Data-Centric Engineering (2024), 5: e47
doi:10.1017/dce.2024.38

RESEARCH ARTICLE

Vortex gust mitigation from onboard measurements using deep reinforcement learning

Brice Martin, Thierry Jardin, Emmanuel Rachelson and Michael Bauerheim

ISAE-SUPAERO, Université de Toulouse, France
Corresponding author: Brice Martin; Email: brice.martin@isae-supaero.fr

Received: 27 October 2023; **Revised:** 07 June 2024; **Accepted:** 02 August 2024

Keywords: vortex gust mitigation; deep reinforcement learning; Kalman filter; aerodynamics; observability

Abstract

This paper proposes to solve the vortex gust mitigation problem on a 2D, thin flat plate using onboard measurements. The objective is to solve the discrete-time optimal control problem of finding the pitch rate sequence that minimizes the lift perturbation, that is, the criterion $J = \sum_t |C_{L,t}|$, where C_L is the lift coefficient obtained by the unsteady vortex lattice method. The controller is modeled as an artificial neural network, and it is trained to minimize J using deep reinforcement learning (DRL). To be optimal, we show that the controller must take as inputs the locations and circulations of the gust vortices, but these quantities are not directly observable from the onboard sensors. We therefore propose to use a Kalman particle filter (KPF) to estimate the gust vortices online from the onboard measurements. The reconstructed input is then used by the controller to calculate the appropriate pitch rate. We evaluate the performance of this method for gusts composed of one to five vortices. Our results show that (i) controllers deployed with full knowledge of the vortices are able to mitigate efficiently the lift disturbance induced by the gusts, (ii) the KPF performs well in reconstructing gusts composed of less than three vortices, but shows more contrasted results in the reconstruction of gusts composed of more vortices, and (iii) adding a KPF to the controller recovers a significant part of the performance loss due to the unobservable gust vortices.

Impact Statement

The development of urban aerial mobility has gained significant momentum in recent years, with notable initiatives such as Airbus' plan to introduce aerial taxis for the Paris 2024 Olympic Games and Amazon's exploration of drone delivery services. Within this emerging field, one critical challenge lies in mitigating disturbances experienced by aerial vehicles. The onboard systems must detect aerological perturbations and minimize their impact on the aerodynamic loads (e.g., lift, drag). We propose a new comprehensive framework harnessing data-driven techniques for detecting perturbations and optimizing control actions to effectively mitigate atmospheric disturbances. This research has far-reaching implications for urban air mobility, as it contributes to ensure safe and reliable operations in the presence of disturbed urban aerological conditions.

1. Introduction

Urban air mobility (UAM) is promised to have a flourishing future in the coming years (Johnson et al., 2018; Straubinger et al., 2020). Johnson et al. (2018) draw the guidelines for future NASA research for UAM development. Among other challenges, such as electric/hybrid propulsion efficiency, safety, and structural integrity, the issue of disturbance rejection by the onboard flight control system is central to

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

UAM development. Current research describes any type of aerodynamic disturbance of an air vehicle as a *gust.* Jones et al. (2022) proposed to categorize gusts into three types: (i) streamwise gusts, (ii) transverse gusts, and (iii) vortex gusts. Streamwise gusts are perturbations parallel to the inflow, transverse gusts are perturbations perpendicular to the inflow, and vortex gusts are bi-directional perturbations stemming from a vortex singularity of circulation Γ_g.

The gust mitigation literature focuses mainly on transverse gusts, yet it is worth noting that these three types of gusts present specific issues that require dedicated mitigation methods.

Firstly, streamwise gusts induce a time-dependant change in effective Reynolds number, which is not critical at high Reynolds numbers but may induce a strongly non-linear lift response to the perturbation at low to moderate Reynolds numbers. This can be viewed as a quasi-steady effect. Moreover, this effect may be modulated by the presence of a streamwise pressure gradient (supported by the temporal perturbation) that affects flow separation, laminar-to-turbulent transition, and reattachment of the separated shear layer, if any, at low and moderate Reynolds numbers. Furthermore, added mass effects and unsteady effects resulting from feedback effects of the developing wake on the airfoil also come into play, typically by introducing attenuation and phase lag to the quasi-steady lift response. The latter are however general to all types of gusts.

Secondly, in addition to added mass and wake effects, as well as potential changes in effective Reynolds numbers, transverse gusts induce a modification of the effective Angle of Attack (AoA) of the vehicle. For example, a transverse gust with an upwash $\tilde{v}$ impinging an airfoil flying at a zero incidence and velocity U_∞ yields an effective AoA perturbation $\tilde{v}/U_\infty$ at first order (Von Karman and Sears, 1938). The main strategy proposed in the literature consists of actively pitching the airfoil to reject the gust-induced disturbance (Andreu-Angulo and Babinsky, 2020, 2021, 2022; Sedky et al., 2020a, 2020b, 2022). For instance, Andreu-Angulo and Babinsky (2021) determine the adequate AoA temporal sequence $\{\alpha_t\}_{t\in[1,T]}$ to mitigate a transverse gust according to low-order models, by minimizing the lift coefficient at each time step, independently of the consequences of choosing α_t on future lift coefficients. Alternatively, Poudel et al. (2021) investigate an open-loop control strategy using oscillating airfoils to mitigate long-lived transverse gusts.

Thirdly, vortex gusts introduce new challenges that could previously be ignored for the mitigation of the two other gust types, which we discuss below:

- Transverse gusts are described by a predefined spatial distribution of upwash disturbances, which are convected at a fixed velocity towards the airfoil, that is, the airfoil pitch motion does not affect the incoming flow perturbation. This made the minimization of the current lift coefficient at every time step by Andreu-Angulo and Babinsky (2021) a reasonable approximation of an optimal control sequence. Conversely, the sequence of AoAs of an airfoil moving through a gust of vortices affects the vortices' future positions, which, in turn, requires accounting for the future system's evolution when choosing α_t, making the vortex gust control more delicate (Jones et al., 2022).
- The main effect of transverse gusts on the lift is due to the modification of the AoA, so that only the upwash perturbation at the leading edge is required to establish the proper control law. This information can be obtained by a simple velocity measurement on the airfoil, yet accurate estimations of AoA in turbulent flows are still an open challenge in practice (Gavrilovic et al., 2018). In contrast, vortices have long-distance effects via their induced velocity on the geometry (Biot–Savart law). This suggests that the position and circulation of the vortices—which are parameters that cannot be directly measured with only onboard sensors (da Silva and Colonius, 2018; Le Provost and Eldredge, 2021)—should be known to compute an optimal control.

While experimental control of a wing in disturbed flow by DRL has been investigated by Renn and Gharib (2022), methods to mitigate vortex gusts are still scarce in the literature. On the one hand, Herrmann et al. (2022) propose a feedforward and feedback strategy controlling the flap of the trailing edge to mitigate the vortex-induced disturbances. On the other hand, Kazarin et al. (2021) aim to solve the robust control problem, which consists of controlling the roll and yaw behavior of an aircraft subjected to unknown and bounded vortex disturbances. However, none of them address the question of the vortex estimation from the onboard measurements. Specifically, Herrmann et al. (2022) estimate experimentally the vortices by a velocity sensor upstream of the airfoil, while Kazarin et al. (2021) consider the vortex gust as an unknown disturbance. To address simultaneously the two difficulties encountered when mitigating vortex gusts, the main objective of our work is to develop a new closed-loop non-linear controller. It is built as an artificial neural network (ANN) that actively pitches a flat plate to mitigate the unsteady lift induced by a vortex gust. To do so, the flow behavior is modeled according to the unsteady vortex lattice method (UVLM). The present work proposes to learn this ANN controller using a deep reinforcement learning (DRL) (Sutton and Barto, 2018) method. DRL is meant to compute optimal controllers for fully observable systems, that is, it requires observing the positions and circulations of the gust vortices at each control time step. To circumvent this issue, a Kalman particle filter (KPF) is proposed on top of the DRL controller to estimate the full state from only onboard measurements.

This paper is organized as follows: Section 2 covers related works which put our contribution in perspective. Section 3 introduces the gust mitigation problem, as well as the UVLM model. Section 4 describes the optimal control problem underlying the mitigation of a vortex gust, as well as the DRL and the KPF algorithms used for its resolution. Finally, in Section 5, we evaluate the performance of the DRL-trained controller, taking the vortices estimated by the KPF as input.

2. Related work

2.1. Deep reinforcement learning for optimal control

In the present work, we model the unsteady lift response to the gust-induced disturbance using the unsteady vortex lattice method (UVLM) (Katz and Plotkin, 2001). The controller is based on an artificial neural network (ANN) (Goodfellow et al., 2016), which is trained to mitigate the vortex gust through DRL (Sutton and Barto, 2018). In a DRL approach, the ANN controller is optimized to mitigate the vortex gust through interactions with the UVLM model. The interaction follows a discrete time evolution. At each interaction step, the controller observes the environment's state s_t. Then, it computes an action a_t, which is applied to the flat plate, triggering a transition in the UVLM model from a state s_t to s_{t+1}. Finally, the controller receives a reward r_t depending on the new state of the environment s_{t+1}. Based on a collection of samples $\{(s_t, a_t, r_t, s_{t+1})\}$, the DRL algorithm optimizes the controller parameters such that the state-control sequence $(s_0, a_0, \dots, s_t, a_t, \dots, s_{T_f}, a_{T_f})$ minimizes the criterion $J = \sum_{t=0}^{T_f} r_t$, where T_f is the final time of the simulation. This optimization procedure allows us

- **to assess the performance loss due to vortex unobservability**. Although DRL theory clearly states that s_t must be a Markov state, that is, contain vortex circulations and vortex positions, we are nevertheless able to train the controller with inputs limited to on-board measurements (non-Markov state) but with no guarantee on the control law optimality.
- **to evaluate multiple levels of control law complexity**. DRL allows us to model the control function as either a simple linear combination of its input or as a complex nonlinear function, modeled as an ANN.

The potential application of ANNs for unsteady fluid dynamics has been established since late 1990 (Faller and Schreck, 1997). In the last few years, ANNs have solved many fluid-related tasks (Brenner et al., 2019; Brunton et al., 2020; Garnier et al., 2021; Brunton and Kutz, 2022). For instance, DRL has been used for a large variety of fluid control tasks, such as reducing the drag coefficient of the flow around a cylinder (Rabault et al., 2019), or diminishing the effort of swimmers leveraging the wake generated by the swimmer ahead (Novati et al., 2017). Regarding aerodynamics and flight control, DRL has been used to solve a variety of aerodynamic control problems. For instance, Waldock et al. (2018) showed that DRL agents are able to manage a landing autonomously, and Bøhn et al. (2019) showed that DRL algorithms have a better ability than standard linear controllers to control the pitch, roll, and yaw of an unmanned

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

aerial vehicle. For the specific question of gust mitigation, DRL offers three advantages over the control method described above. First, DRL is designed to optimize a sequence of actions, which is precisely one of the main difficulties when mitigating a vortex gust compared with transverse perturbations. Second, DRL is able to learn a controller directly from the UVLM model, whereas traditional approaches use low-order models to derive the controller (e.g., Andreu-Angulo and Babinsky, 2021). Third, ANNs allow for an efficient approximation of strongly nonlinear control laws as well as linear control functions.

2.2. Partial observation and state estimation

In the perspective of a real-world deployment, the DRL-trained controller must have full access to the environment state s_t, yet some quantities of s_t, namely the circulations and locations of the gust vortices in the present problem, are not directly measurable with the onboard sensors. This is known as the problem of partial observability, which may lead to performance loss of the controllers (Gunnarson et al., 2021, Paris et al., 2021). The DRL community proposes adding a recurrent ANN (Rumelhart et al., 1986; Elman, 1990; Sherstinsky, 2020) on top of the ANN controller architecture to recover the performance loss due to partial knowledge of the environment state. Hausknecht and Stone (2015), Meng et al. (2021), and Yang and Nguyen (2021) show that this method does indeed recover the performance obtained, compared to when the ANN controller has full access to the environment state. In the specific context of vortex estimation, several strategies are investigated in the literature to determine vortex parameters from the onboard measurements (da Silva and Colonius, 2018; Darakananda et al., 2018; Hou et al., 2019; Fukami et al., 2020; Le Provost and Eldredge, 2021), but these works focus mostly on configurations with downstream wake vortices, rather than incoming gust vortices. A first approach is to learn the correspondence between the unknown vortices and the measurement using end-to-end machine learning techniques (Hou et al., 2019; Fukami et al., 2020). A second considers the vortex parameters as the internal variables x_t of a known transition model $\dot{x}_t = f(x_t)$, and attempts to estimate them from the sensor measurements $y_t = h(x_t)$ using Kalman filters (KFs) (da Silva and Colonius, 2018, Darakananda et al., 2018; Le Provost and Eldredge, 2021). Here f is the UVLM model, and h is the operator computing the pressure on the flat plate. This filtering problem consists in finding an estimation sequence $(\widehat{x}_0, \dots, \widehat{x}_t)$ of $(x_0, \dots, x_t)$ so that the associated measurement sequence $(\widehat{y}_0, ..\widehat{y}_t) = (\mathrm{h}(\widehat{x}_0, a_0), \dots, \mathrm{h}(\widehat{x}_t, a_t))$ fits the actual sequence of measurements $(y_0, .., y_t)$. In the UVLM framework, f and h are both non-linearly dependant on x_t, so the classical KF derived by Kalman and Bucy (1961) is not well suited for the present application. To overcome this issue, Julier and Uhlmann (2004) propose the unscented KF that is able to handle the complete nonlinear model. Nevertheless, the computational cost of this approach becomes quickly untractable when the state dimension increases (da Silva and Colonius, 2018; Le Provost and Eldredge, 2021). The alternative to make the computation tractable lies in discretizing the internal state space with a set of particles. KPF (Ristic et al., 2003; Elfring et al., 2021) pairs each particle with its so-called likelihood probability, i.e., the probability that the particle corresponds to the actual state x_t. In the present study, a particle represent a possible description of the gust, i.e. it is a set $\left(\Gamma_g, \mathbf{x}_{g,1}, \dots, \mathbf{x}_{g,N_v}\right)$, where Γ_g and $\mathbf{x}_{g,i}$ are espectively a possible circulation and location for the i-th vortex. At each time step, the KPF computes $\widehat{x}_t$ as the average of the particles. Concurrently, Evensen (1994) assumes that the probability distribution of x_t remains Gaussian at any time, and introduces the ensemble KF (EnKF). Because the Gaussian probability distribution is defined by its two first moments, the EnKF only computes the particles' mean and covariance to compute $\widehat{x}_t$. The works aiming at estimating wake vortices conducted by Le Provost and Eldredge (2021), da Silva and Colonius (2018) and Darakananda et al. (2018) use EnKF because the number of vortices to estimate is consequent (only the EnKF remains tractable in such a situation). In our work, we prefer to use a KPF to estimate the gust vortices because it is known to achieve better estimation accuracy (Pham, 2001; Weerts and El Serafy, 2006; Le Provost and Eldredge, 2021) than EnKF if the number of particles is high enough. It implies that the number of gust vortices to estimate in our configuration remains small compared to that studied by Le Provost and Eldredge (2021).

3. Problem statement

3.1. The gust mitigation problem

We consider a two-dimensional aerodynamic profile, modeled as a flat plate, flying through a vortex gust consisting of N_v vortices. The flat plate quarter chord is at coordinates $(0,0)$ around which rotation can be performed (see Figure 1). Let c, α and $\dot{\alpha}$ be respectively the chord, the incidence and the pitch rate of the flat plate, $\mathbf{U}_\infty$ the uniform flow velocity parallel to the $\mathbf{e}_x$ direction and ρ the fluid density. The gust vortices are modeled according to the so-called Kaufmann model (Kaufmann, 1962; Bhagwat and Leishman, 2002). It aims at adding a viscous core r_c in a potential vortex, with r_c set to $0.2c$ here. The Kaufmann vortices then behave as potential vortices away from the core, while the fluid inside the viscous core undergoes a solid-like rotation. In this work, the gust consists of $N_{v,x}$ trains of $N_{v,z}$ Kaufmann vortices, with each train being spaced by Δ_x and each vortex having the same circulation Γ_g. We denote the position of the j-th vortex of the i-th train as $\mathbf{x}_{g,i,j} = \left(x_{g,i,j}, z_{g,i,j}\right)$. Such a vortex induces a velocity perturbation at the point $\mathbf{x} = (x,z)$ modeled as $\mathbf{V}_{g,i,j} = V_{g,i,j}\mathbf{e}_\theta$ where $\mathbf{e}_\theta$ is the unit vector normal to $\mathbf{x} - \mathbf{x}_{g,i,j}$ oriented clockwise, and $V_{g,i,j}$ is given as

$$V_{g,i,j}(\mathbf{x}) = \frac{\Gamma_g}{2\pi}\frac{R_g}{R_g^2 + r_c^2} \text{ where } R_g = \sqrt{\left(x - x_{g,i,j}\right)^2 + \left(z - z_{g,i,j}\right)^2}. \tag{1}$$

We define the so-called radius of influence R_I of a vortex as the maximum distance such that the vortex-induced velocity is, arbitrarily, greater than $0.03U_\infty$. In the following, we denote the discrete convective time $t^* = tU_\infty/c$, the normalized gust vortex circulation $\Gamma_g^* = \Gamma_g/(cU_\infty)$, the normalized spacing between the vortices $\Delta_x^* = \Delta_x/R_I$, the flat plate's lift L, the lift coefficient $C_L = L/\left(1/2\rho U_\infty^2 c\right)$, the normalized lift coefficient $C_L^* = C_L c U_\infty/\Gamma_g$, the normalized incidence $\alpha^* = \alpha c U_\infty/\Gamma_g$ and its temporal derivative $\dot{\alpha}^* = \dot{\alpha} c U_\infty/\Gamma_g$.

In this work, we consider six types of vortex gusts, classified in three categories:

- The **one-vortex gust**, the **two-vortices gust**, the **three-vortices gust**, and the **five-vortices gust** (Figure 2a–d). We refer to these gusts as **1**, **2**, **3**, and **5**, respectively. These gusts are composed of vortices of circulation Γ_g. For each of the gusts, the i-th vortex is initialized to the coordinate $\mathbf{x}_{g,i,1} = (-R_I - 0.25c - (i-1)\Delta_x, \Delta_z)$ where $i \leq 5$ and Δ_z is randomly chosen uniformly in $[-0.1c, 0.1c]$.
- The **2×3-vortices gust** (Figure 2e). This gust is composed of two trains of three vortices of circulation Γ_g that are initialized at coordinates $\mathbf{x}_{g,i,j} = (-R_I - 0.25c - (i-1)\Delta_x, (j-1)0.1c + \Delta_z)$ where $i \leq 2, j \leq 3$ and Δ_z is randomly chosen uniformly in $[-0.1c, 0.1c]$. We refer to this gust as **2×3** in the following.
- The **3×2-vortices gust** (Figure 2f). This gust is composed of three trains of two vortices of circulation Γ_g that are initialized at coordinates $\mathbf{x}_{g,i,j} = (-R_I - 0.25c - (i-1)\Delta_x, \pm 0.1c + \Delta_z)$ where $i \leq 3, j \leq 2$ and Δ_z is randomly chosen unifomly in $[-0.1c, 0.1c]$. We refer to this gust as **3×2** in the following.

