arXiv:2006.11005v2 [physics.flu-dyn] 22 Jun 2020

Robust flow control and optimal sensor placement using deep reinforcement learning

Romain Paris[1]†, Samir Beneddine[1] and Julien Dandois[1]

[1]ONERA DAAA, 8 rue des Vertugadins, 92190 Meudon, France

(Received xx; revised xx; accepted xx)

This paper focuses on a drag-reducing control strategy on a 2D-simulated laminar flow past a cylinder. Deep reinforcement learning algorithms have been implemented to discover efficient control schemes, using two synthetic jets located on the cylinder's poles as actuators and pressure sensors in the wake of the cylinder as feedback observation. The present work focuses on the efficiency and robustness of the identified control strategy and introduces a novel algorithm (S-PPO-CMA) to optimise the sensor layout. An energy-efficient control strategy reducing drag by 18.4% at Reynolds number 120 is obtained. This control policy is shown to be robust both to the Reynolds number in the range $[100, 216]$ and to measurement noise, enduring signal to noise ratios as low as 0.2 with negligible impact on performance. Along with a systematic study on sensor number and location, the proposed sparsity-seeking algorithm has achieved a successful optimisation to a reduced 5-sensor layout while keeping state-of-the-art performance. These results highlight the interesting possibilities of reinforcement learning for active flow control and pave the way to efficient, robust and practical implementations of these control techniques in experimental or industrial systems.

1. Introduction

Improvement of aerodynamic characteristics on air vehicles has mainly been achieved through shape optimisation in the past decades, with drag reduction as the primary goal. Passive control devices have long been the centrepiece of flow control (Selby *et al.* 1992; Gutmark & Grinstein 1999; Marquet *et al.* 2008), thanks to their ease of use. Yet, their overall low efficiency advocates for active forms of control, which split into two categories: open-loop strategies (see for instance Sipp (2012)), and closed-loop approaches (Sipp & Schmid 2016), the latter being known to display greater performance and robustness, taking advantage of state measurements. In this context, linear techniques have recently been investigated for active flow control. But they have shown some limitations on nonlinear systems, whether on performance, robustness, or computation complexity (Sipp *et al.* 2010).

These linear approaches often rely on a reduced-order model, mainly via proper orthogonal decomposition (POD) (Gerhard *et al.* 2003; Bergmann *et al.* 2005) or resolvent analyses (Leclercq *et al.* 2019), which provide frameworks to apply linear control techniques. Numerous studies and various approaches have been proposed: Fujisawa *et al.* (2001) and Siegel *et al.* (2003) used variable phase proportional and differential control to reduce the drag of low Reynolds number cylinder flows. Several studies implemented robust H_2 or H_∞ control methods based on resolvent analysis (Jin *et al.* 2019) or using an iterative strategy (Leclercq *et al.* 2019). Other mathematical frameworks, such as the adjoint approach (He *et al.* 2000), have also shown efficient control but at a high

† Email address for correspondence: romain.paris@onera.fr

computational cost. These are a few examples of a large body of work dedicated to techniques that rely on local linear approximations, and thus, often pertain to constant or periodic forcing on the flow. Such control strategies are adapted to weakly nonlinear systems where the linear approach remains valid. They have been nonetheless often applied on nonlinear systems, despite their limitations, due to the lack of robust and efficient methods to tackle nonlinear high-dimensional systems, such as encountered in actual fluid mechanics applications.

In the context of the development of new and promising machine learning (ML) techniques, efficient nonlinear active flow control appears increasingly viable, as emphasised by Brunton & Noack (2015) and Brunton *et al.* (2020). The use of artificial neural networks (NN) as universal function approximators that can be trained efficiently has already proven significant capabilities for solving complex problems such as translation (Cho *et al.* 2014; Sutskever *et al.* 2014) or image recognition (He *et al.* 2016). Coupled with reinforcement methods, that use interactions with the controlled environment to improve performance, these techniques achieve autonomous learning of complex tasks (Baker *et al.* 2019; Kaiser *et al.* 2019), and often perform better than human experts (Mnih *et al.* 2015). The present study focuses particularly on *on-policy* Deep Reinforcement Learning (DRL). From the first methods (Williams 1992) to the most recent algorithms such as Trust Region Policy Optimisation (Schulman *et al.* 2015*a*) or Proximal Policy Optimisation (PPO) and its variants (Schulman *et al.* 2017; Hmlinen *et al.* 2018), DRL has demonstrated the ability to efficiently learn non-trivial control strategies (named policies) in complex and high-dimensional environments. By leveraging stochastic estimation and Markov processes, these algorithms optimise both sample and policy efficiencies.

The use of ML techniques in fluid mechanics enables efficient and more straightforward nonlinear control strategies. In the wake of nonlinear auto-regressive models (Kim *et al.* 2006; Dandois *et al.* 2013), ML algorithms are used either for black-box or model-based feedback control (Seidel *et al.* 2009; Cohen *et al.* 2012), leveraging the flexibility of neural network structures, using for instance the artificial neural network estimator (ANNE) method (Nrgrd *et al.* 2000). Semi-supervised learning methods, such as model predictive control (Nair *et al.* 2020) or DRL for control (Rabault *et al.* 2019), also meet increasing success. However, neural network's known lack of extrapolation capabilities and thus weak robustness for supervised learning applications highlights the robustness of DRL methods as a potential weak point. This issue is explored by Tang *et al.* (2020) who successfully controlled a 2D-cylinder wake across a large range of Reynolds numbers. In this study, the potentialities of deep reinforcement learning in active flow control performance, power efficiency and robustness are investigated on a similar test case. One important contribution of the paper relates to the introduction of a variant of PPO which is shown to outperform state-ot-the-art DRL approaches on the cylinder case.

Another important contribution relates to optimal sensor placement. Reducing sensor requirements while keeping optimal control performances is key to the potential transposition of these techniques to experimental and industrial cases. Rabault *et al.* (2019) and Tang *et al.* (2020) respectively use 151 and 236 probes to control the flow past a 2D-cylinder. Their work is therefore a first step for DRL control that needs to be continued and further improved, which is precisely the purpose of this work. The present study builds on these existing papers to propose new techniques and algorithms that reduce the gap between DRL capabilities for flow control and the requirements for a future experimental or industrial implementation. Having less sensors means at the same time less hardware requirements, less potential failure modes and less computational power needs, especially in a context of embedded systems with strong real-time computing

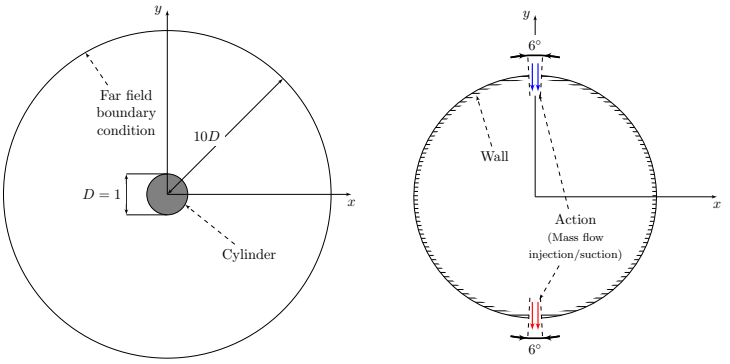

FIGURE 1. Flow domain geometry. (left): Full domain, not at true scale. The far field boundary condition is a characteristic-based inflow/outflow boundary condition modelling free-stream flow. (right): Boundary conditions on the cylinder.

constraints. This issue of optimal measurement location has been investigated by many authors outside of the context of DRL, for instance by Mons *et al.* (2017, 2016); Foures *et al.* (2014); Verma *et al.* (2020) for data assimilation. Bright *et al.* (2013) took advantage of compressed sensing to perform flow reconstruction using a limited number of sensors. The optimal estimation of a reduced order state, usually POD modes, has been used by Cohen *et al.* (2006); Seidel *et al.* (2009) and echoes the assumption that accurate flow estimation is an essential feature of efficient control. However, as stressed by Oehler & Illingworth (2018), control does not systematically require faithful flow reconstruction (in the sense of POD), the partial knowledge of relevant "hidden" variables may be sufficient. This idea is, in the linear framework, conveyed by the notion of observability Gramian. Empirical observability Gramians were used by Singh & Hahn (2005); DeVries & Paley (2013) for flow estimation and a version balancing both observability and controlability Gramians was successfully implemented by Manohar *et al.* (2018) to design an optimal H_2 control of a linearised Ginzburg-Landau model. One of the main contributions of our work is to propose a new method, leveraging a previously learned DRL policy to optimise sensor location.

In the following, the simulated case study is first introduced, then the DRL algorithm used to derive the control strategy is described and a new sparsity-seeking variant of this algorithm, aiming at optimising sensor number and location is proposed. The main results and discussions are developed in section 4. The issue of control efficiency and robustness to both Reynolds number variations and measurement noise is addressed in this section. Finally, the optimal sensor layouts derived using our proposed method is discussed.

2. Description of the flow configuration and numerical methods

The studied configuration is a bi-dimensional (2D) flow past a cylinder. The geometry is made non-dimensional by setting the cylinder diameter D to 1. The centre of the cylinder is located at the origin $(0, 0)$ of the flow domain. Figure 1 displays the computed flow domain, which spans over $10D$, and shows the orientation of axes x and y.

2.1. *Numerical setup*

The flow is described by the compressible Navier-Stokes equations. The free-stream flow is uniform at a Mach number M_∞ of 0.15, oriented along x. In the following, all

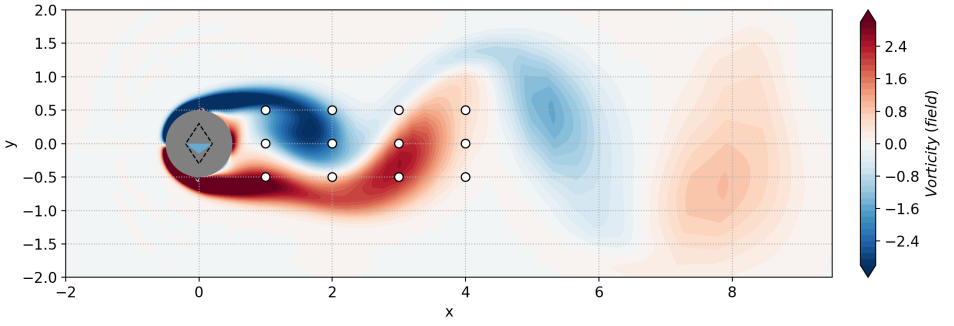

FIGURE 2. Instantaneous vorticity flow field with action $a_t = -0.15$ at $Re = 120$. White dots represent the sensor locations. The coloured triangle in the cylinder depicts the action, its height and colour representing its amplitude. The dashed diamond shape marks off maximum actions (both positive and negative).

quantities are made non-dimensional by the characteristic length D, the inflow density ρ_∞, the velocity U_∞ and the static temperature T_∞. Note that the Mach number is very low, such that the density fluctuations in the whole domain are negligible, and the flow is therefore quasi-incompressible. The Reynolds number Re, defined as $U_\infty D/\nu$ (ν being the kinematic viscosity), is varied in the article, but the first sections of the paper focus on a reference configuration at $Re = 120$. The flow field is computed using ONERA's FastS finite volume method solver (Dandois *et al.* 2018) for both steady and unsteady computations. For unsteady computations, a global numerical time step $dt = 5 \times 10^{-3}$ is chosen. Additional numerical details are available in Appendix B. The C-shaped structured mesh is made of 25200 nodes and is refined in the vicinity of the cylinder. The boundary conditions are specified in figure 1.

In the context of active flow control, and as shown in figure 1, injection or suction is performed on the cylinder's poles through two 6°-wide jet inlets. A control step Δt is defined as the number of numerical iterations during which the control command is held constant. In this study a control step lasts for 50 numerical time steps, thus $\Delta t = 0.25$ non-dimensional time units. For each control step, an action command a_t (positive or negative) is translated into a blowing/suction using a 20-iteration interpolation ramp in order to avoid abrupt changes of boundary conditions and non-physical values due to the numerical schemes, similarly to what Rabault *et al.* (2019) did. Formally, for the i^{th} numerical iteration of the control step, the mass flow per unit area q_i is:

$$q_i = \rho_\infty U_\infty \left(a_{t-1}(1 - r_i) + a_t r_i\right), \text{ with } r_i = \begin{cases} i/20 & \text{if } i < 20 \\ 1 & \text{otherwise} \end{cases} \tag{2.1}$$

To ensure an instantaneous zero-net-mass-flux for every action, the two poles act reciprocally: $+q_i$ is imposed on the top inlet surface and $-q_i$ on the bottom inlet. Note that in several studies, the actuators are such that they are able to inject streamwise momentum, which may directly reduce the cylinder drag. In the present study, the actuators are designed such that they can only inject cross-stream momentum, thus making any "direct" drag reduction impossible.

Several sensors record the pressure of the flow at predefined locations at the end of every control step. The output measurement is a pressure fluctuation, defined as the difference between the local non-dimensional static pressure and the reference inflow static pressure p_∞. Figure 2 illustrates a standard setup for this case.

