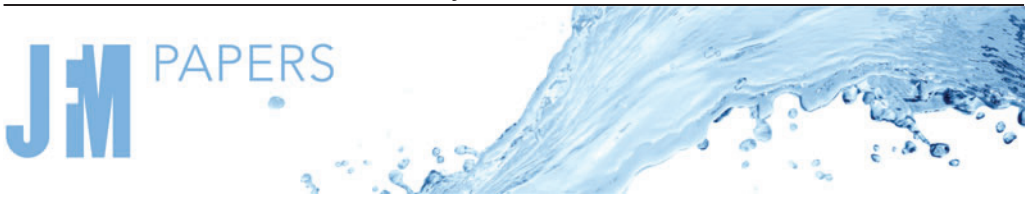
J. Fluid Mech. (2024), *vol*. 981, A17, doi:10.1017/jfm.2024.69



Active flow control for bluff body drag reduction using reinforcement learning with partial measurements

Chengwei Xia[1], **Junjie Zhang**[1], **Eric C. Kerrigan**[1,2] **and Georgios Rigas**[1,†]

[1]Department of Aeronautics, Imperial College London, London SW7 2AZ, UK

[2]Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK

(Received 24 July 2023; revised 7 December 2023; accepted 7 January 2024)

Active flow control for drag reduction with reinforcement learning (RL) is performed in the wake of a two-dimensional square bluff body at laminar regimes with vortex shedding. Controllers parametrised by neural networks are trained to drive two blowing and suction jets that manipulate the unsteady flow. The RL with full observability (sensors in the wake) discovers successfully a control policy that reduces the drag by suppressing the vortex shedding in the wake. However, a non-negligible performance degradation ($\sim$50 % less drag reduction) is observed when the controller is trained with partial measurements (sensors on the body). To mitigate this effect, we propose an energy-efficient, dynamic, maximum entropy RL control scheme. First, an energy-efficiency-based reward function is proposed to optimise the energy consumption of the controller while maximising drag reduction. Second, the controller is trained with an augmented state consisting of both current and past measurements and actions, which can be formulated as a nonlinear autoregressive exogenous model, to alleviate the partial observability problem. Third, maximum entropy RL algorithms (soft actor critic and truncated quantile critics) that promote exploration and exploitation in a sample-efficient way are used, and discover near-optimal policies in the challenging case of partial measurements. Stabilisation of the vortex shedding is achieved in the near wake using only surface pressure measurements on the rear of the body, resulting in drag reduction similar to that in the case with wake sensors. The proposed approach opens new avenues for dynamic flow control using partial measurements for realistic configurations.

Key words: wakes, machine learning, drag reduction

† Email address for correspondence: g.rigas@imperial.ac.uk

1. Introduction

Up to 50 % of total road vehicle energy consumption is due to aerodynamic drag (Sudin *et al.* 2014). In order to improve vehicle aerodynamics, flow control approaches have been applied targeting the wake pressure drag, which is the dominant source of drag. Passive flow control has been applied (Choi, Lee & Park 2014) through geometry/surface modifications, e.g. boat tails (Lanser, Ross & Kaufman 1991) and vortex generators (Lin 2002). However, passive control designs do not adapt to environmental changes (disturbances, operating regimes), leading to sub-optimal performance under variable operating conditions. Active open-loop techniques, where predetermined signals drive actuators, typically are energy inefficient since they target mean flow modifications. Actuators employed typically are synthetic jets (Glezer & Amitay 2002), movable flaps (Beaudoin *et al.* 2006; Brackston *et al.* 2016) and plasma actuators (Corke, Enloe & Wilkinson 2010), among others. Since the flow behind vehicles is unsteady and subject to environmental disturbances and uncertainty, active feedback control is required to achieve optimal performance. However, two major challenges arise in feedback control design, which we aim to tackle in this study: the flow dynamics is (i) governed by the infinite-dimensional, nonlinear and non-local Navier–Stokes equations (Brunton & Noack 2015); and (ii) partially observable in realistic applications due to sensor limitations. This study aims to tackle these challenges, focusing particularly on the potential of model-free control for a partially observable laminar flow, characterised by bluff body vortex shedding, as a preliminary step towards more complex flows and applications.

1.1. *Model-based active flow control*

Model-based feedback control design requires a tractable model for the dynamics of the flow, usually obtained by data-driven or operator-driven techniques. Such methods have been applied successfully to control benchmark two-dimensional (2-D) bluff body wakes, obtaining improved aerodynamic performance, e.g. vortex shedding suppression and drag reduction. For example, Gerhard *et al.* (2003) controlled the circular cylinder wake at low Reynolds numbers based on a low-dimensional model obtained from the Galerkin projection of Karhunen–Loeve modes on the governing Navier–Stokes equations. Protas (2004) applied linear quadratic Gaussian control to stabilise vortex shedding based on a Föppl point vortex model. Illingworth (2016) applied the eigensystem realization algorithm as a system identification technique to obtain a reduced-order model of the flow, and used robust control methods to obtain feedback control laws. Jin, Illingworth & Sandberg (2020) employed resolvent analysis to obtain a low-order input–output model from the Navier–Stokes equations, based on which feedback control was applied to suppress vortex shedding.

Model-based flow control has also been applied at high Reynolds numbers to control dominant coherent structures (persisting spatio-temporal symmetry breaking modes) that contribute to drag, including unsteady vortex shedding (Pastoor *et al.* 2008; Dahan, Morgans & Lardeau 2012; Dalla Longa, Morgans & Dahan 2017; Brackston, Wynn & Morrison 2018) and steady spatial symmetry breaking modes (Brackston *et al.* 2016; Li *et al.* 2016). Typically, for inhomogeneous flows in all three spatial dimensions, low-order models fail to capture the intractable and complex turbulent dynamics, leading inevitably to sub-optimal control performance when used in control synthesis.

1.2. *Model-free active flow control by reinforcement learning*

Model-free data-driven control methods bypass the above limitations by using input–output data from the dynamical system (environment) to learn the optimal control law (policy) directly without exploiting information from a mathematical model of the underlying process (Hou & Xu 2009).

Model-free reinforcement learning (RL) has been used successfully for controlling complex systems, for which obtaining accurate and tractable models can be challenging. The RL learns a control policy based on observed states, and generates control actions that maximise a reward by exploring and exploiting state–action pairs. The system dynamics governing the evolution of the states for a specific action (environment) is assumed to be a Markov decision process (MDP). The policy is parametrised by artificial neural networks as a universal function approximator that can be optimised to an arbitrary control function with any order of complexity. The RL with neural networks can also be interpreted as parametrised dynamic programming with the feature of universal function approximation (Bertsekas 2019). Therefore, RL requires only input–output data from complex systems in order to discover control policies using model-free optimisation.

Effectively, RL can learn to control complex systems in various types of tasks, such as robotics (Kober, Bagnell & Peters 2013) and autonomous driving (Kiran *et al.* 2021). In the context of chaotic dynamics related to fluid mechanics, Bucci *et al.* (2019) and Zeng & Graham (2021) applied RL to control the chaotic Kuramoto–Sivashinsky system. In the context of flow control for drag reduction, Rabault *et al.* (2019) and Rabault & Kuhnle (2019) used RL control for the first time in 2-D bluff body simulations at a laminar regime. The RL algorithm discovered a policy that, using pressure sensors in the wake and near the body, drives blowing and suction actuators on the circular cylinder to decrease the mean drag and wake unsteadiness. Tang *et al.* (2020) trained RL-controlled synthetic jets in the flow past a 2-D cylinder at several Reynolds numbers (100, 200, 300, 400), and achieved drag reduction in a range of Reynolds number from 60 to 400, showing the generalisation ability of RL active flow control. Paris, Beneddine & Dandois (2021) applied the 'S-PPO-CMA' RL algorithm to control the wake behind a 2-D cylinder and optimise the sensor locations in the near wake. Li & Zhang (2022) augmented and guided RL with global linear stability and sensitivity analyses in order to control the confined cylinder wake. They showed that if the sensors cover the wavemaker region, then the RL is robust and successfully stabilises the vortex shedding. Paris, Beneddine & Dandois (2023) proposed an RL methodology to optimise actuator placement in a laminar 2-D flow around an aerofoil, addressing the trade-off between performance and the number of actuators. Xu & Zhang (2023) used RL to suppress instabilities in both the Kuramoto–Sivashinsky system and 2-D boundary layers, showing the effectiveness and robustness of RL control. Pino *et al.* (2023) compared RL and genetic programming algorithms to global optimisation techniques for various cases, including the viscous Burger's equation and vortex shedding behind a 2-D cylinder. Chen *et al.* (2023) applied RL in the flow control of vortex-induced vibration of a 2-D square bluff body with various actuator layouts. The vibration and drag of the body were both reduced and mitigated effectively by RL policies.

Recently, RL has been used to control complex fluid systems, such as flows in turbulent regimes, in both simulations and experiments, addressing the potential of RL flow control in realistic applications. Fan *et al.* (2020) extended RL flow control to a turbulent regime in experiments at Reynolds numbers of $O(10^5)$, achieving effective drag reduction by controlling the rotation speed of two cylinders downstream of a bluff body. The RL discovered successfully the global optimal open-loop control strategy that was found

previously from a laborious non-automated, systematic grid search. The experimental results were verified further by high-fidelity numerical simulations. Ren, Rabault & Tang (2021) examined RL-controlled synthetic jets in a weakly turbulent regime, demonstrating effective control at Reynolds number 1000. This flow control problem of drag reduction of a 2-D cylinder flow using synthetic jets was extended to Reynolds number 2000 by Varela *et al.* (2022). In their work, RL discovered a strategy of separation delay via high-frequency perturbations to achieve drag reduction. Sonoda *et al.* (2023) and Guastoni *et al.* (2023) applied RL control in numerical simulations of turbulent channel flow, and showed that RL control can outperform opposition control in this complex flow control task.

Some RL techniques have been applied also to various flow control problems with different geometries, such as flow past a 2-D cylinder (Rabault *et al.* 2019), vortex-induced vibration of a 2-D square bluff body (Chen *et al.* 2023), and a 2-D boundary layer (Xu & Zhang 2023). However, model-free RL control techniques also have several drawbacks compared to model-based control. For example, it is usually challenging to tune the various RL hyperparameters. Also, typically model-free RL requires large amounts of training data through interactions with the environment, which makes RL expensive and infeasible for certain applications. Further information about RL and its applications in fluid mechanics can be found in the reviews of Garnier *et al.* (2021) and Vignon, Rabault & Vinuesa (2023).

1.3. *Maximum entropy RL*

In RL algorithms, two major branches have been developed: 'on-policy' learning and 'off-policy' learning. The RL algorithms can also be classified into value-based, policy-based and actor–critic methods (Sutton & Barto 2018). The actor–critic architecture combines advantages from both value-based and policy-based methods, so the state-of-the-art algorithms use mainly actor–critic architecture.

The state-of-the-art on-policy algorithms include trust region policy optimisation (Schulman *et al.* 2015), asynchronous advantage actor–critic (Mnih *et al.* 2016) and proximal policy optimisation (Schulman *et al.* 2017). On-policy algorithms require fewer computational resources than off-policy algorithms, but they are demanding in terms of available data (interactions with the environment). They use the same policy to obtain experience in the environment and update with policy gradient, which introduces a high self-relevant experience that may restrict convergence to a local minimum and limit exploration. As the amount of data needed for training grows with the complexity of applications, on-policy algorithms usually require a long training time for collecting data and converging.

By contrast, off-policy algorithms usually have both behaviour and target policies to facilitate exploration while retaining exploitation. The behaviour policy usually employs stochastic behaviour to interact with an environment and collect experience, which is used to update the target policy. There are many off-policy algorithms emerging in the past decade, such as deterministic policy gradient (Silver *et al.* 2014), deep deterministic policy gradient (DDPG; Lillicrap *et al.* 2015), actor–critic with experience replay (Wang *et al.* 2016), twin delayed deep deterministic policy gradient (Fujimoto, Hoof & Meger 2018), soft actor–critic (SAC; Haarnoja *et al.* 2018*a*,*b*) and truncated quantile critic (TQC; Kuznetsov *et al.* 2020). Due to the behaviour-target framework, off-policy algorithms are able to exploit past information from a replay buffer to further increase sample efficiency. This 'experience replay' suits a value-function-based method (Mnih *et al.* 2015), instead

of calculating the policy gradient directly. Therefore, most of the off-policy algorithms implement an actor–critic architecture, e.g. SAC.

One of the challenges of off-policy algorithms is the brittleness in terms of convergence. Sutton, Szepesvári & Maei (2008) and Sutton *et al.* (2009) tackled the instability issue of off-policy learning with linear approximations. They used a Bellman-error-based cost function together with the stochastic gradient descent to ensure the convergence of learning. Maei *et al.* (2009) extended this method further to nonlinear function approximation using a modified temporal difference algorithm. However, some algorithms nowadays still experience the problem of brittleness when using improper hyperparameters. Adapting these algorithms for control in various environments is sometimes challenging, as the learning stability is sensitive to their hyperparameters, such as DDPG (Duan *et al.* 2016; Henderson *et al.* 2018).

To increase sample efficiency and learning stability, off-policy algorithms were developed within a maximum entropy framework (Ziebart *et al.* 2008; Haarnoja *et al.* 2017), known as 'maximum entropy reinforcement learning'. Maximum entropy RL solves an optimisation problem by maximising the cumulative reward augmented with an entropy term. In this context, the concept of entropy was introduced first by Shannon (1948) in information theory. The entropy quantifies the uncertainty of a data source, which is extended to the uncertainty of the outputs of stochastic neural networks in the RL framework. During the training phase, the maximum entropy RL maximises rewards and entropy simultaneously to improve control robustness (Ziebart 2010) and increase exploration via diverse behaviours (Haarnoja *et al.* 2017). Further details about maximum entropy RL and two particular algorithms used in the present work (SAC and TQC) are introduced in § 2.2.