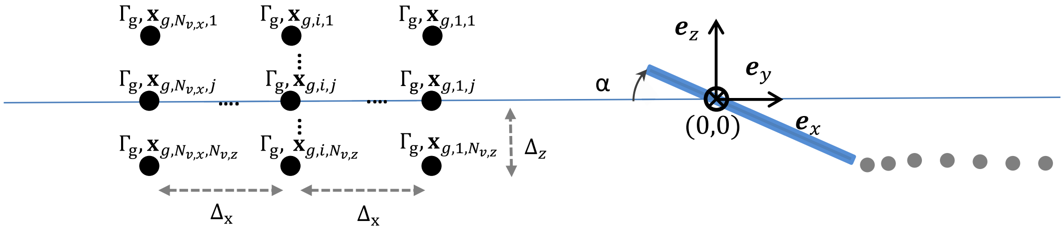
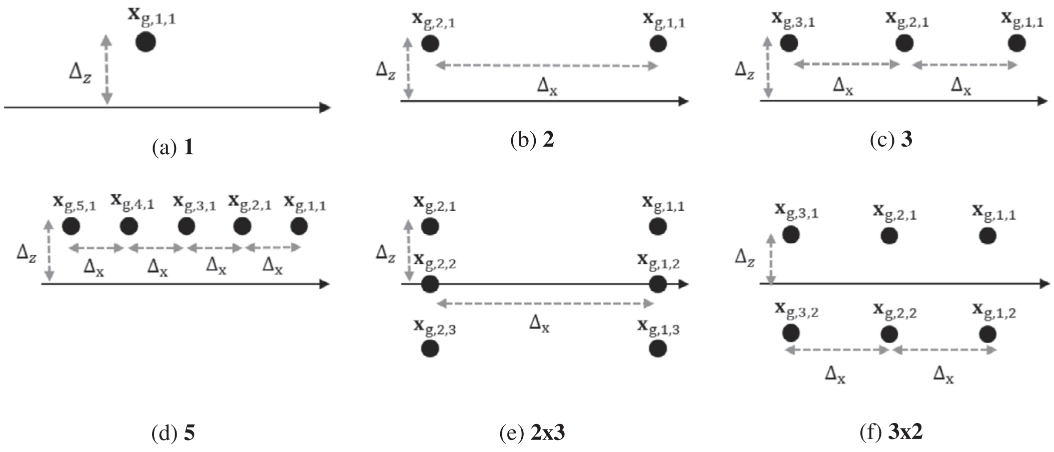

$\Gamma_g, \mathbf{x}_{g,N_{v,x},1}$
$\Gamma_g, \mathbf{x}_{g,i,1}$
$\Gamma_g, \mathbf{x}_{g,1,1}$
$\Gamma_g, \mathbf{x}_{g,N_{v,x},j}$
$\Gamma_g, \mathbf{x}_{g,i,j}$
$\Gamma_g, \mathbf{x}_{g,1,j}$
$\Gamma_g, \mathbf{x}_{g,N_{v,x},N_{v,z}}$
$\Gamma_g, \mathbf{x}_{g,i,N_{v,z}}$
$\Gamma_g, \mathbf{x}_{g,1,N_{v,z}}$
Δ_z
Δ_x
Δ_x
α
$\boldsymbol{e}_z$
$\boldsymbol{e}_y$
$\boldsymbol{e}_x$
(0,0)

Figure 1. *Geometry of the problem.*

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press



(a) **1**

(b) **2**

(c) **3**

(d) **5**

(e) **2x3**

(f) **3x2**

Figure 2. *Sketch of the vortex gusts. Panels (a–d) represent the one-vortex gust, the two-vortices gust, the three-vortices gust and the five-vortices gust, respectively. Panel (e) represents the 2× 3-vortices gust, and (f) represents the 3× 2-vortices gust.*

The objective of this paper is to mitigate the lift perturbations induced by each of these gusts parameterized by Γ_g^* and Δ_x^*. These parameters are chosen uniformly in $[-0.5,0.5]$ and $[1,1.3]$ respectively. R_I is evaluated with the maximum value of Γ_g^*, i.e. $\Gamma_g^* = 0.5$, which corresponds to $R_I/c \approx 3$ (Eq. 1). The control is performed by pitching the flat plate around its quarter chord, at a control frequency $1/dt$, so that the criterion J described as follows is minimal:

$$J = \sum_{0<t^*<T_f^*} |C_{L,t^*} - C_{L,target}|, \tag{2}$$

where T_f^* is the final convective simulation time, and the desired lift coefficient is here chosen as $C_{L,target} = 0$ for the sake of simplicity.

3.2. Environment modeling

The temporal evolution of the gust-induced lift is computed using the UVLM (Katz and Plotkin, 2001). It is a medium-fidelity tool, valid under the assumption of an incompressible potential flow, which in particular, involves an infinite Reynolds number. The method approximates the overall flow as the sum of the incident flow $\mathbf{U}_\infty$ and the induced velocities due to a discrete set of vortex singularities. Among these vortices, N_v are the gust vortices, N are located along the flat plate's chord (bound vortices), and N_w are shed from the Trailing Edge (TE). In the following, we refer to the set of flat plate, wake and gust vortices as $\mathcal{P}$, $\mathcal{W}$ and $\mathcal{G}$ respectively. We respectively denote by Γ_i, $\mathbf{x}_i$ the circulation and the position of the i-th profile vortex, and by $\Gamma_{w,j}$, $\mathbf{x}_{w,j}$ the circulation and the position of the j-th wake vortex. The flow induced by a single gust vortex is modeled as in Eq. 1, whereas the flow induced by the i-th vortex of the profile (resp. of the wake) $\mathbf{V}_i(\mathbf{x})$ (resp. $\mathbf{V}_{w,i}(\mathbf{x})$) is obtained using the Biot-Savart law:

$$\mathbf{V}_i(\mathbf{x}) = -\frac{\Gamma_i}{4\pi}\int \frac{(\mathbf{x}-\mathbf{x}_i)\times \mathbf{dl}}{\|\mathbf{x}-\mathbf{x}_i\|^3}. \tag{3}$$

At each time step, we seek the circulation of the profile vortices so that the non-penetration condition is satisfied:

$$\left(\sum_{i\in\mathcal{P}}\mathbf{V}_i(\mathbf{x}) + \sum_{i\in\mathcal{W}}\mathbf{V}_{w,i}(\mathbf{x}) + \sum_{i\in\mathcal{G}}\mathbf{V}_{g,i}(\mathbf{x}) + \mathbf{U}_\infty\right)\cdot\mathbf{n} = \mathbf{V}_r(\mathbf{x})\cdot\mathbf{n}, \tag{4}$$

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

with $\mathbf{x} \in \mathcal{P}$ and $\mathbf{n}$ is the normal to the flat plate. $\mathbf{V}_r$ is the velocity induced by the pitch motion about the quarter chord: $\mathbf{V}_r(\mathbf{x}) = \dot{\alpha}\mathbf{e}_y \times (\mathbf{x} - \mathbf{x}_{QC})$ where $\mathbf{x}_{QC} = (0,0)$ is the coordinate of the quarter chord.

The circulation of the profile vortices $\Gamma = (\Gamma_i)_{i \leq N}$ is computed by recasting Eq. 4 into a linear system:

$$A\Gamma = -\left[\left(\mathbf{U}_\infty + \sum_{j \in \mathcal{W}} \mathbf{V}_{w,j}(\mathbf{x}_i) + \sum_{j \in \mathcal{G}} \mathbf{V}_{g,j}(\mathbf{x}_i) - \mathbf{V}_r(\mathbf{x}_i)\right) \cdot \mathbf{n}\right]_{i \in \mathcal{P}}, \tag{5}$$

with A the so-called influence matrix, whose size is (N,N), and the components $a_{i,j}$ represent the velocity induced by the j-th profile vortex on the i-th profile vortex projected along $\mathbf{n}$. Once Γ is computed, the conservation of the circulation (Kelvin condition) is enforced by shedding a new wake element from the trailing edge. Its circulation γ_w is given according to the Kelvin equation:

$$\sum_{i \in \mathcal{P}} \Gamma_i + \gamma_w + \sum_{j \in \mathcal{W}} \Gamma_{w,j} = 0. \tag{6}$$

Note that the wake (and gust) circulations remain constant during the simulation because the viscous dissipation of the vortices is neglected.

The transition between time t^* and time $t^* + dt^*$ is performed by advecting the wake vortices and the gust vortices by the total flow velocity $\mathbf{V}(\mathbf{x}) = \mathbf{U}_\infty + \sum_{i \in \mathcal{P}} \mathbf{V}_i(\mathbf{x}) + \sum_{i \in \mathcal{W}} \mathbf{V}_{w,i}(\mathbf{x}) + \sum_{i \in \mathcal{G}} \mathbf{V}_{g,i}(\mathbf{x})$ using a first-order difference scheme:

$$\frac{c}{U_\infty} \frac{\partial \mathbf{x}_{w,i \in \mathcal{W}}}{\partial t^*} = \mathbf{V}(\mathbf{x}_{w,i \in \mathcal{W}}), \tag{7}$$

$$\frac{c}{U_\infty} \frac{\partial \mathbf{x}_{g,i \in \mathcal{G}}}{\partial t^*} = \mathbf{V}\left(\mathbf{x}_{g,i \in \mathcal{G}}\right). \tag{8}$$

Finally, the lift is computed according to Katz and Plotkin (2001) as:

$$\mathbf{L} = \rho U_\infty \sum_{i \in \mathcal{P}} \Gamma_i + \rho \int_{LE}^{c} \frac{\partial}{\partial t} \Gamma(x,t) dx \mathbf{e}_z, \tag{9}$$

where $\Gamma(x,t) = \int_{LE}^{x} \gamma(x_0,t) dx_0$, γ being the circulation density, and the integration of $\frac{\partial}{\partial t}\Gamma(x,t)$ is performed according to the Simpson's method.

In the present work, the number of profile vortices is $N = 10$ and the convection time increment is set to $dt^* = 0.1$. In Appendix A, we assess our UVLM solver on several test cases consisting of (i) an impulsive pitch motion, (ii) a sinusoidal pitch motion, and (iii) an interaction of the (uncontrolled) airfoil with a vortex disturbance.

4. Methods

We aim to solve the optimal control problem by finding the control law for the flat plate's pitch rate $\dot{\alpha}$ that minimizes the optimality criterion J defined in Eq. 2, when the profile flies through the vortex gusts described in Section 3. In addition, the inputs of the control law are limited to an observation vector o_{t^*} composed of onboard measurements:

$$o_{t^*} = (\Gamma_{LE,t^*}, \alpha_{t^*}, \dot{\alpha}_{t^*}, \mathbf{U}_\infty), \tag{10}$$

where Γ_{LE,t^*} refers to the circulation at the Leading Edge (LE) at time t^*. According to Katz and Plotkin (2001), Γ_{LE,t^*} is related to a pressure measurement as $\Gamma_{LE,t^*} = c\Delta P_{LE,t^*} / (\rho U_\infty N)$, where $\Delta P_{LE,t^*}$ is the pressure difference between the upper and lower surface at the flat plate's LE. We consider that at time $t^* = 0$, the profile has no wake, no circulation, no initial pitch rate and that the angle of attack is set at 0°.

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

4.1. Resolution of the gust mitigation problem

We model this optimal control problem as a finite horizon deterministic Markov decision problem (MDP) (Puterman, 2014). At each discrete time t^*, the controller takes a new $\dot{\alpha}_{t^*}$ based on the state of the environment s_{t^*}, that triggers the transition of the environment state from s_{t^*} to $s_{t^*+dt^*}$ according to the UVLM model described in Section 3.2. To ensure the problem is a MDP, the Markovian property should be satisfied. It implies that determining $s_{t^*+dt^*}$ requires only the knowledge of s_{t^*} and $\dot{\alpha}_{t^*}$, regardless of previous values of s and $\dot{\alpha}$. Thus, according to the UVLM model equations (Eqs. 5–8), s_{t^*} must contain:

$$s_{t^*} = \left[(\Gamma_{w,i}, \mathbf{x}_{w,i})_{i\in\mathcal{W}}, (\Gamma_g, \mathbf{x}_{g,i})_{i\in\mathcal{G}}, \mathbf{U}_\infty, \alpha_{t^*}, \dot{\alpha}_{t^*}\right]. \tag{11}$$

Note that the airfoil vortices $(\Gamma_i)_{i\le N}$ are not part of the environment state, since they are computed from s_{t^*} according to Eq. 5. After each transition, the controller observes a user-defined reward of this transition: $r_{t^*}(s_{t^*}, \dot{\alpha}_{t^*}) = -|C_{L,t^*} - C_{L,target}|$. The resolution of this finite-horizon MDP consists of finding the decision function $\dot{\alpha}(s)$ such that the induced trajectory $d_{s_0} = \left(s_0, \dot{\alpha}(s_0), \dots, s_{T_f^*}, \dot{\alpha}\left(s_{T_f^*}\right)\right)$ minimizes J.

The MDP described above is solved using DRL (Sutton and Barto, 2018) where the control function, i.e the flat plate controller, is an artificial neural network (ANN) (Goodfellow et al., 2016). The input of the ANN controller is the state of the environment s_{t^*} and it outputs a pitch rate $\dot{\alpha}_{t^*}$. The DRL algorithm chosen here is the twin delayed deep deterministic policy gradient algorithm (TD3) (Fujimoto et al., 2018). In parallel to the ANN controller, TD3 learns the value function $J(s_0)$ also as an ANN. Here, these ANNs are fully-connected networks, composed of zero, one, or two hidden layers of 400 neurons each. To optimize its behavior, the ANN controller undergoes a variety of simulated gusts, that is, the gust parameters Γ_g^* and Δ_x^* are randomly and uniformly initialized in $[-0.5, 0.5]$ and $[0.1, 1.6]$. At each convective time increment $dt^* = 0.1$, the ANN controller takes a new $\dot{\alpha}_{t^*}$, triggering a transition in the MDP. The result of this transition is gathered by the algorithm as a sample $(s_{t^*}, \dot{\alpha}_{t^*}, r_{t^*}, s_{t^*+dt^*})$. Then, based on a random batch from the samples collected so far, the parameters of both ANNs are optimized using the stochastic gradient descent algorithm on the TD3 loss functions. The ANN controller being deterministic, a Gaussian exploration noise of mean zero and constant standard deviation σ_n is added to the ANN controller output. In this case, a training run lasts approximately between 8 hours, for the one-vortex gust and 24 hours, for the five-vortex gust, on two CPUs.

4.2. Estimation of non-observable quantities

The ANN controller trained using TD3 must take as input the environment state vector s_{t^*} (Eq. 11), but some variables of s_{t^*}, namely $(\Gamma_g, \mathbf{x}_{g,i})_{i\in\mathcal{G}}$, $(\Gamma_{w,i}, \mathbf{x}_{w,i})_{i\in\mathcal{W}}$ are not part of the available onboard measurements (Eq. 10). To render the problem more realistic, and because the impact of the wake on the dynamics of the environment can be assumed to be negligible compared to that of the gust, $(\Gamma_{w,i}, \mathbf{x}_{w,i})_{i\in\mathcal{W}}$ is not estimated in the present work. We estimate $(\Gamma_g, \mathbf{x}_{g,i})_{i\in\mathcal{G}}$ from o_{t^*} (Eq. 10) using a KPF (Ristic et al., 2003; Elfring et al., 2021). We denote by $\left(\widehat{\Gamma}_g, \widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}$ the estimate of the circulation and position of the gust vortices. We also denote $\widehat{s}_{t^*}$ the environment state estimated by the KPF:

$$\widehat{s}_{t^*} = \left[\left(\widehat{\Gamma}_g, \widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}, \Gamma_{LE}, \alpha_{t^*}, \dot{\alpha}_{t^*}\right]. \tag{12}$$

This $\widehat{s}_{t^*}$ vector features the state variables measured by the onboard measurements as well as the KPF estimate of the gust vortices. As stated in the introduction, the KPF computes $\left(\widehat{\Gamma}_g, \widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}$ as the internal variables of a system of first-order differential equations: $d\left(\widehat{\Gamma}_g, \widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}/dt = \mathbf{f}\left(\left(\widehat{\Gamma}_g, \widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}, \dot{\alpha}_{t^*}\right)$, that corresponds here to the dynamics described by the UVLM (Section 3.2). This system of equations is paired with a measurement operator h that computes the circulation $\widehat{\Gamma}_{LE}$ at the LE, if $\left(\widehat{\Gamma}_g, \widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}$

were the actual vortices: $\widehat{\Gamma}_{LE} = h\left(\left(\widehat{\Gamma}_g, \widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}\right)$. According to Eq. 8, the non-viscous flow assumption and the above wake simplification, $\mathbf{f}$ is given as follows:

$$d\widehat{\Gamma}_g/dt = 0, d\left(\widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}/dt = \mathbf{U}_\infty + \sum_{j\in\mathcal{P}} \widehat{\mathbf{V}}_j\left(\left(\widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}\right) + \sum_{j\in\mathcal{G}} \widehat{\mathbf{V}}_{g,j}\left(\left(\widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}\right), \tag{13}$$

where $\widehat{\mathbf{V}}_{g,j}$ is the flow stemming from the j-th estimated gust vortex (Eq. 1), and $\Sigma_{j\in\mathcal{P}}\widehat{\mathbf{V}}_j$ is the flow stemming from the estimated flat plate vortices. It is computed according to Eq. 3 with the circulation of the estimated vortices $\widehat{\Gamma}$ calculated as in Eq. 5 but neglecting the wake:

$$A\widehat{\Gamma} = -\left[\left(\mathbf{U}_\infty + \sum_{j\in\mathcal{G}}\widehat{\mathbf{V}}_{g,j}(\mathbf{x}_i) - \mathbf{V}_r(\mathbf{x}_i)\right)\cdot\mathbf{n}\right]_{i\in\mathcal{P}}. \tag{14}$$

In particular, the h operator computes $\widehat{\Gamma}_{LE}$ as the component of $\widehat{\Gamma}$ at the LE.

The KPF computes the gust vortex estimation sequence $\left[\left(\widehat{\Gamma}_g, \widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}\right]_{0:t^*}$ such that the associated sequence $\widehat{\Gamma}_{LE,0:t^*} = \left(\widehat{\Gamma}_{LE,0}, \dots, \widehat{\Gamma}_{LE,t^*}\right)$ maximizes its so-called likelihood probability, i.e. $\mathbb{P}\left(\widehat{\Gamma}_{LE,0:t^*} = \Gamma_{LE,0:t^*}\right) = \Pi_{k=0}^{n}\mathcal{N}\left(\widehat{\Gamma}_{LE,kdt^*}, \Gamma_{LE,kdt^*}, \sigma\right)$ where $\mathcal{N}(X; m, \sigma)$ is the Gaussian distribution with mean m and standard deviation σ. We briefly present here how the KPF proceeds to compute $\left[\left(\widehat{\Gamma}_g, \widehat{\mathbf{x}}_{g,i}\right)_{i\in\mathcal{G}}\right]_{0:t^*}$. The KPF discretizes the gust vortex space of size $2N_v + 1$ as a set of N_p particles. At the simulation initialization, the KPF initializes randomly the particles. As the simulation continues, the KPF (i) propagates each particle through $\mathbf{f}$, (ii) computes the LE circulation associated to each particle, (iii) assimilates a new measure of Γ_{LE}, (iv) calculates the estimate. For the KPF implementation, we used the practical implementation provided by Elfring et al. (2021). The only difference is that instead of using a constant value of σ, we set $\sigma = \tilde{\sigma}\max(1e^{-2}, |\Gamma_{LE,t^*}|)$ (see Appendix C).