Both drag and lift coefficients (C_x and C_l) are computed on the cylinder via the

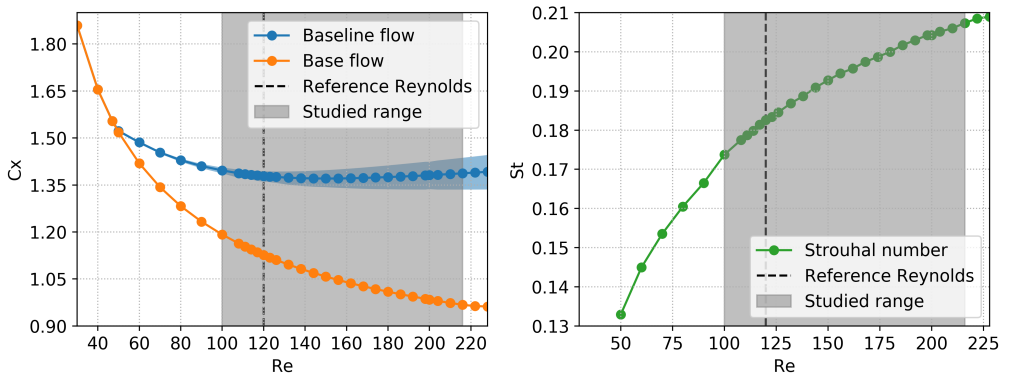

FIGURE 3. (left): Evolution of the time-averaged drag coefficient of the baseline flow (blue) and the base flow (orange) with the Reynolds number. The blue shaded area indicates the variation range of the drag coefficient C_x. (right): Evolution of the Strouhal number of the vortex shedding with the Reynolds number.

resulting force of the flow $\boldsymbol{F}$:

$$\boldsymbol{F} = \int_{\text{cylinder}} \sigma.\boldsymbol{n} dS \tag{2.2}$$

$$C_x = \frac{\boldsymbol{F}.\boldsymbol{e}_x}{\frac{1}{2}\rho_\infty U_\infty^2 D} \tag{2.3}$$

$$C_l = \frac{\boldsymbol{F}.\boldsymbol{e}_y}{\frac{1}{2}\rho_\infty U_\infty^2 D} \tag{2.4}$$

where $\boldsymbol{n}$ is the unitary cylinder surface normal vector, σ is the stress tensor, $\boldsymbol{e}_x = (1, 0)$ and $\boldsymbol{e}_y = (0, 1)$.

2.2. *Uncontrolled flow*

The uncontrolled configuration, denoted in the following as the baseline flow, displays a well-documented vortex shedding behaviour (Williamson 1996) that appears for Reynolds numbers above 46, and which is due to a Hopf bifurcation where the steady solution of the Navier-Stokes equations (the base flow) becomes unstable. Thus, the flow becomes unsteady and follows a stable limit cycle associated with vortex shedding.

As presented in figure 3, values of the drag coefficient and Strouhal number (defined as $St = fD/U_\infty$, with f being the vortex shedding frequency) have been computed for a wide range of Reynolds numbers to ensure consistency with other studies (Nishioka & Sato 1978; Braza *et al.* 1986; Williamson 1996; Henderson 1997; He *et al.* 2000; Bergmann *et al.* 2005). For $Re = 120$, the drag coefficient is 1.379 with fluctuations of amplitude 0.018, and $St = 0.18$, which is in agreement with the literature (Barkley 2006; Sipp *et al.* 2010). Note that, since the simulation solves the 2D Navier-Stokes equations, the flow remains laminar across the studied Reynolds number range and does not undergo any additional stability bifurcation.

According to Protas & Wesfreid (2002), the total drag $C_{x,0}$ of the baseline flow can be decomposed into two contributions. The drag of the base flow $C_{x,BF}$, which is constant and the drag correction due to the flow unsteadiness $C_{x,U}$. If $\langle\cdot\rangle_T$ denotes the time average over a vortex shedding period T, then:

$$\langle C_{x,0}\rangle_T = C_{x,BF} + \langle C_{x,U}\rangle_T \tag{2.5}$$

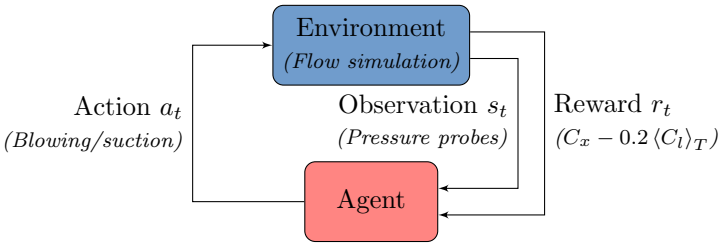

FIGURE 4. Reinforcement learning feedback loop

Using the base flow performance as a reference, the drag gain μ_{C_x} which measures the drag reduction due to the control strategy is computed as a fraction of the drag reduction achieved by the base flow:

$$\mu_{C_x} = \frac{\langle C_{x,0} \rangle_T - C_x}{\langle C_{x,U} \rangle_T} \tag{2.6}$$

Thus a drag gain μ_{C_x} of 100% corresponds to a drag reduction equivalent to a complete suppression of the vortex shedding. Protas & Wesfreid (2002) also asked whether a negative mean drag correction $\langle C_x - C_{x,BF} \rangle_T$ could be reached with a periodic forcing, implying a drag gain larger than 100%. Examples from the literature, such as the work of He *et al.* (2000) who achieved a drag gain of 108%, show that this is possible. But in their case, this performance comes at the cost of a significantly modified mean flow and a large actuation. As shown later, the present study also achieves drag gains slightly higher than 100%, while both preserving the base flow structure and being energy efficient.

3. Reinforcement learning algorithms

3.1. *A short description of on-policy reinforcement learning*

Reinforcement learning considers an environment in interaction with an agent as illustrated in figure 4. At each control step t, the agent receives partial state observations s_t and a reward r_t quantifying the current performance of the environment. The agent then takes an action a_t based on the observations, through a policy π: $a_t \sim \pi(\cdot, s_t)$. This policy can either be deterministic or stochastic. The objective of the training is to derive a policy π^* that maximises the cumulative reward (called return) throughout time.

In the present study, the environment is the previously described numerical case and the goal is to minimise drag. Thus the reward is a defined as:

$$r_t = -C_x - \alpha |\langle C_l \rangle_T| \tag{3.1}$$

with $\langle C_l \rangle_T$ being a moving average of the lift coefficient on the previous 22 control steps corresponding to the duration of a vortex shedding period, and α the corresponding regularisation coefficient. Here $\alpha = 0.2$ (this value is justified in section 4.1). This penalisation ensures a nearly "zero time-averaged lift" policy. Pressure sensors provide the observed information, actions are computed within a valid range ($a_t \in [-2, 2]$) and then implemented on the environment as previously described. The code architecture relies on Tensorflow software library (Abadi *et al.* 2016) and is interfaced with the simulation environment through Cassiope application programming interface (Benoit *et al.* 2015) using Python programming language. Interface standards are inspired from Open AI's Gym toolkit (Brockman *et al.* 2016).

All learning algorithms aim at solving the exploration-exploitation dilemma, meaning achieving the best performances at a minimum learning cost. Efficient exploration of

the state-action subspace is a key factor in the learning algorithm performance. The exploration is performed through the introduction of randomness in actions. However, too much randomness deteriorates the learning speed and thus reduces exploitation performances for a given learning budget. The careful control of the exploration variance is thus crucial. *On-policy* algorithms try to circumvent this issue using the most recent (and best so far) version of the policy to collect experience, thus sparing inefficient exploration in sub-optimal regions of the state-action subspace.

One of the state-of-art learning approaches is the Proximal Policy Optimisation (PPO) introduced by Schulman *et al.* (2017). PPO has been successfully used by Rabault *et al.* (2019); Rabault & Kuhnle (2019); Rabault *et al.* (2020) on a very similar case study. This on-policy actor-critic algorithm uses a dual neural network structure (with around $270,000$ parameters each in our case), an "actor" (π) and a "critic" (V), as agent. Both take the observations s_t as input. The actor outputs an optimal action $\mu_t = \mu_\theta(s_t)$, θ being the weights and biases of π. Then, using a predefined standard deviation σ, an action $a_t \sim \pi_\theta(\cdot|s_t) = \mathcal{N}(\cdot|\mu_\theta, \sigma)$ is sampled, $\mathcal{N}$ being a normal distribution. The critic outputs an estimate of the value $V_t = V_\phi(s_t)$ of the observed state s_t, ϕ being the weights and biases of V. This value is an estimator of the expected return $R_t = \sum_{\tau=t}^{\infty} \gamma^\tau r_\tau$, $\gamma \in]0;1[$ being a discount factor. V_t is used during the update of π_θ to improve learning. The learning phase is performed using a surrogate objective, the updated weights θ_{new} of the actor aim at making the most successful actions more likely, using the probability ratio $\frac{\pi_{\theta_{new}}(a_t,\mu_t,\sigma)}{\pi_\theta(a_t,\mu_t,\sigma)}$. However, to prevent excessively large policy updates, the surrogate objective is clipped. As a consequence, the exploration variance, which is due to both σ and the variability of μ_t, often shrinks prematurely as explained by Hmlinen *et al.* (2018). The next section presents a variant of the PPO algorithm, used in this paper, which addresses this limitation. Refer to Appendix A for more detail on PPO.

3.2. *Standard PPO-CMA*

Proximal Policy Optimisation with Covariance Matrix Adaptation (PPO-CMA) (Hmlinen *et al.* 2018) is a variant of PPO that prevents the premature vanishing of the exploration variance using the covariance matrix adaptation technique introduced by the CMA-ES evolutionary algorithm (Hansen *et al.* 2003; Hansen 2016). Unlike PPO, the covariance matrix σ used to sample the action $a_t \sim \mathcal{N}(\cdot, \mu, \sigma)$, is an output of the actor π, as shown in figure 5 and the surrogate objective used for updates is not clipped. A more detailed description of PPO-CMA is provided in Appendix A. PPO-CMA is used for all the results introduced in parts 4.1 to 4.4. As shown in the following, it yields significantly improved performances than the "vanilla" PPO algorithm used by Rabault *et al.* (2019).

3.3. *Sparse surrogate actor*

A novel algorithm called Sparse PPO-CMA (S-PPO-CMA), selecting relevant observations and discarding non-necessary or redundant sensor information, while preserving the optimality of the learned control strategy as much as possible, is introduced. Note that S-PPO-CMA is only used in section 4.5 to optimise the number and location of sensors, the other results presented in this study being obtained with standard PPO-CMA. This method splits into two separate phases: training a conventional PPO-CMA actor-critic structure (described in part 3.2), then deriving a sparse surrogate actor. The sparse training phase relies on the previously trained PPO-CMA actor-critic policy denoted by π^* for the actor and V^* for the critic. As described by figure 6, the sparse

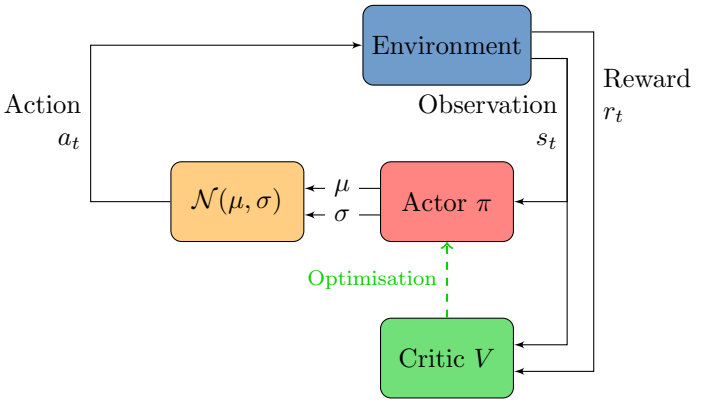

FIGURE 5. Proximal Policy Optimisation with Covariance Matrix Adaptation (PPO-CMA). The actor π receives observations s_t from the environment and outputs μ and σ, which are used to sample action a_t. The critic V estimates the value V_t of the observed state s_t. During the update phase, V_t is used to update the actor and the critic is updated using supervised learning on the effective state values $R_t = \sum_{\tau=t}^{\infty} \gamma^{\tau} r_\tau$ (observed returns).

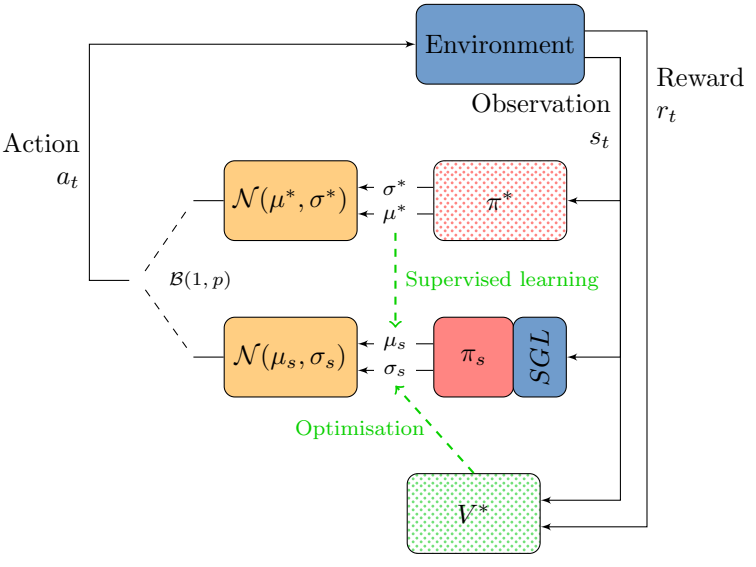

FIGURE 6. Sparse Proximal Policy Optimisation with Covariance Matrix Adaptation (S-PPO-CMA). Actions are either sampled using the reference actor π^* or the sparse actor π_s via a Bernoulli choice $\mathcal{B}(1, p)$. π_s is updated via learning on σ_s using values from V^* and by supervised learning on μ_s using μ^* values. The parameters of the stochastic gated layer (SGL) are also updated during this phase.

actor π_s is composed of a dense neural network having the same structure (architecture and activation functions) as π^*, to which a stochastic gated input layer (SGL) is added.