1.4. *Partial measurements and POMDP*

In most RL flow control applications, RL controllers have been assumed to have full-state information (the term 'state' is in the context of control theory) or a sensor layout without any limitations on the sensor locations. In this study, it is denoted as 'full measurement' (FM) when measurements contain full-state information. In practical applications, typically measurements are obtained on the surface of the body (e.g. pressure taps), and only partial-state information is available due to the missing downstream evolution of the system dynamics. This is denoted as 'partial measurement' (PM), comparatively. Such PM can lead to control performance degradation compared to FM because the sensors are restricted from observing enough information from the flow field. In the control of vortex shedding, full stabilisation can be achieved by placing sensors within the wavemaker region of bluff bodies, which is located approximately at the end of the recirculation region. In this case, full-state information regarding the vortex shedding is available to sensors. Placing sensors far from the recirculation region, for example, on the rear surface of the bluff body (denoted as PM in this work), introduces a convection delay of vortex shedding sensing and partial observation of the state of the system.

In the language of RL, control with PM can be described as a partially observable Markov decision process (POMDP; Cassandra 1998) instead of an MDP. In POMDP problems, the best stationary policy can be arbitrarily worse than the optimal policy in the underlying MDP (Singh, Jaakkola & Jordan 1994). In order to improve the performance of RL with POMDP, additional steps are required to reduce the POMDP problem to an MDP problem. This can be done trivially by using an augmented state known as a 'sufficient statistic' (Bertsekas 2012), i.e. augmenting the state vector with past measurements and

actions (Bucci *et al.* 2019; Wang *et al.* 2023), or recurrent neural networks, such as long short-term memory (LSTM; Verma, Novati & Koumoutsakos 2018). Theoretically, LSTM networks and augmented state approaches can yield comparable performance in partially observable problems (see Cobbe *et al.* (2020), supplementary material). Practically, the augmented state methodology provides notable benefits, including reduced training complexity and ease in parameter tuning, provided that the control state dynamics are tractable and short-term correlated.

In the specific case for which flow field information is available, a POMDP can also be reduced to an MDP by flow reconstruction techniques based on supervised learning. For instance, Bright, Lin & Kutz (2013) estimates the full state based on a library containing the reduced-order information from the full flow field. However, there might be difficulties in constructing such a library as the entire flow field might not be available in practical applications.

1.5. *Contribution of the present work*

The present work uses RL to discover control strategies of partially observable fluid flow environments without access to the full flow field/state measurements. Fluid flow systems typically exhibit more complex sampling in higher-dimensional observation space compared to other physical systems, necessitating a robust exploration strategy and rapid convergence in the optimisation process. To address these challenges, we employ off-policy maximum entropy RL algorithms (SAC and TQC) that identify efficiently nearly optimal policies in the large action space inherent to fluid flow systems, especially for cases with PM and observability.

We aim to achieve two objectives related to RL flow control for bluff body drag reduction problems. First, we aim to improve the RL control performance in a PM environment by reducing a POMDP problem to an MDP problem. More details about this method are introduced in § 2.4. Second, we present investigations on different reward functions and key hyperparameters to develop an approach that can be adapted to a broader range of flow control applications. We demonstrate the proposed framework and its capability to discover nearly optimal feedback control strategies in the benchmark laminar flow of a square 2-D bluff body with fixed separation at the trailing edge, using sensors only on the downstream surface of the body.

The paper is structured as follows. In § 2, the RL framework is presented, which consists of the SAC and TQC optimisation algorithms interacting with the flow simulation environment. A hyperparameter-free reward function is proposed to optimise the energy efficiency of the dynamically controlled system. Exploiting past action state information converts the POMDP problem in a PM environment to an MDP, enabling the discovery of nearly optimal policies. Results are presented and discussed in § 3. The convergence study of RL is first introduced. The degradation of RL control performance in PM environments (POMDP) is presented, and the improvement is addressed by exploiting a sequence of past action measurement information. At the end of this section, we compare the results from TQC with SAC, addressing the advantages of using TQC as an improved version of SAC. In § 4, we provide conclusions for the current research and discuss future research directions.

2. Methodology

We demonstrate the RL drag reduction framework on the flow past a 2-D square bluff body at laminar regimes characterised by 2-D vortex shedding. We study the canonical

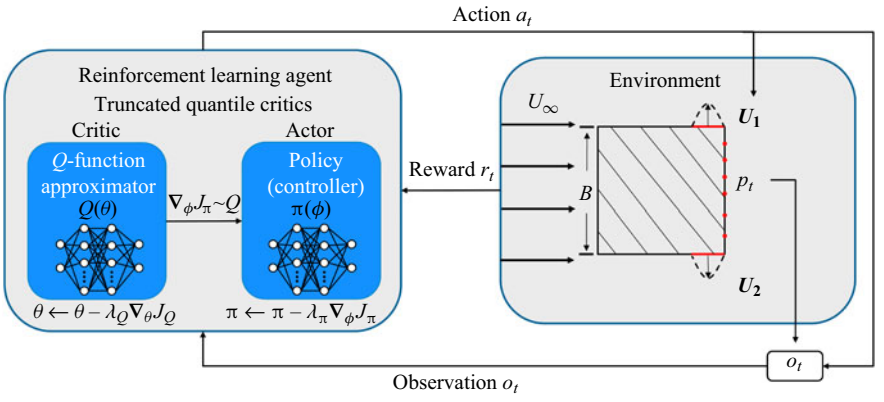


Figure 1. The RL framework. The RL agent and flow environment, and the interaction between them, are demonstrated. The PM case is shown, where sensors are located on the downstream surface of the square bluff body: 64 sensors are placed by default, and the red dots show only a demonstration with a reduced number of sensors. Two jets located upstream of the rear separation points are trained to control the unsteady wake dynamics (vortex shedding).

flow behind a square bluff body due to the fixed separation of the boundary layer at the rear surface, which is relevant to road vehicle aerodynamics. Control is applied by two jet actuators at the rear edge of the body before the fixed separation, and partial- or full-state observations are obtained from pressure sensors on the downstream surface or near-wake region, respectively. The RL agent handles the optimisation, control and interaction with the flow simulation environment, as shown in figure 1. The instantaneous signals a_t, o_t and r_t denote actions, observations and rewards at time step t.

Details of the flow environment are provided in § 2.1. The SAC and TQC RL algorithms used in this work are introduced in § 2.2. The reward functions based on optimal energy efficiency are presented in § 2.3. The method to convert a POMDP to an MDP by designing a dynamic feedback controller for achieving nearly optimal RL control performance is discussed in § 2.4.

2.1. *Flow environment*

The environment is 2-D direct numerical simulations (DNS) of the flow past a square bluff body of height B. The velocity profile at the inflow of the computational domain is uniform with freestream velocity U_∞. Length quantities are non-dimensionalised with the bluff body height B, and velocity quantities are non-dimensionalised with the freestream velocity U_∞. Consequently, time is non-dimensionalised with B/U_∞. The Reynolds number, defined as $Re = U_\infty B/\nu$, is 100. The computational domain is rectangular with boundaries at $(-20.5, 26.5)$ in the streamwise x direction and $(-12.5, 12.5)$ in the transverse y direction. The centre of the square bluff body is at $(x, y) = (0, 0)$. The flow velocity is denoted as $\boldsymbol{u} = (u, v)$, where u is the velocity component in the x direction, and v is the component in the y direction.

The DNS flow environment is simulated using FEniCS and the Dolfin library (Logg, Wells & Hake 2012), based on the implementation of Rabault *et al.* (2019) and Rabault & Kuhnle (2019). The incompressible unsteady Navier–Stokes equations are solved using a finite element method and the incremental pressure correction scheme (Goda 1979). The DNS time step is $\mathrm{d}t = 0.004$. More simulation details are presented in Appendix A, including the mesh and boundary conditions.

Two blowing and suction jet actuators are placed on the top and bottom surfaces of the bluff body before separation. The velocity profile $\boldsymbol{U_j}$ of the two jets ($j = 1, 2$, where 1 indicates the top jet, and 2 indicates the bottom jet) is defined as

$$\boldsymbol{U_j} = \left(0, \frac{3Q_j}{2w}\left[1 - \left(\frac{2x_j - L + w}{w}\right)^2\right]\right), \tag{2.1}$$

where Q_j is the mass flow rate of the jet j, and $L = B$ is the streamwise length of the body. The width of the jet actuator is $w = 0.1$, and the jets are located at $x_j \in [L/2 - w, L/2]$, $y_j = \pm B/2$. A zero mass flow rate condition of the two jets enforces momentum conservation as

$$Q_1 + Q_2 = 0. \tag{2.2}$$

The mass flow rate of the jets is also constrained as $|Q_j| \leqslant 0.1$ to avoid excessive actuation.

In PM environments, N vertically equispaced pressure sensors are placed on the downstream surface of the bluff body, the coordinates of which are given by

$$\boldsymbol{P}_{surf,k} = \left(\frac{B}{2}, \frac{-B}{2} + k\,\frac{B}{N+1}\right), \tag{2.3}$$

where $k = 1, 2, \ldots, N$, and $N = 64$ unless specified otherwise. In FM environments, 64 pressure sensors are placed in the wake region, with a refined bias close to the body. The locations of sensors in the wake are defined with sets $\boldsymbol{x}_s = [0.25, 0.5, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0]$ and $\boldsymbol{y}_s = [-1.5, -1.0, -0.5, -0.25, 0.25, 0.5, 1.0, 1.5]$, following the formula

$$\boldsymbol{P}_{wake,i,j} = \left(\frac{B}{2} + x_{s,i}, y_{s,j}\right), \tag{2.4}$$

where $i = 1, 2, \ldots, 8$ and $j = 1, 2, \ldots, 8$.

The bluff body drag coefficient C_D is defined as

$$C_D = \frac{F_D}{\frac{1}{2}\rho_\infty U_\infty^2 B}, \tag{2.5}$$

and the lift coefficient C_L as

$$C_L = \frac{F_L}{\frac{1}{2}\rho_\infty U_\infty^2 B}, \tag{2.6}$$

where F_D and F_L are the drag and lift forces, defined as the surface integrals of the pressure and viscous forces on the bluff body with respect to the x and y coordinates, respectively.

2.2. *Maximum entropy reinforcement learning of an MDP*

Reinforcement learning can be defined as policy search in an MDP, with a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$, where $\mathcal{S}$ is a set of states, and $\mathcal{A}$ is a set of actions. Here, $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ is a state transition function that contains the probability from current state s_t and action a_t to the next state, s_{t+1}, and $\mathcal{R}(s, a)$ is a reward function (cost function) to be maximised. The RL agent collects data as states $s_t \in \mathcal{S}$ from the environment, and a policy $\pi(a_t \mid s_t)$ executes actions $a_t \in \mathcal{A}$ to drive the environment to the next state, s_{t+1}.

A state is considered to have the Markov property if the state at time t retains all the necessary information to determine the future dynamics at $t+1$, without any information from the past (Sutton & Barto 2018). This property can be presented as

$$\mathcal{P}\{r_{t+1}, s_{t+1} \mid s_t, a_t\} \equiv \mathcal{P}\{r_{t+1}, s_{t+1} \mid s_0, a_0, r_1, \ldots, s_{t-1}, a_{t-1}, r_t, s_t, a_t\}. \tag{2.7}$$

In the present flow control application, the control task can be regarded as an MDP if observations o_t contain full-state information, i.e. $o_t = s_t$, and satisfy (2.7).

We use SAC and TQC as two maximum entropy RL algorithms in the present work; TQC is used by default since it is regarded as an improved version of SAC. Generally, the maximum entropy RL maximises

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}[r_t(s_t, a_t) + \alpha\, \mathcal{H}(\pi(\,\cdot \mid s_t))], \tag{2.8}$$

where r_t is the reward (reward functions given in § 2.3), and α is an entropy coefficient (known as 'temperature') that controls the stochasticity (exploration) of the policy. For $\alpha = 0$, the standard maximum reward optimisation in conventional RL is recovered. The probability distribution (Gaussian by default) of a stochastic policy is denoted by $\pi(\,\cdot \mid s_t)$. The entropy of $\pi(\,\cdot \mid s_t)$ is by definition (Shannon 1948)

$$\mathcal{H}(\pi(\,\cdot \mid s_t)) = \mathbb{E}[-\log \pi(\,\cdot \mid s_t)] = -\int_{\hat{a}_t} \pi(\hat{a}_t \mid s_t) \log \pi(\hat{a}_t \mid s_t)\, \mathrm{d}\hat{a}_t, \tag{2.9}$$

where the term $-\log \pi$ quantifies the uncertainty contained in the probability distribution, and $\hat{a}_t$ is a distribution variable of the action a_t. Therefore, by calculating the expectation of $-\log \pi$, the entropy increases when the policy has more uncertainties, i.e. the variance of $\pi(\hat{a}_t \mid s_t)$ increases.

We develop SAC based on soft policy iteration (Haarnoja *et al.* 2018*b*), which uses a soft Q-function to evaluate the value of a policy, and optimises the policy based on its value. The soft Q-function is calculated by applying a Bellman backup operator $\mathcal{T}^\pi$ as

$$\mathcal{T}^\pi Q(s_t, a_t) \triangleq r_t(s_t, a_t) + \gamma\, \mathbb{E}_{s_{t+1} \sim \mathcal{P}}[V(s_{t+1})], \tag{2.10}$$

where γ is a discount factor (here $\gamma = 0.99$), and $V(s_{t+1})$ satisfies

$$V(s_t) = \mathbb{E}_{a_t \sim \pi}[Q(s_t, a_t) - \log \pi(a_t \mid s_t)]. \tag{2.11}$$

The target soft Q-function can be obtained by repeating $Q = \mathcal{T}^\pi Q$, and the proof of convergence can be referred to as soft policy evaluation (Lemma 1 in Haarnoja *et al.* 2018*b*). With soft Q-function rendering values for the policy, the policy optimisation is given as soft policy improvement (Lemma 2 in Haarnoja *et al.* 2018*b*).