To summarize, we propose to solve the gust mitigation problem described in Section 3 by training an ANN controller using DRL. During the training, the ANN controller has full access to the gust state $\left(\Gamma_g, \mathbf{x}_{i,g}\right)_{i\in\mathcal{G}}$ and to the onboard measurement o_{t^*} (Eq. 10). However, during deployment, the ANN controller does not have access to the actual gust state, instead, the ANN controller takes as input the environment state estimate $\widehat{s}_{t^*}$ (Eq. 12) performed by a KPF. In the following, we refer to this controller as the KPF controller. A scheme of the KPF controller is presented in Figure 3.

5. Results

In this section, we take advantage of DRL's versatility on both control law's shape and inputs to assess the performance gap between (i) linear controllers and non-linear controllers and (ii) full knowledge of the gust vortices and inputs restricted to onboard measurements. Furthermore, we investigate the recoverability of the performance loss due to partial observations using online estimations of the gust vortices performed by a KPF. In this perspective, we present the performance of the linear and non-linear KPF controllers, described in Section 4, on the gust mitigation problems that consist in mitigating the gust

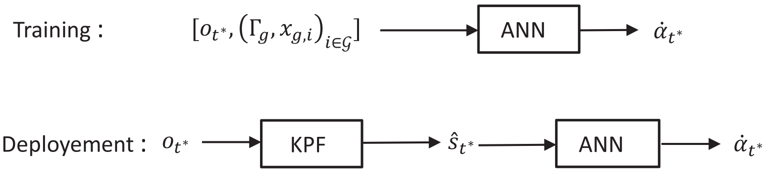


Figure 3. *Scheme of the KPF controller.*

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

configurations **1**, **2**, **3**, **5** described in Figure 2. As a comparison, we also introduce (i) the Full Observation ANN controller (FO controller), that is, an idealized controller taking as inputs the actual values of $\left(\Gamma_g, \mathbf{x}_{i,g}\right)_{i \in \mathcal{G}}$ instead of the KPF's gust reconstruction, and (ii) the Partial Observation ANN controller (PO controller) which stems from training TD3 with inputs limited to the on-board measurements o_{t^*}. Note that the latter does not solve an MDP because o_{t^*} and $\dot{\alpha}_{t^*}$ are not sufficient to compute $o_{t^*+dt^*}$. The corresponding system is a partially observable MDP, for which there is no guarantee that there is an optimal control defined only on o_{t^*}. This section is organized as follows: in Section 5.1, we conduct a cross-analysis of the controller performance regarding its shape complexity and inputs, then in Section 5.2 we present the gust reconstruction and the performance recovery performed by the KPF controller, in Section 5.3 we present in detail the control law learned through DRL and their associated lift mitigation. Finally, in Section 5.4 we extend the discussion to gusts composed of more vortices than estimated by the KPF.

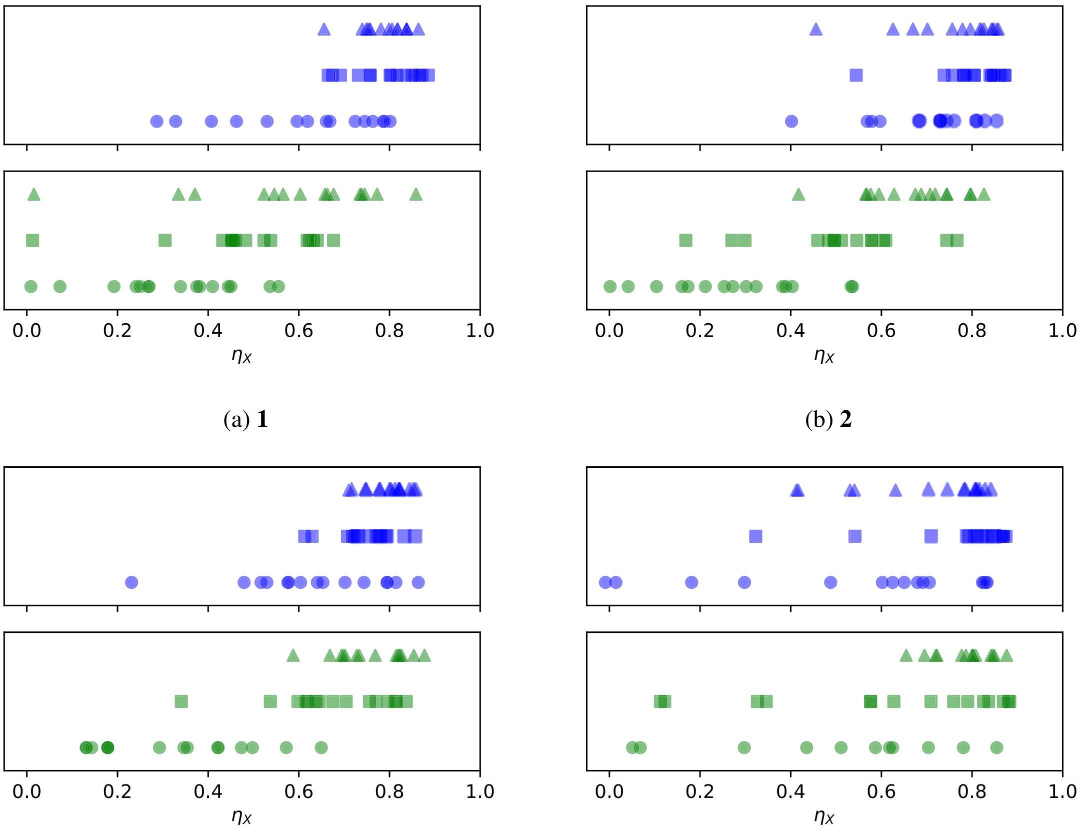
5.1. Shape of the optimal control function

The aim of this section is to explore the characteristics of the optimal control function for the present gust mitigation problem. Specifically, we analyze the complexity of the control function necessary for this problem and examine how the control functions depend on the unobservable gust vortex position and circulation. Three controller architectures are evaluated, with respectively zero, one, and two hidden layers, the hidden layer activation function is ReLu, and the output layer activation function is tanh. Note that the zero hidden layer architecture corresponds to a linear combination of the inputs with the tanh function applied on top. We designate the FO (resp. PO) controller with i hidden layers as the NL_i-FO controller (resp. NL_i-PO controller). The performance of those controllers is evaluated across gust configurations **1**, **2**, **3**, and **5**, as illustrated in Figure 2, with initial gust transverse position set to zero for the sake of repeatability. Note that separate controllers have been trained to mitigate each of these vortex gust configurations. To ensure a fair comparison, each controller is trained using the same protocol: the same number of simulations and the same sequence of initial states. Additionally, the same set of hyper-parameters, as detailed in Appendix B, is used for each training run. These hyper-parameters are inspired by the default parameters in the OpenAI Spinning Up DRL library. It is important to note that the stochastic noise used during ANN optimization can lead to variations in ANN controller performance (Berger et al., 2024). To mitigate this effect, we present the performance of 15 controllers trained with different stochastic noises. These controllers are selected based on the lowest J criterion evaluation during their respective training processes (i.e., the final one is not necessarily chosen). We present in Appendix B, the learning curves for **1** and under NL_0, NL_2 control.

To compare different types of controllers, we introduce the mitigation efficiency as: $\eta_X = (J_{CF} - J_X)/J_{CF}$, where J_{CF} represents the J criterion obtained under control-free conditions, and J_X represents the J criterion obtained using control method X (i.e., NL_i-FO, NL_i-PO, NL_i-KPF). The closer η_X is to one, the more effectively control method X mitigates the uncontrolled lift disturbance. Figure 4 shows η_X for the NL_i-FO, NL_i-PO control methods for each of the 15 controllers across various gusts. Each controller is assigned a single η_X value, which is the averaged performance over $\Delta_x^* = 0.3, 0.6, 1.0, 1.3$ and $\Gamma_g^* = \pm 0.5, \pm 0.4, \pm 0.3, \pm 0.2, \pm 0.1$. It is important to note that the NL_1-FO, NL_2-FO controllers exhibit catastrophic behavior ($\eta_{FO} \sim 0$) for case **5** with $\Delta_x^* = 0.3$ and $\Gamma_g^* \leq 0$. Since this behavior is only observed in a very specific scenario and over a small range of Δ_x^* and Γ_g^*, we do not include these configurations in the results presented here and leave this issue for future investigation.

Linearity of the optimal command law with respect to $[\mathbf{o_{t^*}}, (\mathbf{\Gamma_g}, \mathbf{x_{g,i}})_{\mathbf{i} \in \mathcal{G}}]$. Figure 4 shows that the best linear and non-linear FO controllers achieve mitigation efficiency higher than 0.8 for each of the gust, therefore they are all able to efficiently mitigate the gust-induced lift. For **2**, **3**, **5** the efficiency gap between the best FO controllers seems marginal, but for **1** the mitigation efficiency gap between the best NL_0-FO and NL_1, NL_2-FO controllers is significant, about 10%. Adding complexity to the control function also significantly reduce the variability of performance between training runs. In fact, the variability between η_{NL_2-FO} values (▲) is far less significant than for η_{NL_0-FO} values (●). However, the $NL_2 - FO$ controller

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press



(a) **1**

(b) **2**

(c) **3**

(d) **5**

Figure 4. *Controller performances across the gusts.* ●, ■, ▲, *respectively, represent the efficiency of the* NL_0*-FO controller, the* NL_1*-FO controller, the* NL_2*-FO controller.* ●, ■, ▲, *respectively, represent the efficiency of the* NL_0*-PO controller, the* NL_1*-PO controller, the* NL_2*-PO controller. The efficiency values are displayed for **1** in (a), **2** in (b), **3** in (c), and **5** in (d); each value has been obtained as the averaged* η *for a single controller and over* $\Gamma_g^* = \pm 0.5 \pm 0.4, \pm 0.3, \pm 0.2, \pm 0.1$ *and* $\Delta_x^* = 0.3, 0.6, 1.0, 1.3$.

does not show any clear advantage over the $NL_1 - FO$ controller (■) neither regarding the maximum mitigation efficiency nor its variability. From those results, we conclude that the optimal control function is well represented by an ANN with no hidden layers, even if the DRL algorithm is not able to accurately approximate it at each training run.

Non-linearity of the optimal command law with respect to $\mathbf{o}_{t^*}$. Figure 4 shows that, unlike the NL_0-FO controller, the NL_0-PO controller is not able to mitigate the lift coefficient induced by **1**, **2**, and **3**. Precisely η_{NL_0-PO} ranges, with high variability, between 0.0 and 0.6. Interestingly, for 5, the NL_0-PO controller still observes strong disparity but also observes a significant improvement of the best controller $(\max(\eta_{NL_0-PO}) \sim 0.8$, against $\max(\eta_{NL_0-PO}) \sim 0.6$ for **1**, **2**, **3**), which is a mitigation efficiency comparable to that of the best FO controllers. Unlike the FO controllers, adding non-linearities in the PO controller architecture significantly impacts the maximum efficiency reached by the NL_0-PO controllers. One can see that the $NL_1 - PO$ controller reaches, for **1**, **2**, **3** maximum mitigation efficiencies significantly higher than the NL_0-PO controller but with a comparable disparity. The efficiency loss between the $NL_1 - PO$ controller and the $NL_1 - FO$ controller is still remarkable for **1** and **2** but is less clear for **3**. Furthermore, adding more non-linearities in the PO controller architecture both significantly increases the maximum efficiency and reduces its variability. In fact, one can see that for **1**, **2**, **3** the maximum

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

efficiencies of the NL_2-PO controllers outperform the maximum efficiencies of the NL_1-PO controllers, and the efficiency variability seems to have also been reduced for **2**, **3**, **5**. In addition, one can see that, except for **1**, the efficiency of the NL_2-PO controllers appears similar to that of the NL_2-FO controllers, both in terms of maximum performance and variability. We therefore conclude from these results that the optimal control function can be (i) well approximated by the NL_0-FO controller, or (ii) quite well approximated by a nonlinear combination of o_{t^*}.

5.2. Recoverability of the FO controller performances

Now that the performance of the FO and PO controllers has been outlined, we examine the ability of the KPF controller to reconstruct the vortex gusts and restore the FO controller's mitigation level.

As explained in Section 4.2, the KPF controller estimates the environment state $\widehat{s}_{t^*}$ to compute $\dot{\alpha}_{t^*}$, while the controller is kept the same as the FO controller (no retraining). Consequently, to retrieve the FO controller's level of performance, it is imperative that the KPF estimates the gust vortices as accurately as possible. The estimation of the KPF depends on two parameters: the number of particles N_p and the measurement noise $\tilde{\sigma}$. One can notice that the set of particles at time t^* is included in the initial set of particles convected by the flow (Section 4.2). Thus, to accurately estimate the gust vortices, it is imperative that particles similar to the actual gust vortices (in terms of location and circulation) are represented in the initial set of particles. Therefore, we aim to initialize, within the limit of $4e^6$ particles, a hundred particles similar to the gust vortices, that is, with one-chord uncertainty in the vortex positions and 10% uncertainty in the vortex circulation. Appendix C shows the probability of a particle to be initialized as each gust with a one-chord position uncertainty and a 10% circulation uncertainty. These probabilities vary from $1e^{-2}$ for **1** to $1e^{-4}$ for **3**. For **5** this probability varies from $4e^{-6}$ for $\Delta_x^* = 1.3$ to $1e^{-9}$ for $\Delta_x^* = 0.3$. Consequently, we decide to use $4e^4$, $4e^5$, $1e^6$ and $4e^6$ particles for **1**, **2**, **3**, and **5**, respectively. Note that for **5**, even with the entire particle budget allocated, only a few particles (resp. no particle) are initialized near the actual vortices for the case $\Delta_x^* = 1.3$ (resp. $\Delta_x^* = 0.3$). The measurement noise $\tilde{\sigma}$ is set accordingly, which means that when the vortices are well represented in the particle set, we use a low measurement noise and a large measurement noise otherwise. Precisely $\tilde{\sigma} \sim 0.3$ for **1**, **2**, and **3**, and $\tilde{\sigma} \sim 1.0$ for **5**.

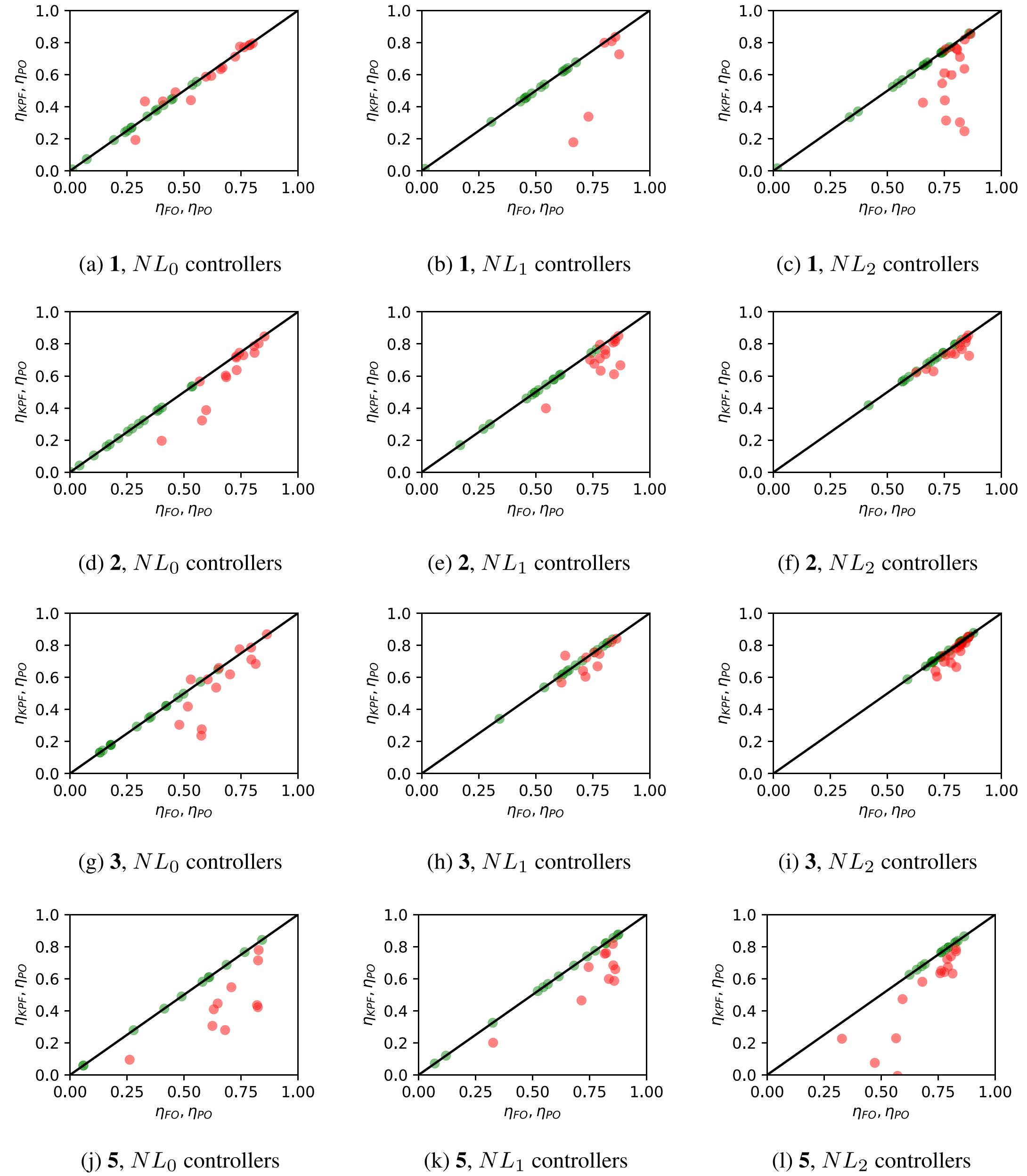
5.2.1. KPF controller's performance recovery

To visualize the KPF controller performance recovery, Figure 5 shows η_{KPF} for each of the three controller architectures depending on the associated η_{FO}, when the controller undergoes **1, 2, 3, and 5** (Figure 5). Each controller is represented by a red dot whose x-component represents η_{FO} and the y-component represents η_{KPF}, so the closer the red dot is to the line $y = x$, the better the KPF controller recovers the mitigation level of the associated FO controller. In addition, we display the efficiency of each PO controller as a green dot located along the line $y = x$. Thus, controllers verifying $\eta_{FO} \geq \eta_{PO}$ see the red dot to the right of the green dot, and controllers verifying $\eta_{KPF} \geq \eta_{PO}$ see the red dot above the green dot. Note that the axis limits are set to $[0,1]$, so controllers with negative η are not displayed.

These results show that adding a KPF to a DRL-trained controller is a good methodology for the problem of mitigating vortex gusts from onboard measurements. Figure 5 shows that, for **1**, **2**, **3**, some KPF controllers recover well the efficiency of the associated FO controller. This translates into a significant improvement in the performance of the best NL_0-KPF controllers and the best NL_1-KPF controllers compared with the best NL_0-PO controllers and the best NL_1-PO controllers. However, no clear advantage of the best NL_2-KPF controllers is found over the best NL_2-PO controllers, but this is mostly because the performance gap between the best NL_2-FO controller and the best NL_2-PO controller is smaller than for the two other architectures.

