Lateral gust avoidance via Reinforcement-learning control

Akshat Jain
5260183
https://github.com/akstamimech/LateralGustAvoidance.git

Introduction

The following report explores a possible bio-inspired intervention against prolonged exposure to pockets of lateral clear-air turbulence (CAT). CAT has perplexed meteorologists as well as aerospace engineers for a number of years, due to it's lack of detectability as well as possible intensity [1]. Turbulent wind patterns form where two layers of air meet while moving at different speeds, where the movement of air between these masses causes erratic wind patterns in pockets. CAT occurs in clear atmospheres, at altitudes in the upper troposphere and is expected to grow stronger and more prevalent as results of climate change, with severe events of CAT growing up to 150% as CO2 in our atmosphere doubles [6]. Moreover, Aircraft pilots often lack the visual cues that depict possible turbulent atmospheric conditions and a sudden encounter can provide significant stress to the airframe, making it worthwhile to find interventions against prolonged exposure to CAT.

There are many deeper explanations for the causes of CAT, primarily within the domain of instabilities within layers of air with different velocities, such as the collapse of the Kelvin-Helmholtz instability wave [2], but ultimately, it's effects are local in nature. Considering this and the fact that turbulence is observable post experience through accelerations on the chassis of the aircraft, a possible solution covered in this report is to change position as soon as effects are first noticeable. For instance, if there are pockets of lateral gusts that the aircraft is due to face, then as soon as lateral acceleration due to gust is noticeable, elevator deflection allows for change in pitch, and as a result altitude. The problem can then be complicated further towards real-world conditions, by adding higher dimensional turbulence, better aircraft dynamics.

What complicates the problem further is that it is preferable for the purposes of navigation that the aircraft stay within a given lateral corridor. There is essentially a threefold problem at hand: Avoid high turbulence regions by changing flight level, and stay in the given lateral corridor, all while ensuring passenger safety and comfort by maintaining safe pitch angle levels.

A control strategy for immediate responses to CAT needs to be robust in order to meet the objectives in an uncertain environment, which provides inspiration to use reinforcement learning as an approach, such that the controller ”learns” the cues that describe entering a high turbulence zone. In this study a reinforcement learning controller was allowed to learn a policy of action, such that we allow for stable flight with the aircraft returning to trim eventually, while avoiding pockets of asymmetric turbulence. It was also ensured, for the sake of comfort of the passengers that the pitch angles are not extremely high (a maximum of 14 degrees is preferable [7]). Moreover we would like to reach our goal of traversing our lateral path with the least number of required steps, and to not make any unnecessary detours. The policy was rewarded through a reward function with many facets as will be discussed further, as well as a soft Actor-critic approach in off-line training, which provided path exploration with variety.

If the goal was purely to travel on an optimized path in a field of turbulence, reinforcement learning would not be a preferred method. However a few points complicate this problem a bit further. Firstly, it is assumed through the nature of CAT that the distribution of turbulence is unknown to the aircraft beforehand, it is ”flying blind” in terms of turbulence. Secondly, in reality, the location of turbulence pockets are not static, rather they are quite dynamic, making path optimization quite difficult.

Background information

Aircraft modelling and assumptions

Due to limitations of time and cost, there are some assumptions that are considered. While the simulation is in 3D space, in the body frame of the aircraft, the turbulence acting on the aircraft is laterally constant. Moreover, the aircraft dynamics used in this study are linearized and set to a specific trim velocity. As a result, the simulation is also restricted to an environment within ±200 meters of the trim altitude. That being said, it is not difficult to expand upon the method of simulation if more time is available. The set of equations used to represent model dynamics are essentially the product of the asymmetric equations of motion and the dryden filter[3] attached to the symmetric equations of motion:

$$\dot{x} = \begin{bmatrix} \mathbf{A}_{\text{asym}} \cdot \mathbf{F}_{\text{Dryden}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{\text{sym}} \end{bmatrix} x \quad + \mathcal{B}u \tag{1}$$

The parameters involved in the linearized model of the Cessna Ce500 at cruise altitude were taken from the Flight Dynamics