In SAC, a stochastic soft Q-function $Q_\theta(s_t, a_t)$ and a policy $\pi_\phi(a_t \mid s_t)$ are parametrised by artificial neural networks θ (critic) and ϕ (actor), respectively. During training, $Q_\theta(s_t, a_t)$ and $\pi_\phi(a_t \mid s_t)$ are optimised with stochastic gradients $\nabla_\theta J_Q(\theta)$ and $\nabla_\phi J_\pi(\phi)$ designed corresponding to soft policy evaluation and soft policy improvement, respectively (see (6) and (10) in Haarnoja *et al.* 2018*b*). With these gradients, SAC updates the critic and actor networks by

$$\theta \leftarrow \theta - \lambda_Q \nabla_\theta J_Q(\theta), \tag{2.12}$$

$$\phi \leftarrow \phi - \lambda_\pi \nabla_\phi J_\pi(\phi), \tag{2.13}$$

where λ_Q and λ_π are the learning rates of Q-function and policy, respectively. Typically, two Q-functions are trained independently, then the minimum of the Q-functions is

brought into the calculation of stochastic gradient and policy gradient. This method is also used in our work to increase the stability and speed of training. Also, SAC supports automatic adjustment of temperature α by optimisation:

$$\alpha^* = \arg\min_{\alpha} \mathbb{E}_{a_t \sim \pi^*}[-\alpha \log \pi^*(a_t \mid s_t; \alpha) - \alpha \bar{\mathcal{H}}]. \tag{2.14}$$

This adjustment transforms a hyperparameter tuning challenge into a trivial optimisation problem (Haarnoja *et al.* 2018*b*).

We can regard TQC (Kuznetsov *et al.* 2020) as an improved version of SAC as it alleviates the overestimation bias of the Q-function on the basic algorithm of SAC. Also, TQC adapts the idea of distributional RL with quantile regression (Dabney *et al.* 2018) to format the return function $R(s, a) := \sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)$ into a distributional representation with Dirac delta functions as

$$R_\psi(s, a) := \frac{1}{M} \sum_{m=1}^{M} \delta(z_\psi^m(s, a)), \tag{2.15}$$

where $R(s, a)$ is parametrised by ψ, and $R_\psi(s, a)$ is converted into a summation of M 'atoms' as $z_\psi^m(s, a)$. Here, only one approximation of $R(s, a)$ is used for demonstration. Then only the k smallest atoms of $z_\psi^m(s, a)$ are preserved as a truncation to obtain truncated atoms

$$y_i(s, a) := r(s, a) + \gamma[z_\psi^i(s', a') - \alpha \log \pi_\phi(a' \mid s')], \quad i = 1, \ldots, k, \tag{2.16}$$

where $s' \sim \mathcal{P}(\cdot \mid s, a)$ and $a' \sim \pi(\cdot \mid s')$. The truncated atoms form a target distribution as

$$Y(s, a) := \frac{1}{k} \sum_{i=1}^{k} \delta(y_i(s, a)), \tag{2.17}$$

and the algorithm minimises the 1-Wasserstein distance between the original distribution $R_\psi(s, a)$ and the target distribution $Y(s, a)$ to obtain a truncated quantile critic function. Further details, such as the design of loss functions and the pseudocode of TQC, can be found in Kuznetsov *et al.* (2020).

In this work, SAC and TQC are implemented based on Stable-Baselines3 and Stable-Baselines3-Contrib (Raffin *et al.* 2021). The RL interaction runs on a longer time step $t_a = 0.5$ compared to the numerical time step dt. This means that RL-related data o_t, a_t and r_t are sampled every t_a time interval. With a different numerical step and an RL step, control actuation c_{n_s} for every numerical step should be distinguished from action a_t in RL. There are $t_a/\mathrm{d}t = 125$ numerical steps between two RL steps, and control actuation is applied based on a first-order hold function as

$$c_{n_s} = a_{t-1} + (a_t - a_{t-1}) \frac{n_s \, \mathrm{d}t}{t_a}, \tag{2.18}$$

where n_s denotes the number of numerical steps after generating the current action a_t and before the next action a_{t+1} is generated. Equation (2.18) smooths the control actuation with linear interpolation to avoid numerical instability. Unless specified, the neural network configuration is set as three layers of 512 neurons for both actor and critic. The entropy coefficient in (2.8) is initialised to 0.01 and tuned automatically based on (2.14) during training. See table 3 in Appendix B for more details of RL hyperparameters.

2.3. *Reward design for optimal energy efficiency*

We propose a hyperparameter-free reward function based on net power saving to discover energy-efficient flow control policies, calculated as the difference between the power saved from drag reduction ΔP_D and the power consumed from actuation P_{act}. Then the power reward ('PowerR') at the RL control frequency is

$$r_t = \underbrace{\Delta P_D}_{power\ saved} - \underbrace{P_{act}}_{power\ spent}. \tag{2.19}$$

The power saved from drag reduction is given by

$$\Delta P_D = P_{D0} - P_{Dt} = (\langle F_{D0}\rangle_T - \langle F_{Dt}\rangle_a)U_\infty, \tag{2.20}$$

where P_{D0} is the time-averaged baseline drag power without control, $\langle F_{D0}\rangle_T$ is the time-averaged baseline drag over a sufficiently long period, and P_{Dt} denotes the time-averaged drag power calculated from the time-averaged drag $\langle F_{Dt}\rangle_a$ during one RL step t_a. Specifically, $\langle\,\cdot\,\rangle_a$ quantities are calculated at each RL step using 125 DNS samples. The jet power consumption of actuation P_{act} (Barros *et al.* 2016) is defined as

$$P_{act} = \sum_{j=1}^{2} |\rho_\infty \langle U_j\rangle_a^3 S_j| = \sum_{j=1}^{2} \left| \frac{\langle a_t\rangle_a^3}{\rho_\infty^2 S_j^2} \right|, \tag{2.21}$$

where $\langle U_j\rangle_a$ is the average jet velocity, and S_j denotes the area of one jet.

The reward function given by (2.19) quantifies the control efficiency of a controller directly. Thus it guarantees the learning of a control strategy that simultaneously maximises the drag reduction and minimises the required control actuation. Additionally, this energy-based reward function avoids the effort of hyperparameter tuning.

All the cases in this work use the power-based reward function defined in (2.19) unless specified otherwise. For comparison, a reward function based on drag and lift coefficient ('ForceR') is also implemented, as suggested by Rabault *et al.* (2019) with a pre-tuned hyperparameter $\epsilon = 0.2$, as

$$r_t^a = C_{D0} - \langle C_{Dt}\rangle_a - \epsilon\,|\langle C_{Lt}\rangle_a|, \tag{2.22}$$

where C_{D0} and $\langle C_{Dt}\rangle_a$ are calculated from a constant baseline drag and RL-step-averaged drag and lift. The RL-step-averaged lift $|\langle C_{Lt}\rangle_a|$ is used to penalise the amplitude of actuation on both sides of the body, avoiding excessive lift force (i.e. the lateral deflection of the wake reduces the drag but increases the side force), and indirectly penalising control actuation and the discovery of unrealistic control strategies. Here, ϵ is a hyperparameter designed to balance the penalty on drag and lift force.

The instantaneous versions of these two reward functions are also investigated for practical implementation purposes (both experimentally and numerically) because they can significantly reduce memory used during computation and also support a lower sampling rate. These instantaneous reward functions are computed only from observations at each RL step. In comparison, the reward functions above take into account the time history between two RL steps, while the instantaneous version of the power reward

('PowerInsR') is defined as

$$r_{t,ins} = \Delta P_{D,ins} - P_{act,ins}, \tag{2.23}$$

where $\Delta P_{D,ins}$ is given by

$$\Delta P_{D,ins} = (\langle F_{D0} \rangle_T - F_{Dt}) U_\infty, \tag{2.24}$$

and $P_{act,ins}$ is defined as

$$P_{act,ins} = \sum_{j=1}^{2} |\rho_\infty \overline{U_j}^3 S_j| = \sum_{j=1}^{2} \left| \frac{a_t^3}{\rho_\infty^2 S_j^2} \right|. \tag{2.25}$$

Notice that the definition of reward in (2.23)–(2.25) is similar to (2.19)–(2.25), and the only difference is that the average operator $\langle \cdot \rangle_a$ is removed. Similarly, the instantaneous version of the force-based reward function ('ForceInsR') is defined as

$$r_{t,ins}^a = C_{D0} - C_{Dt} - \epsilon |C_{Lt}|. \tag{2.26}$$

In § 3.5, we present results on the study of different reward functions, and compare the RL performance.

2.4. *The POMDP and dynamic feedback controllers*

In practical applications, the Markov property (2.7) is often not valid due to noise, broken sensors, partial state information and delays. This means that the observations available to the RL agent do not provide full or true state information, i.e. $o_t \neq s_t$, while in MDP, $o_t = s_t$. Then RL can be generalised as a POMDP defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{Y}, O)$, where $\mathcal{Y}$ is a finite set of observations o_t, and O is an observation function that relates observations to underlying states.

With only PM available in the flow environments (sensors on the downstream surface of the body instead of in the wake), the spatial information is missing along the streamwise direction. Takens' embedding theorem (Takens 1981) states that the underlying dynamics of a high-dimensional dynamical system can be reconstructed from low-dimensional measurements with their time history. Therefore, past measurements can be incorporated into a sufficient statistic. Furthermore, convective delays may be introduced in the state observation since the sensors are not located in the wavemaker region of the flow. According to Altman & Nain (1992), past actions are also required in the state of a delayed problem to reduce it into an undelayed problem. This is because implicitly, a typical delayed MDP subverts the Markov property, as the past measurements and actions encapsulate only partial information.

Therefore, combining the ideas of augmenting past measurements and past actions, we form a sufficient statistic (Bertsekas 2012) for reducing the POMDP problem to an MDP, defined as

$$\boldsymbol{I}_k = [p_0, \ldots, p_k, a_0, \ldots, a_{k-1}], \tag{2.27}$$

which consists of the time history of pressure measurements $p_0, \ldots, p_k$ and control actions $a_0, \ldots, a_{k-1}$ at time steps $0, \ldots, k$. This enlarged state at time k contains all the information known to the controller at time k.

However, the size of the sufficient statistic in (2.27) grows over time, leading to a non-stationary closed-loop system, and introducing a challenge in RL since the number of inputs to the networks varies over time. This problem can be solved by reducing

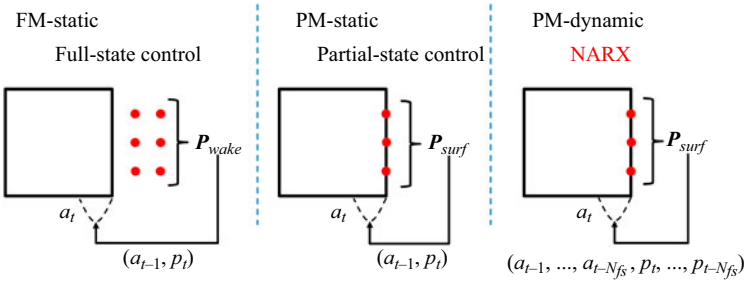


Figure 2. Demonstration of an FM environment with a static feedback controller ('FM-Static'), a PM environment with a static feedback controller ('PM-Static'), and a PM environment with a dynamic feedback controller formulated as an NARX model (case 'PM-Dynamic'). The dashed curve represents the bottom blowing/suction jet, and the red dots demonstrate schematically the locations of the sensors.

(2.27) to a finite-history approximation (White & Scherer 1994). The controller using this finite-history approximation of the sufficient statistic is usually known as a 'finite-state' controller, and the error of this approximation converges as the size of the finite history increases (Yu & Bertsekas 2008). The trade-off is that the dimension of the input increases based on the history length required. The nonlinear policy, which is parametrised by a neural network controller, has an algebraic description

$$a_t \sim \pi_\phi(a_t \mid \underbrace{a_{t-1}, a_{t-2}, \ldots, a_{t-N_{fs}-1}}_{\textit{past actions}}, p_t, \underbrace{p_{t-1}, p_{t-2}, \ldots, p_{t-N_{fs}}}_{\textit{past measurements}}), \tag{2.28}$$

where p_t represents pressure measurements at time step t, and N_{fs} denotes the size of the finite history. The above expression is equivalent to a nonlinear autoregressive exogenous (NARX) model.

A 'frame stack' technique is used to feed the 'finite-history sufficient statistic' to the RL agent as input to both the actor and critic neural networks. The frame stack constructs the observation o_t from the latest actions and measurements at step t as a 'frame' $o_t = (a_{t-1}, p_t)$, and piles up the finite history of N_{fs} frames together into a stack. The number of stacked frames is equivalent to the size of the finite history N_{fs}.

The neural network controller trained as an NARX model benefits from past information to approximate the next optimised control action since the policy has been parametrised as a nonlinear transfer function. Thus a controller parametrised as an NARX model is denoted as a 'dynamic feedback' controller because the time history in the NARX model contains dynamic information of the system. Correspondingly, a controller fed with only the latest actions a_{t-1} and current measurements p_t is denoted as a 'static feedback' controller because no past information from the system is fed into the controller.

Figure 2 demonstrates three cases with both FM and PM environments that will be investigated. In the FM environment, sensors are located in the wake as $\boldsymbol{P}_{surf}$ given by (2.3). In the PM environment, sensors are placed only on the back surface of the body as $\boldsymbol{P}_{wake}$ given by (2.4). The static feedback controller is employed in the FM environment, and both static and dynamic feedback controllers are applied in the PM environment. Results will be shown with $N_{fs} = 27$, and in § 3.3, a parametric study of the effect of the finite-history length is presented.

