric, similarly to the methodology of Teirlinck and Van Kampen (2022).

The hyperparameters are shown in Table (1). The linearly decreasing learning rate, γ, $\bar{h}$, $|\mathcal{B}|$, and ζ have been set to the values found optimal by Dally and Van Kampen (2022). The buffer size $|\mathcal{D}|$ is increased to 1e6 to stabilize learning. Ψ is set to identity for risk-neutral and Wang(-0.5) for risk-averse agents.

Table 1. Hyperparameters.

Hyperparameter	Notation	Value
Learning rate	α_{LR}	4.4e-4 $\rightarrow$ 0
Entropy target	$\bar{\mathcal{H}}$	$-m = -3$
Discount factor	γ	0.99
Dense network activation		`ReLU`
Hidden neurons	$\bar{h}$	64×64
Memory buffer size	$\lvert\mathcal{D}\rvert$	$1,000,000$
Mini-batch size	$\lvert\mathcal{B}\rvert$	256
Interpolation step-size	ζ	0.995
CAPS-smoothing	$\lambda_{S,T}$	400
Smoothing proximity	$\bar{\sigma}$	5e$-$2
Nr. of quantiles	N	8
Nr. of cosine neurons	C	64
Nr. of quantiles for Q_k^{Ψ}	T	16
Risk-averse parameter	ξ	-0.5

In order to ensure reproducibility, the pseudo-random stochasticity during training is controlled for both environment and agent. The source code and algorithm [1] are available online.

4. RESULTS AND DISCUSSION

4.1 Learning and tracking performance

The agents are trained for 7.5e5 samples, i.e. 250 episodes. The learning curves are shown in Figure (5), for the first 1.5e5 samples, due to early convergence. DSAC demonstrates an increase in sample efficiency of 33.9% for the risk-neutral, and 28.5% for the risk-averse agents.

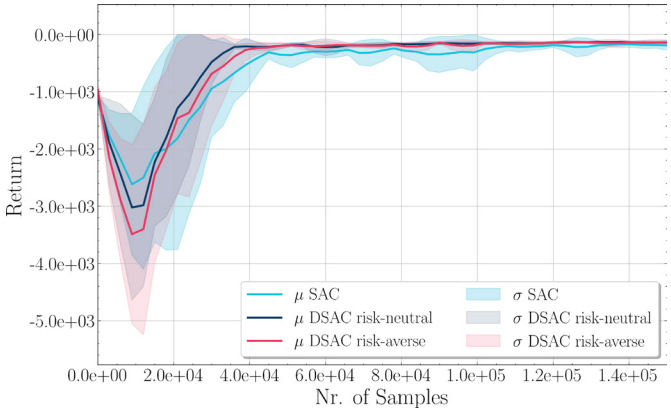

Fig. 5. Mean learning curve and standard deviation (n=10).

In addition to improved sample efficiency, two further improvements can be observed: higher converged average rewards and a reduction in variance.

Table (2) summarizes the converged average return μ and standard deviation of return σ for each variant, indicating the relative improvement with respect to the SAC baseline. Additionally, nMAE values are shown as a metric of tracking performance. The p-value is shown for the hypotheses that DSAC achieves higher returns, achieves lower variance, and performs better in tracking control.

Table 2. Average return, variance, and nMAE. Bold shows significant improvement (p<5e-2).

	SAC	R.N. DSAC		R.A. DSAC	
		Value (Rel.)	p	Value (Rel.)	p
$\mu[Z]$	-149	-122 (+18%)	3e-1	-118 (+21%)	3e-1
$\sigma[Z]$	177	**32 (-82%)**	**5e-6**	**31 (-82%)**	**5e-6**
nMAE	12.5	12.4 (-1.5%)	5e-1	**12.0 (-4.0%)**	**8e-3**

It can be seen that the reduced variance of DSAC agents is statistically significant. This indicates improved learning stability and improved robustness to the stochasticity of the environment. Furthermore, the improvement in learning characteristics is achieved while maintaining similar tracking performance.

The sample efficiency of all three agent variants have increased by 95% relative to the approach of Dally and Van Kampen (2022), which required 1e6 samples. This improvement is likely due to the omission of the incremental architecture, which reduces the size of the observation space from $\mathbb{R}^9$ to $\mathbb{R}^7$.

Example time-domain responses of the evaluation are shown in Figure (6). Adequate attitude tracking can be observed for both θ and ϕ, while β is successfully regulated within $[-1°, 1°]$, for all realizations of the agents. At $t \approx 10$ (s), the pitch error increases as banking is initiated.