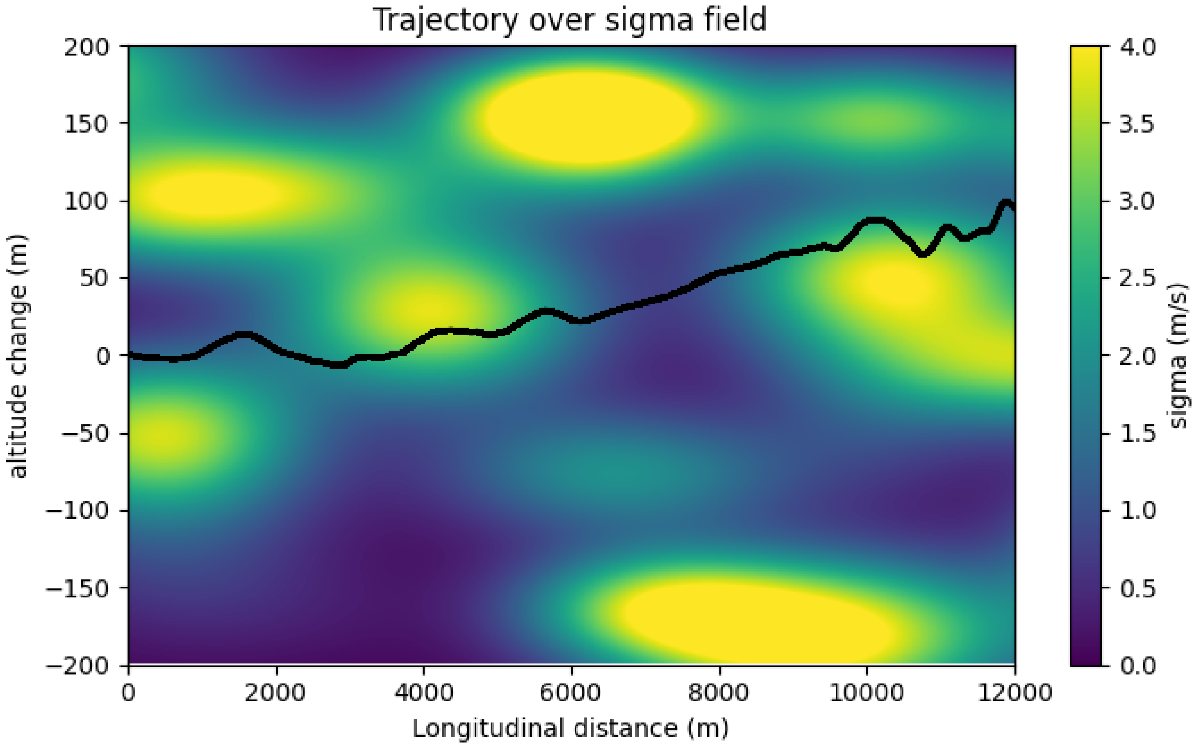

Figure 1: A working RLC flight controller avoiding turbulence.

course material at TU Delft[4]. The integral scale of turbulence, L_g was maintained at 150 meters. The aircraft also has natural oscillatory modes that are very mildly stable, that have been corrected for via simple gains. This results in the closed loop system as in figure 2 below.

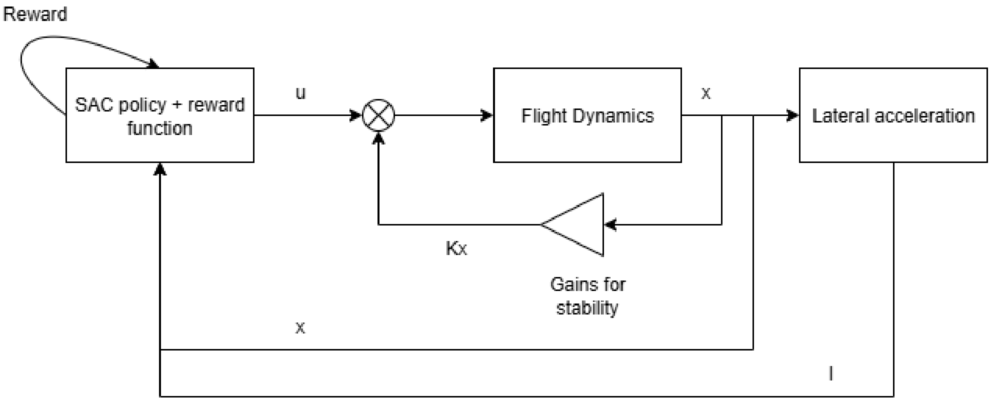

Figure 2: Block diagram of the closed loop system. States x and measured lateral acceleration (root-mean-squared over time window) I feed into the agent, which learns and produces an action u. This action feeds into the state-feedback closed-loop flight dynamics.

Environment Design, gust modeling

Turbulent gust, is by nature, turbulent and varies continuously unpredictably. However, over a scale of time and multiple samples,the frequency of turbulent gust follows a spectrum in frequency space (dryden/ von Karman). It is possible to generate an emulation of gust by using shaping filters (the Dryden spectra was used in this assessment), driven by white noise. As such, lateral gust v_g of root mean square intensity σ is described by:

The gust inputs are then fed into a linearized model of a Cessna Citation ce500. Since there is a direct proportionality of the intensity in turbulence faced and σ, it was possible to define a scalar field $\sigma(x, z)$ over the aircraft's longitudinal plane. The RL environment then provides the state values, sideslip angle β, roll angle ϕ, roll rate p, yaw rate r, longitudinal gust state u_g, longitudinal gust rate $\dot{u}_g$, angle-of-attack gust state a_g, angle-of-attack gust rate $\dot{a}_g$, lateral gust state b_g, lateral gust rate $\dot{b}_g$, forward-velocity perturbation u, vertical-velocity perturbation w, pitch angle θ, and pitch rate q. Attached, and also part of the state space are the position states, x, z, y.

The first layer of training was performed on a proof-of-concept distribution the aircraft travels in an unrealistic field, with a high sigma outside of a sinusoidal valley. The goal is for the aircraft to avoid high lateral acceleration as much as possible, and training on the proof of concept map was performed in order to understand if the RL controller sufficiently provides control actions on the elevator, aileron and rudder surfaces for when the lateral turbulence intensity changes. For the second layer of training, the environment provided changed every episode (one run through of the environment simulation), and was defined by the some of randomly placed gaussians with random intensity (clipped at 4m/s maximum) to simulate clear air turbulence as well as possible.