Environment	Algorithm	N_c	$R_{ep,c}$	(Layers, neurons)	N_{fs}	Number of inputs
FM-Static	TQC	325	37.72	(3, 512)	0	$64p_t + 2a_{t-1}$
PM-Static	TQC	1235	21.87	(3, 512)	0	$64p_t + 2a_{t-1}$
PM-Dynamic	TQC	715	34.35	(3, 512)	27	$N_{fs}(64p_t + 2a_{t-1})$

Table 1. Number of episodes N_c required for RL convergence in different environments. The episode reward $R_{ep,c}$ at the convergence point, the configuration of the neural network and the dimension of inputs are presented for each case. Here, N_{fs} is the finite-horizon length of past actions measurements.

3. Results of RL active flow control

In this section, we discuss the convergence of the RL algorithms for the three FM and PM cases (§ 3.1) and evaluate their drag reduction performance (§ 3.2). A parametric analysis of the effect of NARX memory length is presented (§ 3.3), along with the isolated effect of including past actions as observations during the RL training and control (§ 3.4). Studies of reward function (§ 3.5), sensor placement (§ 3.6) and generalisability to Reynolds number changes (§ 3.7) are presented, followed by a comparison of SAC and TQC algorithms (§ 3.8).

3.1. *Convergence of learning*

We perform RL with the maximum entropy TQC algorithm to discover control policies for the three cases shown in figure 2, which maximise the net-power-saving reward function given by (2.19). During the learning stage, each episode (one set of DNS) corresponds to 200 non-dimensional time units. To accelerate learning, 65 environments run in parallel.

Figure 3 shows the learning curves of the three cases. Table 1 shows the number of episodes needed for convergence and relevant parameters for each case. It can be observed from the curve of episode reward that the RL agent is updated after every 65 episodes, i.e. one iteration, where the episode reward is defined as

$$R_{ep} = \sum_{k=1}^{N_k} r_k, \tag{3.1}$$

where k denotes the kth RL step in one episode, and N_k is the total number of samples in one episode. The root mean square (RMS) value of the drag coefficient, C_D^{RMS}, at the asymptotic regime of control, is also shown to demonstrate convergence, defined as $C_D^{RMS} = \sqrt{(\mathcal{D}(\langle C_D \rangle_{env}))^2}$, where the operator $\mathcal{D}$ detrends the signal with a 9th-order polynomial and removes the transient part, and $\langle \cdot \rangle_{env}$ denotes the average value of parallel environments in a single iteration.

In figure 3, it can be noticed that in the FM environment, RL converges after approximately 325 episodes (five iterations) to a nearly optimal policy using a static feedback controller. As will be shown in § 3.2, this policy is optimal globally since the vortex shedding is fully attenuated and the jets converge to zero mass flow actuation, thus recovering the unstable base flow and the minimum drag state. However, with the same static feedback controller in a PM environment (POMDP), the RL agent fails to discover the nearly optimal solution, requiring approximately 1235 episodes for convergence but obtaining only a relatively low episode reward. Introducing a dynamic feedback controller in the PM environment, the RL agent converges to a near-optimal solution in 735 episodes.

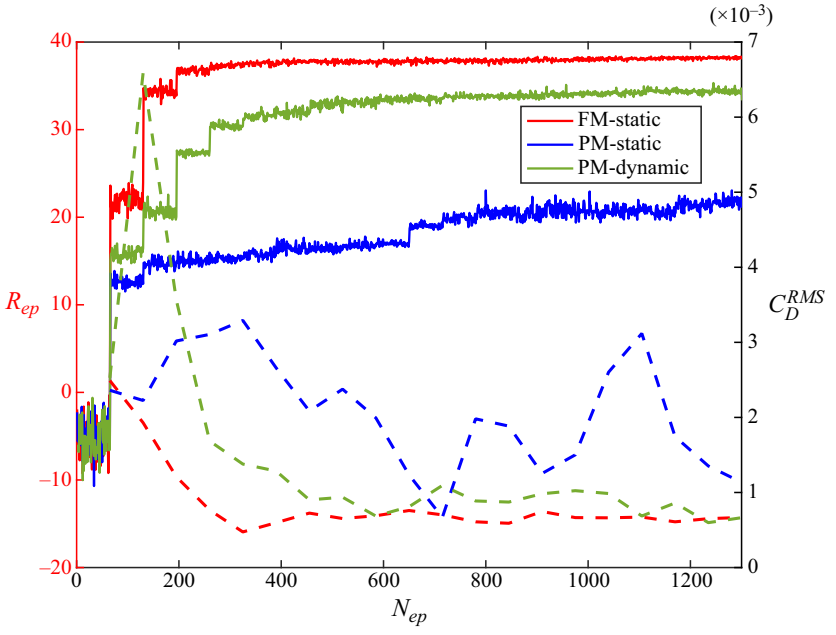


Figure 3. Episode rewards (solid lines) and RMS of drag coefficient (dashed lines) against episode number during the maximum entropy RL phase with TQC.

The dynamic feedback controller trained by RL achieves a higher episode reward (34.35) than the static feedback controller in the PM case (21.87), which is close to the FM case (37.72). The learning curves illustrate that using a finite horizon of past actions measurements ($N_{fs} = 27$) to train a dynamic feedback controller in the PM case improves learning in terms of speed of convergence and accumulated reward, achieving nearly optimal performance with only wall pressure measurements.

3.2. *Drag reduction with dynamic RL controllers*

The trained controllers for the cases shown in figure 2 are evaluated to obtain the results shown in figure 4. Evaluation tests are performed for 120 non-dimensional time units to show both transient and asymptotic dynamics of the closed-loop system. Control is applied at $t = 0$ with the same initial condition for each case, i.e. steady vortex shedding with average drag coefficient $\langle C_{D0}\rangle \approx 1.45$ (baseline without control). Consistent with the learning curves, the difference in control performance in the three cases can be observed from both the drag coefficient C_D and the actuation Q_1. The drag reduction is quantified by a ratio η using the asymptotic time-averaged drag coefficient with control $C_{Da} = \langle C_D\rangle_{t\in[80,120]}$, the drag coefficient C_{Db} of the base flow (details presented in Appendix D), and the baseline time-averaged drag coefficient without control $\langle C_{D0}\rangle$, as

$$\eta = \frac{\langle C_{D0}\rangle - C_{Da}}{\langle C_{D0}\rangle - C_{Db}} \times 100\,\%. \tag{3.2}$$

(i) FM-Static. With a static feedback controller trained in a full-measurement environment, a drag reduction $\eta = 101.96\,\%$ is obtained with respect to the base flow (steady unstable fixed point; maximum drag reduction). This indicates that an

RL controller informed with full-state information can stabilise the vortex shedding entirely, and cancel the unsteady part of the pressure drag.

(ii) PM-Static. A static/memoryless controller in a PM environment leads to performance degradation and a drag reduction $\eta = 56.00\,\%$ in the asymptotic control stage, i.e. after $t = 80$, compared to the performance of FM-Static. This performance loss can also be observed from the control actuation curve, as Q_1 oscillates with a relatively large fluctuation in PM-Static, while it stays near zero in the FM-Static case. The discrepancy between FM and PM environments using a static feedback controller reveals the challenge of designing a controller with a POMDP environment. The RL agent cannot fully identify the dominant dynamics with only PM on the downstream surface of the bluff body, resulting in sub-optimal control behaviour.

(iii) PM-Dynamic. With a dynamic feedback controller (NARX model presented in § 2.4) in a PM environment, the vortex shedding is stabilised, and the dynamic feedback controller achieves $\eta = 97.00\,\%$ of the maximum drag reduction after time $t = 60$. Although there are minor fluctuations in the actuation Q_1, the energy spent in the synthetic jets is significantly lower compared to the PM-Static case. Thus a dynamic feedback controller in PM environments can achieve nearly optimal drag reduction, even if the RL agent collects information only from pressure sensors on the downstream surface of the body. The improvement in control indicates that the POMDP due to the PM condition of the sensors can be reduced to an approximate MDP by training a dynamic feedback controller with a finite horizon of past actions measurements. Furthermore, high-frequency action oscillations, which can be amplified with static feedback controllers, are attenuated in the case of dynamic feedback control. These encouraging and unexpected results support the effectiveness and robustness of model-free RL control in practical flow control applications, in which sensors can be placed only on a solid surface/wall.

In figure 5, snapshots of the velocity magnitude $|\boldsymbol{u}| = \sqrt{u^2 + v^2}$ are presented for baseline without control, PM-Static, PM-Dynamic and FM-Static control cases. Snapshots are captured at $t = 100$ in the asymptotic regime of control. A vortex shedding structure of different strengths can be observed in the wake of all three controlled cases. In PM-Static, the recirculation area is lengthened compared to the baseline flow, corresponding to base pressure recovery and pressure drag reduction. A longer recirculation area can be noticed in PM-Dynamic due to the enhanced attenuation of vortex shedding and pressure drag reduction. The dynamic feedback controller in the PM case renders a 326.22 % increase of recirculation area with respect to the baseline flow, while only a 116.78 % increase is achieved by a static feedback controller. The FM-Static case has the longest recirculation area, and the vortex shedding is almost fully stabilised, which is consistent with the drag reduction shown in figure 4.

Figure 6 presents first- and second-order base pressure statistics for the baseline case without control and PM cases with control. In figure 6(*a*), the time-averaged value of base pressure, $\bar{p}$, demonstrates the base pressure recovery after control is applied. Due to flow separation and recirculation, the time-averaged base pressure is higher at the middle of the downstream surface, which is retained with control. The base pressure increase is linked directly to pressure drag reduction, which quantifies the control performance of both static and dynamic feedback controllers. Up to 49.56 % of the pressure increase at the centre of the downstream surface is obtained in the PM-Dynamic case, while only 21.15 % can be achieved by a static feedback controller. In figure 6(*b*), the base pressure RMS is

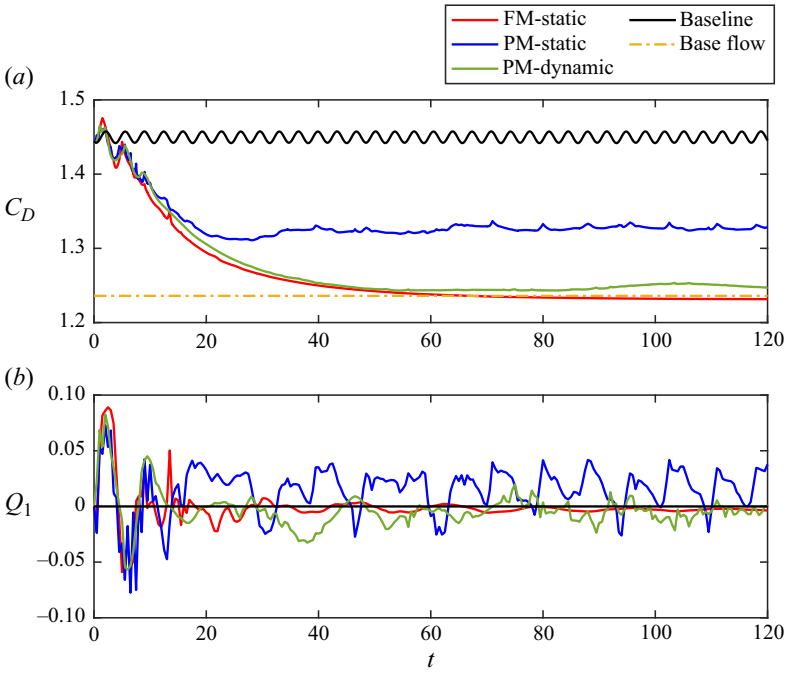


Figure 4. (*a*) Drag coefficient C_D without control ('Baseline') and with active flow control by RL in both FM and PM cases. In PM cases, control results with a dynamic and static feedback controller are presented. The dash-dotted line represents the base flow C_{Db}. (*b*) The mass flow rate Q_1 of one of the blowing and suction jets.

shown. For the baseline flow, strong vortex-induced fluctuations of the base pressure can be noticed around the top and bottom on the downstream surface of the bluff body. In the PM-Static case, the RL controller partially suppresses the vortex shedding, leading to a sub-optimal reduction of the pressure fluctuation. The sensors close to the top and bottom corners are also affected by the synthetic jets, which change the RMS trend for the two top and bottom measurements. In the PM-Dynamic case, the pressure fluctuations are nearly zero for all the measurements on the downstream surface, highlighting the success of vortex shedding suppression by a dynamic RL controller in a PM environment.

The differences between static and dynamic controllers in PM environments are elucidated further in figure 7 by examining the time series of pressure differences Δp_t from surface sensors (control input) and control actions a_{t-1} (output). The pressure differences are calculated from sensor pairs at $y = \pm y_{sensor}$, where y_{sensor} is defined in (2.3). For $N = 64$, there are 32 time series of Δp_t for each case. During the initial stages of control ($t \in [0, 11]$), the control actions are similar for the two PM cases and they deviate for $t > 11$, resulting in discernible control performance at the asymptotic regime. At the initial stages, the controllers operate in nearly anti-phase to Δp_t, in order to eliminate the antisymmetric pressure component due to vortex shedding. The inability of the static controller to have a frequency-dependent amplitude (and phase) manifests as well through the amplification of high-frequency noise. For $t > 11$, the static feedback controller continues to operate in nearly anti-phase to the pressure difference, resulting in partial stabilisation of unsteadiness. However, the dynamic feedback controller adjusts its phase and amplitude significantly, which attenuates the antisymmetric fluctuation of base pressure and drives Δp_t to near zero.