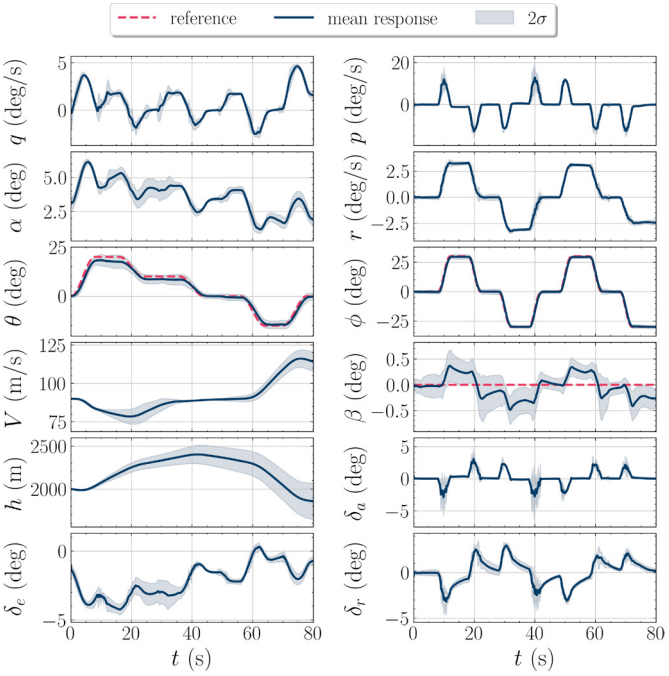

Fig. 6. Mean time-domain response of risk-averse DSAC agents with ∼95% (2σ) confidence interval.

4.2 Risk-sensitive learning

In order to investigate the effect of risk-averse learning on the synthesized control law, the agents are evaluated on a high-risk task to follow a sustained pitch-up manoeuvre to near-stall, extreme flight conditions. Such a situation is chosen for two reasons. Firstly, it is expected that the uncertainty of such conditions is high, due to the lack of exploration and the unobservable dynamics that depend on airspeed and altitude. Secondly, such situations connect the uncertainty of return directly to *flight risk*, as stall conditions are considered hazardous and may lead to loss-of-control.

The responses of both risk-neutral and risk-averse agents are shown in Figure (7), depicting only the longitudinal states of the system ($\phi_r = 0$). The sustained high pitch reference causes the aircraft to lose airspeed and gain altitude. The 45° pitch reference at 60 (s) is unattainable without entering stall-induced oscillations and instability. The risk-neutral DSAC agent responds by further deflecting the elevator, inducing undesirable oscillations. On the other hand, the risk-averse agent avoids the stall-induced oscillations and keeps the angle of attack at 10°. The risk-neutral agent loses altitude, whereas the risk-averse agent maintains a stable climb.

Figure (7) also shows that while the risk-averse agent achieves higher rewards compared to the risk-neutral agent

[1] https://github.com/peter-seres/dsac-flight

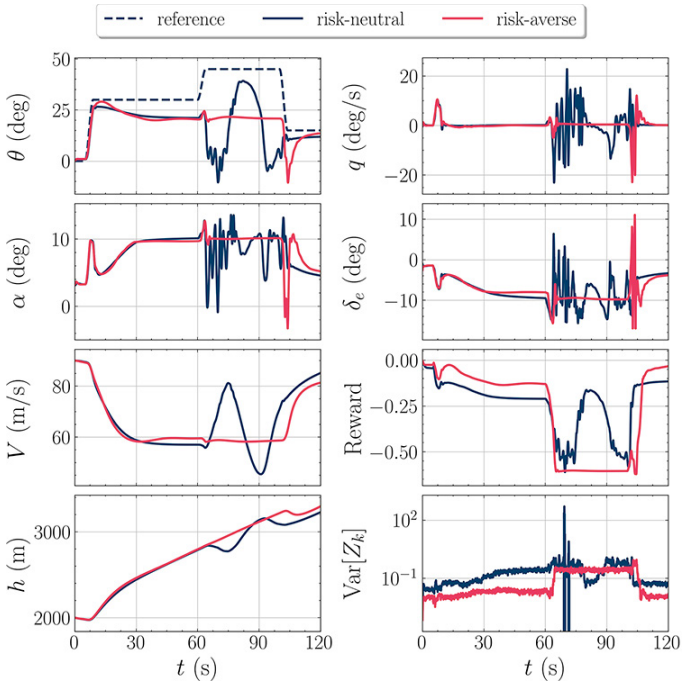

Fig. 7. Near-stall response of risk-neutral (blue) and risk-averse (red) DSAC agents.

in the first half of the episode, it chooses more conservative actions following the 45° pitch reference at 60 (s), resulting in a decrease in rewards. The risk-averse agent achieves a reduced end-of-episode return, but manages to avoid stall-induced oscillations. It is shown that the variance can be used as a metric of uncertainty, and that the risk-averse controller sacrifices immediate rewards to avoid states with high uncertainty, thus increasing the safety of the flight.