Even if the best KPF controllers perform well at recovering the FO controllers' mitigation level for **1, 2, 3**, we notice a strong variability in the mitigation efficiency of the KPF controllers. For example, the NL_2-KPF controller recovers well the attenuation level of the NL_2-FO controller for **2** and **3**, whereas most NL_1-KPF controllers perform poorly for **1**. Furthermore, the variability does not seem to be linked to any

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press



(a) **1**, NL_0 controllers (b) **1**, NL_1 controllers (c) **1**, NL_2 controllers

(d) **2**, NL_0 controllers (e) **2**, NL_1 controllers (f) **2**, NL_2 controllers

(g) **3**, NL_0 controllers (h) **3**, NL_1 controllers (i) **3**, NL_2 controllers

(j) **5**, NL_0 controllers (k) **5**, NL_1 controllers (l) **5**, NL_2 controllers

Figure 5. *KPF controllers' performance recovery.* η_{PO}, η_{FO}, η_{KPF} *are displayed, for* ***1, 2, 3, 5*** *under* NL_0, NL_1, NL_2 *controllers.* ● *represent* $\eta_{KPF}=f(\eta_{FO})$ *for each of the 15 controllers and* ● *represent* η_{PO} *displayed along the line* $x=y$ *(——). Each of the* η *value given for* ***1, 2, 3*** *(resp.* ***5****) are averaged over* $\Delta_x^*=0.3,0.6,1.0,1.3$ *(*$\Delta_x^*=0.6,1.0,1.3$*) and* $\Gamma_g^*=\pm 0.1, \pm 0.2, \pm 0.3, \pm 0.4, \pm 0.5$. *Note that the axis' limits have been set to* $[0,1]$ *and therefore controllers with* $\eta<0$ *are not displayed.*

specific gust type or control architecture. For example, the same NL_2-KPF controller has low variability for **2** (Figure 5f), but high variability for **1** (Figure 5c), and **1** can be attenuated either with low variability by the NL_0-KPF controller, or with high variability by the NL_1-KPF controller.

The gust **5** must be viewed from a different perspective than the other three gusts. The first difference is that Δ_x^* plays an important role. Specifically, for the cases $\Delta_x^* = 0.6, 1.0, 1.3$ (Figure 5j–l), the best KPF controllers maintain a reasonable mitigation efficiency since the best η_{KPF} are close to their associated η_{FO}. But in the case of $\Delta_x^* = 0.3$, the performance recovery is significantly reduced compared to the cases with higher Δ_x^*. Furthermore, their performance is inferior to the best PO controllers. The NL_0-KPF controllers also see their recovery capability reduced compared to the other gusts.

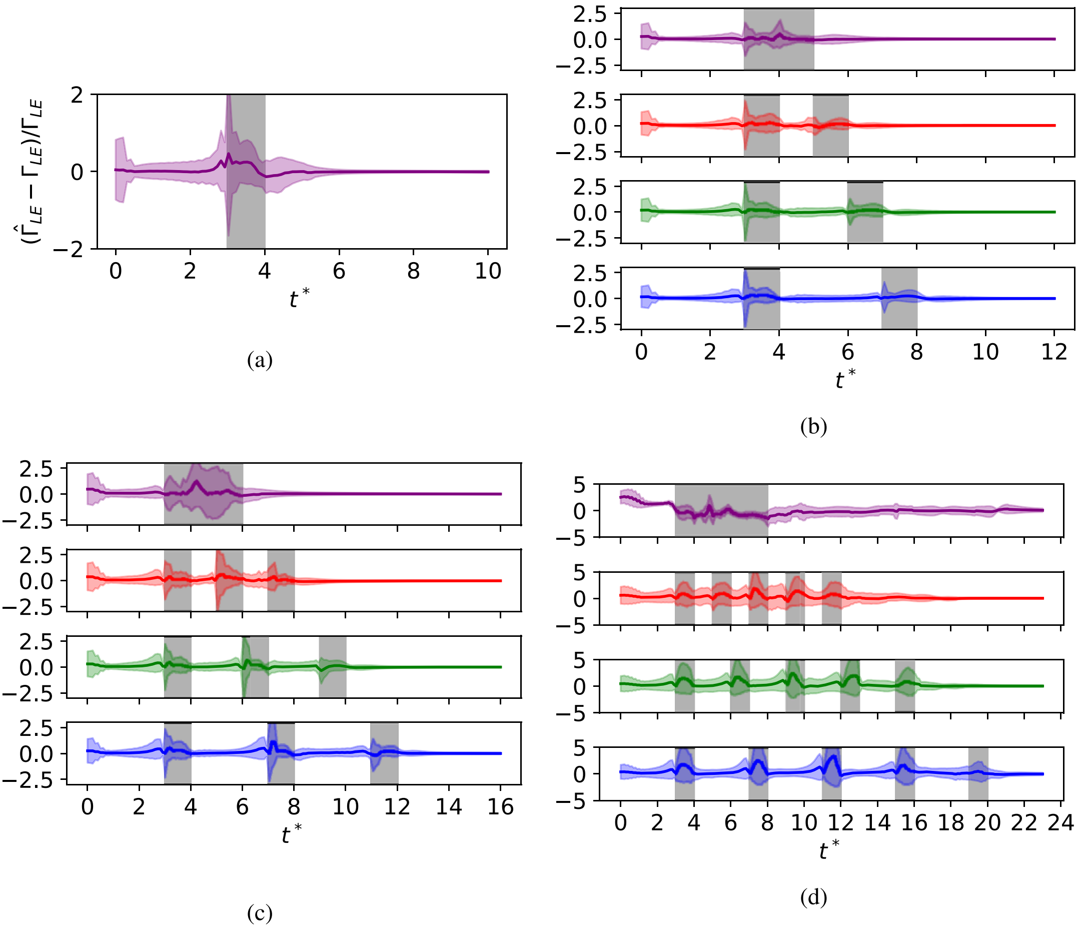
5.2.2. Gust reconstruction

To further analyse the results of the KPF controller, we study the reconstruction of both the circulation at the LE and the position and circulation of the gusts. We present here the reconstructions performed by the five NL_2-KPF controllers, which are built on top of the five NL_2-FO controllers that achieve the lowest J criterion evaluation. (For **1**, we present the reconstruction averaged over only four controllers because the fifth is not representative.) We restrict our results to the best controllers because, as explained in the next section, the accuracy of the estimation can vary depending on the controller chosen. We refer to the position of the i-th gust vortex as $\mathbf{x}_{g,i} = \left(x_{g,i}, z_{g,i}\right)$ and to its KPF estimate as $\widehat{\mathbf{x}}_{g,i} = \left(\widehat{x}_{g,i}, \widehat{z}_{g,i}\right)$.

We display in Figure 6 $\left(\widehat{\Gamma}_{LE} - \Gamma_{LE}\right)/\max(\Gamma_{LE}, 1e^{-2})$ for each of the gusts with $\Delta_x^* = 0.3, 0.6, 1.0, 1.3$ and $\Gamma_g^* = \pm 0.1, \pm 0.2, \pm 0.3, \pm 0.4, \pm 0.5$, where Γ_{LE} (resp. $\widehat{\Gamma}_{LE}$) is the circulation at the LE induced by the actual gust vortices (resp. the estimated gust vortices) under KPF control. Note that the denominator is clipped under $1e^{-2}$, because the measurement used for the particles update is itself clipped under $1e^{-2}$ (see Section 4.2). For **1**, the relative error between $\widehat{\Gamma}_{LE}$ and Γ_{LE} has three remarkable stages (Figure 6-a): (i) for $t^* < 3$ the error is around 20%, (ii) for $3 < t^* < 5$ the error is characterized by sharp variations of high amplitudes, and (iii) for $t^* > 5$ the error is of low amplitude, that is, lower than 5%. The decrease in estimation error throughout the simulation is due to the KPF methodology (Section 4.2). Indeed, when initializing the simulation, the KPF computes $\widehat{\Gamma}_{LE}$ as the LE circulation induced by a gust vortex estimated as the mean of a random distribution (since no data is available.) As the simulation continues, the KPF acquires new observations of Γ_{LE} and refines the estimate of the vortex so that the sequence of $\widehat{\Gamma}_{LE}$ corresponds to the sequence of Γ_{LE}. However, we notice high amplitude errors when the gust vortex is close to the profile ($3 < t^* < 5$ in Figure 6a). We can advance two reasons to explain this error: (i) when the vortices are close to the profile, the velocity perturbation induced by the vortices varies sharply with the position of the vortices, and is therefore more sensitive to estimation errors (Eq. 1), (ii) wake vortices are no more negligible in Γ_{LE}, while they were neglected in the derivation of $\widehat{\Gamma}_{LE}$. The trend observed for **1** is still valid for the other three gusts. However, a singular case is worth to be mentioned. For **5**, the MSE between $\widehat{\Gamma}_{LE,0:T_f^*}/(cU_\infty)$ and $\Gamma_{LE,0:T_f^*}/(cU_\infty)$ is two orders of magnitude higher than for **1**, **2**, and **3**. This result correlates with the discussion of the particle budget above: for **5**, the number of particles is not high enough, which reduces the accuracy of the estimate of Γ_{LE} in this case.

As a complementary result, we also present in Figures 7 and 8 the gust reconstruction performed by the KPF controller. Those reconstructions have been made for $\Delta_x^* = 0.3, 0.6, 1.0, 1.3$ and $\Gamma_g^* = \pm 0.1, \pm 0.2, \pm 0.3, \pm 0.4, \pm 0.5$. The estimation of the gust circulation Γ_g and the estimation of the longitudinal position of the gust vortices $\left(x_{g,i}\right)_{i \leq N_{v,x}}$ are shown. Here, the transverse distance of the vortices from the profile is negligible compared to the longitudinal distance, so we approximate the positions of the vortices to their longitudinal positions and present the estimate of the transverse positions in Appendix D. First, the KPF needs time to accurately estimate the gusts. According to Figures 7 and 8, it estimates accurately **1**, **2** and **3** after 3, 5 and 7 units of convective time respectively. Note that the KPF accurately estimates the gust vortices when they pass by the LE. The reason for this estimation delay is that, at the initialization of the simulation, the estimated gust vortex is considered to be the mean of a random distribution (Section 4.2), and then the KPF assimilates observations of Γ_{LE} to refine its reconstruction. Thus, prior to the convection times mentioned above, the KPF estimates the gust vortices far upstream of their actual positions and with an overestimated circulation. Then, the KPF estimates the vortices with an error on Γ_g of less than 10% and an error on $x_{g,i}$ of less than $0.5c$. Note that

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press



(a)

(b)

(c)

(d)

Figure 6. *Error* $\left(\widehat{\Gamma}_{LE}-\Gamma_{LE}\right)/\Gamma_{LE}$ *for* ***1*** *(a),* ***2*** *(b),* ***3*** *(c), and* ***5*** *(d). The dashed lines represent the cases* $\Delta_x^*=0.3,0.6,1.0,1.3$. $\left(\widehat{\Gamma}_{LE}-\Gamma_{LE}\right)/\max\left(\Gamma_{LE},1e^{-2}\right)$ *has been evaluated for* $\Gamma_g^*=\pm 0.5,\pm 0.4,\pm 0.3,\pm 0.2\pm 0.1$. *Solid lines represent the average quantities, and the colored areas are the associated standard deviations. The gray areas report the presence of a gust vortex near the profile.*

this accuracy holds for each pair Δ_x^*,Γ_g^*. Conversely, the accuracy of the estimate in **5** depends mainly on Δ_x^*. For the case $\Delta_x^*=1.3$ (Figure 8h), the observations made for the other gusts remain valid: each of the five vortices is accurately estimated when the vortices reach the profile. However, these conclusions are not valid for the cases $\Delta_x^*=0.3,0.6$. In particular, for $\Delta_x^*=0.3$, the four leading vortices are only accurately estimated in the last quarter of the simulation and the fifth gust vortex is not estimated (Figure 8f). Note also that the low performance of the KPF controller for $\Delta_x^*=0.3$ observed in Section 5.2.1 correlates with the poor estimation of the gust vortices.

5.2.3. Variability of the KPF controller recovery

Now that the gust reconstruction performed by the Kalman filter has been presented, we further discuss the η_{KPF} variability presented in Section 5.2.1. The η_{KPF} variability on **5** is left out because the particle budget is not high enough and therefore leads to a worse vortex estimation compared to the other gust. To explain the variability in KPF controller performance recovery, we investigate two potential sources of variability: (i) the intrinsic KPF variability depending on Γ_g^*,Δ_x^* and the range of Γ_{LE}^* observed by the controller, and (ii) the variability in the ANN controller optimization. Indeed, it is well known that

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

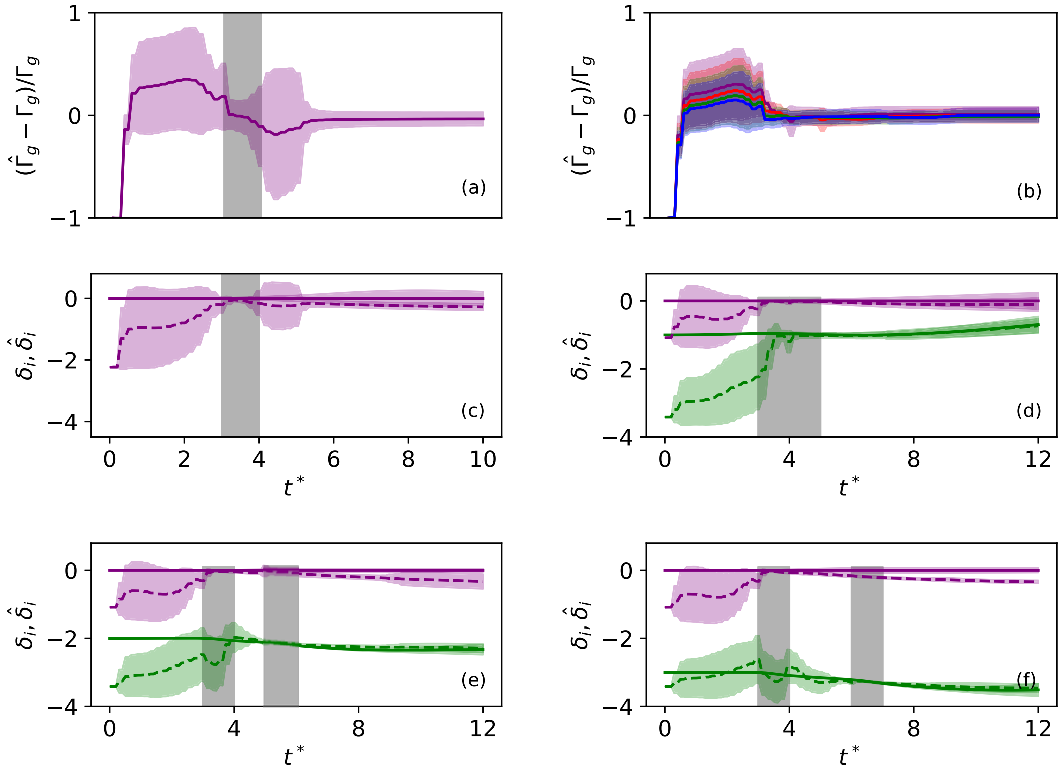


***Figure 7.** KPF's reconstruction of **1** and **2**. The gust intensity estimation error $\left(\widehat{\Gamma}_g - \Gamma_g\right)/\Gamma_g$ is displayed for **1** (a) and **2** (b) with Δ_x^*=0.3 (—), 0.6 (—), 1.0 (—), 1.3 (—). The actual i-th vortex position $\delta_i = \left(x_{g,i} - x_{g,1}\right)/c$ and its estimate $\widehat{\delta}_i = \left(\widehat{x}_{g,i} - x_{g,1}\right)/c$ are displayed for **1** (c) and **2** with $\Delta_x^* = 0.3\,(\mathrm{d}), 0.6\,(\mathrm{e}), 1.0\,(\mathrm{f})$. δ_1, δ_2, $\widehat{\delta}_1$, $\widehat{\delta}_2$ are respectively represented by —, —, - - -, - - -. Each quantity has been evaluated for the five KPF controllers with $\Gamma_g^* = \pm 0.5 \pm 0.4, \pm 0.3, \pm 0.2 \pm 0.1$. Solid lines represent the average quantities and the colored areas are the associated standard deviations. The gray areas indicate the presence of a gust vortex near the profile.*

different training runs can lead to different local minima of the objective function and thus produce ANN controllers with significantly different parameters. We also introduce an objective recovery criterion $r = |1 - \eta_{KPF}/\eta_{FO}|$, the closer r is to 0, the better the KPF controller recovers the associated η_{FO}. The gust **1** is the most symptomatic of the variability of r, because the NL_0-KPF controllers recover the associated η_{FO} well (Figure 5a), while the NL_1-KPF controllers show a high variability in r (Figure 5b). We therefore focus our analysis on this particular gust.

To visualize the potential correlation between KPF controller performance and vortex estimation accuracy, we plot in Figure 9a η_{KPF} under the 15 $NL_1 - KPF$ controllers depending on the *rMSE* between Γ_g^* and $\widehat{\Gamma}_g^*$ and $x_{g,1}^*$ and $\widehat{x}_{g,1}^*$. We chose the $NL_1 - KPF$ controller because it exhibits the highest variations in r (Figure 5b). It is shown that low η_{KPF} are associated with bad gust vortex estimates. To complement this result, we show in Figure 9b η_{FO} under the 15 $NL_1 - FO$ controllers depending on the *rMSE* between Γ_g^* and $\widehat{\Gamma}_g^*$ and $x_{g,1}$ and $\widehat{x}_{g,1}$. This experiment is performed as before, but instead of using the reconstructed vortex to compute the next action, we use the actual vortex. It is shown that there is no particular correlation between η_{FO} under NL_1 control and the accuracy of the estimate. We conclude that the variability in r is not due to the variability of the estimator, since the estimation accuracy is stable for each of the $NL_1 - FO$ controllers.