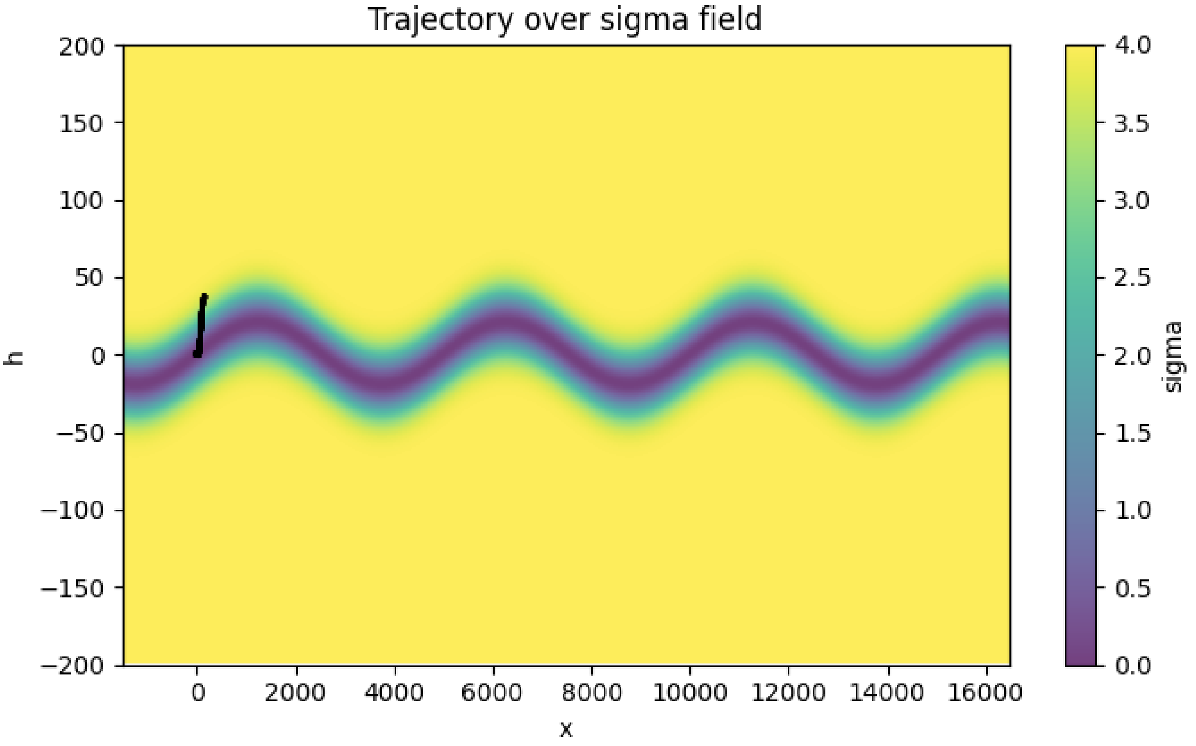

Figure 3: Initial deterministic turbulence field for proof-of-concept.

By traveling along the trim path, in steady wings level flight, the aircraft then senses lateral acceleration, measured by:

$$I_{rms} = \sqrt{(V_{trim}(\dot{\psi} + \dot{\beta}))^2} \tag{2}$$

as can be seen in figure 4. The mean value was taken from a history maintained of the past three values measured in time-steps of the lateral acceleration, at every point starting 3 time-steps in. By doing so, the variable I is history-aware and was thought to be useful in creating a controller that is reactive. The aircraft senses this lateral acceleration by taking gyroscopic measurements (through it's IMU), determining the root-mean-square intensity and using that as an environment variable.

State update is performed through a Runge-Kutta integration step in a dead-reckoning format, where the initial states and kinematics are used in order to determine the next state. The lateral acceleration of the next state is then calculated.

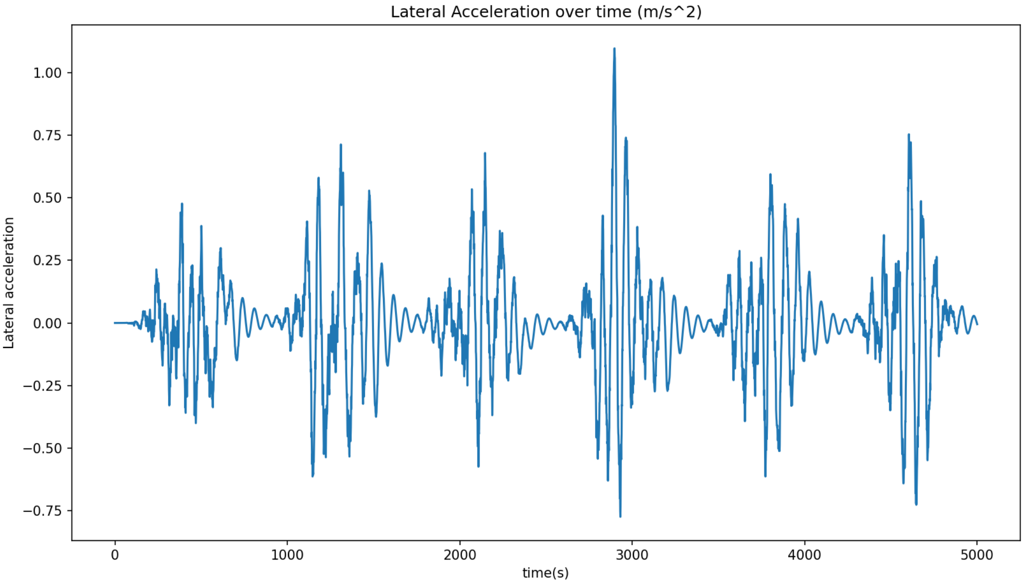

Figure 4: Resultant lateral acceleration in steady, level flight

0.0.1. Reinforcement learning in a continuous setting

The entire idea of reinforcement learning is for an agent to learn a manner of acting, i.e. a policy for an *environment*such that its reward for its behaviour is maximized. The environment is defined as a finite, time-continuous system, defined by the tuple $< X, U, f, \rho >$ where: X is a finite, continuous state space and U is a finite, continuous action space. $f : X \times U \rightarrow X$ is a state transition function and $\rho : X \times U \rightarrow \Re$ is a scalar reward function. The agent then learns a policy:

$$\pi : X \rightarrow U \tag{3}$$

by observing the state space, action space, and reward. The next state is a result of following this policy. It then improves its behaviour from experience of multiple runs.

The goal is then, to learn a policy π that maximizes the rewards earned over the future horizon, considering the current starting point.

$$V^{\pi}(x_0) = \Sigma_{k=0}^{\inf} \gamma^k \rho(x_k, \pi(x_k)) \tag{4}$$

where γ is a discount factor, which is to the power of the index of the next state. γ allows us to encode increasing uncertainty about the future, as well as parametrize how much the expected reward of the future states, actions matter in comparison to the current states and actions. It is beneficial for defining how ”greedy” we would like our agent to be.