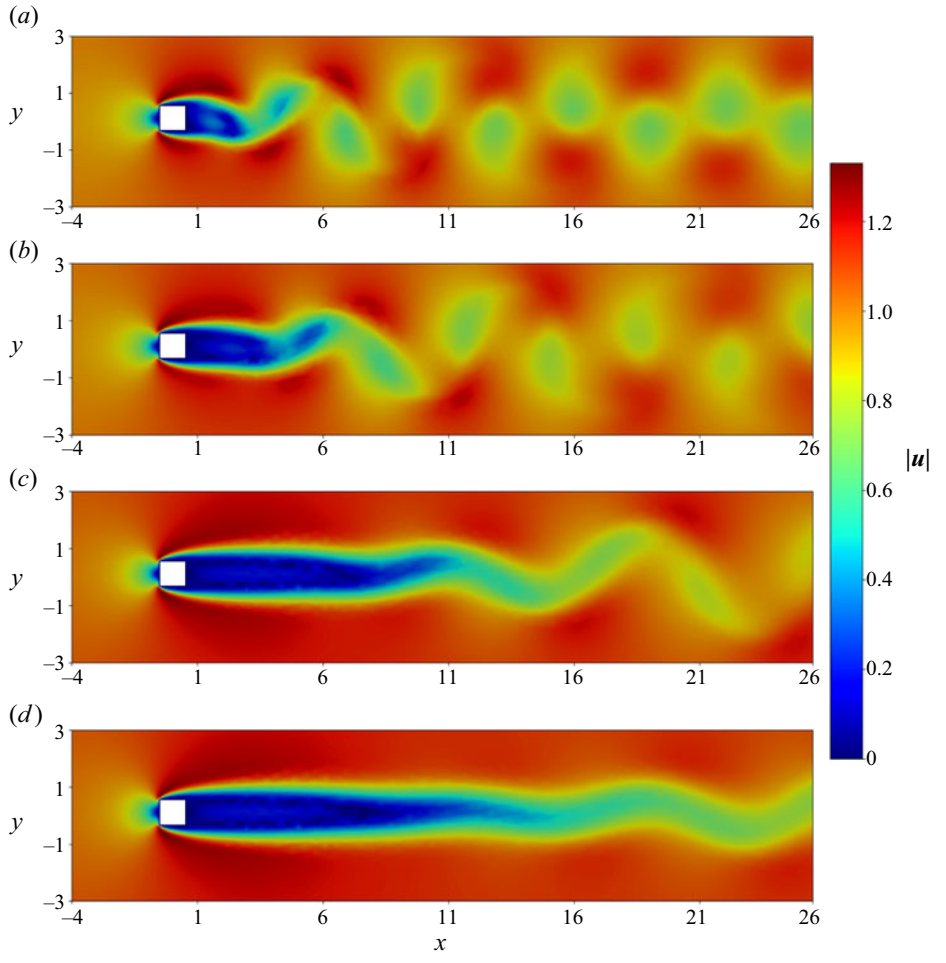


Figure 5. Contours of velocity magnitude $|\boldsymbol{u}|$ in the asymptotic regime of control (at $t = 100$). Areas of $(-4, 26)$ in the x direction and $(-3, 3)$ in the y direction are presented for visualisation: (*a*) baseline (no control), (*b*) PM-Static, (*c*) PM-Dynamic, (*d*) FM-Static.

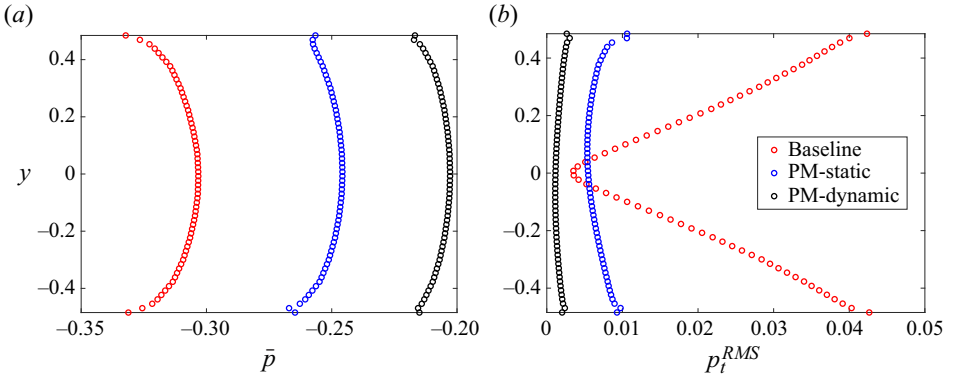


Figure 6. (*a*) Mean and (*b*) RMS base pressure for controlled and uncontrolled cases from the 64 wall sensors on the downstream surface of the bluff body base.

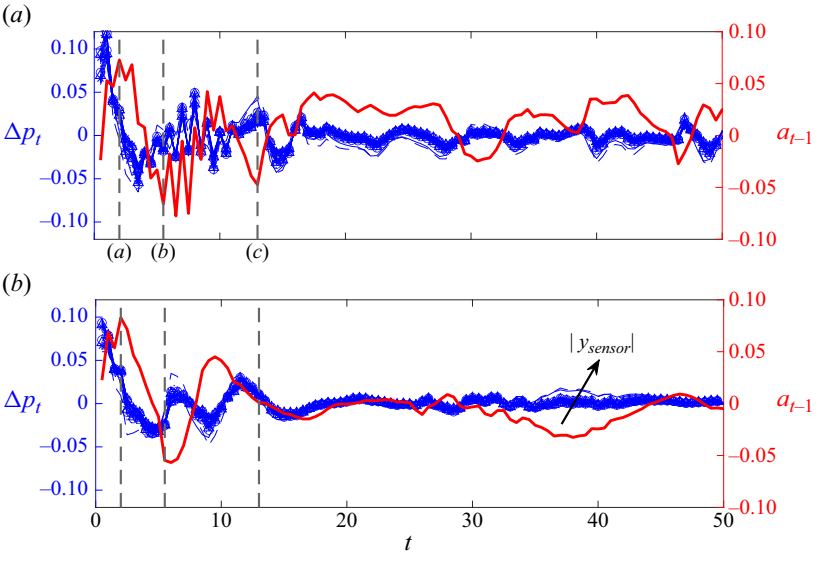


Figure 7. Time series of pressure differences Δp_t (blue) and action a_{t-1} (red) for (*a*) PM-Static and (*b*) PM-Dynamic cases. Control is applied at $t = 0$. The arrows are pointing from low to high values of $|y_{sensor}|$ among Δp_t curves. The vertical dashed lines mark the time instances of the vorticity snapshots in figure 8.

Figure 8 shows instantaneous vorticity contours for PM-Dynamic and PM-Static cases, showing both similarities and discrepancies between the two cases. At $t = 2$, flow is expelled from the bottom jet for both cases, generating a clockwise vortex, termed V1. This V1 vortex, shown in black, works against the primary anticlockwise vortex labelled as P1, depicted in red, emerging from the bottom surface. At $t = 5.5$, a secondary vortex, V2, forms from the jets to oppose the primary vortex shedding from the top surface (labelled as P2). At $t = 13$, the suppression of the two primary vortices near the bluff body is evident in both cases, indicated by their less tilted shapes compared to the previous time instances. At $t = 13$, the PM-Dynamic adjusted the phase of the control signal, which corresponds to a marginal action at this time instance at figure 7. Consequently, no additional counteracting vortex is formed in PM-Dynamic. However, in the PM-Static scenario, the jets generate a third vortex, labelled V3, which emerges from the top surface. This corresponds to a peak in the action of the PM-Static controller at this time. The inability of the PM-Static controller to adapt the amplitude/phase of the input–output behaviour results in suboptimal performance.

3.3. *Horizon of the finite-history sufficient statistic*

A parametric study on the horizon of the finite history in the NARX model (2.28), i.e. the number of frames stacked N_{fs}, is presented in this subsection. Since the NARX model uses a finite horizon of past actions measurements in (2.27), the horizon of the finite history affects the convergence of the approximation (Yu & Bertsekas 2008). This approximation affects the optimisation during the learning of RL because it determines whether the RL agent can observe sufficient information to converge to an optimal policy.

Since vortex shedding is the dominant instability to be controlled, the choice of N_{fs} should link intuitively to the time scale of the vortex shedding period. The 'frames' of observations are obtained every RL step (0.5 time units), while the vortex shedding period

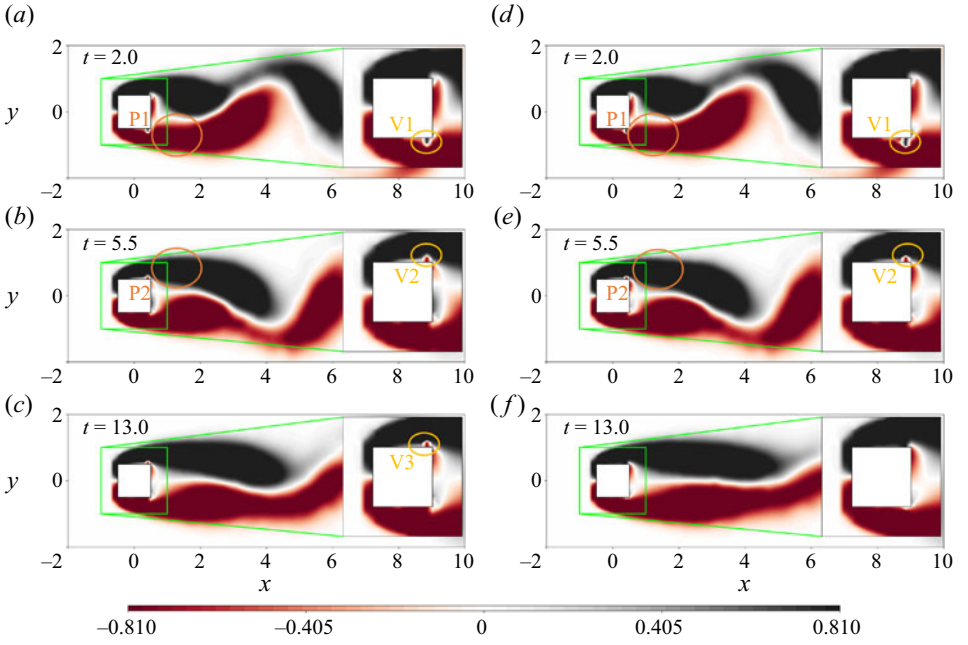


Figure 8. Vorticity snapshots at the transient phase of control: (*a–c*) PM-Static, (*d–f*) PM-Dynamic.

Number of VS periods	Non-dimensional time units	History length (N_{fs})
0.5	3.43	7
1	6.85	14
2	13.70	27
3	20.55	41
4	27.40	55
5	34.25	68

Table 2. Correspondence between the number of vortex shedding (VS) periods and frame stack (history) length in samples N_{fs}. The RL control step size is $t_a = 0.5$, and N_{fs} is rounded to an integer.

is $t_{vs} \approx 6.85$ time units. Thus N_{fs} is rounded to integer values for different numbers of vortex shedding periods, as shown in table 2.

The results of time-averaged drag coefficients $\langle C_D \rangle$ after control, and the average episode rewards $\langle R_{ep} \rangle$ in the final stage of training, are presented in figure 9. As N_{fs} increases from 0 to 27, the performance of RL control improves, resulting in a lower $\langle C_D \rangle$ and a higher $\langle R_{ep} \rangle$. We examine $N_{fs} = 2$ especially, because the latent dimension of the vortex shedding limit cycle is 2. However, the control performance with $N_{fs} = 2$ is improved marginally to the one with $N_{fs} = 0$, i.e. a static feedback controller. This result indicates that the horizon consistent with the vortex shedding dimension is not long enough for the finite horizon of past action measurements. The optimal history length to achieve stabilisation of the vortex shedding in PM environments is 27 samples, which are equivalent to 13.5 convective time units or ∼2 vortex shedding periods.

With $N_{fs} = 41$ and $N_{fs} = 55$, the drag reduction and episode rewards drop slightly compared to $N_{fs} = 27$. The decline in performance is non-negligible as N_{fs} increases

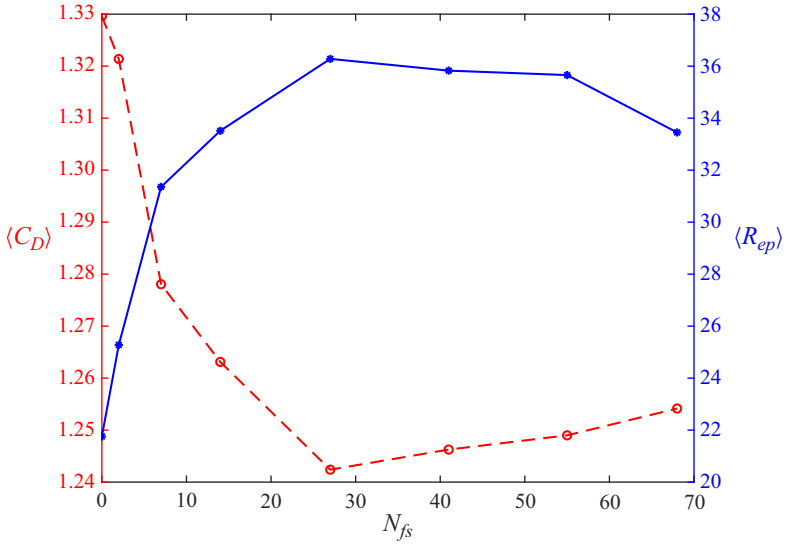


Figure 9. Average drag coefficient $\langle C_D \rangle$ and average episode reward $\langle R_{ep} \rangle$ in PM cases against number history length (numbers of stacked frames) N_{fs}. Here, $\langle C_D \rangle$ is obtained from the asymptotic regime of control, and $\langle R_{ep} \rangle$ is calculated from two episodes after convergence of RL.

further to 68. This decline shows that excessive inputs to the neural networks (see table 1) may impede training because more parameters need to be tuned or larger neural networks need to be trained.

3.4. *Observation sequence with past actions*

Past actions (exogenous terms in NARX) facilitate reducing a POMDP to an MDP problem, as discussed in § 2.4. In the near-optimal control of a PM environment using a dynamic feedback controller with inputs $(o_t, o_{t-1}, \ldots, o_{t-N_{fs}})$, a sequence of observations $o_t = \{p_t, a_{t-1}\}$ at step t is constructed to include pressure measurements and actions. In the FM environment, due to the introduction of one-step delayed action due to the first-order hold interpolation given by (2.18), the inclusion of the past action along with the current pressure measurement, meaning $o_t = \{p_t, a_{t-1}\}$, is required even when the sensors are placed in the wake and cover the wavemaker region.