We now seek to understand to what extent the variability within the 15 $NL_0 - KPF$, $NL_1 - KPF$, $NL_2 - KPF$ controllers explains the variability of r. In this perspective, we emphasize that the KPF

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

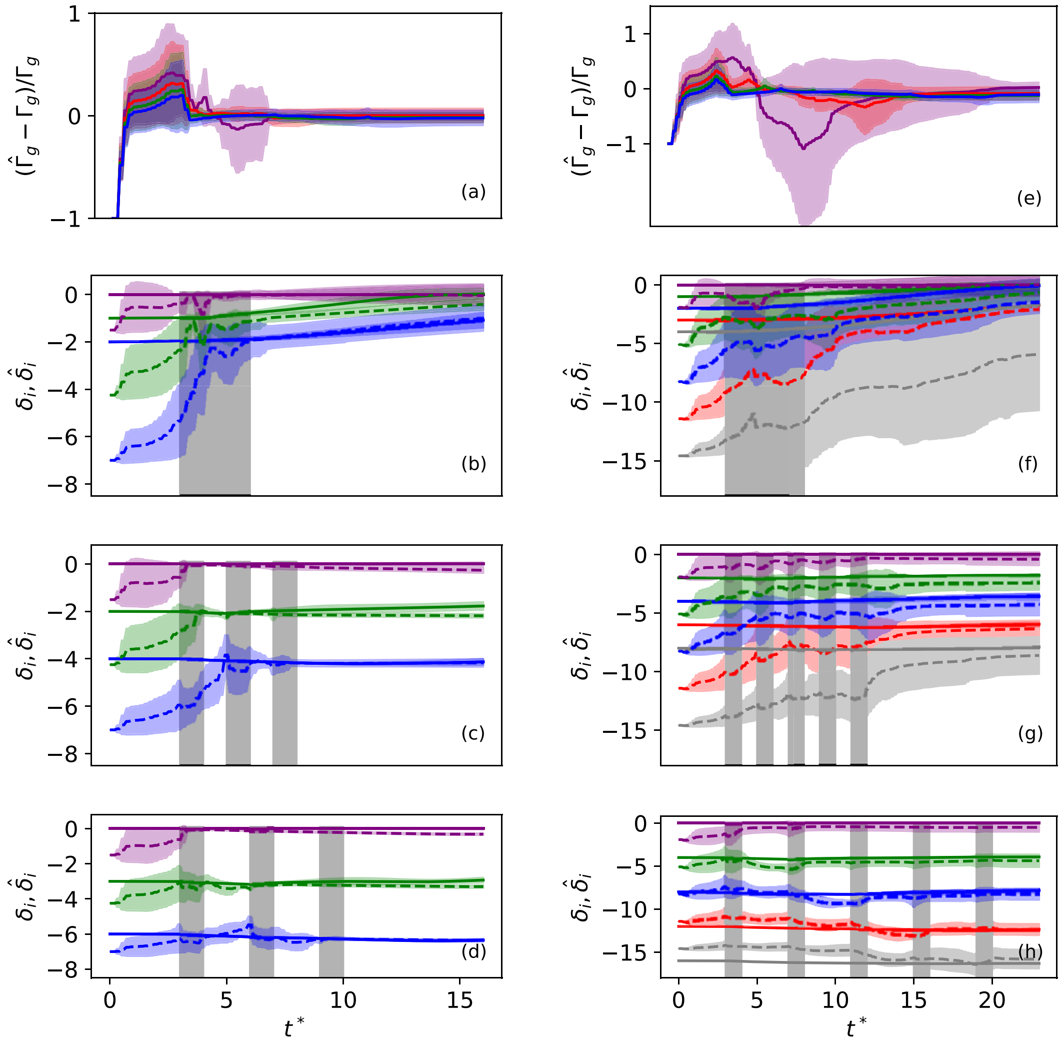


__Figure 8.__ KPF's reconstruction of __3__ (left) and __5__ (right). The gust intensity estimation error $\left(\widehat{\Gamma}_g - \Gamma_g\right)/\Gamma_g$ *is displayed for __3__ (a) and __5__ (e) with* Δ_x^**=0.3 (—), 0.6 (—), 1.0 (—), 1.3 (—). The actual* i-th *vortex position* $\delta_i = \left(x_{g,i} - x_{g,1}\right)/c$ *and its estimate* $\widehat{\delta}_i = \left(\widehat{x}_{g,i} - x_{g,1}\right)/c$ *are displayed for __3__ and __5__ with* $\Delta_x^* = 0.3\,(\mathrm{b,f}), 0.6\,(\mathrm{c,g}), 1.0\,(\mathrm{d}), 1.3\,(\mathrm{h})$. δ_1, δ_2, δ_3, δ_4, δ_5, $\widehat{\delta}_1$, $\widehat{\delta}_2$, $\widehat{\delta}_3$, $\widehat{\delta}_4$, $\widehat{\delta}_5$ *are respectively represented by* —, —, —, —, —, - - -, - - -, - - -, - - -, - - -. *Each quantity has been evaluated for the five KPF controllers with* $\Gamma_g^* = \pm 0.5, \pm 0.4, \pm 0.3, \pm 0.2 \pm 0.1$. *Solid lines represent the average quantities and the colored areas are the associated standard deviations. The gray areas indicate the presence of a gust vortex near the profile.*

controllers are asked to mitigate the lift induced by **1** based on gust reconstruction with high error in both vortex circulation and position when $t^* \leq 3$ (see Figure 7a,c). Therefore, we can assume that the sensitivity of the KPF controller to the reconstructed input plays a role in the overall performance recovery of the KPF controller. The notion of KPF controller sensitivity is evaluated by the gradient of the controller π with respect to its inputs evaluated along the reconstructed gust trajectory $\nabla_s \pi|_{\widehat{s}}$. In Figure 10, we plot the sum of $|\nabla_s \pi|_{\widehat{s}}|$ along a simulation performed under $NL_0 - KPF$, $NL_1 - KPF$, $NL_2 - KPF$ control depending on the recovery $r = |1 - \eta_{KPF}/\eta_{FO}|$. It shows a strong correlation between the sensitivity of the controller and its recovery. More precisely, the $NL_0 - KPF$ controllers have $r \sim 0$ and small sensitivity,

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

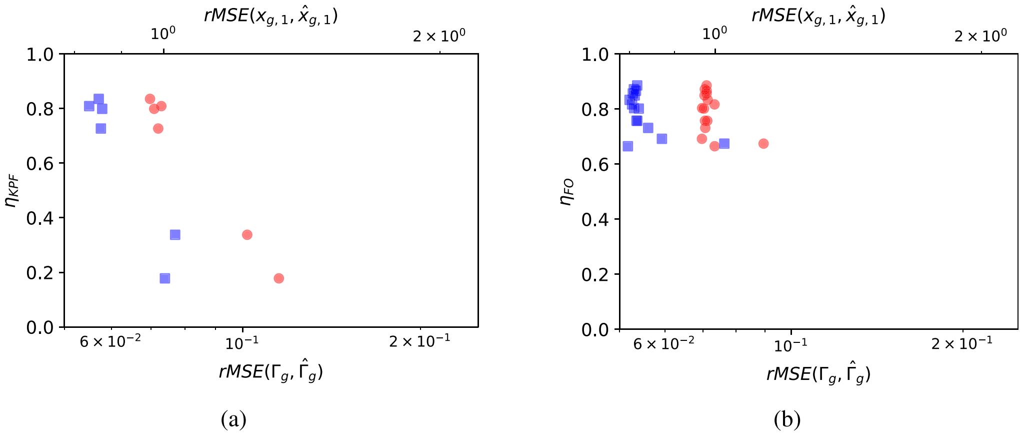


Figure 9. *Impact of η_{KPF} and η_{FO} on the estimation accuracy. We display in (a) (resp. b) η_{KPF} (resp. η_{FO}) obtained under NL$_1$-KPF (resp. NL$_1$-FO) control depending on $rMSE\left(\widehat{\Gamma}_g,\Gamma_g\right)$ and $rMSE\left(\widehat{x}_{g,1},x_g\right)$. ●, ■ represents $rMSE\left(\widehat{\Gamma}_g,\Gamma_g\right)$, and $rMSE\left(\widehat{x}_{g,1},x_{g,1}\right)$, respectively. Each value is averaged over $\Gamma_g^* = \pm0.5, \pm0.4, \pm0.3, \pm0.2, \pm0.1$.*

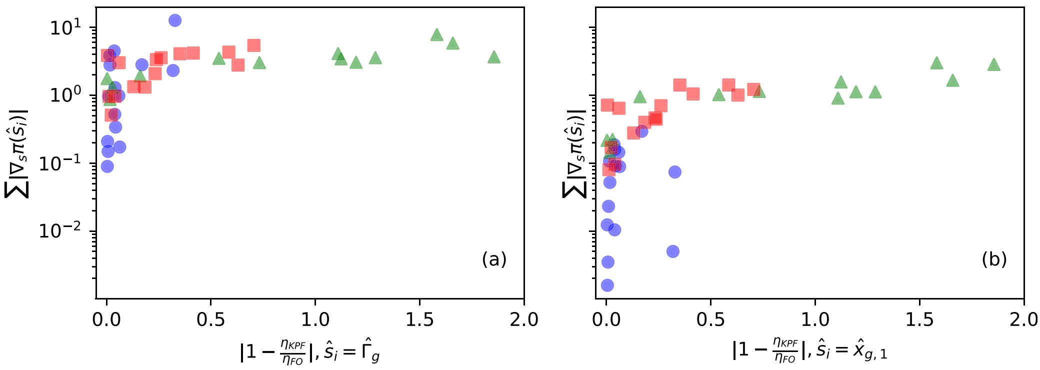


Figure 10. *Impact of the controller sensitivity on the KPF controllers' recovery. We display the sensitivity of a KPF controller π $\sum|\nabla_s\pi|_{\widehat{s}_i}|$ depending on the recovery $r = |1-\eta_{KPF}/\eta_{FO}|$, where $\widehat{s}_i$ is either $\widehat{\Gamma}_g$ or $\widehat{x}_{g,1}$.●, ▲, ■ represent the $NL_0 - KPF$ controllers, the $NL_1 - KPF$ controllers, and the $NL_2 - KPF$ controllers, respectively. Each value is averaged over $\Gamma_g^* = \pm0.5, \pm0.4, \pm0.3, \pm0.2, \pm0.1$.*

the $NL_1 - KPF$ (resp. $NL_2 - KPF$) controllers with r close to zero have small sensitivity, and, the greater is r, the greater is the sensitivity. Besides, regardless of the architecture of the KPF controller, $r \sim 0$ is associated with low sensitivity and $r \sim 1$ with high sensitivity.

We conclude from these results that the variability in r of the KPF controllers observed for **1** is related to the variability within the 15 controllers learned by DRL and not to the variability in the KPF's gust estimation. Therefore, we can consider in future work to constrain the ANN controllers to learn functions that are less sensitive to the reconstruction error in order to reduce the variability of r.

5.3. Control law analysis

To complement the performance of the FO, KPF, and PO controllers described in the above sections, we present here the control laws for each controller and their associated lift mitigation. These are obtained under the control of the five controllers, out of the 15 presented in Sections 5.1 and 5.2, achieving the minimum J evaluation. (For **1**, we present the result averaged over only four controllers because the fifth is not representative of the other four.) We have shown in Section 5.1 that the best NL_0-FO controllers achieve a mitigation comparable to the NL_1, NL_2-FO controllers, but with a much larger variability. The NL_0-PO controllers are found to be significantly less efficient than the NL_0-FO controllers. However, the efficiencies of the PO controllers are significantly increased when nonlinear controllers are used, that is, adding one and two hidden layers to the PO controllers significantly increases their efficiency. In fact $NL_2 - PO$ controllers see their best efficiencies to be close to that of the best $NL_2 - FO$ controllers but with a much higher variability (except for **5** where the $NL_2 - PO$ controllers are of a lower variability).

We present the normalized lift coefficients C_L^* and the normalized control laws $\dot{\alpha}^*$ under each NL_0 controller (resp. NL_2 controller) in Figures 11 and 12 (resp. Figures 11 and 13). In addition, we display the quantity $\Phi_{t^*} = d(J_{t^*}/J)/dt^*$, where $J_{t^*} = \sum_{t' \leq t^*} |C_{L,t'}|$, with $t' = ndt^*, n \in \mathbb{N}$. This derivative highlights regions where the controller lacks of efficiency, which generates positive values of J_{t^*}. Thus, when $t^* \to T_f$, it is expected that $\lim_{t^* \to T_f} J_{t^*} = J$ and that $\lim_{t^* \to T_f} d(J_{t^*}/J)/dt^* = 0$. Similarly, at early times and for gust vortices far from the profile, it is expected that $\lim_{t^* \to 0} J_{t^*} = 0$ and $\lim_{t^* \to 0} d(J_{t^*}/J)/dt^* = 0$.

FO control laws. To analyze the behavior of the FO controller in more detail, we show in Figures 11 and 12 (resp. Figures 11 and 13) the NL_0-FO control laws (respectively, NL_2-FO control laws) and their associated lift coefficients. This analysis shows that the FO controllers mitigate the lift perturbation well when the gust vortices are far from the profile (i.e., typically when the gust vortices are more than half a chord from the edges of the flat plate), but are less effective when the vortices are close to the profile (i.e., when the gust vortices are within half a chord of the edges of the flat plate). This observation is reflected in the quantity Φ_{t^*}, which increases sharply when the vortices are within half a chord of the profile edges, but otherwise remains relatively stable. We deepen the analysis of the FO control laws in Appendix E, where we compare the FO controller with an open-loop controller derived according to the thin-airfoil theory.

PO control laws. To analyze the behavior of the PO controllers in more detail, we present in Figures 11 and 12 (respectively, Figures 11 and 13) the NL_0-PO control laws (respectively NL_2-PO control laws) and the associated lift coefficients. When the PO controller undergoes **1** or **2** and **3** or **5**, respectively. First, the architecture of the ANN controller has a significant impact on the performance of the PO controller. In fact, for gusts **1**, **2**, and **3**, the NL_0-PO control laws are of much smaller amplitude than the NL_0-FO control laws. As a consequence, the mitigation of the lift coefficient is poor at any t^*. Conversely, the NL_2 controllers approximate the FO control laws more accurately than the NL_0 controllers. In fact, for **1** and **2**, the NL_2-PO controller is close to the NL_2-FO control laws, except when the vortices are close to the airfoil. This is also materialized by a small difference between the lift coefficients under NL_2-FO and NL_2-PO control when the vortices are close to the airfoil (see Figure 11). **5** is a gust apart from the other three. Indeed, both the NL_0-PO and the NL_2-PO controllers produce control laws similar to that of their FO counterparts. The main difference between the NL_0-PO and NL_2-PO controllers is that the lift coefficient deviation under NL_0-PO control is much larger than that under NL_2-PO control.

KPF controller. To analyze the behavior of the KPF controller, we present in Figures 11 and 12 the NL_0-KPF control laws and the corresponding lift coefficients. It can be seen that the NL_0-KPF controller recovers well the NL_0-FO control laws for **1**, **2**, and **3**. Consequently, the lift coefficients under the NL_0-KPF control appear to be similar to those of the NL_0-FO control and of significantly smaller magnitude than the lift coefficients obtained under NL_0-PO control. It is worth noting that for **2** and **3** (see Figures 11c and 12a), a small shift is observed between the lift coefficients obtained under NL_0-KPF and NL_0-FO control, while the corresponding control laws appear to be very similar. However, we observe a shift in the incidence laws for these cases (see Figures 11d and 12b), which highlights error accumulation, probably due to small estimation errors.

To complete this analysis, we present in Figures 11 and 13 the NL_2-KPF control laws and the corresponding lift coefficients. The NL_2-KPF controller also recovers well the NL_2 control laws for **1**, **2**, and **3**. Consequently, the lift coefficients under the NL_2 control appear to be similar to those of the

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

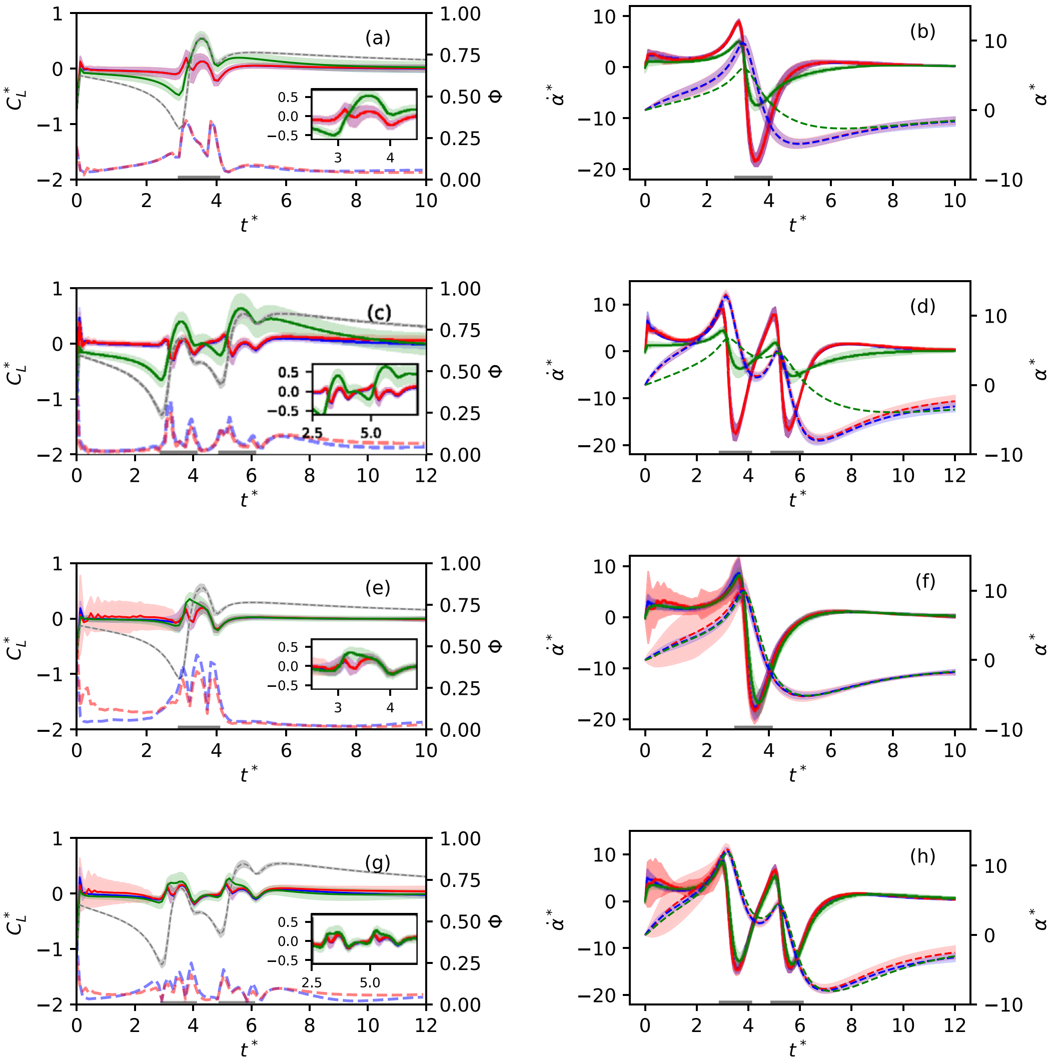


*Figure 11. The normalized lift coefficients C_L^* (left, —) and the normalized control laws $\dot{\alpha}^*$ (right, —) and α^* (right, - - -) under the NL_0 (resp. NL_2)-KPF (red), the NL_0 (resp. NL_2)-FO (blue) and the NL_0 (resp. NL_2)-PO (green) controllers, as well as the control free (gray) for **1** in a,b (resp. e,f) and **2** with $\Delta_x^* = 0.6$ in c,d $(resp. g,h)$. - - -, - - - on the left column represents the quantity Φ under KPF and FO control. Solid lines represent the averaged quantities, and colored areas represent the associated standard deviations. Gray areas on the t^*-axis indicate the presence of a gust vortex near the profile.*

NL_0-FO control and of comparable magnitude to the lift coefficients obtained under the NL_2-PO controller. It can be noted that the NL_2-KPF controller recovers well the lift coefficient under NL_2-FO control for **1** and **2** (see Figure 11) with the vortices close to the airfoil, while the NL_2-PO controller is less efficient. We also notice for **1** and **2** a noisy actuation and a noisy lift coefficient for $t^* \leq 3$. According to Figure 7, this correlates with a high amplitude error in the vortex estimation made for $t^* \leq 3$. However, we do not believe that this noisy control is due to the quality of the estimator because (i) the NL_0-KPF controller also faces high amplitude estimation errors but does not show this control