Off Policy learning

Before the aircraft experiences real-time clear air turbulence (CAT), it would be best for it to learn from experiencing a variety of turbulent distributions. It was decided to move forward with a method of deep reinforcement learning, which would combine the capabilities of training a neural network to send controller commands, as well as repetitive attempts to reinforce understanding. The soft-actor critic (SAC) controller learns from experience how to maximize its own performance and was implemented as a method for learning.

The SAC algorithm relies on the training of two networks, simultaneously, the *Actor* network, which learns a policy that maps the environment state to an action as in equation 3, as well as a twin - critic network, which is the agent’s estimation of the value function 4 mapping continuous state to an estimated reward. Both of these networks amount to the *agent*.

The core idea is to improve the critic’s estimation of what value function will be, i.e. improve it’s estimation of current as well as future rewards, and use it to execute an optimal policy through the actor. Learning occurs through the actors taking actions and errors being made, where the critic has an approximation of the value function, $V_w(x)$. The way it learns is through Temporal differences, which can be derived from a method similar to bellmans equation:

$$\delta_t = \big(r_t + \gamma V_w(s_{t+1})\big) - V_w(s_t) \tag{5}$$

This error can then be treated like a mean squared error loss in order to update the critic parameters, w. The temporal difference δt is fed directly into the loss function for the actor:

$$\mathcal{L}_{\text{actor}}(\theta) = -\log \pi_\theta(a_t \mid s_t)\,\delta t \tag{6}$$

Both of these networks backpropagate as necessary in order to learn their behaviours. The action space and state space are as a result both parametrized and don't need to be discretized to be used, which is quite beneficial when working with continuous control spaces such as control surface deflection in the Cessna.

At the beginning, the actor network follows a random policy, and seems to take nonsensical actions. This irrational behaviour is beneficial for the purpose of gathering different rewards. This past behaviour is used by the critic network to gain a better understanding of how rewards are distributed in (state, action) space. Then, we can use the training critique, i.e. value function, which is an estimation of future reward, to modify the actor network as need be.

The issue with conventional actor-critic algorithms are largely in exploration, as well as generalization. conventionally, the actor critic setup seems to converge to a local optimum, but may not as a result escape and achieve globally optimal solutions. Moreover it is difficult for the actor network to generalize to unconventional and unseen situations, which are likely to occur in the situation of clear air turbulence. It is difficult to simulate in a usual off-policy all the subtleties that the actor may encounter on-line.

Soft Actor critic (SAC) methods provide value function estimates while Actors behave in a stochastic manner, and requests maximizing the entropy of the followed policy, resulting in more chanced to achieve global optimality and incentive to explore alternative strategies and to recover from unexpected situations.

In SAC, the policy is trained to maximize a trade-off between expected return and a certain level of randomness (i.e. entropy) in a policy [5]. This results in an extension of the problem:

$$\pi^* = \arg\max_\pi \left[\sum_{t=0}^{\infty} \gamma^t \Big(R(s_t, a_t, s_{t+1}) + \alpha\, H\big(\pi(\cdot \mid s_t)\big)\Big)\right] \tag{7}$$

The entropy coefficient α is then smartly reduced over time as a clear control strategy is generated and the actor's behaviour matches the critic's standard. The overall learning pipeline is as described in flowchart 5 below, which is akin to the block diagram in figure 2:

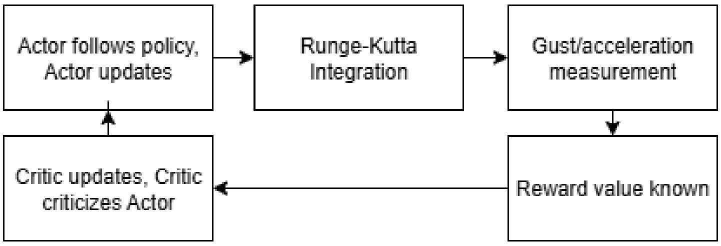
Actor follows policy, Actor updates
Runge-Kutta Integration
Gust/acceleration measurement
Reward value known
Critic updates, Critic criticizes Actor

Figure 5: Flowchart of learning process

Rewarding our aircraft for good behaviour

lateral turbulence avoidance was a surprisingly difficult task to reward engineer for. There were significant attempts by the learning method to reward-hack, i.e. achieve a high reward without appropriately achieving the goals of the controller. The goals were namely to:

1. travel in the longitudinal direction.
2. Not have a pitch angle beyond 14 degrees unless there is high intensity turbulence.

3. Avoiding high variance lateral acceleration as much as possible, and maneuvering towards a low lateral acceleration region.
4. Maintaining lateral position within a certain range.

The reward function that lead to relatively robust results was made of several terms:

$$R^{\pi}(\text{state}) = P_x - P_h + P_{oob} + P_\theta - P_y - P_{vh} - P_{\Delta I} - P_I \tag{8}$$

Where there are rewards for forward longitudinal movement P_x, and *weak* punishments for large altitudes, and pitches, P_h, P_θ to avoid going out of bounds. We also want to avoid extremely rapid climbs and descents for the comfort of the passengers, so we add a very weak vertical velocity punishment P_{vh}. We also add a very strong out-of-bounds penalty P_{oob}. Finally, we add the strong important terms $P_{\Delta I}, P_I$ which define moving towards the region with the least measured root mean square time-windowed lateral acceleration I.