During the sparse training phase, the action a_t is either sampled using the optimal policy π^* or π_s, using a Bernoulli random variable. Both training of π^* and V^* are stopped, but their outputs are used to train π_s and the SGL that make up the sparse version of π^*.

The SGL mechanism used here is inspired by the stochastic gate model proposed by Louizos *et al.* (2017). Let n be the number of sensors (or the dimension of the observation space). The SGL, presented in figure 7 is a special simply connected layer that provides

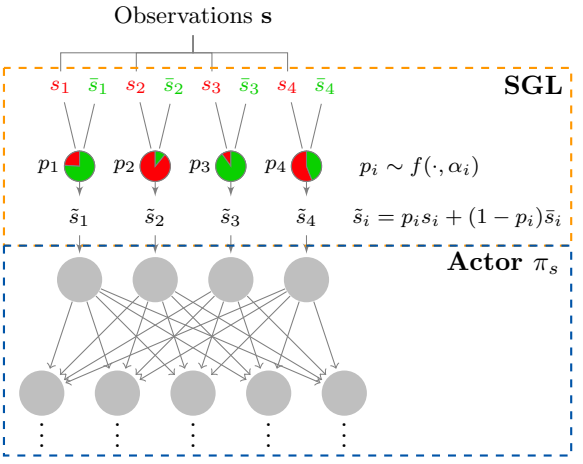

FIGURE 7. Stochastic Gated input Layer (SGL). Received observation s_i is either passed on to the actor π_s if $p_i = 1$, combined with its substitute value $\bar{s}_i$ if $p_i \in]0;1[$, or replaced by $\bar{s}_i$ if $p_i = 0$. p_i is sampled using a "gating" function f parameterised by α_i, which is updated during the second phase the S-PPO-CMA method.

inputs to π_s and that contains substitute values $\bar{\boldsymbol{s}} = (\bar{s}_1, \bar{s}_2, ..., \bar{s}_n)$ for each observation component. Every time an observation vector $\boldsymbol{s} = (s_1, s_2, ..., s_n)$ is received, the SGL samples a random vector $\boldsymbol{p} \in [0;1]^n$ that determines its output $\tilde{\boldsymbol{s}}$ such as:

$$\tilde{\boldsymbol{s}} = \boldsymbol{p} \odot \boldsymbol{s} + (1 - \boldsymbol{p}) \odot \bar{\boldsymbol{s}} \tag{3.2}$$

where $\odot$ represents the elementwise product. Thus, $p_i = 0$ outputs the observation s_i whereas $p_i = 1$ gives its substitute value $\bar{s}_i$, and any value in-between provides a linear combination of s_i and $\bar{s}_i$. Similarly to Louizos *et al.* (2017), $\boldsymbol{p}$ is sampled over a "gating" function:

$$\boldsymbol{u} \sim \mathcal{U}^n(0,1) \tag{3.3}$$

$$\boldsymbol{p} = f(\boldsymbol{u}, \boldsymbol{\alpha}) = \text{clip}\left((\zeta - \gamma)\text{Sigmoid}\left[\frac{1}{\beta}(\log \boldsymbol{u} - \log(1 - \boldsymbol{u}) + \boldsymbol{\alpha})\right] + \gamma, 0, 1\right) \tag{3.4}$$

with $\mathcal{U}^n(0,1)$ denoting a uniform distribution on $[0,1]^n$, β, γ, ζ being fixed numerical parameters, $\boldsymbol{\alpha}$ being a trainable vector steering the expectation on $\boldsymbol{p}$ and $\text{clip}(a,b,c) = \min(\max(a,b),c)$. f can be seen as a "soft" Bernoulli choice distribution enabling values of $\boldsymbol{p}$ in $[0;1]^n$. The L_0 complexity of the SGL, giving the expected number of observation components s_i for which $p_i > 0$, can be written as:

$$\mathcal{L}_c(\boldsymbol{\alpha}) = \sum_{i=1}^{n} P(p_i > 0) = \sum_{i=1}^{n} \text{Sigmoid}\left[\alpha_i - \beta \log \frac{-\gamma}{\zeta}\right] \tag{3.5}$$

During testing, $\boldsymbol{p}^*$, the most likely value of $\boldsymbol{p}$ is chosen deterministically as:

$$\boldsymbol{p}^* = \text{clip}\left((\zeta - \gamma)\text{Sigmoid}\left[\boldsymbol{\alpha}\right] + \gamma, 0, 1\right). \tag{3.6}$$

Both $\boldsymbol{p}$ and $\boldsymbol{p}^*$ can take values between 0 and 1 (included), thus modelling a fully "open" or fully "closed" gate while still allowing for a gradient-based optimisation using the loss $\mathcal{L}_c$.

3.4. *Sparse actor training*

The weights θ_s of the sparse actor π_s are initialised using the weights θ^* of π^* and updated every epoch both by the training of σ_s and μ_s. σ_s is trained in the same way σ^* has been trained (refer to Appendix A). Concerning μ_s however, a supervised learning using the optimal action μ^* is performed with the loss:

$$\mathcal{L}_{\pi_s}(\theta_s, \boldsymbol{\alpha}) = ||\mu_s(\boldsymbol{s}, \theta_s, \boldsymbol{\alpha}) - \mu^*(\boldsymbol{s}, \theta^*)||_1 \tag{3.7}$$

For the SGL, $\boldsymbol{\alpha}$ and $\bar{\boldsymbol{s}}$ are trained in the same process as θ_s, allowing π_s to "adapt" to the variations of input $\tilde{\boldsymbol{s}}$ caused by the updates of the SGL. $\bar{\boldsymbol{s}}$ is slowly updated using the observation values $\boldsymbol{s}$ at every epoch and updates of $\boldsymbol{\alpha}$ are based on the following loss $\mathcal{L}_{\text{sparse}}$:

$$\mathcal{L}_{\text{sparse}} = \mathcal{L}_{\pi_s}(\theta_s, \boldsymbol{\alpha}) + \lambda\left[\mathcal{H}_1(\mathcal{L}_c(\boldsymbol{\alpha})) + \Gamma\boldsymbol{\alpha}\right] \tag{3.8}$$

where λ is the regularisation parameter, $\mathcal{H}_1$ is a unitary Huber loss and Γ can be seen as a Tikhonov matrix that accounts for strong correlations between observations. Its purpose is to penalise α_i whose observation s_i is correlated with any other $s_{j\neq i}$ and thus is redundant (refer to Appendix A for more details). The choice of λ drives the equilibrium between sparsity and control performance.

4. Results and discussion

Unless otherwise stated, all results are obtained using the reference case at $Re = 120$, with the 12-sensor layout described by figure 2 and PPO-CMA as learning algorithm. Figure 8 illustrates a standard learning process. A large variation of mean C_x values can be observed in the first epochs of training, then C_x values concentrate more around their moving average. This is caused by PPO-CMA decreasing the exploratory variance σ when performance stabilises. For all the following results, training is performed over 200 epochs of 480 steps each. A standard training epoch requires around 180s on 4 CPU cores, most of the CPU time being used to run the environment.

4.1. *Control performance and efficiency*

At a Reynolds number of 120, the time-averaged baseline flow drag coefficient is $\langle C_{x,0}\rangle = 1.379$. Performance in terms of drag reduction is computed as a percentage of the average baseline flow drag coefficient $\langle C_{x,0}\rangle$ and also using the drag gain μ_{C_x} introduced in section 2.2. Figure 9 shows the instantaneous drag coefficient C_x, the corresponding action a_t and instantaneous lift coefficient C_l throughout control steps. A first phase, from time $t = 25$ (control starting) to $t = 50$ approximately, shows a rapid transient from the fully developed vortex shedding instability to the controlled flow. This transient corresponds to approximately 4.5 vortex shedding periods. During that phase, actions have a large amplitude and do not seem to follow any simple pattern. In a second phase, from time 50 to the end, the drag coefficient is stabilised to a value below $C_{x,BF}$. This represents a drag reduction of about 18.4% and a drag gain μ_{C_x} around 100.6%. Actions have a significantly reduced amplitude compared to the first phase, and they appear to have a slightly non-zero average. Starting from $t = 150$, a periodic action pattern seems to appear in the form of modulated bursts. These last two points are further discussed in section 4.2.

For other Reynolds number values, drag reduction has also been measured. Results are presented in table 1 and confronted to other comparable studies made on the same case. For $Re = 100$, a drag gain slightly larger than 100% is also reached. The observed mean flow is similar to the base flow. For $Re = 200$, the results from He *et al.* (2000) slightly

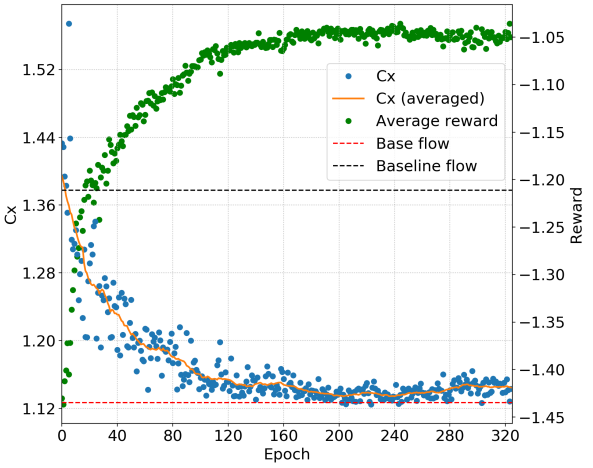

FIGURE 8. Standard learning process. Each C_x value is averaged over the whole epoch, including the transient from developed vortex shedding to controlled flow. This explains the discrepancy with pure performance values on C_x later introduced. The yellow curve is a 20-epoch moving average of C_x values. The average reward (green dots, reward r_t averaged over the current epoch) shows a quasi-monotonic growth that saturates from epoch 200 onward.

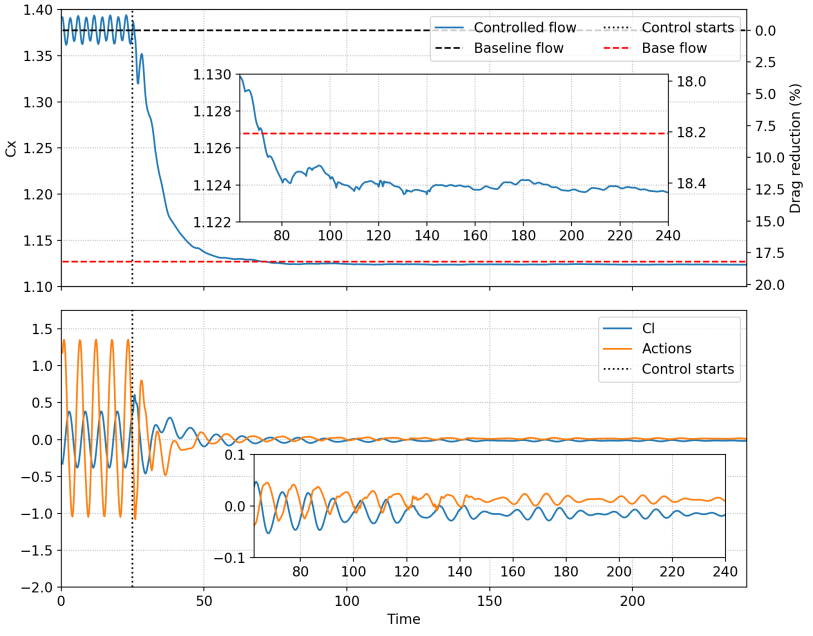

FIGURE 9. Performance of the active flow control strategy. (top): Evolution of the instantaneous drag coefficient C_x. (bottom): Evolution of both action and lift coefficient C_l.

outperform those of the present study in terms of drag reduction. But their drag gain is obtained at the cost of an important mean flow modification, as previously mentioned in section 2.2. Some existing studies on the control of the cylinder flow have not been included in table 1 due to the significant discrepancies of their test case compared to ours, which prevents any straightforward comparison. For instance, Min & Choi (1999), using

Re	Drag red. (%)	Drag gain (%)	PSR	Learning type	Action type	Reference
100	8.0	54.6	-	Gradient descent	Blowing	Leclerc *et al.* (2006)
	8.0	92.7	-	DRL	Blowing	Rabault *et al.* (2019)*
	5.7	66.1	-	DRL	Blowing	Tang *et al.* (2020)*
	14	95.5	-	ANN/ARX	Translation	Siegel *et al.* (2003) Seidel *et al.* (2009)
	14.9	101.7	173	DRL	Blowing	**Present study**
150	4	17.5	-	Parameter study	MHD	Singha & Sinhamahapatra (2011)
	15	65.7	51	Parameter study	Rotation	Protas & Styczek (2002)
	21.2	92.9	20	DRL	Blowing	**Present study**
200	**31**	107.9	-	Adjoint NS	Rotation	He *et al.* (2000)
	28.6	99.6	0.07	POD-based	Rotation	Bergmann & Cordier (2008)
	24.5	85.3	0.26	POD-based	Rotation	Bergmann *et al.* (2005)
	21.6	104.6	-	DRL	Blowing	Tang *et al.* (2020)*
	28.6	99.6	9.2	DRL	Blowing	**Present study**

TABLE 1. Drag reduction and performance comparison. *These cases are slightly different since walls parallel to the flow are added. Action types: "Blowing": Blowing on cylinder poles, "Translation": Vertical translation of the cylinder, "Rotation": Rotation of the cylinder, "MHD": magneto-hydrodynamic forcing

a 360° blowing/suction actuation, end up artificially reducing the equivalent diameter of their cylinder, and while their results and methodology are interesting, comparison of performances is however not relevant here. The work of Arakeri & Shukla (2013), who impose the tangential velocity on the cylinder surface and force a quasi upstream-downstream symmetric flow, is not included in the comparison for similar reasons. Among the related – yet not directly comparable – interesting work, we can also cite Sohankar *et al.* (2015) who achieve a significant drag reduction for a square cylinder flow at $Re = 100$, the paper from Muddada & Patnaik (2010) which shows rather important gains using two small rotating rods in the vicinity of the cylinder, or the results from Chen & Aubry (2005) using magneto-hydrodynamic forcing to stabilise a cylinder flow at $Re = 200$. For a more exhaustive list on the topic, one may refer to the review from Rashidi *et al.* (2016).