This risk-averse behaviour is achieved without the addition of human-domain knowledge, e.g. reward shaping. Whether the critic's estimate of uncertainty is due to parametric or intrinsic uncertainty in the environment is not pertinent to the safety of control and decision making. Instead, in order to reduce flight risk, both unexplored and highly stochastic state-action pairs are to be assigned a lower risk-distorted action-value to avoid uncertainty.

5. CONCLUSION

This research contributes to the synthesis of risk-averse model-free flight controllers for non-linear fully-coupled aerospace systems, and lays the foundation for DRL flight controllers that approximate the uncertainty of the environment. We show that the DSAC algorithm significantly improves learning consistency by reducing the variance of returns, while achieving similar tracking performance. In addition, we show that training risk-averse policies results in control laws that prioritize state-action pairs with low uncertainties, indicated by smaller variance in the return distribution.

However, the DSAC algorithm used in this paper does not utilize the return distribution estimate post training, although it is demonstrated that the variance of the return is a valuable predictor of flight risk. Future work is needed to make use of the trained critic networks, in a potentially adaptive, continually learning setting.

To sum up, the risk-sensitive behaviour in this paper is achieved through goal-oriented interaction. The risk-averse agents avoid both intrinsic and parametric uncertainty, which inherently increases the safety of RL-based flight control.

REFERENCES

Bellemare, M.G., Dabney, W., and Munos, R. (2017). A Distributional Perspective on Reinforcement Learning. ICML. https://arxiv.org/abs/1707.06887.

Dabney, W., Ostrovski, G., Silver, D., and Munos, R. (2018). Implicit Quantile Networks for Distributional Reinforcement Learning. PMLR. http://arxiv.org/abs/1806.06923.

Dally, K. and Van Kampen, E. (2022). Soft Actor-Critic Deep Reinforcement Learning for Fault Tolerant Flight Control. In AIAA SCITECH. San Diego, CA & Virtual. https://arc.aiaa.org/doi/10.2514/6.2022-2078.

Fujimoto, S., Hoof, H., and Meger, D. (2018). Addressing Function Approximation Error in Actor-Critic Methods. In ICML. PMLR. https://proceedings.mlr.press/v80/fujimoto18a.html.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S. (2019). Soft Actor-Critic Algorithms and Applications. https://arxiv.org/abs/1812.05905.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. https://arxiv.org/abs/1509.02971.

Liu, C., van Kampen, E., and de Croon, G.C.H.E. (2022). Adaptive Risk Tendency: Nano Drone Navigation in Cluttered Environments with Distributional Reinforcement Learning. https://arxiv.org/abs/2203.14749.

Ma, X., Xia, L., Zhou, Z., Yang, J., and Zhao, Q. (2020). DSAC: Distributional Soft Actor Critic for Risk-Sensitive Reinforcement Learning. https://arxiv.org/abs/2004.14547.

Mysore, S., Mabsout, B., Mancuso, R., and Saenko, K. (2021). Regularizing Action Policies for Smooth Control with Reinforcement Learning. IEEE ICRA. https://arxiv.org/abs/2012.06644.

Pollack, T. and Van Kampen, E. (2022). Robust Stability and Performance Analysis of Incremental Dynamic Inversion-based Flight Control Laws. In AIAA SCITECH. San Diego, CA & Virtual. https://arc.aiaa.org/doi/10.2514/6.2022-1395.

Teirlinck, C. and Van Kampen, E. (2022). Reinforcement Learning for Flight Control: Hybrid Offline-Online Learning for Robust and Adaptive Fault-Tolerance. Msc Thesis, TU Delft. http://resolver.tudelft.nl/uuid:dae2fdae-50a5-4941-a49f-41c25bea8a85.

Van den Hoek, M., de Visser, C., and Pool, D. (2018). Identification of a Cessna Citation II model based on flight test data. In Advances in Aerospace Guidance, Navigation and Control. Springer. https://doi.org/10.1007/978-3-319-65283-2_14.

Wang, S.S. (2000). A class of distortion operators for pricing financial and insurance risks. Journal of risk and insurance, 15–36.