The Rewards/Punishments were defined by:

$$P_x = (w_x \tanh \Delta x \cdot\ e^{-I}), \quad P_{vh} = (w_{vh} \cdot |\Delta h / \Delta t|) \tag{9}$$

A Tanh function was used in the forward velocity reward P_x in order to saturate rewards earned by forward velocity, as well as making it proportional to a decay term based on the current lateral acceleration to make the term turbulence-aware. Without this shape, it was seen that the aircraft would often find it more rewarding to ”push-through” a heavily turbulent patch in comparison to changing pitch, as it would find more reward in doing so. P_{vh} was a standard proportionality.

$$P_h = (w_h \max(0, |h| - 75)^2), \quad P_y = (w_y \cdot \max(0, |y| - 50)), \quad P_{\Delta I} = (w_I \cdot \max(0, \Delta I)) \tag{10}$$

The other position based rewards were based on allowed corridors of exploration. Some leeway was allowed up till 75 meters ascent and descent, as well as 50 meters lateral position change as a result of lateral acceleration, as long as the controller used the control surfaces to stay within these corridors. It was also preferable to only punish increases in lateral acceleration rather than rewarding decreases in the term $P_{\Delta I}$, as the actor would reward-hack by purposely travelling into high turbulence regions in order to harvest a reward by changing elevation again. Due to the behaviour of the Soft Actor-Critic algorithm, even if good behaviour such as traveling in the direction of decreasing turbulence wasn't rewarded, the actor would eventually find itself in a turbulence minima. The goal then is to keep the aircraft transport within the minimum valley. The regular lateral acceleration term P_I was also a standard proportionality.

The pitch punishment was piecewise, because we wanted to punish/reward pitch in a context-aware manner: let $a = |\theta|$ be the absolute pitch angle, with thresholds $F = 0.15$ rad and $M = 0.25$ rad. Define the constants $R_{\text{small}} = 0.01$, $W_{\text{mid}} = 0.01$, and $W_{\text{large}} = 0.10$. Then the pitch term is

$$r_\theta = \begin{cases} W_{\text{small}} \left(1 - \dfrac{a}{F}\right), & 0 \le a \le F, \\ -W_{\text{mid}}\,(a - F), & F < a \le M, \\ -W_{\text{mid}}(M - F) - W_{\text{large}}(a - M), & a > M. \end{cases} \tag{11}$$

Small pitches are rewarded, especially because we want to promote pitch change in order to avoid turbulence. However larger pitches are undesirable and are punished accordingly.

Results and sensitivity analysis

Once it was found that the aircraft was relatively comfortable with following low turbulence paths in the given sinusoidal environment, as can be seen by figure 6, further training was performed by exposure over fields of sums of gaussians with random characteristics, as can be seen by figure 7. The total training amounts to $\approx$ 3.25 million steps, or 90.25 flight hours, done in chunks while altering the reward function weights for better performance. Previous training on inaccurate weights was not discarded, rather training continued after tweaking the weights as best parameters were saved after every set of training.

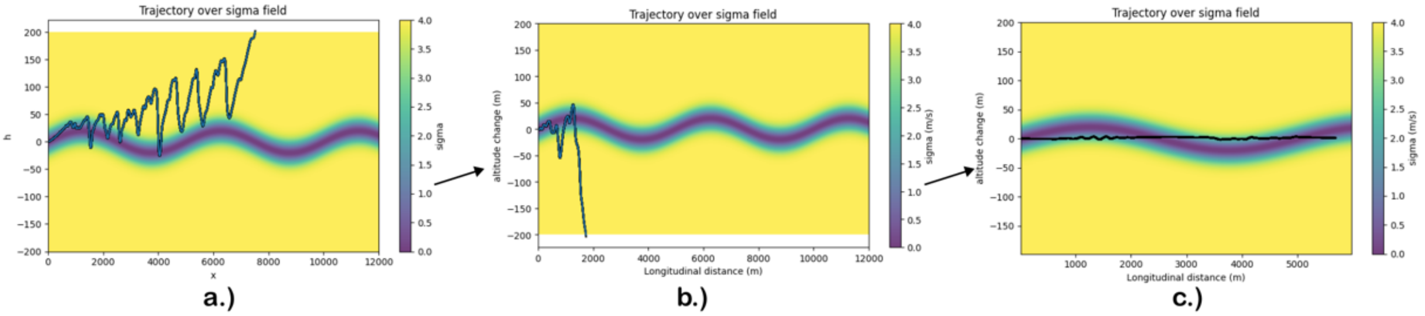

Figure 6: Evolution over time for learning over sinusoidal distribution. Figure a.): Learning for 50000 steps, and the initial reward function. Figure b.): Learning for 200K steps, and changes in the reward function. Figure c.): Learning for 1.25 Million steps, and limiting the possible vertical (z) and lateral(y) positions.

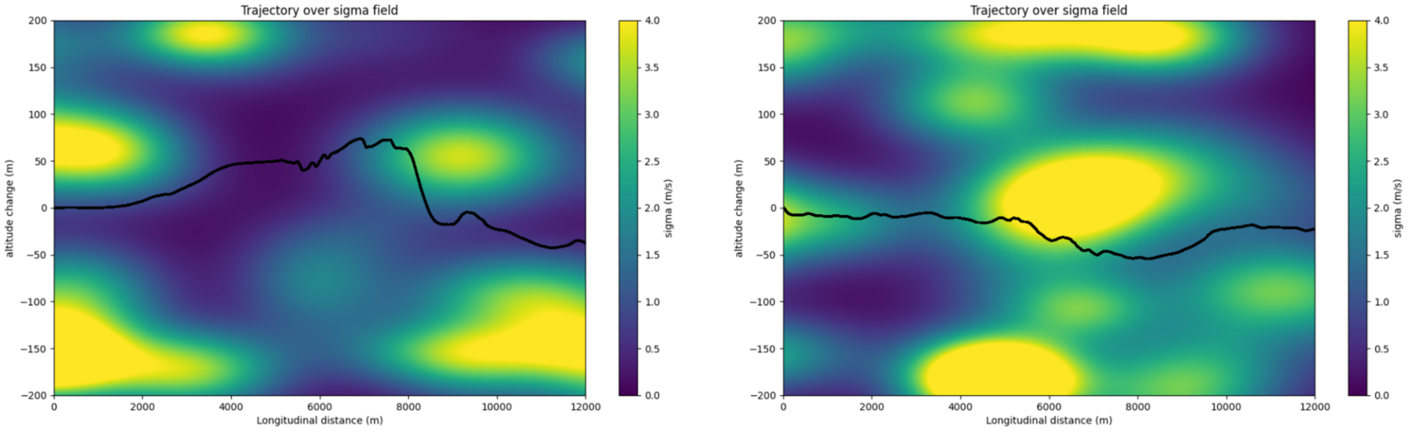

Figure 7: Aircraft successfully traversing a simulated turbulent environment, and taking appropriate altitude changing decisions at the appropriate time.