Another important indicator of the performance of the control is the energy required for drag reduction. Considering the time-averaged baseline flow drag power ($P_0 = \frac{1}{2}\rho U_\infty^3 D \langle C_{x,0} \rangle$) as reference, the actuation power peaks at 22% of P_0 in the early stages of the first control phase, but only represents less than 0.3% of P_0 on average in the second phase (see figure 10). Thus the total power expenditure (necessary to both counteract drag and implement action), is temporarily higher than for the baseline flow but is quickly counterbalanced by the significant decrease of both drag and actuation powers during the second control phase. In the example shown in figure 10, the energy trade-off starts being beneficial 13 time steps after the control starts, which is long before the flow stabilisation.

The Power Saving Ratio (PSR) was introduced by Protas & Wesfreid (2002) and is defined as the ratio of the gain in drag power to the control power. In quasi-steady controlled regime, $PSR \approx 71$ for $Re = 120$, showing that the control obtained here is highly energy-efficient. For other Reynolds numbers, PSR are reported in table 1, and

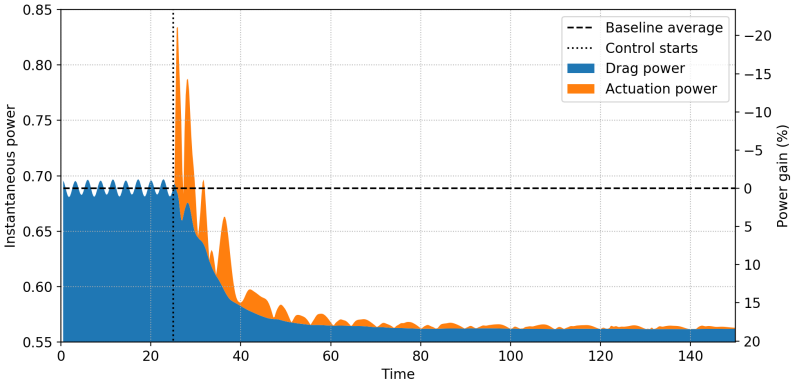

FIGURE 10. Evolution of power expenditure throughout control. The drag power is the power necessary to withstand drag forces, the actuation power represents the power spent on action implementation.

the values found are significantly higher than 1 even for the highest Re considered. It can be noticed that for $Re = 150$, Protas & Styczek (2002) achieved extremely energy-efficient control that actually outperform the present study in terms of PSR (but with a lesser net drag reduction). However, their actuation is made through cylinder rotation, and the actuation power does not consider the inertia of the rotating cylinder (the mass of the cylinder is considered null). This highlights that power-based comparisons between different actuation types, especially in the case of cylinder rotation, may have a limited relevance and should be considered carefully.

It is interesting to note that, despite its high energy-efficiency, the control policy is obtained without any explicit penalisation of the instantaneous control power. The actuation power expenditure is not directly included into the measured performance during learning. However, the reward r_t is penalised by the time-averaged lift coefficient, which ensures parsimonious actions since a strong action generates strong lift. Even though C_l is averaged over one vortex shedding period, the periodicity of the flow varies (or even vanishes) during training which makes a perfect compensation of positive and negative actions' effect on C_l very unlikely. As described early on, slightly non-zero-average actions are systematically observed during the second control phase. Thus, a lack of convergence cannot account for this fact. Instead, an increase in the penalisation on lift through α causes a reduction of this constant component.

However, an increase in α has downsides. By trying several values within the range $[0; 5]$, it has been observed that, for $Re = 120$, the chosen value $\alpha = 0.2$ is close to the optimal trade-off between pure performance and energy consumption. For both an increase or a decrease of α, the PSR decreases and the drag reduction shows a very slight decline. The slight negative effect on the PSR when α decreases below 0.2 can be explained by the reduction of the penalisation on large actions, thereby increasing the control power expenditure. On the other hand, an overly large value of α reduces the observed exploratory variance due to the strong disadvantage put on large amplitude actions that are necessary in the early stages of the control to achieve a near-stabilisation of the flow. The search of the optimal α value has only been performed for $Re = 120$, and the value of 0.2 has been retained for other Reynolds values. Therefore, the PSR values presented in table 1 may not be optimal and might be improved by a careful choice of α. But from the results obtained for $Re = 120$, it appears that α is not a sensitive

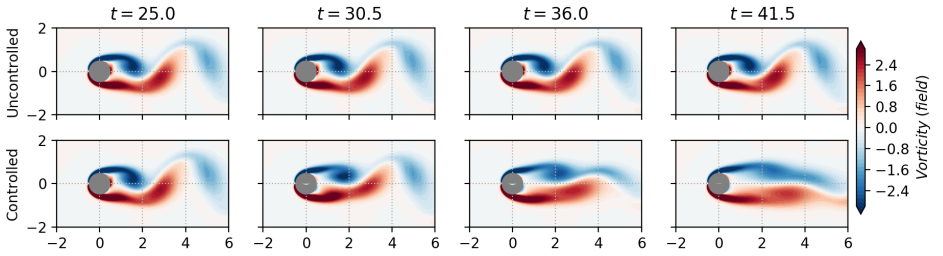

FIGURE 11. Comparison of uncontrolled (top) and controlled (bottom) flows in the transient phase of the control strategy.

parameter: it leads to negligible changes in drag reduction performance, and for a wide range of α values, the PSR remains significantly higher than 1.

4.2. *Analysis of the controlled flow*

A common difficulty with deep learning approaches is the physical understanding of the results. Unfortunately, no simple action pattern has been noticed throughout the evaluations of the control strategy, whether it is for the first or second control phase. Unsuccessful attempts to reproduce this action behaviour with simpler linear controllers (simple gain and delayed response) might indicate that complexity is required to reach the observed control efficiency. While it is hard to precisely explain how the control policy acts on the flow to reduce the drag, the present section nonetheless attempts to describe the control based on an *a posteriori* analysis of the flow.

As studied by Nair *et al.* (2020), who used cylinder rotation or momentum injection parallel to the flow to impose an energy optimal phase-shift control, the drag reduction seen in the transient phase, is caused by the delay in vortex shedding. This generates "elongated vortex structures", that also stabilise the instantaneous recirculation bubble. Similar observations were made in our case. As shown by figure 11, the first phase of the control strategy is a fast transient from fully developed vortex shedding to a stabilised cylinder wake, where the actions trigger the shedding of vortices slightly earlier than the natural shedding. This results into longitudinally stretched and weaker vortical structures.

Once the flow has been stabilised and is nearly steady, its drag coefficient is very close to $C_{x,BF}$. Figure 12 compares the convergence of C_x with the length of the instantaneous recirculation bubble. This length is multiplied by more than 2.5 during the control phase and peaks at 99.5% of the base flow recirculation bubble length. The correlation of both the increase of the length of the recirculation bubble and the drag reduction is a well-known fact (Protas & Wesfreid 2002; Rabault *et al.* 2019). The recirculation bubble lengths found in this study are in good agreement with the reference literature (Zielinska *et al.* 1997; Protas & Wesfreid 2002). From time step 100 onward, both base flow and controlled flow have a very similar recirculation bubble, as illustrated by figure 13. The "tail" of the controlled bubble slowly flaps vertically with a very moderate displacement amplitude ($\Delta y < 0.3$) at $St \approx 0.12$. This confirms that the control policy tends to lead the flow towards the base flow, the latter being an unstable optimum with respect to drag. The controlled flow reaches a small amplitude cycle around this equilibrium point.

As shown in figure 14 (left), the spectral analysis of the action during the second phase reveals two main oscillating components $St_1 \approx 0.11$ and $St_2 \approx 0.14$ and three secondary peaks at Strouhal numbers $\delta_{St} = St_2 - St_1$, $St_3 = St_1 + St_2$ and $St_4 = 2St_2$, the latter

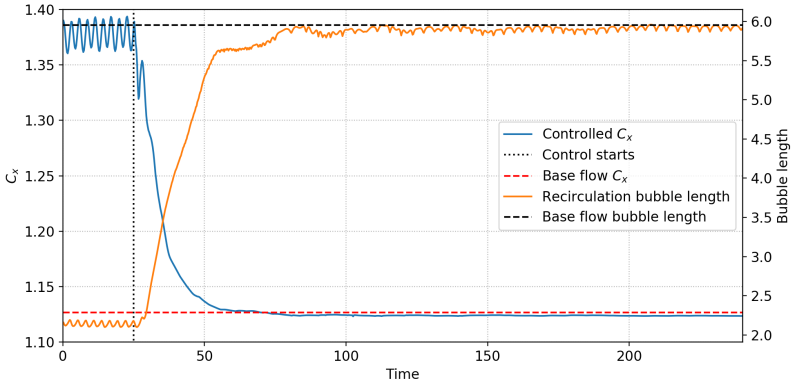

FIGURE 12. Evolution of C_x and of the length of the instantaneous recirculation bubble throughout time.

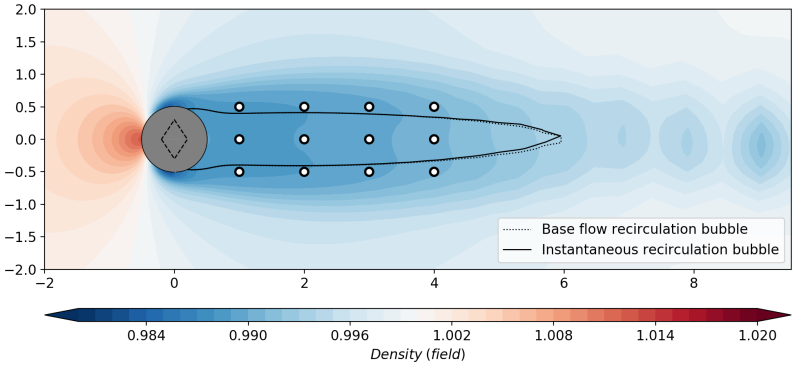

FIGURE 13. Comparison of the base flow and the controlled flow recirculation bubbles once the flow is stabilised.

having amplitudes at least two orders of magnitude lower than the main components. Since δ_{St} and St_3 are the marks of nonlinear coupling between the two main components, one can assume a nearly interaction-free superimposition of the two main waves at St_1 and St_2. Their corresponding Fourier modes (not shown here) peak near the location of the "tail" of the recirculation bubble.

Note that in the stabilised phase, the state is close to the base flow and the actions are small, such that the flow evolves in a linear regime. The dominant Strouhal numbers of this phase are significantly lower than the natural vortex shedding frequency $St = 0.18$. This may be easily understood by performing a resolvent analysis, which describes the frequency-response of the flow in the vicinity of a steady state. If $\boldsymbol{q}$ denotes the flow state, $\boldsymbol{f}$ an external forcing, and $\mathcal{N}$ the Navier-Stokes operator, then Navier-Stokes equations may be written in the compact form:

$$\frac{\partial \boldsymbol{q}}{\partial t} = \mathcal{N}(\boldsymbol{q}) + \boldsymbol{f}. \tag{4.1}$$

Decomposing the flow as the sum of the base flow $\boldsymbol{q_{BF}}$ and of a small perturbation $\boldsymbol{q'}$ and since the base flow is a steady solution ($\mathcal{N}(\boldsymbol{q_{BF}}) = 0$), the fluctuations $\boldsymbol{q'}$ are governed

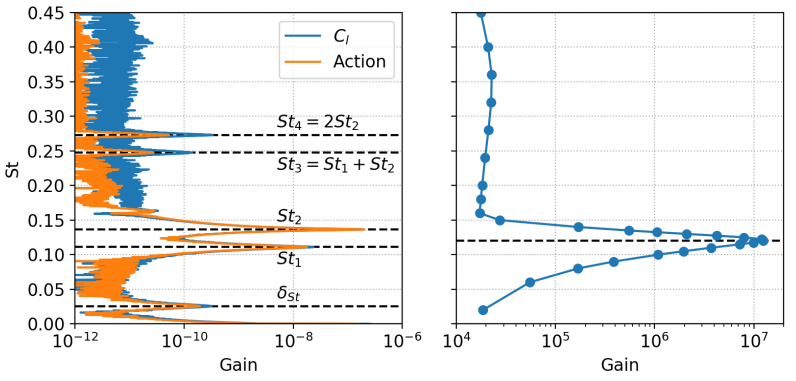

FIGURE 14. (left): Spectra of the lift coefficient C_l and of the action during the second control phase. $\delta St = St_2 - St_1$ (right): Evolution of the optimal gain of the base flow resolvent operator with the external forcing Strouhal number at $Re = 120$.

by the following first-order approximation:

$$\frac{\partial \boldsymbol{q'}}{\partial t} - \boldsymbol{J}\boldsymbol{q'} = \boldsymbol{f} \quad \text{with } \boldsymbol{J} = \frac{\partial \mathcal{N}}{\partial \boldsymbol{q}}(\boldsymbol{q_{BF}}). \tag{4.2}$$

The previous equation may then be Fourier decomposed. Denoting the angular frequency by ω, the identity matrix by $\mathcal{I}$ and the Fourier-transformed variables by a hat notation:

$$(i\omega\mathcal{I} - \boldsymbol{J})\,\hat{\boldsymbol{q}}' = \hat{\boldsymbol{f}} \quad \rightarrow \quad \hat{\boldsymbol{q}}' = \underbrace{(i\omega\mathcal{I} - \boldsymbol{J})^{-1}}_{\mathcal{R}}\,\hat{\boldsymbol{f}}, \tag{4.3}$$

with $\mathcal{R}$ being the resolvent operator. The highest singular value of $\mathcal{R}$, which is a function of ω, gives the highest linear gain σ that may be achieved through an external forcing (see for instance Beneddine (2017)). Formally, it reads:

$$\sigma^2(\omega) = \max_{\hat{\boldsymbol{f}}} \frac{||\hat{\boldsymbol{q}}'||_{\boldsymbol{q}}}{||\hat{\boldsymbol{f}}||_{\boldsymbol{f}}}, \tag{4.4}$$

with $||\cdot||_{\boldsymbol{q}}$ and $||\cdot||_{\boldsymbol{f}}$ representing norms on the response and forcing spaces respectively (classically associated with the kinetic energy for the response, and the L_2-norm for the forcing). As illustrated by figure 14 (right), the highest optimal gain is obtained for $St = 0.12$ (consistently with Barkley (2006); Jin *et al.* (2019)) and the flow is responsive to only a narrow range of Strouhal numbers (below 0.15). It is therefore not surprising that the values associated with the control fall within this range. But interestingly, the control avoids the highest gain frequency and the particular selection of the two specific frequencies $St_1 = 0.11$ and $St_2 = 0.14$ remains an open question. To our knowledge, this is not reminiscent of any existing work related to the linear control of the vortex shedding near the base flow.