Figure 10 presents the control performance for the same environment with and without past actions included. In the FM case, there is no apparent difference between RL control with $o_t = \{p_t, a_{t-1}\}$ or $o_t = \{p_t\}$, which indicates that the inclusion of the past action is negligible to the performance. This is the case when the RL sampling frequency is sufficiently faster than the time scale of the vortex shedding dynamics. In PM cases, if exogenous action terms are not included in the observations but only the finite history of pressure measurements is used, then the RL control fails to converge to a near-optimal policy, with only $\eta = 67.45\,\%$ drag reduction. With past actions included, the drag reduction of the same environment increases to $\eta = 97.00\,\%$.

The above results show that in PM environments, sufficient statistics cannot be constructed from only the finite history of measurements. Missing state information needs to be reconstructed by both state-related measurements and control actions.

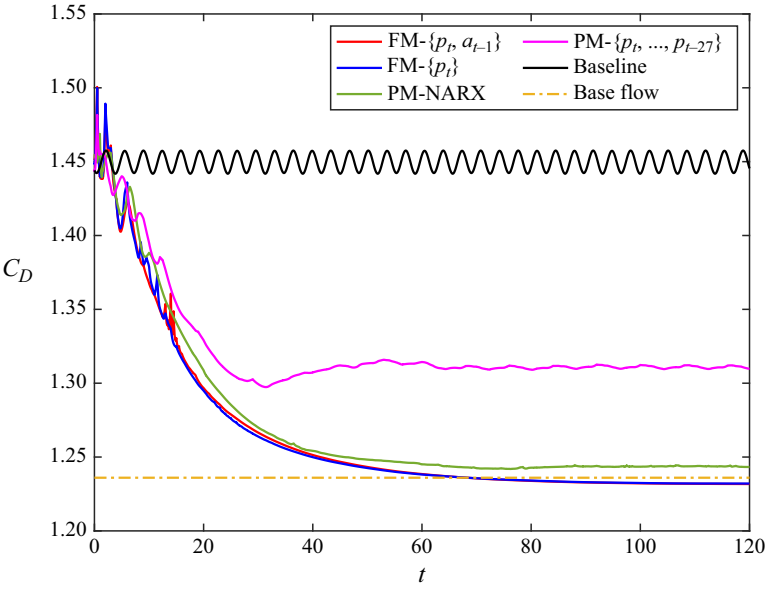


Figure 10. Curves of drag coefficients after control applied in both FM and PM environments. Results from FM cases are presented as references, while a performance difference can be observed in the PM cases with and without past actions included.

3.5. *Reward study*

In § 3.2, a power-based reward function given by (2.19) has been implemented, and stabilising controllers can be learned by RL, as shown. In this subsection, RL control results with other forms of reward functions (introduced in § 2.3) are provided and discussed.

The control performance of RL control with the different reward functions is evaluated based on the drag coefficient C_D shown in figure 11. Static feedback controllers are trained in FM environments, and dynamic feedback controllers are trained in PM environments. In FM cases, control performance is not sensitive to the choice of reward function (power or force-based). In PM cases, the discrepancies between RL-step time-averaged and instantaneous rewards can be observed in the asymptotic regime of control. The controllers with both rewards (power or force-based) achieve nearly optimal control performance, but there is some unsteadiness in the cases using instantaneous rewards due to slow statistical convergence of the rewards and limited correlation to the partial observations.

All four types of reward functions studied in this work achieve nearly optimal drag reduction at approximately 100 %. However, the energy-based reward ('PowerR') offers an intuitive reward design, attributable to its physical properties and the dimensionally consistent addition of the constituent terms of the reward function. Further enhancing its practicality, since the power of the actuator can be measured directly, it avoids the necessity for hyperparameter tuning, as in the force-based reward. Additionally, the results show similar performance with both time-averaged between RL steps and instantaneous rewards, avoiding the necessity for faster sampling for the calculation of the rewards. This choice of reward function can be extended to various RL flow control problems, and can be beneficial to experimental studies.

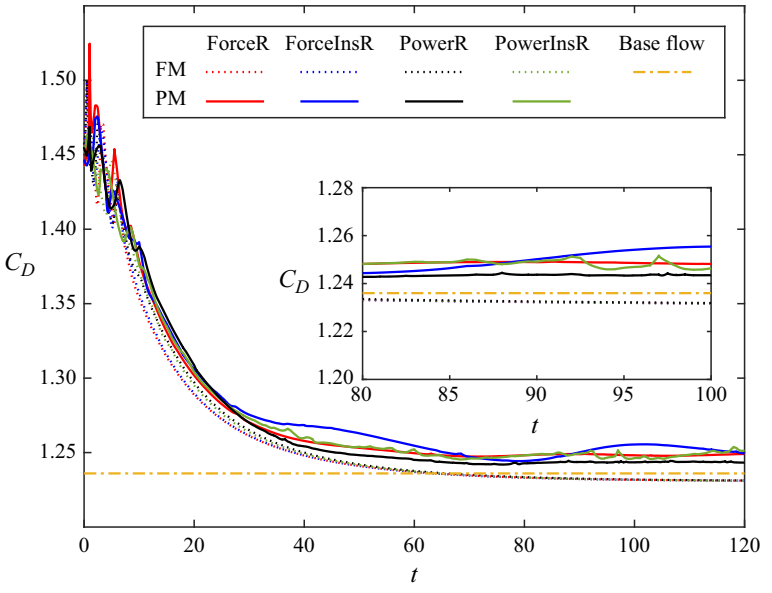


Figure 11. Tests of RL-trained controllers with various reward functions. Drag coefficient C_D curves are presented for each case. Dotted lines denote the cases with FM environments, while solid lines denote PM environments. The dash-dotted line represents C_D in the base flow, which has no vortex shedding. Control starts at $t = 0$, with the same initial conditions for every case.

3.6. *Sensor configuration study with PM*

In the PM environment, the configuration of sensors (number and location on the downstream surface) may also affect the information contained in the observations, and thus control performance. Control results of drag coefficient C_D for different sensor configurations in PM-Dynamic cases are presented in figure 12. In the configuration with $N = 2$, two sensors are placed at $y = \pm 0.25$, and for $N = 1$, only one sensor is placed at $y = 0.25$. Other configurations are consistent with (2.3).

The C_D curves in figure 12 show that as the number of sensors is reduced from 64 to 2, RL control achieves the same level of performance with minor discrepancies due to randomness in different learning cases. However, if RL control uses observations from only one sensor at $y = 0.25$, performance degradation can be observed in the asymptotic stage with 19.79 % on average less drag reduction. The inset presents the relationship between the number of sensors and asymptotic drag coefficient $\langle C_D \rangle$. These results indicate a limit on sensor configuration for the use of the NARX-modelled controller to stabilise the vortex shedding.

To understand the cause of performance degradation in the $N = 1$ case, the pressure measurements from two sensors in both baseline and PM-Dynamic cases are presented in figure 13. In the baseline case, two sensors are placed at the same location as the $N = 2$ case ($y = \pm 0.25$) only for observations. It can be observed that the pressure measurements from two sensors are antisymmetric since they are placed symmetrically on the downstream surface. In the PM-Dynamic case, the NARX controller is used, and control is applied at $t = 0$. In this closed-loop system, the antisymmetric relationship between two sensors (from the symmetric position) is broken by the control actuation, and no correlation is evident. This can be seen during the transient dynamics, e.g. in $t \in [0, 10]$. Therefore, when the number of sensors is reduced to $N = 1$ by removing one sensor from

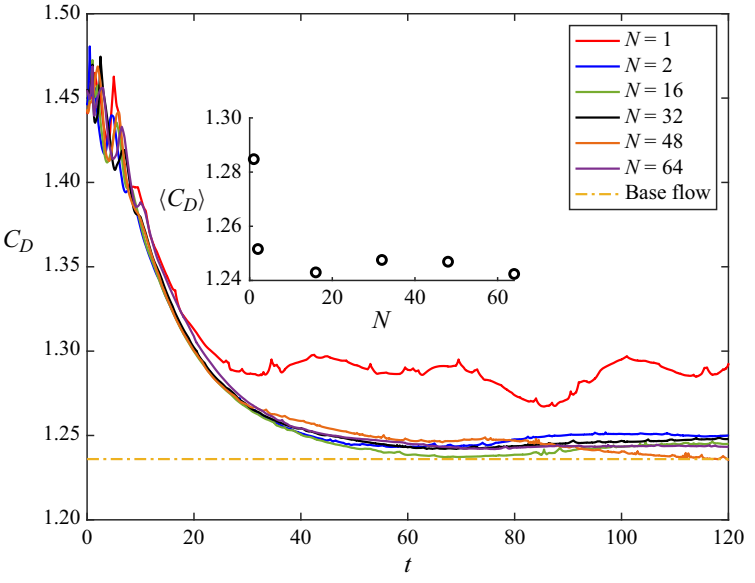


Figure 12. Curves of drag coefficients after control applied at $t = 0$ in PM-Dynamic cases. Sensor configurations with different sensor numbers $N = 1, 2, 16, 32, 48, 64$ are tested. The dash-dotted line presents C_D from the base flow. The inset shows the asymptotic drag coefficient $\langle C_D \rangle$ (time-averaged value after $t = 80$) and probe number N.

the $N = 2$ case, the dynamic feedback from the removed sensor cannot be reflected fully by the remaining sensor in the closed-loop system. This loss of information affects the fidelity of the control response to the dynamics of the sensor-removing side, causing sub-optimal drag reduction in the $N = 1$ scenario.

It should be noted that the configuration of 64 sensors is not necessary for control, as $N = 2$ or $N = 16$ also achieves nearly optimal performance. The number of sensors $N = 64$ in PM-Static environments is used for comparison with the FM-Static configuration (2.4), which eliminates the effect from different input dimensions between two static cases. Also, 64 sensors cover the downstream surface of the bluff body sufficiently to avoid missing spatial information. The optimal configuration of sensors can be tuned with optimisation techniques such as in Paris *et al.* (2021), but the results in figure 12 indicate that RL adapts with nearly optimal performance to non-optimised sensor placement in the present environment.

3.7. *Performance of RL controllers to unseen Re*

The RL controller is tested at different Reynolds numbers, in order to examine its generalisability to environment changes. The controllers have been trained at $Re = 100$ with both FM and PM conditions, and tested at $Re = 80, 90, 100, 110, 120, 150$. The controllers were trained further at $Re = 150$, denoted as continual learning (CL), and tested again at $Re = 150$.

As shown in figure 14, in both PM-Dynamic and FM-Static cases, the RL controllers are able to reduce drag by $\eta = 64.68\,\%$ in the worst case, when Re is close to the training point at $Re = 100$, i.e. the test cases with $Re = 80, 90, 100, 110, 120$. However, when applying the controllers trained at $Re = 100$ to an environment at $Re = 150$, the

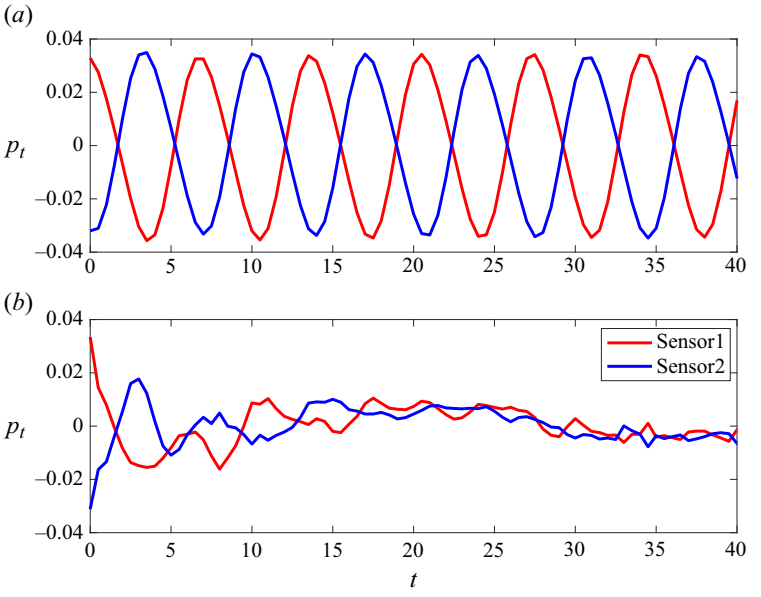


Figure 13. Pressure measurements in $t \in [0, 40]$ (early transient stage in the controlled case) from two surface sensors: (*a*) baseline without control; (*b*) PM-Dynamic with an NARX controller, $N = 2$. All curves are detrended by a fifth-order polynomial to reveal the relationship between measurements from the two sensors.

drag reduction drops to $\eta = 41.98\,\%$ and $\eta = 74.04\,\%$ in the PM-Dynamic and FM-Static cases, respectively.

Performing CL at $Re = 150$, the drag reduction is improved to $\eta = 78.07\,\%$ in PM-Dynamic after 1105 training episodes, while $\eta = 88.13\,\%$ in FM-Static after 390 episodes, with the same RL parameters as the training at $Re = 100$. Overall, the results of these tests indicate that the RL-trained controllers can achieve significant drag reduction in the vicinity of the training point (i.e. $\pm 20\,\%$ *Re* change). If the test point is far from the training point, then a CL procedure can be implemented to achieve nearly optimal control.

3.8. *Comparing TQC to SAC*

Control results with TQC and SAC are presented in figure 15 in terms of C_D, where TQC shows a more robust control performance. In the case of FM, SAC might demonstrate a slightly more stable transient behaviour attributed to the fact that the quantile regression process in TQC introduced complexity to the optimisation process. Both controllers achieved an identical level of drag reduction in the FM case.