Total reward behaviour was also recorded as training progressed, for both the sinusoidal distribution as well as random Gaussians, as can be seen by figure 8. For the sinusoidal distribution, rewards over the evaluation instances quickly reached ≈ maximal status. This was good news and was proof that the training was working as proposed, however too much optimality suggests a lack of robustness and inability to adapt to different environments as need be. After further training on Gaussians, the eventual reward per evaluation stayed in the -1000 to -2000 range. The rewards over time in figure 8 are the un-normalized rewards. During training, rewards were **normalized**. The goal of doing so was to ensure that the actions of the agent were rewarded/punished proportionally to the priority of the reward sub-parts, which were of course weighted as required.

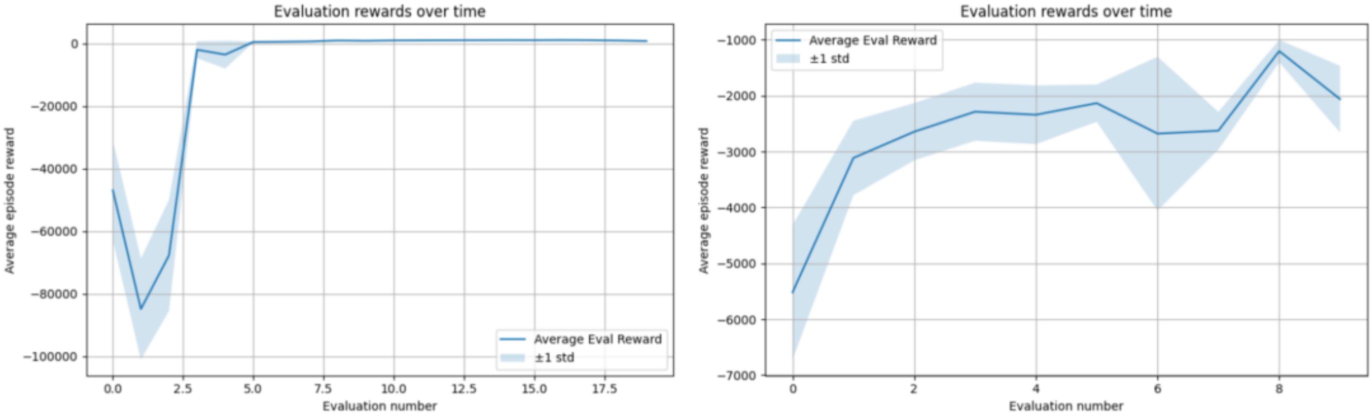

Figure 8: rewards over evaluation instances for sinusoidal distribution **(on the left)**, as well as sum of gaussian distributions (**on the right.**)

Post training, various state variables were extracted as can be seen in figure 10. Reiterating the goals that were mentioned earlier, it was important that pitch angle was maintained below 14 degrees or 0.24 radians. It was also important that lateral position does not deviate significantly from the allotted corridor. These goals were met consistently in every roll-out conducted post training, with the exception of outliers approximately once in thirty attempts, but only by a small margin, such as reaching a lateral position of 54 meters away from trim.

Readings from the control surfaces (aileron, rudder and elevator) in figure 10 show significant noise. This noise comes from attempting to control position in turbulent environments. The RL controller essentially works towards trying to reduce this noise as much as possible, by promoting movement to a less noisy portion of the environment rather than implementing better control over the surfaces.