4.3. *Robustness*

4.3.1. *Reynolds robustness*

An assessment of the control policy robustness across a range of Reynolds numbers has been performed. Unlike Tang *et al.* (2020) who trained their policy on several Reynolds numbers values (100, 200, 300 and 400) and evaluated it on a mix of "seen" and "unseen" Reynolds numbers, our policy has been trained on a single Reynolds value $Re = 120$,

evaluations have been performed on a range spanning from $Re = 100$ to 216 and compared with cases specifically trained on those Reynolds numbers. As illustrated by figure 3, this range of Reynolds numbers corresponds to a variation in vortex shedding Strouhal number (St) of around 18%. Moreover, the non-dimensional amplitude of the pressure fluctuation displays a factor 2 between the two extreme Re values considered, showing that the dynamics of the flow, although not radically different, is still noticeably altered in this range of Reynolds number, such that the robustness is tested in actual off-design conditions.

Figure 15 shows that the control is remarkably robust. Note that, as previously introduced, the flow state is made non-dimensional by the reference density ρ_∞, upstream velocity U_∞ and static temperature T_∞. Reference velocities, pressures and case geometry (cylinder diameter and sensor location) being held constant is decisive for the robustness of the control policy. This ensures indeed a nearly constant convection time between sensors and comparable variation amplitudes both for sensors (on pressure) and for actuators (on mass flow) across the different Reynolds numbers considered. The only varying factor between different Reynolds flows is the change in vortex shape, their relative strength and organisation. The policy, acting only as a function of the current observation s_t, is insensitive to the variation in the von Kármán vortex street convection velocity. It is hence only affected by the change in instantaneous form of the flow structures, and the present results proves that the control law handles very well these changes.

Non-dimensonalisation also circumvents the issue of neural network input normalisation. Once neural networks' weights and biases are tuned to adapt to the range of input values, they remain appropriately tuned as this range does not overly change across Reynolds numbers. Tang *et al.* (2020) used the same non-dimensionalisation scheme. Thus, even though their deep learning algorithm is different, the robustness they observed may be explained by the fact that the policy is robust over a wide range of Reynolds numbers even with a single Reynolds number training. Adding several other Reynolds numbers in the training marginally improves an already strong robustness.

This robustness is very promising for future experimental exploitation of deep learning for flow control, since flow conditions are subject to uncertainties in experiments. Should one attempt to use a CFD-trained policy to control an actual experimental case, it may be interesting to consider transfer learning, which consists in re-training only specific layers of the network rather than the whole model to quickly adapt it to a slightly different configuration, without restarting from scratch. Several studies have been done in the image processing community on the topic (Huh *et al.* 2016) and should be considered for future work.

Figure 15 also shows better robustness for lower Reynolds numbers than for higher ones (compared to the training Re). One of the reasons may be the chosen sensor layout, fixed across all cases but which covers more of the base flow recirculation bubble for lower Reynolds numbers. It has been shown indeed that its length increases with the Reynolds number. The 12-sensor layout spans over 75% of the recirculation region at $Re = 100$, but only 55% at $Re = 216$.

4.3.2. *Observation noise robustness*

Assessing the tolerance of the control strategy to measurement errors is a key point in the transposition of that method to real-world experiments, where measurement noise is unavoidable. Noise robustness is therefore important in the perspective of transfer learning from a numerically trained case (without noise) to an experimental setup. To this end, the robustness of a zero-noise-training policy has been assessed and compared with

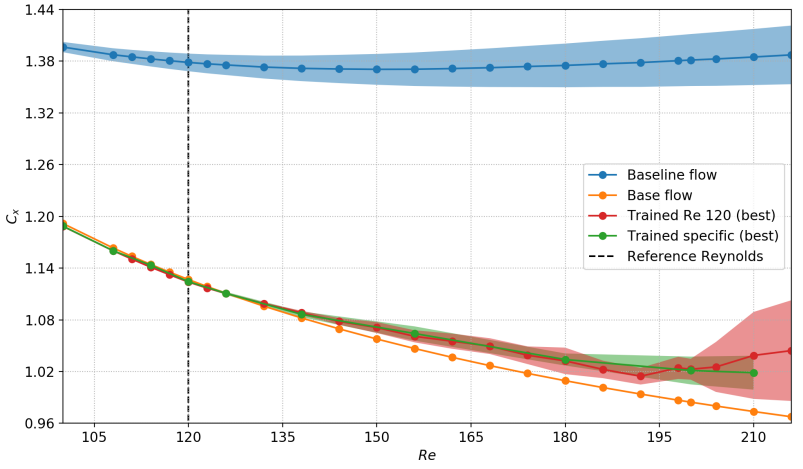

FIGURE 15. Robustness to a Reynolds number variation. The best case (among 10 test cases) trained at $Re = 120$ is evaluated at different Reynolds numbers (red curve) and compared to the best control policy (among 10 test cases) specifically trained on the target Reynolds number (green curve). Shaded areas represent the standard deviation of the controlled drag coefficient C_x.

policies trained on noisy data. Added noise is parameterised, using a relative amplitude σ. Noisy observations $\tilde{s}_t$ are computed as:

$$\tilde{s}_t = s_t + \bar{s}_t \sigma \mathcal{N}(\cdot|0,1) \tag{4.5}$$

where $\bar{s_t}$ is the average pressure over all sensors at time t, which is found to be relatively steady and $\mathcal{N}(\cdot|0,1)$ is a standard random normal probability distribution. Figure 16 compares the performances of policies trained at different noise levels σ and evaluated on a range of noise levels from 0 to 1. One can notice that the level of training noise does not seem to impact performances in a significant manner up to $\sigma = 0.5$, which corresponds to very noisy measurements that certainly exceeds the actual noise one may expect in most experiments (see figure 17). Unexpectedly, figure 16 tends to show that a zero-training-noise policy seems overall slightly more robust to noise than others at different training noise levels. Therefore, in the present case, it is unnecessary to account for measurement noise during the training, which is once again promising for the possible transfer of CFD-trained models to experiments.

Figure 17 illustrates this robustness throughout time for a policy trained with $\sigma = 0$. Despite large noise disturbances, the control policy achieves good performances. Even with extreme noise levels such as $\sigma = 1$, the drag reduction reaches about 12% on average. This is only possible in a closed-loop control strategy, and may be explained by the feedback characteristic of the problem that enables for efficient error correction from one control step to the next. Both observation and action signal-to-noise ratios (SNR) are assessed on the second control phase (having steady statistics) as:

$$SNR = \frac{\text{noise-free variation amplitude}}{\text{noise standard deviation}} \tag{4.6}$$

Action noise is defined as the difference between the action computed using noisy observations and the action based on noise-free measurements. Results are reported in Table 2. Note that apparent discrepancy between SNR and σ values are due to the definition of each quantity: SNRs are computed considering the amplitude of variation

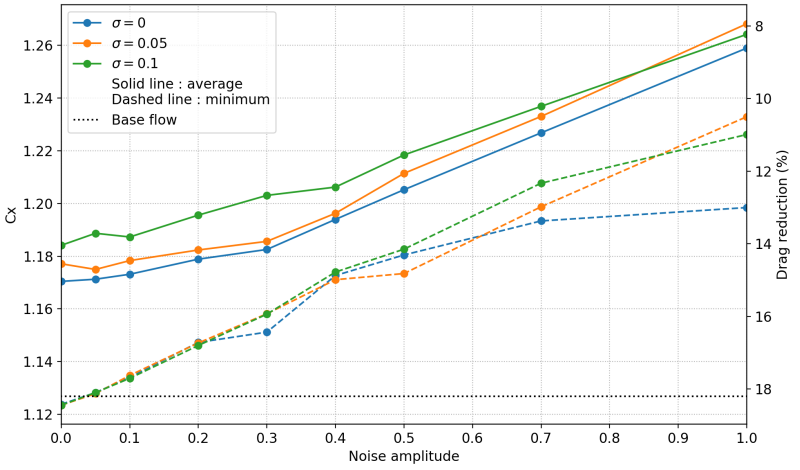

FIGURE 16. Robustness to Gaussian noise on observations. Each curve represents the mean (solid line) or the best (dashed line) performance in drag coefficient of a 20 test-case batch trained with noise levels from $\sigma = 0$ to $\sigma = 0.1$ and evaluated on noise levels ranging from $\sigma = 0$ to $\sigma = 1$.

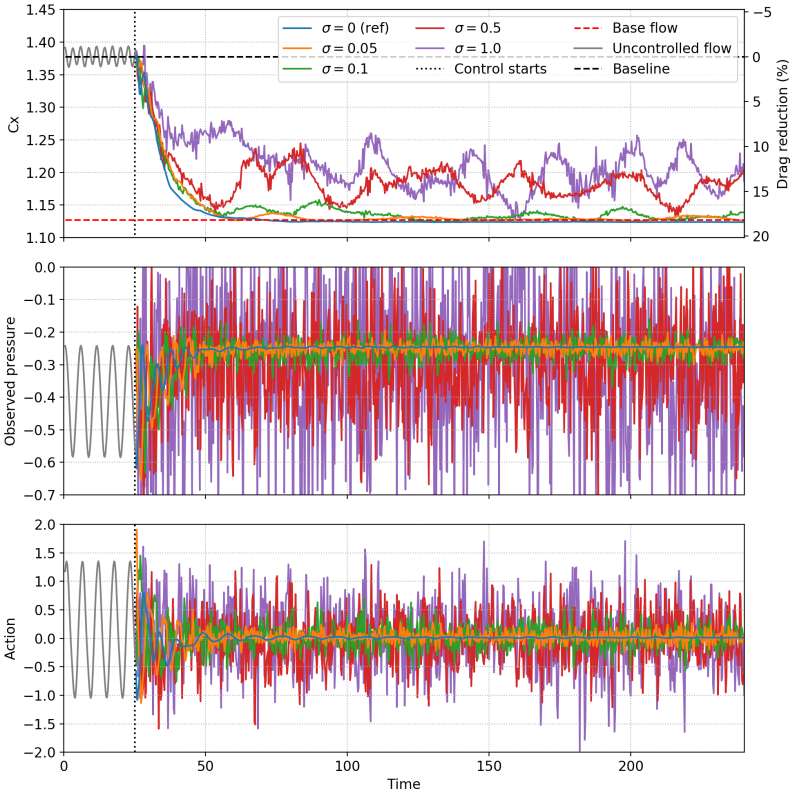

FIGURE 17. Robustness to Gaussian noise on observations for a policy trained without noise ($\sigma = 0$). (top): Evolution of C_x throughout time, for different noise levels. (middle): Noisy pressure signal s_0 located in (1,0.5). (bottom): Corresponding action taken by the actor.

σ	Observation SNR	Action SNR	Average drag reduction (%)
0	∞	∞	18.4
0.05	0.33	0.17	18.1
0.1	0.18	0.09	17.8
0.5	0.04	0.04	14.2
1	0.02	0.03	13.1

TABLE 2. Noise robustness comparison. SNR: Signal-to-Noise Ratio

of the signal, thus excluding the signal's time-averaged value, while the noise level driven by σ is measured as a fraction of the signal value, including its constant component. It can be seen that the action SNR has the same order of magnitude as the observation SNR, their ratio ranges between 0.7 and 2. In particular, the SNR of observations and actions become closer as the level of noise increases. This highlights the robustness of the policy, which does not diverge from optimal actions due to spurious fluctuations within the observations. The errors do not accumulate over time and the closed-loop system appears able to rectify the previous erroneous action to contain the deviation from the optimal controlled flow-state. The policy is therefore sufficiently insensitive to input errors in that range of noise levels to ensure a strong robustness. In addition, it is possible that the decorrelation of these errors between each measurements helps mitigating the effects of the noise.