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

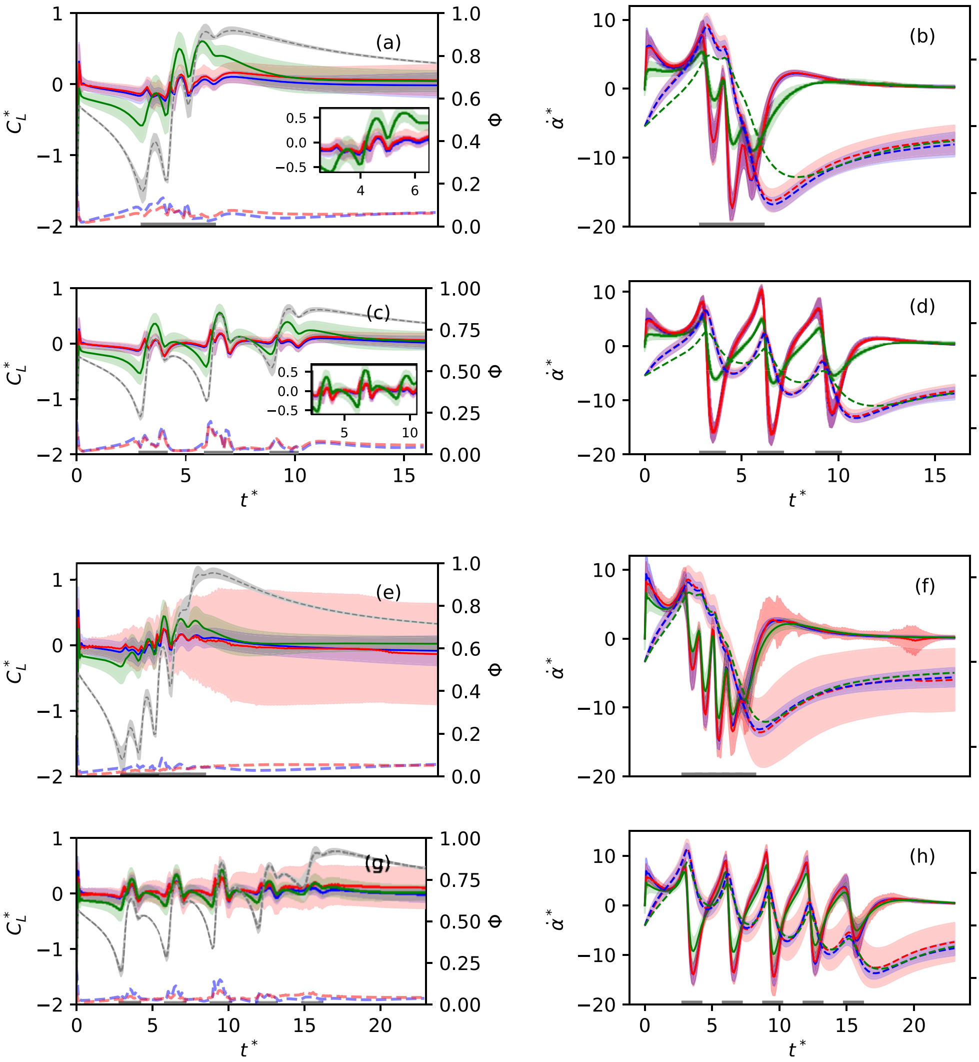


Figure 12. *The normalized lift coefficients C_L^* (left, ——) and the normalized control laws $\dot{\alpha}^*$ (right, ——) and α^* (right, - - -) under the NL_0-KPF (red), the NL_0-FO (blue) and theNL$_0$-PO (green) controllers, as well as the control free (gray) for **3** with $\Delta_x^* = 0.3\,(a,b), 1.0\,(c,d)$ and **5** with $\Delta_x^* = 0.3\,(e,f), 1.0\,(g,h)$. - - -, - - - on the left column represent the quantity Φ under KPF and FO control. Solid lines represent the averaged quantities and colored areas represent the associated standard deviations. Gray areas on the t^*-axis indicate the presence of a gust vortex near the profile.*

behavior, (ii) it is shown in Appendix F that only one NL_2-KPF controller out of five actually produces a noisy control law, the other four well approximate the NL_2-FO control laws. We therefore conjecture that this noisy actuation is attributable to the controller parameters.

The performance of the KPF controllers for **5** must be seen apart from the other three. In fact, as explained in Section 5.2, the number of particles is not high enough to estimate the five vortices for $\Delta_x^* = 0.3$ but it is enough for $\Delta_x^* = 1.3$. Consequently, we can see in Figure 13 that the $NL_2 - KPF$ controller is not able to mitigate efficiently **5** for $\Delta_x^* = 0.3$ but present better mitigation for $\Delta_x^* = 1.3$. The NL_0-KPF

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

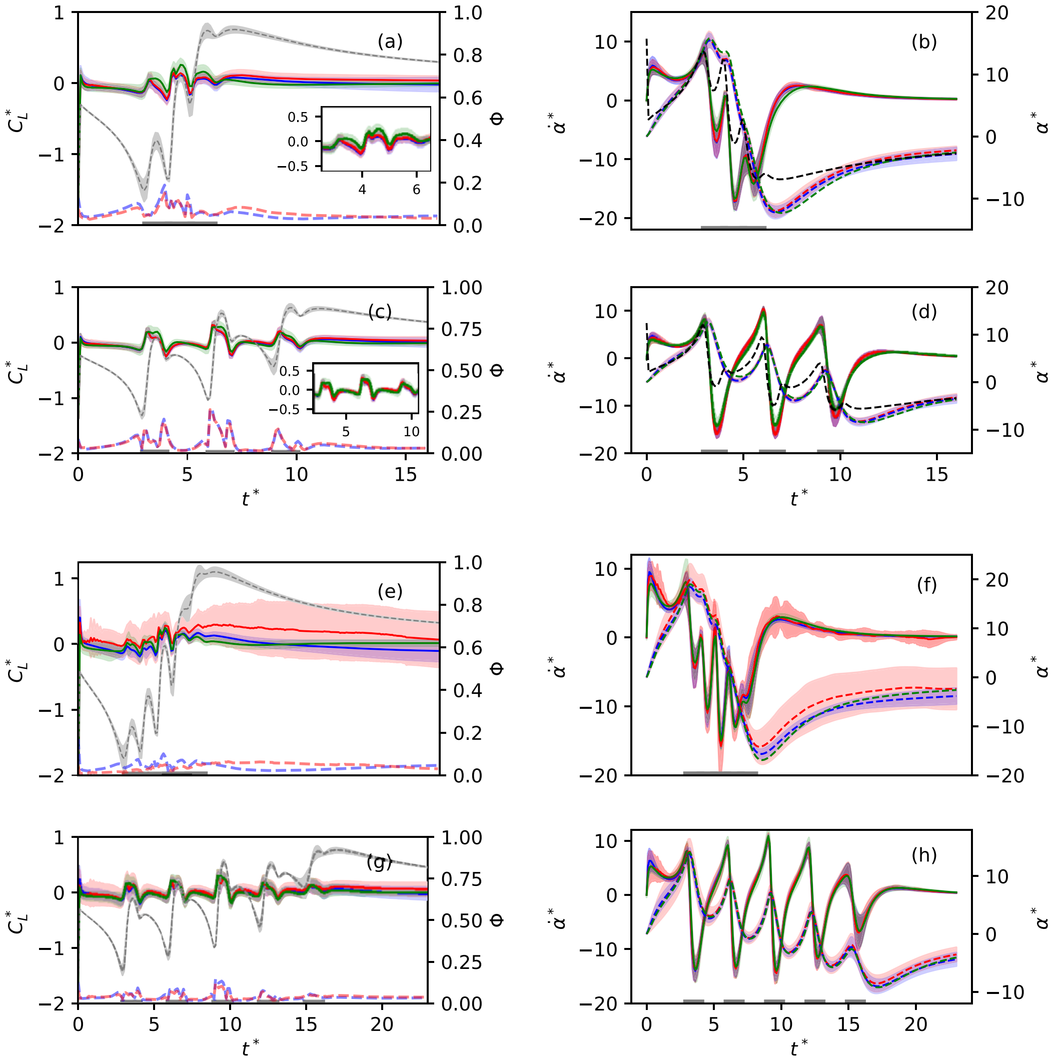


Figure 13. *The normalized lift coefficients C_L^* (left, —) and the normalized control laws $\dot{\alpha}^*$ (right, —) and α^* (right, - - -) under the NL$_2$-KPF (red), the NL$_2$-FO (blue) and the NL$_2$-PO (green) controllers, as well as the control free (gray) for* **3** *with $\Delta_x^* = 0.3\,(a,b), 1.0\,(c,d)$ and* **5** *with $\Delta_x^* = 0.3\,(e,f), 1.0\,(g,h)$. - - -, - - - on the left column represent the quantity Φ under KPF and FO control. Solid lines represent the averaged quantities and colored areas represent the associated standard deviations. Gray areas on the t^*-axis indicate the presence of a gust vortex near the profile.*

controller also suffers from the lack of particles for $\Delta_x^* = 0.3$, however, unlike the $NL_2 - KPF$ controller, it only achieves a very noisy mitigation for $\Delta_x^* = 1.3$.

5.4. Generalisation to unseen gusts

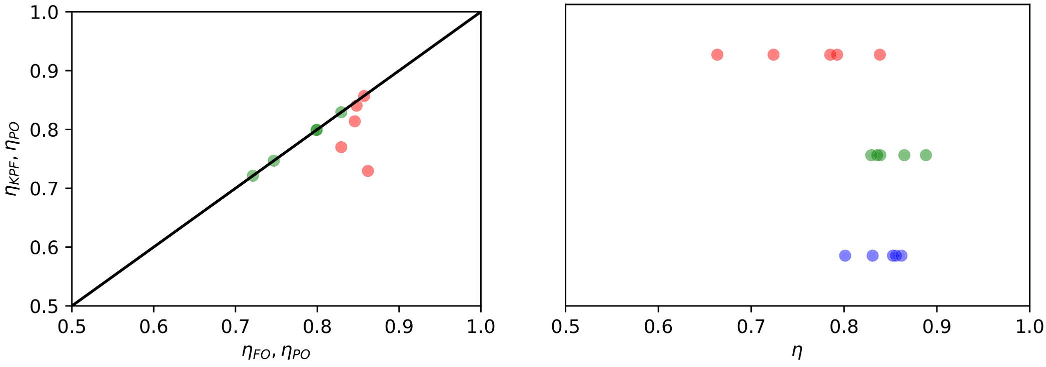
In the above sections, we assumed that the KPF controller was exposed to a vortex gust with the same number of vortices as those estimated by the KPF and used for the ANN controller training. However, in a real deployment, the KPF controller undergoes gusts whose vortex number is unknown. Therefore, in the

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

following discussion, we examine the gust reconstruction performed by the NL_2-KPF controller when it undergoes a gust composed of more vortices than those used for the controller training and for the gust vortex estimation. In this section, we evaluate the KPF controller used for the mitigation of **2** on **2×3** (Figure 2e) and **3×2** (Figure 2f). In these cases, the KPF controller estimates only two gust vortices instead of the six actual vortices. **2×3** (resp. **3×2**) is parameterized by $\Delta_x^* = 0.3, 0.6, 1.0, 1.3$ (respectively, $\Delta_x^* = 0.3, 0.6, 1.0, 1.3$) and $\Gamma_g^* = \pm 0.5/3, \pm 0.4/3, \pm 0.3/3, \pm 0.2/3, \pm 0.1/3$ (respectively, $\Gamma_g^* = \pm 0.5/2 \pm 0.4/2, \pm 0.3/2, \pm 0.2/2, \pm 0.1/2$) so that these gusts induce an uncontrolled lift perturbation similar to **2** (respectively, **3**). We define the position of the i-th gust vortex train as $\overline{x}_{g,i} = \left(\overline{x}_{g,i}, \overline{z}_{g,i}\right) = \left(1/N_{v,z}\Sigma_{j=0}^{N_{v,z}} x_{g,i,j}, 1/N_{v,z}\Sigma_{j=0}^{N_{v,z}} z_{g,i,j}\right)$.

In Figure 14, we present the criteria $\eta_{FO}, \eta_{PO}, \eta_{KPF}$ when the five best NL_2 ANN controllers are subjected to either **2×3** or **3×2**. To determine the performance of the FO controller on **2×3** (resp. **3×2**), we evaluate the above criteria using the FO controller trained on **2** (resp. **3**) and observing $3\Gamma_g, \overline{\mathbf{x}}_{g,1}, \overline{\mathbf{x}}_{g,2}$ (resp. $2\Gamma_g, \overline{\mathbf{x}}_{g,1}, \overline{\mathbf{x}}_{g,2}, \overline{\mathbf{x}}_{g,3}$). For **2×3**, the performance of the three ANN controllers is very similar to that obtained for **2**, that is, the five best FO controllers achieve a mitigation above 0.80, the five best η_{PO} are above 0.70, and the best η_{KPF} recovers well its associated η_{FO}, but with significant variability. However, this result is no longer valid for **3×2**. Although the performance of the FO and PO controllers is very similar to that obtained for **3**, that is, the best η_{FO} and η_{PO} are comparable and range between 0.80 and 0.90. However, the KPF controller does not achieve the same level of performance as **3**, the five η_{KPF} are distributed between 0.65 and 0.85, while for **3** the best η_{KPF} are all close to 0.80.

Here, we investigate the clustering of the gust vortices performed by the KPF controller. We present in Figure 15 the gust reconstruction performed by the KPF controller for respectively **2×3** and **3×2**. Figure 15-a (resp. Figure 15-e) displays the relative error between $\widehat{\Gamma}_g$ and $3\Gamma_g$ (resp. $2\Gamma_g$) when the KPF controller undergoes **2×3** (resp. **3×2**). In addition, Figure 15b–d (resp. Figure 15f–h) compares, respectively, the position of the gust trains with the estimated vortices for $\Delta_x^* = 0.3, 0.6, 1.0$. Figure 15 shows that the KPF controller clusters **2×3** into two vortices of circulation $3\Gamma_g$ and localized in $\overline{\mathbf{x}}_{g,1}, \overline{\mathbf{x}}_{g,2}$. Furthermore, the clustering is made with the same accuracy as the estimation of **2** (Figure 7 in Section 5.2.2), precisely: the KPF controller accurately captures the leading vortex train when it reaches



(a) **2X3**, NL_2 controllers (b) **3X2**, NL_2 controllers

Figure 14. *KPF's controller performance recovery. η_{PO}, η_{FO}, η_{KPF} are displayed, for **2×3**, **3×2**. (a) shows the performance of the controllers for **2×3**, ● stands for $\eta_{KPF} = f(\eta_{FO})$ and ● represent η_{PO} displayed along the line $x = y$ (——). (b) shows the performance of the controllers for **3×2**, ●, ●, ● represent $\eta_{KPF}, \eta_{FO}, \eta_{PO}$ respectively. Note that in (b), η_{FO} and η_{KPF} do not represent the same ANN controller. Each of the η values given for **2×3** (respectively, **3×2**) is averaged over $\Delta_x^* = 0.3, 0.6, 1.0, 1.3$ ($\Delta_x^* = 0.3, 0.6, 1.0, 1.3$) and $\Gamma_g^* = \pm 0.1/3, \pm 0.2/3, \pm 0.3/3, \pm 0.4/3, \pm 0.5/3$ (respectively, $\Gamma_g^* = \pm 0.1/2, \pm 0.2/2, \pm 0.3/2, \pm 0.4/2, \pm 0.5/2$).*

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

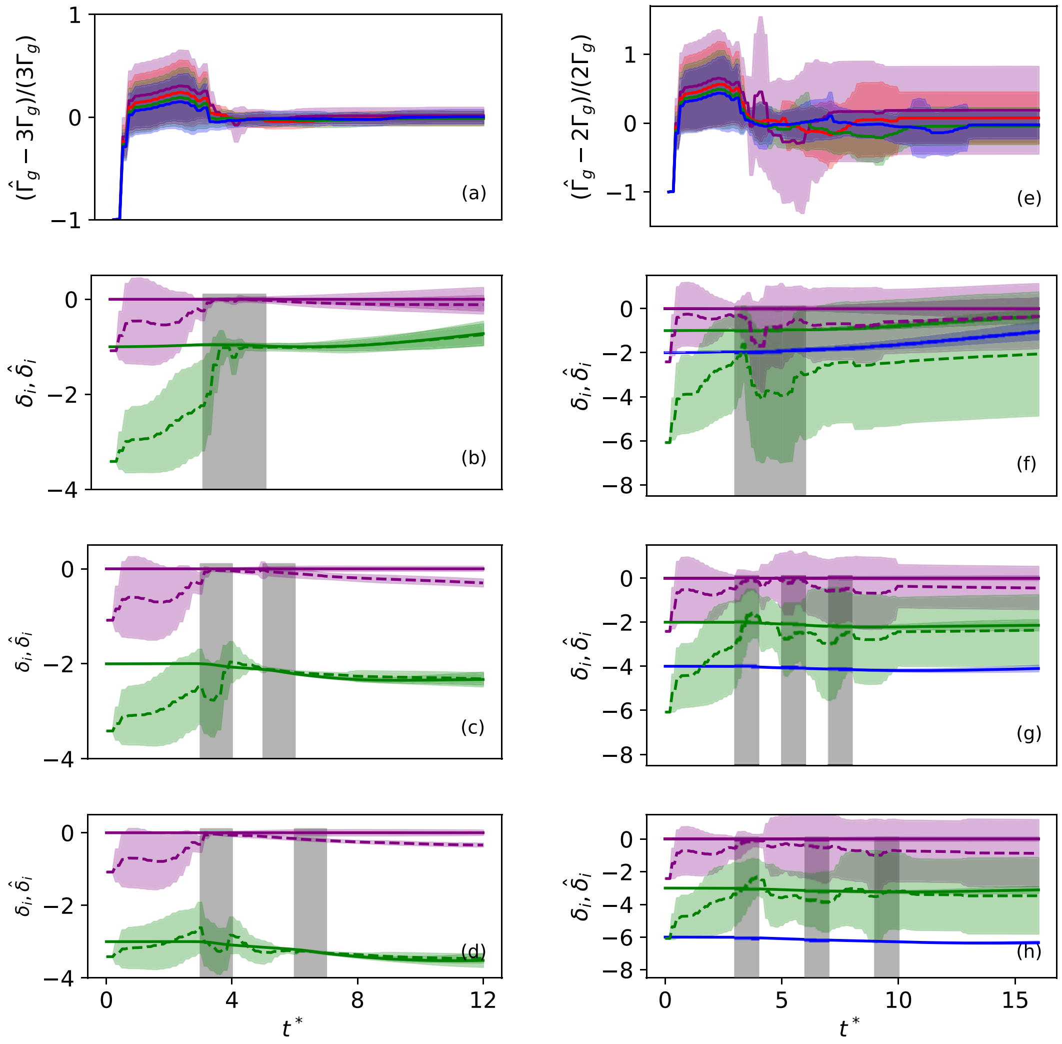


Figure 15. *KPF's reconstruction of* ***2×3*** *and* ***3×2****. The gust intensity estimation error is displayed for* ***2×3*** *(a) and* ***3×2*** *(e) with Δ_x^*=0.3 (—), 0.6 (—), 1.0 (—), 1.3 (—). The actual i-th vortex train position $\delta_i = \left(\bar{x}_{g,i} - \bar{x}_{g,1}\right)/c$ and its estimate $\widehat{\delta}_i = \left(\widehat{x}_{g,i} - \bar{x}_{g,1}\right)/c$ are displayed for* ***2×3*** *with $\Delta_x^* = 0.3\,(b), 0.6\,(c), 1.0\,(d)$ and* ***3×2*** *with $\Delta_x^* = 0.3\,(f), 0.6\,(g),\ 1.0\,(h)$. δ_1, δ_2, δ_3, $\widehat{\delta}_1$, $\widehat{\delta}_2$, are respectively represented by —, —, —, - - -, - - -. Each quantity has been evaluated for the five KPF controllers with $\Gamma_g^* = \pm 0.5/N_{v,z}, \pm 0.4/N_{v,z}, \pm 0.3/N_{v,z}, \pm 0.2/N_{v,z}, \pm 0.1/N_{v,z}$. Solid lines represent the average quantities and the colored areas are the associated standard deviations. The gray areas indicate the presence of a gust vortex near the profile.*

the LE ($t^* = 3$). The second vortex train is accurately captured when the simulation reaches 4 to 6 unit of convective time. $\widehat{\Gamma}_g$ is largely overestimated for $t^* \le 5$, but reaches $3\Gamma_g$ afterwards. However, this conclusion is not valid for **3×2**: the clustering of the gust vortices is not clear for this case. For each of the Δ_x^*, the KPF controller accurately estimates none of the actual vortex trains. Furthermore, no clear pattern emerges from the estimation of Γ_g, except that $\widehat{\Gamma}_g$ reaches $2\Gamma_g$ for none of the Δ_x^*.