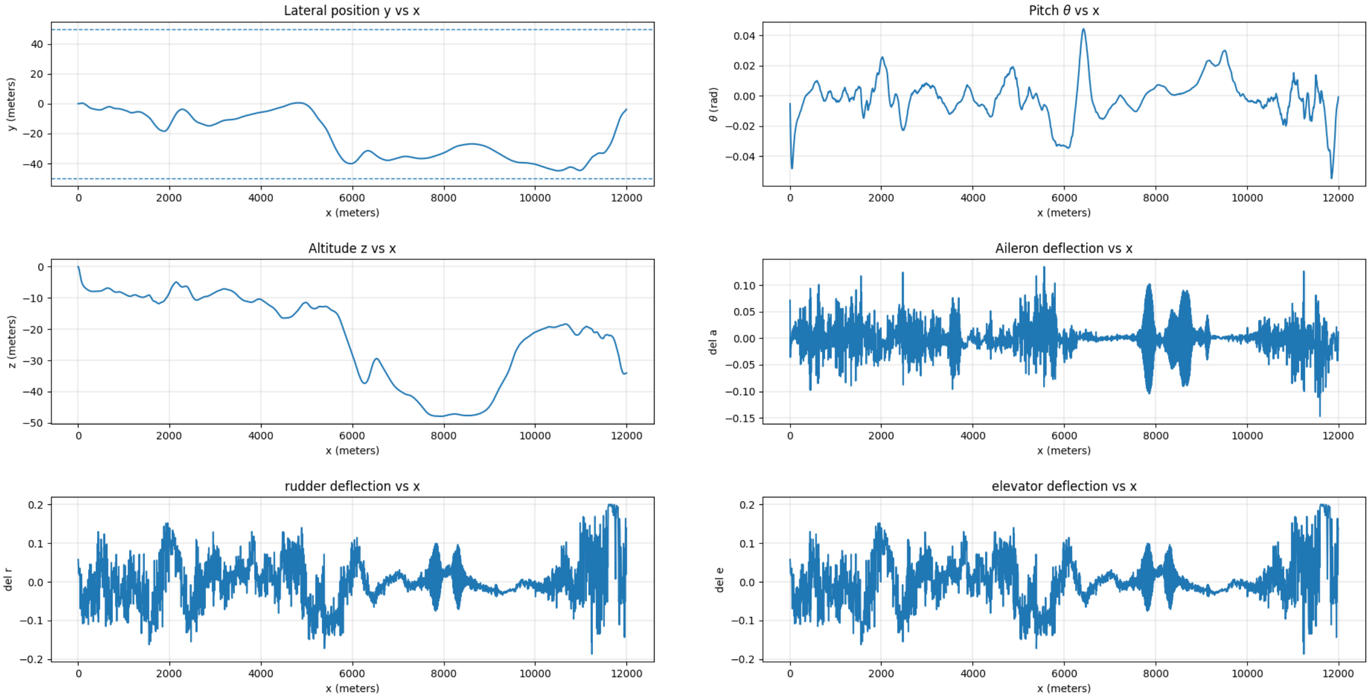

Figure 9: Control surfaces and kinematic changes vs. Lateral Position

The overall control behaviour learned, as can be seen by the paths discovered in figure 7 was reactive in nature, compared to anticipatory. This is expected, the controller does not have any information on the future of the gust states. The best that could be aimed for is for quick intervention and changes made. Making the lateral acceleration variable I history-aware by including the mean value from three past measurements is likely benefiting the robustness of the avoidance mechanism.

All said and done, there are some consistent failures that need to be addressed. The aircraft controller occasionally searches for mildly turbulent regions, in order to gain a reward from then reducing the turbulence. This is reward-hacking behaviour, and although the reward function was corrected against this behaviour, remnant learning from previous iterations still produce this behaviour at a mild level. Fortunately, this can be corrected for with longer training times. There are also certain failure cases where if the region of turbulence is too large, no manouvres can help. The controller is best used before high prevalence.

It was important to identify whether the aircraft controller can meet the demands of the objectives in an ever-growing turbulent environment. Whether the controller is meeting the required objectives or not is essentially quantified as the reward function. By plotting the reward function in figure 10, in comparison to the number of Gaussians used which is a description of the overall concentration of turbulence of the distribution, it is possible to identify how well the aircraft is performing in growing difficult conditions.

While it's clear that the aircraft performs worse in tougher conditions as the average reward grows worse, it doesn't seem to be growing with a strong proportionality (the aircraft's performance worsens at a proportionality less than 1), which implies that it does benefit to have the intelligent controller active in harsh turbulence. Furthermore it may be possible to bring this

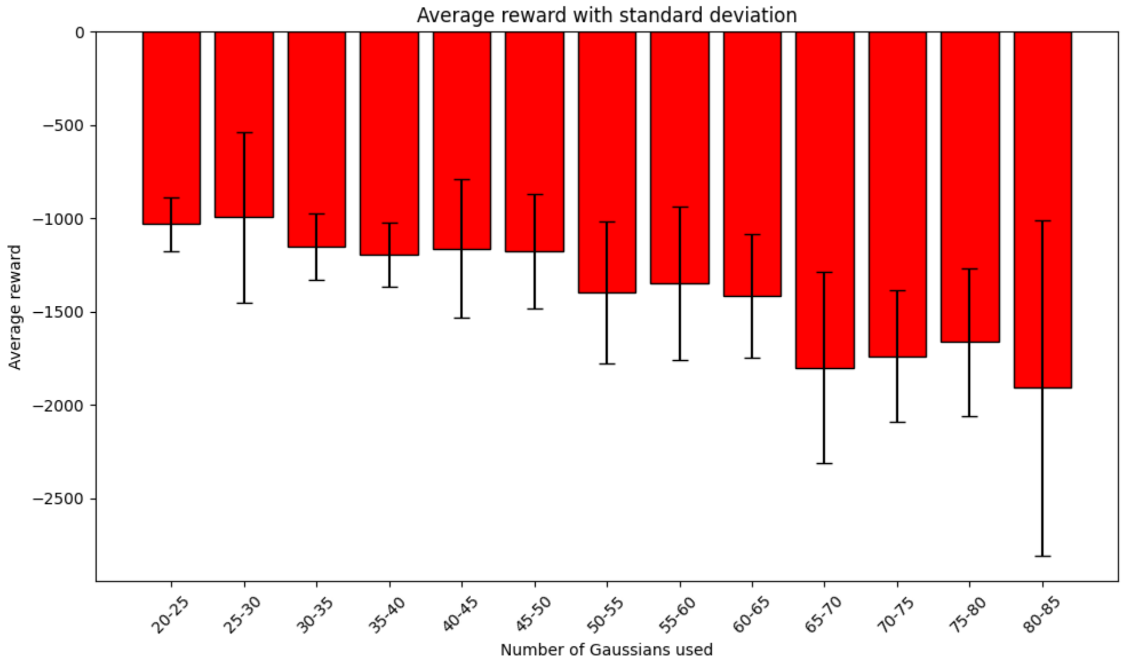

Figure 10: Rewards earned in increasingly concentrated conditions. Rewards were averaged post training and for 15 roll-outs.

proportionality down further, with more training in harsher turbulence. The current training was performed with a sum of between 50-60 Gaussians, and so it would seem that the best performance is below or equal to this value.