4.4. *Impact of the sensor number and location on the control performance*

As introduced previously, the optimisation of both the sensors number and location is a widely explored domain. In this part, a systematic study on sensor configurations within a 3-by-5 grid-like layout is performed. Figure 18 illustrates the learning curves of 10-case batches having from 3 to 15 sensors. The addition of the second and third columns of sensors (located in $x = 2$ and $x = 3$) yields a significant gain in performance, and one can notice that 12 and 15-sensor layouts have a very similar average performance. Thus it is possible to conclude that the three additional sensors (located in $x = 5$) are not useful to the control strategy.

Figure 19 shows the effect of the location of pressure observations, for an array of 6 sensors that are displaced in the streamwise direction. This time, the importance of the first sensor column (located in $x = 1$) is demonstrated by the noticeable gain in drag reduction between the first two layouts (blue and yellow curves). The importance of the third sensor column ($x = 3$) is once again stressed by the decrease in performance between the green and red curves. Within this predefined combinatorial set, this partial study highlights the relevance of sensors closest to the cylinder. These first preliminary tendencies are confronted in the next section with the results from the newly-proposed S-PPO-CMA algorithm that is designed to provide the optimal sensor location for the control.

4.5. *Optimal choice of sensors: results from the S-PPO-CMA algorithm*

The S-PPO-CMA algorithm, described in section 3, is used here to derive optimal sensor placement for any allocated number of sensors ranging from 1 to 9, within the imposed 15-sensor grid-like pattern. The number of sensors is indirectly controlled through the value of the L_0 regularisation constant λ, that balances the gradients of both $\mathcal{L}_{\pi_s}$ (performance loss) and $\mathcal{L}_c$ (complexity loss). Figure 20 shows the achievable drag reduction with respect to the number of sensors i and the corresponding sensor

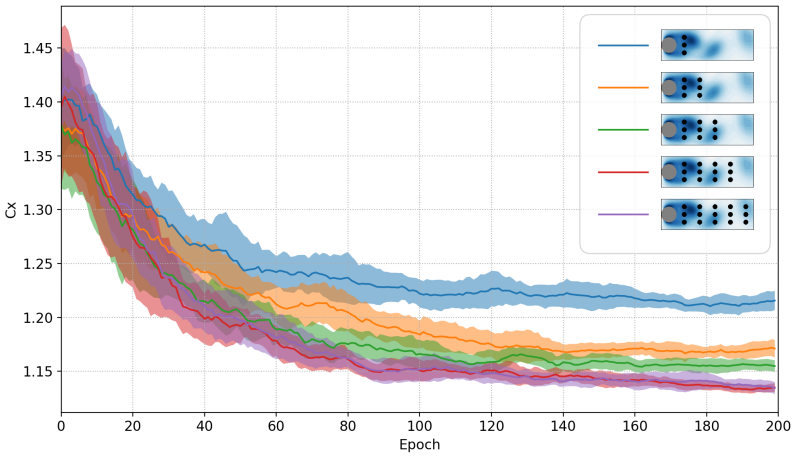

FIGURE 18. 10-case batch-averaged learning curves for different sensor layouts. Shaded areas represent the standard deviation of the corresponding plotted quantities.

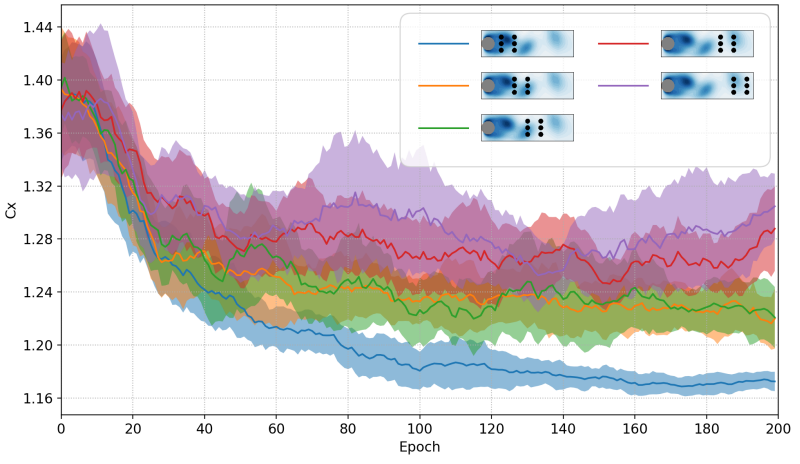

FIGURE 19. 10-case batch-averaged learning curves for different sensor layouts. Shaded areas represent the standard deviation of the corresponding plotted quantities.

layout l_i. Note that due to the symmetry of the configuration, there always exist pairs of symmetric layout that achieve identical performances. The S-PPO-CMA algorithm randomly outputs one of the two optimal layouts for each value of λ, but only one layout is displayed in the figures for simplicity.

With a single sensor, the drag reduction is around 11% and it peaks to approximately 18% for 5 sensors or more. The sensor pattern's tendency to fill without relocating existing sensors, meaning that $l_i \subset l_{j>i}$, is a sign of convexity of this problem in the sense that any combination of the optimal layout set is also part of this set. It is interesting to notice that, starting from the 5-sensor optimal layout, the addition of more sensors does not improve drag reduction, which makes this 5-sensor layout the optimal trade-off between performance and sensor setup complexity. To our knowledge, this layout is not reminiscent of anything used in the large number of existing studies on the control of

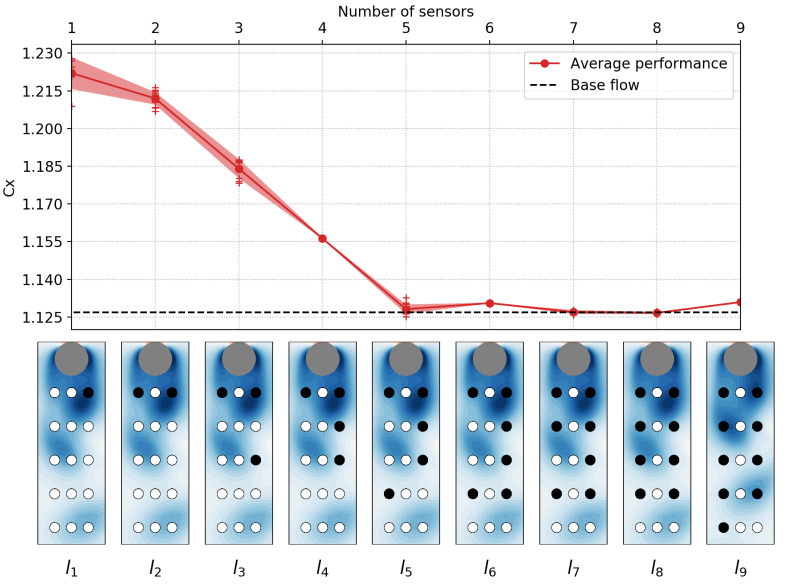

FIGURE 20. Evolution of both drag reduction and optimal sensor layout with the number of sensors. Layout thumbnails l_1 to l_9 are rotated 90° clockwise.

the 2D cylinder wake. Thus, this highlights the usefulness of the S-PPO-CMA algorithm since optimal sensor placement is, even in such a simple case, not particularly intuitive.

The centerline locations ($y = 0$) do not appear relevant for the control since the corresponding sensors are never selected by the algorithm. A possible explanation may be that these sensors cannot provide information relative of the instantaneous asymmetry of flow, and are thus not fit to choose the action's sign. The first two layouts l_1 and l_2 validate the importance of the first sensor column, and the selection of sensors shows a weaker importance of locations beyond $x = 4$. This is in line with the conclusions of section 4.4.

As discussed in the introduction, many studies optimise sensor placement based on the linear framework of POD, with the underlying idea that the better the estimation of mode coefficients is, the better the reconstruction and control performance are. They naturally often choose locations where the POD modes are strong. Figure 21 illustrates the superimposition of the sensor locations with the first three POD modes derived from the natural transient from base flow to fully developed vortex shedding. These three modes account for more than 95% of the transient's energy. Despite that these modes are only valid for control trajectories that stay close to this natural transient, the choice of l_2 seems reasonable as it allows estimations of both shift mode and second vortex shedding mode simultaneously, since sensors are close to the extrema of this modes (refer to left and right panels of figure 21). The second column of sensors appears less able to provide relevant information on the shift mode. Figure 21 also confirms that the centerline sensors are unfit to estimate von Kármán modes, which account for the instantaneous asymmetry of the vortex shedding.

Comparing the second and third layouts l_2 and l_3, it appears that, given the first two probe locations, an additional sensor is preferred in the third column rather than in the second. This might be because this layout provides a better "coverage" of the instantaneous recirculation bubble. Additionally, despite the lack of mean flow symmetry during

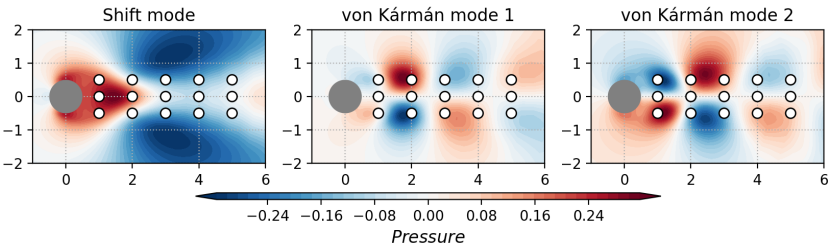

FIGURE 21. Comparison of the sensor locations with the first three POD modes of a natural transient from base flow to fully developed vortex shedding.

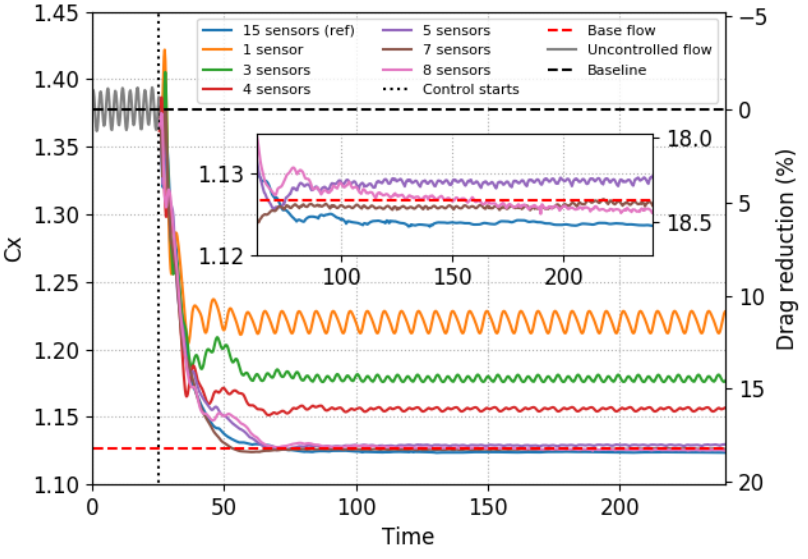

FIGURE 22. Performance comparison for different optimal sensor layouts.

the transient phase of control, sensors tend to concentrate on a single streamwise row. From a POD-based control viewpoint, this may not be optimal for modes reconstruction. This shows a strength of the present approach: an estimation of the full flow field (or the dominant POD modes) is likely not to be needed for the control. Therefore, searching for points that allow such a full reconstruction may be sub-optimal. In that context, favouring a precise vortex tracking, whose centre travels in the vicinity of the external sensor rows over a more complete estimation of the wake may lead to better performance.

Figure 22 compares the performances of some the previously found sensor layouts throughout time. It confirms that as from 5 sensors, optimal performance ($\sim 18\%$) is reached. All configurations show a comparable transient phase, that stops earlier both in time and drag reduction for the sparsest sensor configurations. Despite an already significant drag reduction, layout l_1 seems unable to notably stabilise the vortex shedding instability. A much better steadiness is achieved with 3, 4 and more sensors.

5. Conclusion

In this study, PPO-CMA, implemented on a laminar 2D cylinder case study, discovered efficient nonlinear control strategies with only 12 pressure sensors in the cylinder's wake. A drag reduction of 18.4% is reached for $Re = 120$. Comparable performances are achieved for other values of Re, which match state-of-the-art control performances. The control strategy has been analysed *a posteriori* and split into two distinct phases. After a rapid transient phase where large amplitude actions bring the mean flow close to the base flow, the policy simply keeps the controlled flow on a small amplitude limit cycle with weaker actions, whose temporal spectrum is dominated by two distinct frequencies, both close to the base flow's resonance frequency. This translates to a temporary disadvantageous energy expenditure, that quickly becomes beneficial with a PSR in the order of $O(10)$ for the whole range of Re considered.

The robustness with respect to a variation in Reynolds number and measurement noise has also been quantified. It has been demonstrated that a policy trained at $Re = 120$ shows near-optimal performances that match the drag reductions achieved by specifically trained policies on Reynolds numbers in the range $[100; 216]$. Such robustness is in part explained by the chosen non-dimensionalisation scheme of the case study. The impact of measurement noise has been assessed by both training policies with different noise levels and by evaluation of these policies on different levels of noisy observations. It has been concluded that overall, the control is very robust, yet noise-free trained policies seem slightly more robust than those trained on noisy data and that the actor takes advantage of both the decorrelation of noise between observations and the closed-loop nature of the problem to demonstrate efficient control on noisy environment. Those two aspects show interesting possibilities for direct application of reinforcement-based feedback control on real cases and, in the scope of transfer learning, for a synergistic coupling of numerical simulation and experiments for active flow control.