However, in the context of the PM cases, it is observed that TQC outperforms SAC in drag reduction with both static and dynamic feedback controllers. For static feedback control, TQC achieved an average drag reduction $\eta = 56.00\,\%$, compared to the $\eta = 46.31\,\%$ reduction achieved by SAC. The performance under dynamic feedback control conditions is more compelling, where TQC fully reduced the drag, achieving $\eta = 97.00\,\%$ of drag reduction, reverting it to a near-base-flow scenario. In contrast, SAC managed to achieve average drag reduction $\eta = 96.52\,\%$.

The fundamental mechanism for updating Q-functions in RL involves selecting the maximum expected Q-functions among possible future actions. This process, however, potentially can lead to overestimation of certain Q-functions (Hasselt 2010). In the

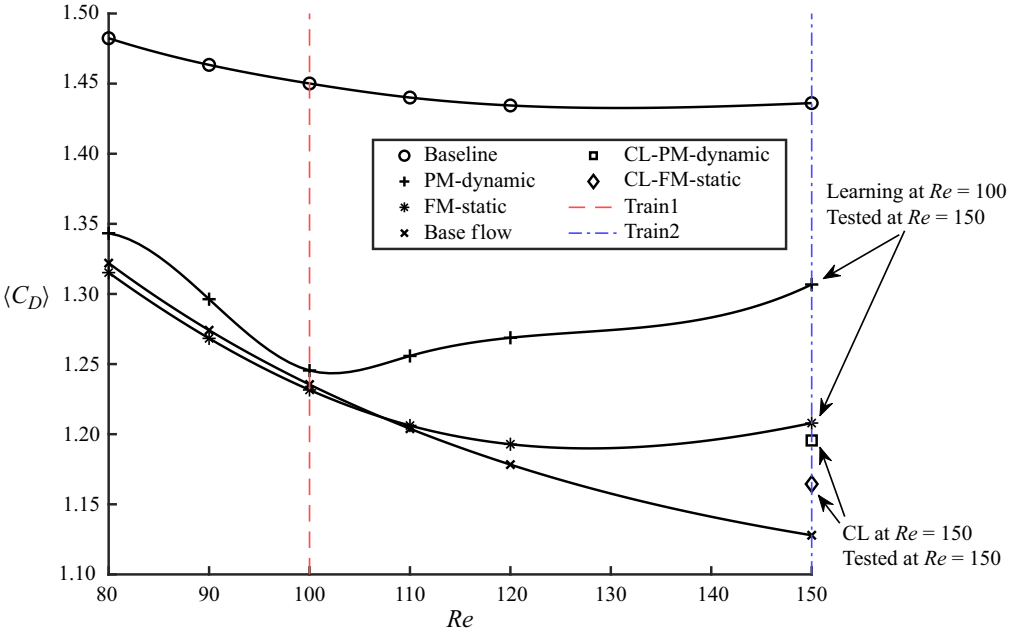


Figure 14. Asymptotic drag coefficient $\langle C_D \rangle$ for baseline, base flow, and tests of RL-trained controllers in both FM and PM environments with different Re. The controllers were trained at $Re = 100$ (dashed line), and tested at $Re = 80, 90, 100, 110, 120, 150$. The controllers were trained again at $Re = 150$ (dash-dotted line) and tested at $Re = 150$ (square and diamond markers). All curves are fitted using a third-order spline.

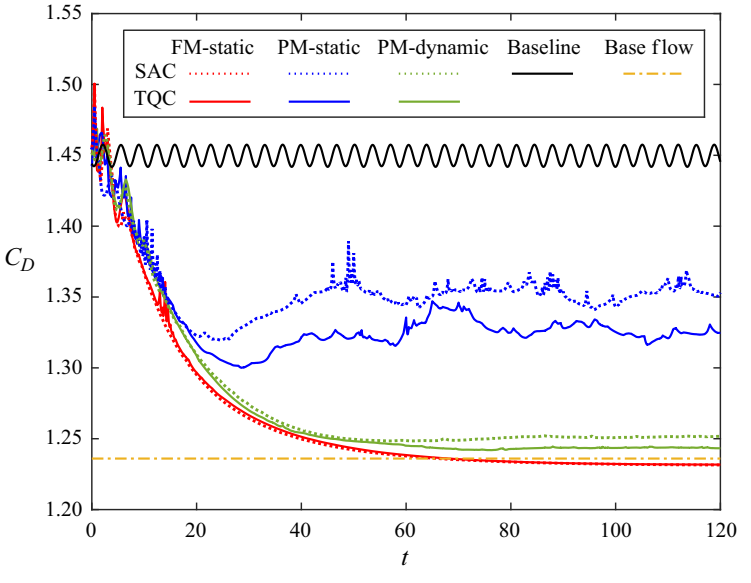


Figure 15. Comparison of control performance in terms of C_D between SAC and TQC. Control starts at $t = 0$. Solid curves show the cases using TQC and baseline, while dotted curves show SAC. The dash-dotted curve corresponds to the base flow C_D.

POMDP, this overestimation bias might be exacerbated due to the inherent uncertainty arising from the partial-state information. Therefore, the Q-learning-based algorithm, when applied to a POMDP, might be more prone to choosing these overestimated values, thereby affecting the overall learning and decision-making process.

As mentioned in § 2.2, the core benefit of TQC under these conditions can be attributed to its advanced handling of the overestimation bias of rewards. By constructing a more accurate representation of possible returns, TQC provides a more accurate Q-function approximation than SAC. This process of modulating the probability distribution of the Q-function assists TQC in managing the uncertainties inherent in environments with only partial-state information. In this case, TQC can adapt more robustly to changes and uncertainties, leading to better performance in both static and dynamic feedback control tasks.

4. Conclusions

In this study, maximum entropy RL with TQC has been performed in an active flow control application with PM to learn a feedback controller for bluff body drag reduction. Neural network controllers have been trained by the RL algorithm to discover a drag reduction control strategy behind a 2-D square bluff body at $Re = 100$. By comparing the control performances in FM environments to PM environments, we showed a non-negligible degradation of RL control performance if the controller is not trained with full-state information. To solve this issue, we proposed a method to train a dynamic neural network controller with an approximation of a finite-history sufficient statistic and formulate the dynamic controller as an NARX model. The dynamic controller was able to improve the drag reduction performance in PM environments and achieve near-optimal performance (drag reduction ratio $\eta = 97\,\%$ with respect to the base flow drag) compared to a static controller ($\eta = 56\,\%$). We found that the optimal horizon of the finite history in NARX is approximately two vortex shedding periods when the sensors are located only on the base of the body. The importance of including exogenous action terms in the observations of RL is discussed, by pointing out the degradation of $\eta = 29.55\,\%$ on drag reduction if only past measurements are used in the PM environment. Also, we proposed a net power consumption design for the reward function based on the drag power savings and the power of the actuator. This power-based reward function offers an intuitive understanding of the closed-loop performance, whereas electromechanical losses can also be added directly, once a specific actuator is chosen. Moreover, its inherent feature of being hyperparameter-free contributes to a straightforward reward function design process in the context of flow control problems. Results from SAC are compared with TQC, and we showed the improvement by TQC, which attenuates overestimation in neural networks.

It was shown that model-free RL was able to discover a nearly optimal control strategy without any prior knowledge of the system dynamics using partial realistic measurements, exploiting only input–output data from the simulation environment. Therefore, this particular study on RL-based active flow control in 2-D laminar flow simulations can be seen as a step towards controlling the complex dynamics of three-dimensional turbulent flows in practical applications by replacing the simulation environment with the experimental set-up. Also, the frame stack method employed here to convert the POMDP to an MDP can be replaced by recurrent neural networks and attention-based architectures, which may further improve control performance in a scenario with complex dynamics.

Funding. We acknowledge support from the UKRI AI for Net Zero grant EP/Y005619/1.

Declaration of interests. The authors report no conflict of interest.

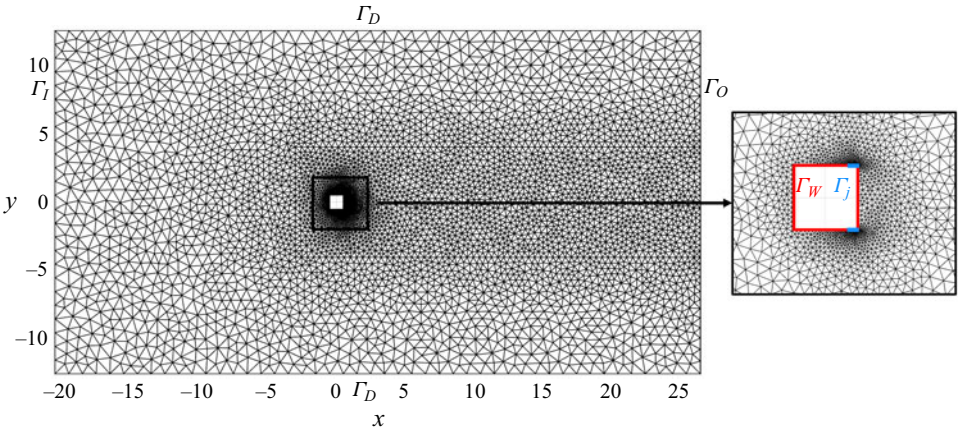


Figure 16. Computational mesh of the simulation domain, $x \in (-20.5, 26.5)$ and $y \in (-12.5, 12.5)$. A zoom-in view around the bluff body is presented in the black rectangle at the right. Boundaries of the simulation domain, bluff body surface and jet area are denoted.

Data availability statement. The open-source code of this project is available on the GitHub repository: https://github.com/RigasLab/Square2DFlowControlDRL-PM-NARX-SB3. The code is developed from the work by Rabault *et al.* (2019) and Rabault & Kuhnle (2019), using a simulation environment by FEniCS v2017.2.0 (Logg *et al.* 2012). The RL algorithm is adapted to a version with SAC/TQC, implemented by Stable-Baselines3 and Stable-Baselines3-contrib (Raffin *et al.* 2021) in a PyTorch (Paszke *et al.* 2019) environment. See Appendix A and the GitHub repository for more details of the simulation.

Author ORCIDs.
Georgios Rigas https://orcid.org/0000-0001-6692-6437.

Appendix A. Details of simulation environment

The simulation environment for solving the governing Navier–Stokes equations is adapted from Rabault *et al.* (2019) to the flow past a square bluff body. The boundary condition at the inflow boundary Γ_I is set as a uniform velocity profile, and a zero-pressure condition is used at the outflow boundary Γ_O. A freestream condition is used at the top and bottom boundary Γ_D of the domain. The boundary on the bluff body is separated into body surface Γ_W and jet area Γ_j, with a no-slip boundary condition and jet velocity profile, respectively. The boundary conditions are formulated as

$$\left.\begin{aligned} u &= U_\infty \quad \text{on } \Gamma_I, \\ p &= 0 \quad \text{on } \Gamma_O, \\ u &= U_\infty \quad \text{on } \Gamma_D, \\ u &= 0 \quad \text{on } \Gamma_W, \\ \boldsymbol{u} &= \boldsymbol{U_j} \quad \text{on } \Gamma_j, j = 1, 2. \end{aligned}\right\} \tag{A1}$$

The mesh of the simulation domain and a zoom-in view of the mesh around the square bluff body are presented in figure 16. The mesh is refined in the wake region with ratio 0.45, and near the body wall with ratio 0.075, with respect to the mesh size of the far field. Near the jet area, the mesh is refined further, with ratio 0.015. More details can be found in the source code (see the GitHub repository).

Hyperparameter	Value
Optimiser	Adam
Learning rate	10^{-4}
Discount (γ)	0.99
Replay buffer size	10^5
Number of hidden layers (both actor and critic)	3
Number of hidden units per layer	512
Number of samples per mini-batch	128
Entropy target	$-\dim(\mathcal{A})$
Activation	*ReLU*
Target smoothing coefficient (τ)	5×10^{-3}
Target update interval	1
Gradient steps	48
Top quantiles to drop per net	2
Number of quantiles per net	25

Table 3. Hyperparameters used by default in TQC. For SAC, 'top quantiles to drop per net' is not used, and other parameters remain the same. For the entropy target, $-\dim(\mathcal{A})$ denotes the dimension of action space $\mathcal{A}$.

Appendix B. Hyperparameters of RL

The RL hyperparameters to reproduce the results section (§ 3) are listed in table 3.

Appendix C. A long-run test of RL-trained controller

In figure 17, the trained policy is tested for a longer time (400 time units) than training (200 time units) to show the control stability outside the training time frame for the dynamic controller in the PM environment. The initial condition of this long-run test is different compared to figure 4, indicating the adaptability of the controller to different initial conditions. Other parameters in this run are consistent with the results in figure 4.

The control performance and behaviour in this test are consistent with the results shown in figure 4 in both the transient stage and the asymptotic stage. The drag coefficient C_D starts from the condition of steady vortex shedding, and drops to the value of the stabilised flow in approximately 120 time units, with minor fluctuations. After training time (200 time units), the controller is still able to prevent triggering vortex shedding and preserve the drag coefficient near the base flow values (minimum drag without vortex shedding). The behaviour of the controller is presented further in the insets of Q_1. The controller creates negligible random mass flow after stabilising the vortex shedding due to the maximum entropy in training.

Appendix D. Base flow simulation

The base flow corresponds to a steady equilibrium of the governing Navier–Stokes equations. This fixed point is unstable to infinitesimal perturbations, giving rise to vortex shedding. The base flow is obtained by simulating only half of the domain, as shown in figure 18, which prevents antisymmetric vortex shedding. The boundary conditions are consistent with (A1), while a symmetric boundary condition is applied on the bottom boundary (symmetry line) of the domain, i.e. on $y = 0$. The symmetric boundary condition is imposed as $v = 0$, $\partial u/\partial y = 0$ and $\partial p/\partial y = 0$.