From figure 11 it was also clear to see the role that the discount factor played in the learning process. The final model was trained with a discount factor $\gamma = 0.99$, which learned the maximal path relatively quickly. $\gamma = 0.95$ also seemed to do so. Increasing or decreasing the discount factor any further seemed to worsen training performance in the first few evaluations, however it could also be seen the other way: they were exploring different, less rewarding paths that could lead to a more robust path strategy were the situation different. If the study were to be expanded in terms of dimensions of turbulence or more control surfaces, changing the discount factor could be worthwhile to look into. Decreasing the discount factor likely lead to a more greedy policy, i.e. the short term gains mattered much more, meaning that short term control decisions were likely more sensitive to incoming gust. But this would also mean more control effort and variation, leading to less rewards from having a steep pitch angle/going off the lateral path. Increasing the discount factor meant that future decisions mattered much more, leading to less sensitivity of the aircraft for small turbulence regions vs. longer term ones. Not reacting as much in this instance also probably lead to collecting larger negative rewards.

Conclusion

This study showed that a reinforcement learning control aimed towards altitude changes in the face of lateral gust can be an effective intervention for mitigating long term impact. By training on synthetic turbulence fields of increasing complexity, the controller learned to react and adjust altitude and attitude to reduce both passenger discomfort and physical stress on the chassis, all while staying within the constraints prescribed.

The policy was occasionally prone to reward-hacking behaviour, and struggled in high-turbulence regions where it does not have a clear understanding of the field gradient further away, but these problems are solvable through extended training times, sufficient reward engineering and further developing the problem. It is possible to say that RL based controllers are a promising avenue where path planning and obstacle avoidance is needed in a dynamic, unpredictable context. Further improvements could come from using a clearer, nonlinear model, and increasing the dimensionality of the problem, as well as using well defined turbulence models.

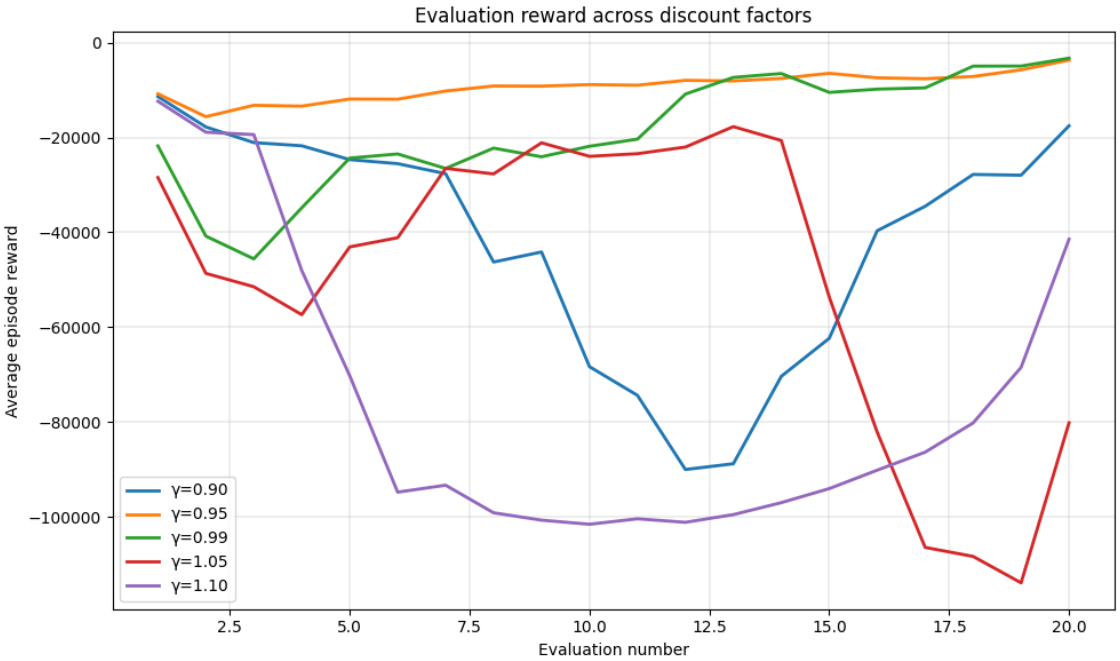

Figure 11: Determining how discount factors affect the growing evaluation reward and learning process.

References

[1] John A. Dutton. “Clear-air turbulence, aviation, and atmospheric science”. In: *Reviews of Geophysics* 9.3 (1971), pp. 613–657. DOI: 10.1029/rg009i003p00613.

[2] “Liutex and Proper Orthogonal Decomposition for Hairpin Vortex Generation”. In: *Liutex and Its Applications in Fluid Dynamics*. Elsevier, 2021. Chap. 6. DOI: 10.1016/B978-0-12-819023-4.00006-9.

[3] J.A. Mulder et al. *Aircraft Responses to Atmospheric Turbulence: Lecture Notes AE4304*. Delft, The Netherlands: Delft University of Technology, Faculty of Aerospace Engineering, 2020.

[4] J.A. Mulder et al. *Flight Dynamics Lecture Notes AE3202*. Delft University of Technology, Faculty of Aerospace Engineering, 2013.

[5] OpenAI. *Soft Actor–Critic — Spinning Up in Deep Reinforcement Learning*. https://spinningup.openai.com/en/latest/algorithms/sac.html. Accessed: 2025-08-16. n.d.

[6] Paul D. Williams and Manoj M. Joshi. “Intensification of winter transatlantic aviation turbulence in response to climate change”. In: *Nature Climate Change* 3.7 (2013), pp. 644–648. DOI: 10.1038/nclimate1866.

[7] Xinhe Yao et al. “Sitting comfort in an aircraft seat with different seat inclination angles”. In: *International Journal of Industrial Ergonomics* 96 (2023), p. 103470. DOI: 10.1016/j.ergon.2023.103470.