An optimisation of the number of sensors has also been performed while preserving performances. After a first coarse systematic study showing the importance of having sensors close the cylinder, S-PPO-CMA has been introduced and implemented on the test case. This new algorithm, discovering optimal sensor layouts for reinforcement-based control, selects the most relevant sensors and discards redundant and irrelevant ones. Thus, the number of sensors has been reduced to only 5 while keeping state-of-the-art performance. The obtained sensor layout has been compared with both the outcomes of our systematic study and conclusions of other linear (mostly POD-based) studies. Several explanations have been proposed to back the observed consistency of these results.

A future study could consider extending this approach to larger sensors layouts for more complex cases. One could try to improve performance on the present case study. As shown by He *et al.* (2000) this could mean seeking for other mean flow configurations with for instance induction of a reverse von Kármán street (Bergmann *et al.* 2006). This would more likely require a system similar to what Tang *et al.* (2020) did and a much lower energy efficiency.

Acknowledgement

This work is funded by the French Agency for Innovation and Defence (AID) via a PhD scholarship. Their support is gratefully acknowledged. The authors would like to thank Jean Rabault for the valuable discussions and advice.

Declaration of Interests

The authors report no conflict of interest.

Appendix A. PPO, PPO-CMA and S-PPO-CMA learning algorithms

Given a partial state s_t and under a policy π, the advantage value $A^\pi(a_t, s_t)$ compares the value (R_t) of a specific action a_t with the expected value of a randomly selected action according to $\pi(\cdot, s_t)$. The latter is simply $V^\pi(s_t)$, the state value computed by the critic neural network V, which is an estimator of the expected return $\mathbb{E}_\pi\left[\sum_\tau \gamma^\tau r_\tau\right]$ following policy π. The advantage estimates "how much better" is a_t compared to the "average" action sampled following π: $A^\pi(a_t, s_t) = R_t - V^\pi(s_t)$. A more stable method of estimating advantage, Generalised Advantage Estimation (GAE) (Schulman *et al.* 2015*b*), depending on an extra parameter λ_{GAE}, is used here. PPO and its variants rely on the estimation of the advantage function and use advantage values as weights for the computation of gradients. Concerning PPO, the policy $\pi_\theta(\cdot, s_t)$ follows a random normal distribution whose mean $\mu_\theta(s_t)$ is the output of a neural network π (θ being its weights/biases) and standard deviation σ is a predefined hyper-parameter. The surrogate loss computed during policy update is:

$$\mathcal{L}_{PPO}(t) = \min\left(r_\theta, \text{clip}(r_\theta, 1-\varepsilon, 1+\varepsilon)\right) A^\pi(a_t, s_t) \tag{A 1}$$

$$\text{with } r_\theta(t) = \frac{\pi_\theta(a_t, s_t)}{\pi_{\theta_{old}}(a_t, s_t)} \quad \text{and } \text{clip}(a, b, c) = \min\left(\max\left(a, b\right), c\right) \tag{A 2}$$

with θ_{old} being the weights of π before update and ε a clipping hyper-parameter. Algorithm 1 describes the steps of PPO, with ϕ being the weights and biases of V.

Algorithm 1 Proximal Policy Optimisation algorithm

$k \leftarrow 0$
Initialise ϕ_0 and θ_0
while θ_k is not converged **do**
  Collect a set of trajectories $\mathcal{D}_k = \{\tau_i\} = \{(s, a, r)_i\}$ of length T using policy π_{θ_k}
  for all $t \in [0, T]$ **do**
    Estimate rewards-to-go $\hat{R}_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$
    Estimate advantage $\hat{A}_t$ through GAE
  end for
  Estimate policy gradient $\hat{g} = \nabla_\theta \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \mathcal{L}_{PPO}(a_t, s_t, \hat{A}_t)$
  Compute policy update $\theta_{k+1} \leftarrow \theta_k + \alpha \hat{g}_k$ (or other gradient technique)
  Fit value function by regression: $\phi_{k+1} \leftarrow \arg\min_\phi \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t\right)^2$
  $k \leftarrow k + 1$
end while

PPO-CMA (Hmlinen *et al.* 2018) relies on a similar loss estimation technique, but uses two unclipped surrogate objectives. The standard deviation of the policy π_θ is, this time, also an output of the actor π. The latter is then trained twice per update phase using two different losses $\mathcal{L}^\sigma_{PPOCMA}$ and $\mathcal{L}^\mu_{PPOCMA}$. PPO being known for instability in policy updates due to negative advantages, PPO-CMA uses a mirroring technique to consider

the information brought by negative advantage samples:

$$\mathcal{L}^{\sigma}_{PPOCMA}(t) = \delta_{A^{\pi}>0} A^{\pi}(a_t, s_t)\pi_{\theta}(a_t, s_t) \tag{A 3}$$

$$\mathcal{L}^{\mu}_{PPOCMA}(t) = \delta_{A^{\pi}>0} A^{\pi}(a_t, s_t)\pi_{\theta}(a_t, s_t) - \delta_{A^{\pi}<0} A^{\pi}(a_t, s_t)\pi_{\theta}(\mu_t - 2a_t, s_t) Z(a_t - \mu_t) \tag{A 4}$$

$$\text{with } \delta_{f(x)} = \begin{cases} 1 & \text{if } f(x) \text{ is True} \\ 0 & \text{otherwise} \end{cases} \tag{A 5}$$

where Z is a Gaussian kernel damping function, that vanishes when $||a_t - \mu_t|| \to \infty$. To ensure a more stable convergence of σ, $\mathcal{L}^{\sigma}_{PPOCMA}$ is estimated on a randomly sampled history buffer $\mathcal{B}$ that contains all the information of the past H epochs of training. Algorithm 2 describes the steps of PPO-CMA.

Algorithm 2 Proximal Policy Optimisation algorithm with Covariance Matrix Adaptation

$k \leftarrow 0$
Initialise ϕ_0 and θ_0
while θ_k is not converged **do**
  Collect a set of trajectories $\mathcal{D}_k = \{\tau_i\} = \{(s, a, r)_i\}$ of length T using policy π_{θ_k}
  for all $t \in [0, T]$ **do**
    Estimate rewards-to-go $\hat{R}_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$
    Estimate advantage $\hat{A}_t$ through GAE
  end for
  Append $\mathcal{D}_k$, $\hat{R}$ and $\hat{A}$ to the history buffer $\mathcal{B}$
  Sample $\mathcal{D}'_k$, $\hat{R}'$ and $\hat{A}'$ on history buffer $\mathcal{B}$
  Estimate σ policy gradient $\hat{g}^{\sigma} = \nabla_{\theta} \frac{1}{|\mathcal{D}'_k|} \sum_{\tau' \in \mathcal{D}'_k} \sum_{t=0}^{|\mathcal{D}'_k|} \mathcal{L}^{\sigma}_{PPOCMA}(a'_t, s'_t, \hat{A}'_t)$
  Compute policy update $\theta_{k+1} \leftarrow \theta_k + \alpha \hat{g}^{\sigma}_k$ (or other gradient technique)
  Estimate μ policy gradient $\hat{g}^{\mu} = \nabla_{\theta} \frac{1}{|\mathcal{D}_k|} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{|\mathcal{D}_k|} \mathcal{L}^{\mu}_{PPOCMA}(a_t, s_t, \hat{A}_t)$
  Compute policy update $\theta_{k+1} \leftarrow \theta_k + \alpha \hat{g}^{\mu}_k$ (or other gradient technique)
  Fit value function by regression: $\phi_{k+1} \leftarrow \arg\min_{\phi} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_{\phi}(s_t) - \hat{R}_t \right)^2$
  $k \leftarrow k + 1$
end while

The training of S-PPO-CMA (described in section 3) is similar the standard PPO-CMA. An additional loss $\mathcal{L}_{\text{sparse}}$ is defined to train $\boldsymbol{\alpha}$ values. This contains a Tikhonov matrix Γ, that penalises correlations between observations. Γ is diagonal and for a predefined correlation threshold δ_{corr}:

$$C_i \equiv \{j \neq i \mid \text{corr}(s_i, s_j) > \delta_{corr}\} \tag{A 6}$$

$$\Gamma_{ii} = \begin{cases} 1 & \text{if } C_i \neq \emptyset \text{ and } \exists\, j \in C_i \ \alpha_j > \alpha_i \\ 0 & \text{otherwise} \end{cases} \tag{A 7}$$

where $\text{corr}(s_i, s_j)$ is the correlation of s_i and s_j over the current epoch.

The substitute vector $\bar{\boldsymbol{s}}$ can be seen as a baseline. There is an advantage in gradient accuracy to choose a slowly updated average of the observation vector as baseline. Let us consider:

$$\nabla_{\alpha} \mathcal{L}_{\pi_s}(\theta_s, \boldsymbol{\alpha}) = \underbrace{\text{sign}\left[\mu_s(\boldsymbol{s}, \theta_s, \boldsymbol{\alpha}) - \mu^*(\boldsymbol{s})\right]}_{\delta} \nabla_{\alpha} \mu_s(\boldsymbol{s}, \theta_s, \boldsymbol{\alpha}) \tag{A 8}$$

Parameter	Symbol	Value	Comment/Reference
Flow simulation setup			
Spatial scheme	-	AUSM+	Edwards & Liou (1998)
Mesh nodes (aziumtally)	-	360	-
Mesh nodes (radially)	-	70	-
Temporal scheme	-	BDF2	Curtiss & Hirschfelder (1952)
Numerical time step	dt	5×10^{-3}	-
Action ramp length	-	20 it.	-
Maximum action amplitude	-	2	-
Control step length	Δt	50 it. = 0.25	-
(S)-PPO-CMA hyper-parameters			
Training epochs	-	200	-
Steps per epoch	-	480	-
Actor architecture	π	(512×512)	2 fully connected layers
Critic architecture	V	(512×512)	2 fully connected layers
Return discount factor	γ	0.99	-
GAE control parameter	λ_{GAE}	0.97	Standard value
Optimiser	-	ADAM	Kingma & Ba (2014)
History buffer depth	H	3 epochs	-
Bernoulli choice parameter on action	p	0.2	-
L_0 regularisation parameter	λ	$[0.1; 10]$	-
Correlation threshold	δ_{corr}	0.99	-

TABLE 3. Additional numerical parameters

$$\nabla_\alpha \mathcal{L}_{\pi_s}(\theta_s, \boldsymbol{\alpha}) = \delta \nabla_{\tilde{\boldsymbol{s}}} \pi_{s,\sigma}(\theta_s, \tilde{\boldsymbol{s}}) \odot \nabla_\alpha \tilde{\boldsymbol{s}} \quad \text{with } \tilde{\boldsymbol{s}} = (\boldsymbol{s} - \bar{\boldsymbol{s}}) \odot f(\boldsymbol{u}, \boldsymbol{\alpha}) + \bar{\boldsymbol{s}} \tag{A 9}$$

$$\nabla_\alpha \tilde{\boldsymbol{s}} = (\boldsymbol{s} - \bar{\boldsymbol{s}}) \odot \nabla_\alpha f(\boldsymbol{u}, \boldsymbol{\alpha}) \tag{A 10}$$

$$\text{Thus :} \quad \nabla_\alpha \mathcal{L}_{\pi_s}(\theta_s, \boldsymbol{\alpha}) = \delta \nabla_{\tilde{\boldsymbol{s}}} \pi_{s,\sigma}(\theta_s, \tilde{\boldsymbol{s}}) \odot \nabla_\alpha f(\boldsymbol{u}, \boldsymbol{\alpha}) \odot (\boldsymbol{s} - \bar{\boldsymbol{s}}) \tag{A 11}$$

If $\bar{\boldsymbol{s}} = 0$, for the values of $\boldsymbol{u}$ where $\nabla_\alpha f(\boldsymbol{u}, \boldsymbol{\alpha}) \neq 0$, the amplitude of the gradient is proportional to the average value of $\boldsymbol{s}$. If there is an important disparity in observation averages (*i.e.* $avg(s_i) \gg avg(s_j)$), α_i will be updated much faster than α_j which is not wished. Furthermore, for "small" batches, any correction based on the batch average (using $avg(\boldsymbol{s})$ instead of $\bar{\boldsymbol{s}}$) introduces a bias due to the lack of convergence of the epoch-averaged estimator avg. A slowly updated baseline for $\bar{\boldsymbol{s}}$, is thus necessary.

Appendix B. Numerical hyper-parameters

Table 3 presents the main numerical parameters of both the simulated case and the learning algorithm, with the notations used in the article (if introduced).

REFERENCES

ABADI, MARTN, BARHAM, PAUL, CHEN, JIANMIN, CHEN, ZHIFENG, DAVIS, ANDY, DEAN, JEFFREY, DEVIN, MATTHIEU, GHEMAWAT, SANJAY, IRVING, GEOFFREY & ISARD, MICHAEL 2016 Tensorflow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)* pp. 265–283.

ARAKERI, JAYWANT H. & SHUKLA, RATNESH K. 2013 A unified view of energetic efficiency in active drag reduction, thrust generation and self-propulsion through a loss coefficient with some applications. *Journal of Fluids and Structures* **41**, 22–32.

Baker, Bowen, Kanitscheider, Ingmar, Markov, Todor, Wu, Yi, Powell, Glenn, McGrew, Bob & Mordatch, Igor 2019 Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv: 1909.07528* .

Barkley, D. 2006 Linear analysis of the cylinder wake mean flow. *EPL (Europhysics Letters)* **75** (5), 750.

Beneddine, Samir 2017 Characterization of unsteady flow behavior by linear stability analysis. PhD thesis.

Benoit, Christophe, Pron, Stphanie & Landier, Sm 2015 Cassiopee: a CFD pre-and post-processing tool. *Aerospace Science and Technology* **45**, 272–283.