These results leave open questions for future work. With a view to real-world deployment, it is imperative that the KPF controller learns an effective mitigation strategy without prior knowledge of the number of incoming gust vortices. This could be achieved by modifying the architecture of the KPF

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

controller: it could seek to estimate the number of vortices along with the positions and circulations of the gust vortices. This approach would also involve modifying the controller's learning procedure, since it would then have to mitigate gusts whose number of vortices is a priori unknown.

6. Conclusion

This paper provides a closed-loop control method for the vortex gust mitigation problem. It consists in controlling the pitch rate $\dot{\alpha}$ of a two-dimensional flat plate so that the criterion $J = \sum_t |C_{L,t}|$ is minimized. DRL is a promising approach to this problem because (i) it models the controller as an artificial neural network, which allows the optimal controller to be searched in a space of functions of varying complexity, and (ii) it derives the optimal controller directly from the aerodynamic model without any prior simplifications. The DRL theory states that the DRL-trained controller must have access to the positions and circulations of the gust vortices to be optimal, but these are not known a priori in a real deployment. To circumvent this issue, we propose to estimate online the gust vortices using data available from the onboard sensors using a KPF. To discuss the advantages and limitations of this approach, we consider three controllers that have all been trained through DRL: (i) the KPF controller that is trained with full knowledge of the gust vortices but is deployed with only knowledge of the gust reconstruction performed by the KPF, (ii) the FO controller that is trained and deployed with full knowledge of the gust vortices and, (iii) the PO controller that is trained and deployed with knowledge limited to the onboard measurements. For each controller, three control laws modeled by zero, one, and two hidden layers are investigated. All controllers are evaluated on gusts composed of one to five vortices. The performance of each controller is evaluated according to the metric $\eta_X = (J_{CF} - J_X)/J_{CF}$, where J_X is the criterion J obtained with controller X that is, KPF, FO or PO and J_{CF} is the criterion J obtained in control-free conditions. Our results show that for the gusts consisting of one to three vortices, the maximum η_{FO} reached by the FO controller is not sensitive to the control law architecture. However, the deeper is the controller, the lower is the variability. The maximum η_{PO} reached by the PO controllers increases significantly as the controller depth increases so that the maximum efficiency of the shallowest PO controller is significantly below that of the FO controllers, but the efficiency of the deepest PO controller is comparable to that of the deepest FO controller. The performance loss observed between the PO and FO controllers is partially recovered by the KPF controllers, that is, the best KPF controllers recover well the performance of their FO counterparts.

However, three limitations arise with the use of the KPF controller. The first is that, for a gust composed of five vortices, the particle requirement (i.e., the computational cost) becomes very high, so the reconstruction of the gust is poorer. Consequently, the performance of the KPF controller becomes poorer to that of the PO controllers. The second is that the KPF controller is trained and deployed with prior knowledge of the number of incoming gust vortices, but this information is inaccessible in a real-world deployment. To examine the impact of this limitation in more detail, we deploy the KPF controller on gusts consisting of more vortices than those estimated by the KPF. Our results show that the KPF controller is able to reconstruct the incoming gust if the number of gust trains matches the number of trains in the estimated gust. However, it performs poorly when the number of estimated gust trains is different from the actual one. Finally, the efficiency of the KPF controller is highly dependent on the parameters of the neural network.

Our work leaves open questions that could be addressed in future work. Indeed, in the perspective of real-world deployment, it is mandatory that the KPF controller operates (i) without prior knowledge of the number of incoming gust vortices and (ii) in such a way that the computational cost of the KPF is adapted to the on-board computational resources. In addition, future work could increase the fidelity of the aerodynamic model by modeling the boundary layers and their interaction with the gust, and then developing an optimal controller, taking advantage of the DRL's ability to learn optimal control laws without prior model simplifications.

Author contribution. Conceptualization: B.M., T.J., E.R., M.B.; Funding acquisition: T.J., M.B.; Investigation: B.M.; Methodology: B.M., T.J., E.R., M.B.; Software: B.M.; Validation: B.M.; Visualization: B.M.; Writing–Original Draft: B.M., T.J., E.R., M.B.; Writing–Review and Editing: B.M., T.J., E.R., M.B. All authors approved the final submitted draft.

Data availability statement. No prior data were used for this article.

Funding statement. No external funding was received for this research.

Competing interest. The authors declare no financial or non-financial interests associated with this research.

Ethical standard. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

Andreu-Angulo I and Babinsky H (2020) Negating gust effects by actively pitching a wing. In *AIAA Scitech 2020 Forum*. Orlando, FL: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2020-1057.

Andreu-Angulo I and Babinsky H (2021) Unsteady modelling of pitching wings for gust mitigation. In *AIAA Scitech 2021 Forum*. Orlando, FL: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2021-1999.

Andreu-Angulo I and Babinsky H. Lift control on pitching wings experiencing gusts. In *AIAA Scitech 2022 Forum*. San Diego, CA: American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2022-0332.

Berger S, **Ramo AA**, **Guillet V**, **Lahire T**, **Martin B**, **Jardin T**, **Rachelson E and Bauerheim M** (2024) Reliability assessment of off-policy deep reinforcement learning: A benchmark for aerodynamics. *Data-Centric Engineering 5*, e2. https://doi.org/10.1017/dce.2023.28.

Bhagwat MJ and Leishman JG (2002) Generalized viscous vortex model for application to free-vortex wake and aeroacoustic calculations. *Annual Forum Proceedings of American Helicopter Society*, *58*(2), 2042–2057.

Bøhn E, **Coates EM**, **Moe S and Johansen TA** (2019) Deep reinforcement learning attitude control of fixed-wing UAVS using proximal policy optimization. In *Proceedings of the IEEE International Conference on Unmanned Aircraft Systems*, pp. 523–533.

Brenner MP, **Eldredge JD and Freund JB** (2019) Perspective on machine learning for advancing fluid mechanics. *Physical Review Fluids 4*, 100501. https://doi.org/10.1103/PhysRevFluids.4.100501.

Brunton SL and Kutz JN (2022) *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press.

Brunton SL, **Noack BR and Koumoutsakos P** (2020) Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics 52*, 477–508.

da Silva AFC and Colonius T (2018) Ensemble-based state estimator for aerodynamic flows. *AIAA Journal 56*(7), 2568–2578.

Darakananda D, **Castro da Silva AF de**, **Colonius T and Eldredge JD** (2018) Data-assimilated low-order vortex modeling of separated flows. *Physical Review Fluids 3*(12), 124701.

Elfring J, **Torta E and van de Molengraft R** (2021) Particle filters: A hands-on tutorial. *Sensors 21*(2), 438.

Elman JL (1990) Finding structure in time. *Cognitive Science 14*(2), 179–211. https://doi.org/10.1016/0364-0213(90)90002-E.

Evensen G (1994) Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans 99*(C5), 10143–10162.

Faller WE and Schreck SJ (1997) Unsteady fluid mechanics applications of neural networks. *Journal of Aircraft 34*(1), 48–55.

Fujimoto S, **Hoof H and Meger D** (2018) Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596.

Fukami K, **Fukagata K and Taira K** (2020) Assessment of supervised machine learning methods for fluid flows. *Theoretical and Computational Fluid Dynamics 34*(4), 497–519.

Garnier P, **Viquerat J**, **Rabault J**, **Larcher A**, **Kuhnle A and Hachem E** (2021) A review on deep reinforcement learning for fluid mechanics. *Computers & Fluids 225*, 104973.

Gavrilovic N, **Bronz M**, **Moschetta J-M and Benard E** (2018) Bioinspired wind field estimation - part 1: Angle of attack measurements through surface pressure distribution. *International Journal of Micro Air Vehicles 10*(3), 273–284. https://doi.org/10.1177/1756829318794172.

Goodfellow IJ, **Bengio Y and Courville A** (2016) *Deep Learning*. MIT Press.

Gunnarson P, **Mandralis I**, **Novati G**, **Koumoutsakos P and Dabiri JO** (2021) Learning efficient navigation in vortical flow fields. *Nature Communications 12*(1), 7143.

Hausknecht M and Stone P (2015) Deep Recurrent Q-Learning for Partially Observable MDPs. In *2015 AAAI Fall Symposium Series*. https://www.aaai.org/ocs/index.php/FSS/FSS15/paper/view/11673.

Herrmann B, **Brunton SL**, **Pohl JE and Semaan R** (2022) Gust mitigation through closed-loop control. ii. feedforward and feedback control. *Physical Review Fluids*, *7*, 024706.

Hou W, **Darakananda D and Eldredge JD** (2019) Machine-learning-based detection of aerodynamic disturbances using surface pressure measurements. *AIAA Journal 57*(12), 5079–5093.

Johnson W, **Silva C**, **Solis E**, **Silva C and Solis E** (2018) Concept vehicles for VTOL air taxi operations. In *AHS Technical Conference on Aeromechanics Design for Transformative Vertical Flight*, San Fransisco, CA.

Jones AR, **Cetiner O and Smith MJ** (2022) Physics and modeling of large flow disturbances: Discrete gust encounters for modern air vehicles. *Annual Review of Fluid Mechanics 54*(1), 469–493. https://doi.org/10.1146/annurev-fluid-031621-085520.

Julier SJ and Uhlmann JK (2004) Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, *92*(3), 401–422.

Kalman RE and Bucy RS (1961) New results in linear filtering and prediction theory. *Journal of Basic Engineering 83*(1), 95–108. https://doi.org/10.1115/1.3658902.

Katz J and Plotkin A (2001) *Low-speed Aerodynamics*. Cambridge University Press.

Kaufmann W (1962) Über die ausbreitung kreiszylindrischer wirbel in zähen (viskosen) flüssigkeiten. *Ingenieur-Archiv 31*(1), 1–9.

Kazarin P, **Golubev V**, **MacKunis W and Moreno C** (2021) Robust nonlinear tracking control for unmanned aircraft in the presence of wake vortex. *Electronics 10*(16).

Le Provost M and Eldredge JD (2021) Ensemble Kalman filter for vortex models of disturbed aerodynamic flows. *Physical Review Fluids 6*(5), 050506.

Meng L, **Gorbet R and Kulić D** (2021) Memory-based deep reinforcement learning for POMDPS. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5619–5626. https://doi.org/10.1109/IROS51168.2021.9636140.

Novati G, **Verma S**, **Alexeev D**, **Rossinelli D**, **Van Rees WM and Koumoutsakos P** (2017) Synchronisation through learning for two self-propelled swimmers. *Bioinspiration & biomimetics 12*.

Paris R, **Beneddine S and Dandois J** (2021) Robust flow control and optimal sensor placement using deep reinforcement learning. *Journal of Fluid Mechanics 913*.

Pham DT (2001) Stochastic methods for sequential data assimilation in strongly nonlinear systems. *Monthly Weather Review 129* (5), 1194–1207.

Poudel N, **Yu M and Hrynuk JT** (2021) Gust mitigation with an oscillating airfoil at low Reynolds number. *Physics of Fluids 33* (10), 101905. https://doi.org/10.1063/5.0065234.

Puterman ML (2014) *Markov decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.

Rabault J, **Kuchta M**, **Jensen A**, **Réglade U and Cerardi N** (2019) Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics 865*, 281–302.

Renn PI and Gharib M (2022) Machine learning for flow-informed aerodynamic control in turbulent wind conditions. *Communications Engineering 1*(1), 45.

Ristic B, **Arulampalam S and Gordon N** (2003) *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House.

Rumelhart DE, **Hinton GE and Williams RJ** (1986) Learning representations by back-propagating errors. *Nature 323*(6088), 533–536.

G Sedky, **Jones AR and Lagor FD** (2020a) Lift regulation during transverse gust encounters using a modified Goman–Khrabrov model. *AIAA Journal 58*(9), 3788–3798. https://doi.org/10.2514/1.J059127.

Sedky G, **Lagor FD and Jones A** (2020b) Unsteady aerodynamics of lift regulation during a transverse gust encounter. *Physical Review Fluids 5*(7), 074701. https://doi.org/10.1103/PhysRevFluids.5.074701.

G Sedky, **Gementzopoulos A**, **Andreu-Angulo I**, **Lagor FD and Jones AR** (2022) Physics of gust response mitigation in open-loop pitching manoeuvres. *Journal of Fluid Mechanics 944*, A38. https://doi.org/10.1017/jfm.2022.509.

Sherstinsky A (2020) Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena 404*, 132306.

Straubinger A, **Rothfeld R**, **Shamiyeh M**, **Büchter K-D**, **Kaiser J and Plötner KO** (2020) An overview of current research and developments in urban air mobility–setting the scene for UAM introduction. *Journal of Air Transport Management 87*, 101852.

Sutton RS and Barto AG (2018) *Reinforcement Learning: An Introduction*. The MIT Press.

Theodorsen T (1935) General theory of aerodynamic instability and the mechanism of flutter. NACA. Technical Report, TR 496.

Von Karman T and Sears WR (1938) Airfoil theory for non-uniform motion. *Journal of the Aeronautical Sciences*, *5*(10), 379–390. https://doi.org/10.2514/8.674

Wagner H (1925) Über die entstehung des dynamischen auftriebes von tragflügeln. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, *5*(1), 17–35. https://doi.org/10.1002/zamm.19250050103.

Waldock A, **Greatwood C**, **Salama F and Richardson T** (2018) Learning to perform a perched landing on the ground using deep reinforcement learning. *Journal of Intelligent & Robotic Systems 92*, 685–704.

Weerts AH and El Serafy GYH (2006) Particle filtering and ensemble Kalman filtering for state updating with hydrological conceptual rainfall-runoff models. *Water Resources Research 42*(9).

Yang Z and Nguyen H (2021) Recurrent off-policy baselines for memory-based continuous control. *Deep RL Workshop, NeurIPS 2021*.

Appendix

A. UVLM validation

This section presents the test cases studied for the assessment of our UVLM model. We present here three test cases: (i) comparison with Wagner's model on an impulsive pitch motion, (ii) comparison with Theodorsen's model on a sinusoidal pitching motion, and (iii) comparison of the uncontrolled lift disturbance induced by **1** modeled according to the UVLM model and numerical simulation of the (inviscid) Euler's equations using a finite volume method.

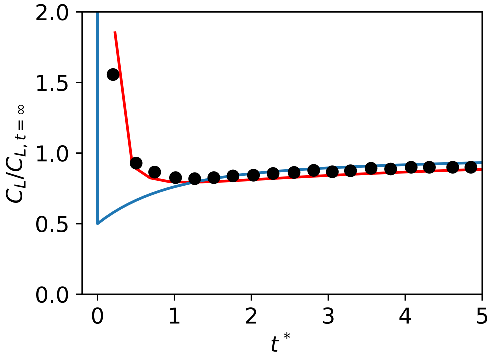


Figure A1. *Impulsive start modeled according to Wagner's theory —, the Katz and Plotkin's (2001) implementation of the lumped vortex method (•) and our lumped vortex method implementation (—).*

Impulsive start

We compute the temporal lift evolution of a flat plate subjected to an AoA step. At $t^* = 0^-$, the AoA is 0° and at $t^* = 0^+$ the AoA is 5°. In Figure A1, we display the temporal evolution of $C_L/C_{L,t=\infty}$ according to Wagner's theory (—) (Wagner, 1925), the lumped vortex method (•) implemented by Katz and Plotkin (2001) and our implementation of the lumped vortex method (—). Note that the lumped vortex method is as the UVLM solver but with $N = 1$, $dt^* = 0.25$ and the lift is computed according to the unsteady Joukovski formula: $\mathbf{L} = \left(\rho U_\infty \sum_{i\in\mathcal{P}}\Gamma_i + \rho c \frac{\partial}{\partial t}\sum_{i\in\mathcal{P}}\Gamma_i\right)\mathbf{e}_z$. It results a good agreement between Wagner's theory and the lumped-vortex method implementations.

Sinusoidal pitch motion

This test case assumes that the flat plate is undergoing a sinusoidal pitch motion defined as $\alpha(t) = 5^\circ \sin(\omega t)$, where ω is the pulsation of the oscillation. In the present work, the maximum $\dot{\alpha}^*$ value chosen by the controllers is $\omega^*_{max} = \omega_{max} c/U_\infty \sim 0.13$. We assess our UVLM model over a sinusoidal pitch motion described as $\alpha(t) = 5^\circ \sin(\omega t)$ with $\omega = \omega_{max}$ and $\omega = 2\omega_{max}$.

In Figure A2 we compare the lift coefficient of the oscillating profile predicted by our UVLM solver with that predicted by Theodorsen's model (Theodorsen, 1935). Theodorsen's theory states that in the case of an oscillating profile, the added-mass effect and the wake development cause changes in both amplitude and phase of the lift coefficient compared to that predicted by the thin airfoil theory $C_L = 2\pi\alpha$. Figure A2 shows a very good agreement between Theodorsen's model and the lift predicted by our UVLM solver for $\omega = \omega_{max}$ and $\omega = 2\omega_{max}$, that is, both the delay and the amplitude modulation are captured by the UVLM solver. Note that

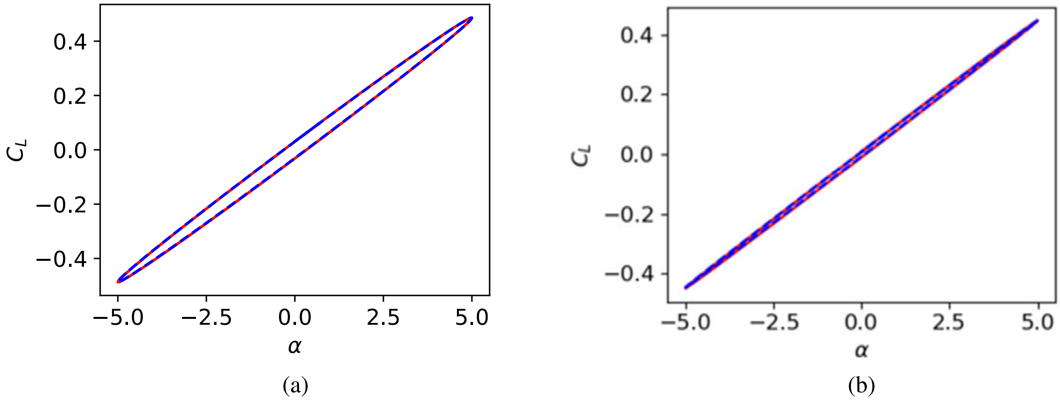
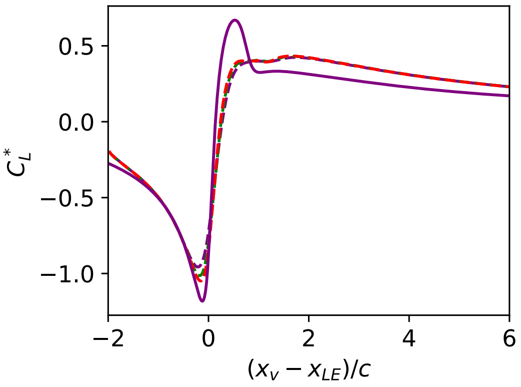


Figure A2. *Lift coefficient of an oscillating flat plate. The pitch motion is defined as $\alpha(t) = 5^\circ \sin(\omega t)$ with $\omega = \omega_{max}$ (left) and $\omega = 2\omega_{max}$ (right). The lift coefficient is computed according to our UVLM solver (—) with $N = 10$ and $dt^* = 0.025$ and compared with Theodorsen's model (- - -).*

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press



(a)

Figure A3. *Comparison of the uncontrolled lift coefficient under UVLM and Euler modeling. — shows the lift coefficient obtained by our UVLM solver with $N = 40$ and $dt^* = 0.01$. - - -, - - -, - - - represent the lift coefficients obtained with the Euler equations with the mesh cell size and the time step dt^* equal to $(r_c/10, 1/80)$, $(r_c/20, 1/160)$, $(r_c/40, 1/320)$, respectively.*

this result was obtained with a time step $dt^* = 0.025$, which is four times smaller than that used for our training runs, but we are unable to train our controllers with this precision because of the computational cost (for example, training would take 4 days for **5**).

Euler comparison

This test case compares the uncontrolled lift coefficient obtained when the flat plate undergoes **1** under UVLM modeling with that obtained using Euler's equations.

The Euler simulation is performed using the computational fluid dynamics software StarCCM+. In both models, we consider a gust vortex with a viscous core $r_c = 0.2c$ and $\Gamma_g^* = 0.5$. The main differences are that (i) in the Euler modeling, the airfoil is a NACA 0012 profile instead of a flat plate, and (ii) in the UVLM modeling, the gust is modeled as a vortex whose position is defined as a point convected by the flow, whereas in the Euler modeling the gust is modeled as a continuous velocity distribution initialized as Eq. 1 and convected by the flow.

Figure A3 shows the lift coefficients C_L^* obtained with each of the methods as a function of the distance between the vortex (x_v) and the leading edge of the profile (x_{LE}). It is shown that the two models agree very well as long as the vortex is upstream of the profile, but they differ when the vortex passes the leading edge. We can argue that this difference is due to the fact that the vortex in UVLM is a vortex singularity, whereas the vortex in Euler is a continuous set of velocities. Consequently, as it passes close to the profile, the Euler vortex is split by the surface, whereas the UVLM vortex remains a single vortex singularity.

B. Hyperparameters and learning curves

Table B1 shows the hyperparameters used to train the controllers. Note that since the UVLM model is deterministic, we chose $\sigma_n = 0$ (Section 4.1).

We also show in Figure B1 the J criterion of the 15 NL_0 and NL_2-FO controllers and the 15 NL_0 and NL_2-PO controllers along the training for **1**. Each of the J values is averaged from ten simulations with Γ_g^*, Δ_x^* randomly chosen. Note that positive values are due to the calculation of the standard deviation and do not actually occur in the training. One can see that the training is noisy for the NL_0-FO controller, but smoother for the NL_2-FO controller. Interestingly, the NL_2-FO controller improves almost continuously along the training, while the NL_2-PO controller achieves its maximum mitigation efficiency during the first few episodes of the training, but shows a poor J criterion afterward.

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

Table B1. *Hyper-parameters*

Hidden layer actor	| [400] |[400,400]
Corresponding hidden layer critic	[400,400] | [400] |[400,400]
Optimizer	Adam
Learning rate critic	$1e^{-3}$
Learning rate actor	$1e^{-3}$
σ_n	0.0
Critic loss	MSE
γ	0.99
τ	0.005
Batch size	64
Max episode	6000
Hidden layer activation function/ Output layer activation function	ReLu/ tanh

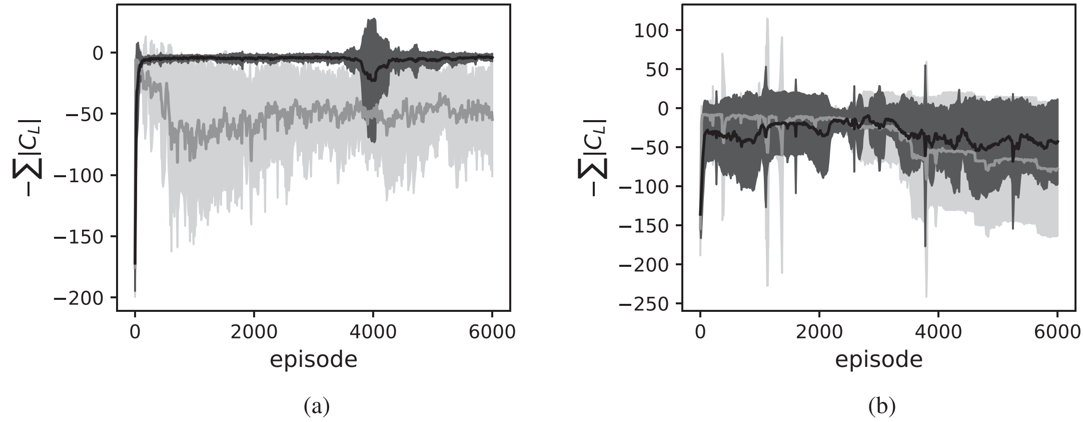


Figure B1. *Learning curves for the mitigation of* ***1*** *under NL_2 control (a) and NL_0 control(b). The FO controllers are displayed in —, and the PO controllers are displayed in —, solid lines represent the average quantities and the colored areas are the associated standard deviations.*

C. KPF initializations

We discuss here the initialization of the KPF particles. To mitigate **1**, **2**, **3**, and **5**, we assume that we know a priori the number of vortices N_v but we know neither Γ_g^* nor Δ_x^*. Thus, to initialize a particle, we randomly choose a value of Γ_g^* uniformly in $[-0.5,0.5]$, we choose randomly N_v longitudinal positions so that $x_{N_v,1} \leq \ldots \leq x_{1,1}$ and we choose randomly N_v transverse positions, uniformly in $[-0.1c,0.1c]$. In Table C1, we compute the probability for a particle to be initialized as the real gust vortices with an uncertainty on the position of one chord and a 10% uncertainty on the circulation. Table C1 shows that the probability for a particle to be initialized

Table C1. *Probability for a particle to be initialized as* ***1****,* ***2****,* ***3****, or* ***5***

Δ_x/R_I	0.3	0.6	1.0	1.3
$N_{v,x},N_{v,z}=(1,1)$	$1e^{-2}$			
$N_{v,x},N_{v,z}=(2,1)$	$2e^{-3}$	$3e^{-3}$	$4e^{-3}$	$5e^{-3}$
$N_{v,x},N_{v,z}=(3,1)$	$5e^{-5}$	$2e^{-4}$	$3e^{-4}$	$6e^{-4}$
$N_{v,x},N_{v,z}=(5,1)$	$8e^{-10}$	$7e^{-8}$	$9e^{-7}$	$4e^{-6}$

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

as the gust vortices decreases very sharply as N_v increases and this probability is larger for high Δ_x^* than for low Δ_x^*. This is due to our constraint on the longitudinal positions, $x_{i,1}$ is s.t. $x_{N_v,1} \leq \ldots \leq x_{i,1} \leq \ldots \leq x_{1,1}$.

We also examine the choice of measurement noise $\sigma = \tilde{\sigma}\max(1e^{-2}, |\Gamma_{LE,t^*}|)$. First, we observe experimentally that Γ_{LE} varies from $1e^{-3}$ to $1e^{-1}$. Consequently, a constant value of σ is not appropriate here, because if $\sigma \sim 1e^{-1}$, the KPF is unable to discriminate particles when $\Gamma_{LE} \sim 1e^{-2}$ and, conversely, if $\sigma \sim 1e^{-2}$, the KPF over-constrains the particles when $\Gamma_{LE} \sim 0.1$. We therefore propose an adaptive measurement noise to take account of this wide range of Γ_{LE} values.

D. Estimation of the gust vortex transverse positions

We discuss here the reconstruction of z_g performed by the KPF for **2** and **3**. In Figure D1, we present the KPF reconstruction of z_g for **2** (Figure D1a) and for **3** (Figure D1b) for $\Delta_x^* = 0.3, 0.6, 1.0, 1.3$. We compare $z_{g,1}$ (—), $z_{g,2}$ (—), $z_{g,3}$ (—) with the KPF estimates $\widehat{z}_{g,1}$ (- -), $\widehat{z}_{g,2}$ (- -), $\widehat{z}_{g,3}$ (- -). It results that the estimation of the vortex transverse positions is a more challenging task than the estimation of the vortex longitudinal positions. As a matter of fact, Figure D1 shows that for both **2** and **3**, the KPF is not able to accurately track the transverse positions. It is important to note that the transverse distances between the gust vortices and the LE are, most of the time, negligible compared to the longitudinal distances. We can therefore assume that the KPF is not able to estimate them because they have a limited impact on Γ_{LE} compared to their longitudinal counterpart (Eq. 5).

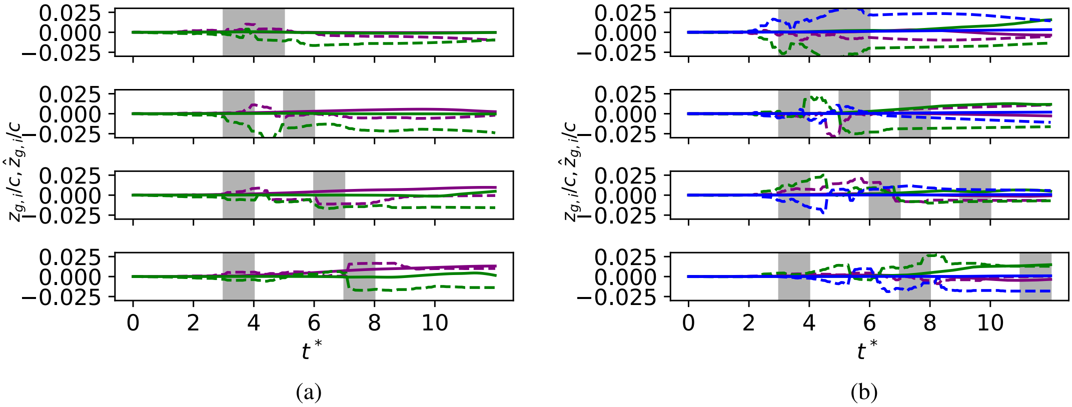
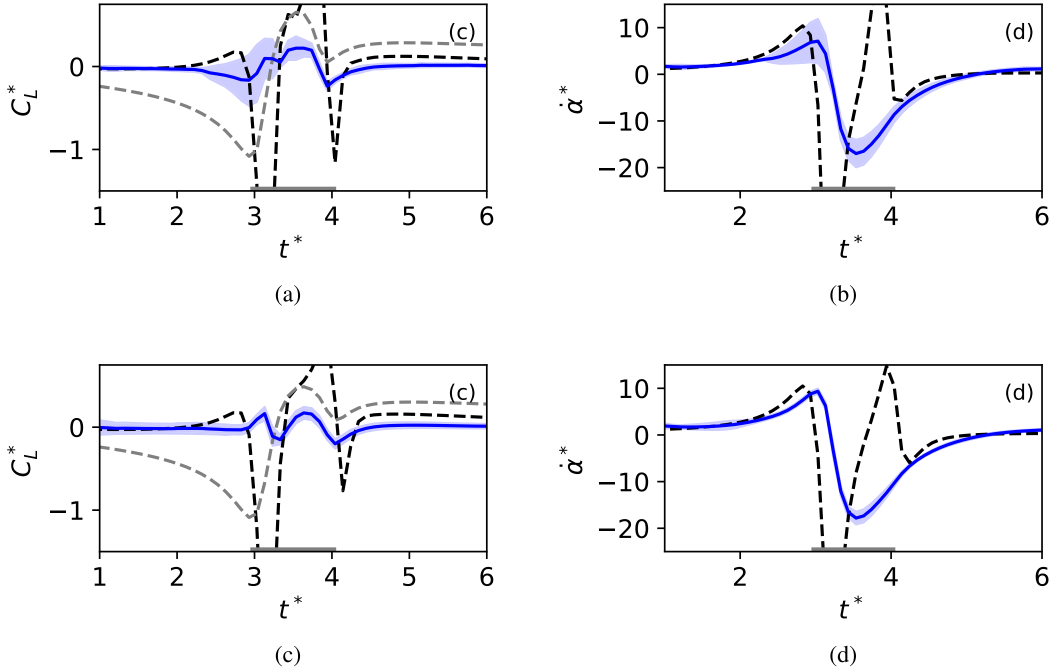


Figure D1. *Reconstruction of z_g performed by the KPF controller.*

E. Comparison with thin airfoil theory based control

In the limit of thin airfoil theory, the lift coefficient in the control free configuration, $C_{L,free}$, can be viewed as the lift response to a perturbation in AoA α_{t^*}, such that $C_{L,free} = 2\pi\alpha_{t^*}$. In the most simple quasi-steady framework, $C_{L,free}$ can thus be mitigated by applying a control law that opposes $C_{L,free}$, i.e. $\alpha_{t^*} = -C_{L,free}/(2\pi)$. This rough approximation assumes that (i) the pitch motion does not affect the trajectories and the circulations of the gust and wake vortices and, (ii) neglects the unsteady effects. In the following, we refer to the open-loop (OL) controller defined as $\alpha_{t^*} = -C_{L,free}/2\pi$ as the OL controller. Note that this controller is only introduced for analysis purposes since $C_{L,free}$ is not known a priori. To compare the difference between the NL_2-FO controller and the OL controller, we display in Figure E1 the FO control laws, the OL control laws and their associated lift coefficients when the controllers undergo **1** with $\Gamma_g^* = \pm 0.5, \pm 0.1$. Figure E1 shows that for $t^* \leq 3$, there is a good agreement between the two control laws. However, for higher t^*, the FO controller differs from the OL controller. This differences involve a poor mitigation of the OL controller and a good mitigation of the FO controller. These observations must be seen in the context of the assumptions made to derive the OL controller. We assume that the pitch motion does not induce a high amplitude $\dot{\alpha}$ but under OL control, this assumption is not valid for $3 \leq t^* \leq 4$. Furthermore, we assume that there is no coupling between the pitching motion and the wake vortices, however, in the ideal case where the FO controller maintains the lift coefficient at $C_{L,target}$, the wake vortices have zero circulation. This result shows the great capability of DRL to build efficient controllers without the need of assumptions or simplifications of the physics, as done for the OL controller.



(a)

(b)

(c)

(d)

Figure E1. *The normalized lift coefficients C_L^* (left) and the normalized control laws $\dot{\alpha}^*$ (right) under the OL controller - - -, the FO controller —, control free - - for **1** with $\Gamma_g^* = \pm 0.5(\mathrm{a,b}), \pm 0.1(\mathrm{c,d})$. Solid lines represent the averaged quantities and colored areas represent the associated standard deviations, and the gray area on the t^*-axis indicates the presence of the gust vortex on the profile.*

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press

F. All control laws for 2

In Section 5.3, we observe that the KPF controller exhibits noisy actuation for gust **f(2)**. To better understand the reasons behind this behavior, Figure F1 displays the five KPF and FO controllers used for the visualizations in Section 5.3. The results show that only one out of the five controllers demonstrates this noisy actuation behavior. We therefore conclude that the noisy actuation is caused by variability within the control functions.

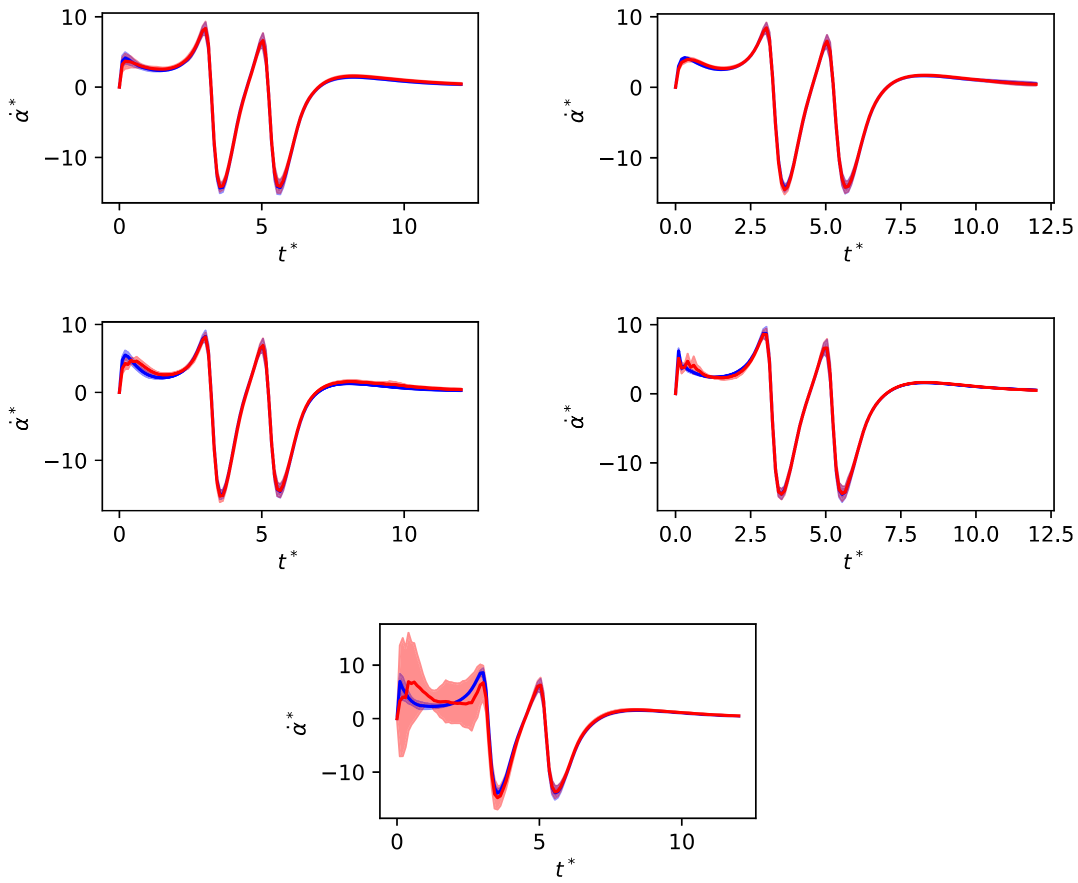


Figure F1. *All control laws used for Figure 11-g.* —, — *respectively represent* $\dot{\alpha}^*$ *under NL*$_2$*-FO control and NL*$_2$*-KPF control, when the controllers undergo* ***2*** *with* $\Delta_x^* = 0.6$*. The results have been computed for* $\Gamma_g^* = \pm 0.5, \pm 0.4, \pm 0.3, \pm 0.2, \pm 0.1$*. Solid lines represent the average control laws, and the colored areas are the associated standard deviations.*

Cite this article: Martin B, Jardin T, Rachelson E and Bauerheim M (2024). Vortex gust mitigation from onboard measurements using deep reinforcement learning. *Data-Centric Engineering*, 5, e47. doi:10.1017/dce.2024.38

https://doi.org/10.1017/dce.2024.38 Published online by Cambridge University Press