Bergmann, Michel & Cordier, Laurent 2008 Optimal control of the cylinder wake in the laminar regime by trust-region methods and pod reduced-order models. *Journal of Computational Physics* **227** (16), 7813–7840.

Bergmann, Michel, Cordier, Laurent & Brancher, J.-P. 2005 Control of the cylinder wake in the laminar regime by trust-region methods and pod reduced order models. *Proceedings of the 44th IEEE Conference on Decision and Control* pp. 524–529.

Bergmann, Michel, Cordier, Laurent & Brancher, Jean-Pierre 2006 On the generation of a reverse von krmn street for the controlled cylinder wake in the laminar regime. *Physics of Fluids* **18** (2), 028101.

Braza, M., Chassaing, P. H. H. M. & Minh, H. Ha 1986 Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder. *Journal of Fluid Mechanics* **165**, 79–130.

Bright, Ido, Lin, Guang & Kutz, J. Nathan 2013 Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. *Physics of Fluids* **25** (12), 127102.

Brockman, Greg, Cheung, Vicki, Pettersson, Ludwig, Schneider, Jonas, Schulman, John, Tang, Jie & Zaremba, Wojciech 2016 OpenAI Gym. *arXiv preprint arXiv:1606.01540* .

Brunton, Steven L & Noack, Bernd R 2015 Closed-loop turbulence control: progress and challenges. *Applied Mechanics Reviews* **67** (5), 050801–.

Brunton, Steven L., Noack, Bernd R. & Koumoutsakos, Petros 2020 Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics* **52**, 477–508.

Chen, Zhihua & Aubry, Nadine 2005 Active control of cylinder wake. *Communications in nonlinear science and numerical simulation* **10** (2), 205–216.

Cho, Kyunghyun, Van Merrinboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger & Bengio, Yoshua 2014 Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv: 1406.1078* .

Cohen, Kelly, Siegel, Stefan & McLaughlin, Thomas 2006 A heuristic approach to effective sensor placement for modeling of a cylinder wake. *Computers & fluids* **35** (1), 103–120.

Cohen, Kelly, Siegel, Stefan, Seidel, Jrgen, Aradag, Selin & McLaughlin, Thomas 2012 Nonlinear estimation of transient flow field low dimensional states using artificial neural nets. *Expert Systems with Applications* **39** (1), 1264–1272.

Curtiss, Charles Francis & Hirschfelder, Joseph O. 1952 Integration of stiff equations. *Proceedings of the National Academy of Sciences of the United States of America* **38** (3), 235.

Dandois, J., Garnier, E. & Pamart, P.-Y. 2013 NARX modelling of unsteady separation control. *Experiments in fluids* **54** (2), 1445.

Dandois, Julien, Mary, Ivan & Brion, Vincent 2018 Large-eddy simulation of laminar transonic buffet. *Journal of Fluid Mechanics* **850**, 156–178.

DeVries, Levi & Paley, Derek A. 2013 Observability-based optimization for flow sensing and control of an underwater vehicle in a uniform flowfield. *2013 American Control Conference* pp. 1386–1391.

Edwards, Jack R. & Liou, Meng-Sing 1998 Low-diffusion flux-splitting methods for flows at all speeds. *AIAA Journal* **36** (9), 1610–1617.

Foures, Dimitry P. G., Dovetta, Nicolas, Sipp, Denis & Schmid, Peter J. 2014

A data-assimilation method for Reynolds-averaged Navier-Stokes-driven mean flow reconstruction. *Journal of Fluid Mechanics* **759**, 404–431.

Fujisawa, N., Kawaji, Y. & Ikemoto, K. 2001 Feedback control of vortex shedding from a circular cylinder by rotational oscillations. *Journal of Fluids and Structures* **15** (1), 23–37.

Gerhard, Johannes, Pastoor, Mark, King, Rudibert, Noack, Bernd, Dillmann, Andreas, Morzynski, Marek & Tadmor, Gilead 2003 Model-based control of vortex shedding using low-dimensional galerkin models. *33rd AIAA Fluid Dynamics Conference and Exhibit* p. 4262.

Gutmark, E. J. & Grinstein, F. F. 1999 Flow control with noncircular jets. *Annual review of fluid mechanics* **31** (1), 239–272.

Hansen, Nikolaus 2016 The cma evolution strategy: A tutorial. *arXiv preprint arXiv: 1604.00772* .

Hansen, Nikolaus, Mller, Sibylle D. & Koumoutsakos, Petros 2003 Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary computation* **11** (1), 1–18.

He, J.-W., Glowinski, R., Metcalfe, R., Nordlander, A. & Periaux, J. 2000 Active control and drag optimization for flow past a circular cylinder: I. oscillatory cylinder rotation. *Journal of Computational Physics* **163** (1), 83–117.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing & Sun, Jian 2016 Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition* pp. 770–778.

Henderson, Ronald D. 1997 Nonlinear dynamics and pattern formation in turbulent wake transition. *Journal of Fluid Mechanics* **352**, 65–112.

Huh, Minyoung, Agrawal, Pulkit & Efros, Alexei A. 2016 What makes imagenet good for transfer learning? *arXiv preprint arXiv: 1608.08614* .

Hmlinen, Perttu, Babadi, Amin, Ma, Xiaoxiao & Lehtinen, Jaakko 2018 PPO-CMA: proximal policy optimization with covariance matrix adaptation. *arXiv preprint arXiv: 1810.02541* .

Jin, Bo, Illingworth, Simon J. & Sandberg, Richard D. 2019 Feedback control of vortex shedding using a resolvent-based modelling approach. *arXiv preprint arXiv:1909.04865* .

Kaiser, Lukasz, Babaeizadeh, Mohammad, Milos, Piotr, Osinski, Blazej, Campbell, Roy H., Czechowski, Konrad, Erhan, Dumitru, Finn, Chelsea, Kozakowski, Piotr & Levine, Sergey 2019 Model-based reinforcement learning for Atari. *arXiv preprint arXiv:1903.00374* .

Kim, Kihwan, Kerr, Murray, Beskok, Ali & Jayasuriya, Suhada 2006 Frequency-domain based feedback control of flow separation using synthetic jets. *2006 American Control Conference* pp. 6–pp.

Kingma, Diederik P. & Ba, Jimmy 2014 Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Leclerc, Eric, Sagaut, Pierre & Mohammadi, Bijan 2006 On the use of incomplete sensitivities for feedback control of laminar vortex shedding. *Computers & fluids* **35** (10), 1432–1443.

Leclercq, Colin, Demourant, Fabrice, Poussot-Vassal, Charles & Sipp, Denis 2019 Linear iterative method for closed-loop control of quasiperiodic flows. *Journal of Fluid Mechanics* **868**, 26–65.

Louizos, Christos, Welling, Max & Kingma, Diederik P. 2017 Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312* .

Manohar, Krithika, Kutz, J. Nathan & Brunton, Steven L. 2018 Optimal sensor and actuator placement using balanced model reduction. *arXiv preprint arXiv:1812.01574* .

Marquet, Olivier, Sipp, Denis & Jacquin, Laurent 2008 Sensitivity analysis and passive control of cylinder flow. *Journal of Fluid Mechanics* **615**, 221–252.

Min, Chulhong & Choi, Haecheon 1999 Suboptimal feedback control of vortex shedding at low reynolds numbers. *Journal of Fluid Mechanics* **401**, 123–156.

Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A., Veness, Joel, Bellemare, Marc G., Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K. & Ostrovski, Georg 2015 Human-level control through deep reinforcement learning. *Nature* **518** (7540), 529.

Mons, Vincent, Chassaing, J.-C., Gomez, Thomas & Sagaut, Pierre 2016 Reconstruction of unsteady viscous flows using data assimilation schemes. *Journal of Computational Physics* **316**, 255–280.

Mons, Vincent, Chassaing, Jean-Camille & Sagaut, Pierre 2017 Optimal sensor placement for variational data assimilation of unsteady flows past a rotationally oscillating cylinder. *Journal of Fluid Mechanics* **823**, 230–277.

Muddada, Sridhar & Patnaik, B. S. V. 2010 An active flow control strategy for the suppression of vortex structures behind a circular cylinder. *European Journal of Mechanics-B/Fluids* **29** (2), 93–104.

Nair, Aditya G., Taira, Kunihiko, Brunton, Bingni W. & Brunton, Steven L. 2020 Phase-based control of periodic fluid flows. *arXiv preprint arXiv:2004.10561* .

Nishioka, Michio & Sato, Hiroshi 1978 Mechanism of determination of the shedding frequency of vortices behind a cylinder at low reynolds numbers. *Journal of Fluid Mechanics* **89** (1), 49–60.

Nrgrd, Peter Magnus, Ravn, Ole, Poulsen, Niels Kjlstad & Hansen, Lars Kai 2000 *Neural networks for modelling and control of dynamic systems-A practitioner's handbook*. Springer-London.

Oehler, Stephan F. & Illingworth, Simon J. 2018 Sensor and actuator placement trade-offs for a linear model of spatially developing flows. *Journal of Fluid Mechanics* **854**, 34–55.

Protas, B. & Styczek, A. 2002 Optimal rotary control of the cylinder wake in the laminar regime. *Physics of Fluids* **14** (7), 2073–2087.

Protas, B. & Wesfreid, J. E. 2002 Drag force in the open-loop control of the cylinder wake in the laminar regime. *Physics of Fluids* **14** (2), 810–826.

Rabault, Jean, Kuchta, Miroslav, Jensen, Atle, Rglade, Ulysse & Cerardi, Nicolas 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of Fluid Mechanics* **865**, 281–302.

Rabault, Jean & Kuhnle, Alexander 2019 Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Physics of Fluids* **31** (9), 094105.

Rabault, Jean, Ren, Feng, Zhang, Wei, Tang, Hui & Xu, Hui 2020 Deep reinforcement learning in fluid mechanics: a promising method for both active flow control and shape optimization. *arXiv preprint arXiv:2001.02464* .

Rashidi, Saman, Hayatdavoodi, Masoud & Esfahani, Javad Abolfazli 2016 Vortex shedding suppression and wake control: A review. *Ocean Engineering* **126**, 57–80.

Schulman, John, Levine, Sergey, Abbeel, Pieter, Jordan, Michael & Moritz, Philipp 2015*a* Trust region policy optimization. *International conference on machine learning* pp. 1889–1897.

Schulman, John, Moritz, Philipp, Levine, Sergey, Jordan, Michael & Abbeel, Pieter 2015*b* High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* .

Schulman, John, Wolski, Filip, Dhariwal, Prafulla, Radford, Alec & Klimov, Oleg 2017 Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

Seidel, Jrgen, Siegel, Stefan, Fagley, C., Cohen, K. & McLaughlin, T. 2009 Feedback control of a circular cylinder wake. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* **223** (4), 379–392.

Selby, G. V., Lin, J. C. & Howard, F. G. 1992 Control of low-speed turbulent separated flow using jet vortex generators. *Experiments in Fluids* **12** (6), 394–400.

Siegel, Stefan, Cohen, Kelly & McLaughlin, Tom 2003 Feedback control of a circular cylinder wake in experiment and simulation. *33rd AIAA Fluid Dynamics Conference and Exhibit* p. 3569.

Singh, Abhay K. & Hahn, Juergen 2005 Determining optimal sensor locations for state and parameter estimation for stable nonlinear systems. *Industrial & engineering chemistry research* **44** (15), 5645–5659.

Singha, Sintu & Sinhamahapatra, K. P. 2011 Control of vortex shedding from a circular cylinder using imposed transverse magnetic field. *International Journal of Numerical Methods for Heat & Fluid Flow* **21** (1), 32–45.

Sipp, Denis 2012 Open-loop control of cavity oscillations with harmonic forcings. *Journal of Fluid Mechanics* **708**, 439–468.

Sipp, Denis, Marquet, Olivier, Meliga, Philippe & Barbagallo, Alexandre 2010 Dynamics and control of global instabilities in open-flows: a linearized approach. *Applied Mechanics Reviews* **63** (3).

Sipp, Denis & Schmid, Peter J. 2016 Linear closed-loop control of fluid instabilities and noise-induced perturbations: A review of approaches and tools. *Applied Mechanics Reviews* **68** (2).

Sohankar, A., Khodadadi, M. & Rangraz, E. 2015 Control of fluid flow and heat transfer around a square cylinder by uniform suction and blowing at low Reynolds numbers. *Computers & Fluids* **109**, 155–167.

Sutskever, Ilya, Vinyals, Oriol & Le, Quoc V. 2014 Sequence to sequence learning with neural networks. *Advances in neural information processing systems* pp. 3104–3112.

Tang, Hongwei, Rabault, Jean, Kuhnle, Alexander, Wang, Yan & Wang, Tongguang 2020 Robust active flow control over a range of reynolds numbers using an artificial neural network trained through deep reinforcement learning. *Physics of Fluids* **32** (5), 053605.

Verma, Siddhartha, Papadimitriou, Costas, Lthen, Nora, Arampatzis, Georgios & Koumoutsakos, Petros 2020 Optimal sensor placement for artificial swimmers. *Journal of Fluid Mechanics* **884**.

Williams, Ronald J. 1992 Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* **8** (3-4), 229–256.

Williamson, Charles H. K. 1996 Vortex dynamics in the cylinder wake. *Annual review of fluid mechanics* **28** (1), 477–539.

Zielinska, B. J. A., Goujon-Durand, S., Dusek, J. & Wesfreid, J. E. 1997 Strongly nonlinear effect in unstable wakes. *Physical review letters* **79** (20), 3893.