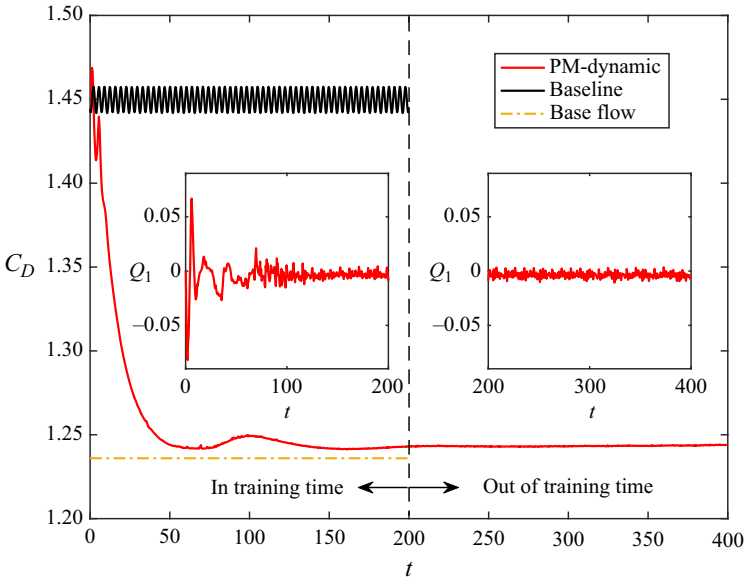


Figure 17. A long evaluation for 400 non-dimensional time units of the RL-trained dynamic controller in a PM environment. Control starts at $t = 0$. Solid curves show the controlled C_D using TQC and baseline without control. The dash-dotted curve corresponds to the base flow C_D. The mass flow rate Q_1 is presented for $t \in [0, 200]$ and $t \in [200, 400]$, respectively.

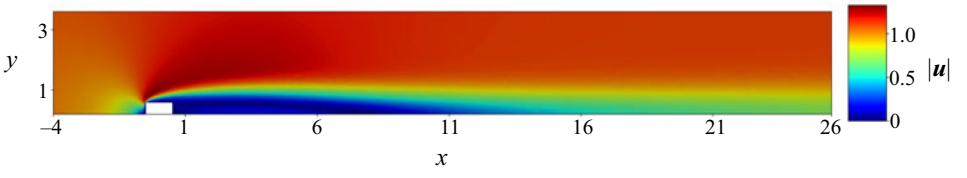


Figure 18. Base flow (steady flow without vortex shedding) obtained with a half-domain simulation, i.e. $y \in [0, 12.5]$. A sub-domain $y \in [0, 3.5]$, $x \in [-4, 26]$ is plotted for demonstration. The symmetric boundary condition is applied on the $y = 0$ boundary. The mesh of the simulation is consistent with figure 16.

In this case, the vortex shedding is not triggered, as shown in the contour of figure 18, and the only cause of the pressure drag is flow separation. Therefore, comparing the pressure drag in a full-domain simulation of uncontrolled flow, where the vortex shedding is triggered, with this base flow, the amount of pressure drag due to flow unsteadiness can be estimated. As only the unsteady component of pressure drag can be reduced effectively by flow control (Bergmann, Cordier & Brancher 2005), the control performance can be evaluated with respect to this base flow (3.2). The drag coefficient of the half square body measures $C_{Dh} = 0.618$, and the base flow drag coefficient of the whole body can be obtained as $C_{Db} = 2C_{Dh} = 1.236$.

References

Altman, E. & Nain, P. 1992 Closed-loop control with delayed information. *ACM Sigmetrics Perform. Eval. Rev.* **20** (1), 193–204.

Barros, D., Borée, J., Noack, B.R., Spohn, A. & Ruiz, T. 2016 Bluff body drag manipulation using pulsed jets and Coanda effect. *J. Fluid Mech.* **805**, 422–459.

BEAUDOIN, J.-F., CADOT, O., AIDER, J.-L. & WESFREID, J.E. 2006 Bluff-body drag reduction by extremum-seeking control. *J. Fluids Struct.* **22** (6–7), 973–978.

BERGMANN, M., CORDIER, L. & BRANCHER, J.-P. 2005 Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model. *Phys. Fluids* **17** (9), 097101.

BERTSEKAS, D. 2012 *Dynamic Programming and Optimal Control: Volume I*. Athena Scientific.

BERTSEKAS, D. 2019 *Reinforcement Learning and Optimal Control*. Athena Scientific.

BRACKSTON, R.D., GARCÍA DE LA CRUZ, J.M., WYNN, A., RIGAS, G. & MORRISON, J.F. 2016 Stochastic modelling and feedback control of bistability in a turbulent bluff body wake. *J. Fluid Mech.* **802**, 726–749.

BRACKSTON, R.D., WYNN, A. & MORRISON, J.F. 2018 Modelling and feedback control of vortex shedding for drag reduction of a turbulent bluff body wake. *Intl J. Heat Fluid Flow* **71**, 127–136.

BRIGHT, I., LIN, G. & KUTZ, J.N. 2013 Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements. *Phys. Fluids* **25** (12), 127102.

BRUNTON, S.L. & NOACK, B.R. 2015 Closed-loop turbulence control: progress and challenges. *Appl. Mech. Rev.* **67** (5), 050801.

BUCCI, M.A., SEMERARO, O., ALLAUZEN, A., WISNIEWSKI, G., CORDIER, L. & MATHELIN, L. 2019 Control of chaotic systems by deep reinforcement learning. *Proc. R. Soc.* A **475** (2231), 20190351.

CASSANDRA, A.R. 1998 A survey of POMDP applications. In *Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes*, vol. 1724. AAAI.

CHEN, W., WANG, Q., YAN, L., HU, G. & NOACK, B.R. 2023 Deep reinforcement learning-based active flow control of vortex-induced vibration of a square cylinder. *Phys. Fluids* **35** (5), 053610.

CHOI, H., LEE, J. & PARK, H. 2014 Aerodynamics of heavy vehicles. *Annu. Rev. Fluid Mech.* **46**, 441–468.

COBBE, K., HESSE, C., HILTON, J. & SCHULMAN, J. 2020 Leveraging procedural generation to benchmark reinforcement learning. In *International Conference on Machine Learning* (ed. H. Daumé III & A. Singh), pp. 2048–2056. PMLR.

CORKE, T.C., ENLOE, C.L. & WILKINSON, S.P. 2010 Dielectric barrier discharge plasma actuators for flow control. *Annu. Rev. Fluid Mech.* **42** (1), 505–529.

DABNEY, W., ROWLAND, M., BELLEMARE, M. & MUNOS, R. 2018 Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, article 353, pp. 2892–2901. AAAI.

DAHAN, J.A., MORGANS, A.S. & LARDEAU, S. 2012 Feedback control for form-drag reduction on a bluff body with a blunt trailing edge. *J. Fluid Mech.* **704**, 360–387.

DALLA LONGA, L., MORGANS, A.S. & DAHAN, J.A. 2017 Reducing the pressure drag of a D-shaped bluff body using linear feedback control. *Theor. Comput. Fluid Dyn.* **31**, 567–577.

DUAN, Y., CHEN, X., HOUTHOOFT, R., SCHULMAN, J. & ABBEEL, P. 2016 Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning* (ed. M.F. Balcan & K.Q. Weinberger), pp. 1329–1338. PMLR.

FAN, D., YANG, L., WANG, Z., TRIANTAFYLLOU, M.S. & KARNIADAKIS, G.E. 2020 Reinforcement learning for bluff body active flow control in experiments and simulations. *Proc. Natl Acad. Sci. USA* **117** (42), 26091–26098.

FUJIMOTO, S., HOOF, H. & MEGER, D. 2018 Addressing function approximation error in actor–critic methods. In *International Conference on Machine Learning* (ed. J. Dy & A. Krause), pp. 1587–1596. PMLR.

GARNIER, P., VIQUERAT, J., RABAULT, J., LARCHER, A., KUHNLE, A. & HACHEM, E. 2021 A review on deep reinforcement learning for fluid mechanics. *Comput. Fluids* **225**, 104973.

GERHARD, J., PASTOOR, M., KING, R., NOACK, B.R., DILLMANN, A., MORZYNSKI, M. & TADMOR, G. 2003 Model-based control of vortex shedding using low-dimensional Galerkin models. In *33rd AIAA Fluid Dynamics Conference*, p. 4262. AIAA.

GLEZER, A. & AMITAY, M. 2002 Synthetic jets. *Annu. Rev. Fluid Mech.* **34** (1), 503–529.

GODA, K. 1979 A multistep technique with implicit difference schemes for calculating two- or three-dimensional cavity flows. *J. Comput. Phys.* **30** (1), 76–95.

GUASTONI, L., RABAULT, J., SCHLATTER, P., AZIZPOUR, H. & VINUESA, R. 2023 Deep reinforcement learning for turbulent drag reduction in channel flows. *Eur. Phys. J.* E **46** (4), 27.

HAARNOJA, T., TANG, H., ABBEEL, P. & LEVINE, S. 2017 Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning* (ed. D. Precup & Y.W. Teh), pp. 1352–1361. PMLR.

HAARNOJA, T., ZHOU, A., ABBEEL, P. & LEVINE, S. 2018*a* Soft actor–critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870.

HAARNOJA, T., ZHOU, A., HARTIKAINEN, K., TUCKER, G., HA, S., TAN, J., KUMAR, V., ZHU, H., GUPTA, A. & ABBEEL, P. 2018*b* Soft actor–critic algorithms and applications. arXiv:1812.05905.

HASSELT, H. 2010 Double Q-learning. In *Advances in Neural Information Processing Systems 23* (ed. J. Lafferty, C. Williams. J. Shawe-Taylor, R. Zemel & A. Culotta). Curran Associates.

HENDERSON, P., ISLAM, R., BACHMAN, P., PINEAU, J., PRECUP, D. & MEGER, D. 2018 Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, article 392, pp. 3207–3214. AAAI.

HOU, Z.S. & XU, J.X. 2009 On data-driven control theory: the state of the art and perspective. Scopus. doi:10.3724/SP.J.1004.2009.00650.

ILLINGWORTH, S.J. 2016 Model-based control of vortex shedding at low Reynolds numbers. *Theor. Comput. Fluid Dyn.* **30** (5), 429–448.

JIN, B., ILLINGWORTH, S.J. & SANDBERG, R.D. 2020 Feedback control of vortex shedding using a resolvent-based modelling approach. *J. Fluid Mech.* **897**, A26.

KIRAN, B.R., SOBH, I., TALPAERT, V., MANNION, P., AL SALLAB, A.A., YOGAMANI, S. & PÉREZ, P. 2021 Deep reinforcement learning for autonomous driving: a survey. *IEEE Trans. Intell. Transp. Syst.* **23** (6), 4909–4926.

KOBER, J., BAGNELL, J.A. & PETERS, J. 2013 Reinforcement learning in robotics: a survey. *Intl J. Rob. Res.* **32** (11), 1238–1274.

KUZNETSOV, A., SHVECHIKOV, P., GRISHIN, A. & VETROV, D. 2020 Controlling overestimation bias with truncated mixture of continuous distributional quantile critics. In *International Conference on Machine Learning* (ed. H. Daumé III & A. Singh), pp. 5556–5566. PMLR.

LANSER, W.R., ROSS, J.C. & KAUFMAN, A.E. 1991 Aerodynamic performance of a drag reduction device on a full-scale tractor/trailer. *SAE Trans.* **100**, 2443–2451.

LI, J. & ZHANG, M. 2022 Reinforcement-learning-based control of confined cylinder wakes with stability analyses. *J. Fluid Mech.* **932**, A44.

LI, R., BARROS, D., BORÉE, J., CADOT, O., NOACK, B.R. & CORDIER, L. 2016 Feedback control of bimodal wake dynamics. *Exp. Fluids* **57** (10), 158.

LILLICRAP, T.P., HUNT, J.J., PRITZEL, A., HEESS, N., EREZ, T., TASSA, Y., SILVER, D. & WIERSTRA, D. 2015 Continuous control with deep reinforcement learning. arXiv:1509.02971.

LIN, J.C. 2002 Review of research on low-profile vortex generators to control boundary-layer separation. *Prog. Aerosp. Sci.* **38** (4), 389–420.

LOGG, A., WELLS, G.N. & HAKE, J. 2012 DOLFIN: a C++/Python finite element library. In *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book* (ed. A. Logg, K.-A. Mardal & G. Wells), pp. 173–225. Springer.

MAEI, H., SZEPESVARI, C., BHATNAGAR, S., PRECUP, D., SILVER, D. & SUTTON, R.S. 2009 Convergent temporal-difference learning with arbitrary smooth function approximation. In *Advances in Neural Information Processing Systems 22* (ed. Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams & A. Culotta). Curran Associates.

MNIH, V., BADIA, A.P., MIRZA, M., GRAVES, A., LILLICRAP, T., HARLEY, T., SILVER, D. & KAVUKCUOGLU, K. 2016 Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning* (ed. M.F. Balcan & K.Q. Weinberger), pp. 1928–1937. PMLR.

MNIH, V., KAVUKCUOGLU, K., SILVER, D., RUSU, A.A., VENESS, J., BELLEMARE, M.G., GRAVES, A., RIEDMILLER, M., FIDJELAND, A.K. & OSTROVSKI, G. 2015 Human-level control through deep reinforcement learning. *Nature* **518** (7540), 529–533.

PARIS, R., BENEDDINE, S. & DANDOIS, J. 2021 Robust flow control and optimal sensor placement using deep reinforcement learning. *J. Fluid Mech.* **913**, A25.

PARIS, R., BENEDDINE, S. & DANDOIS, J. 2023 Reinforcement-learning-based actuator selection method for active flow control. *J. Fluid Mech.* **955**, A8.

PASTOOR, M., HENNING, L., NOACK, B.R., KING, R. & TADMOR, G. 2008 Feedback shear layer control for bluff body drag reduction. *J. Fluid Mech.* **608**, 161–196.

PASZKE, A., *et al.* 2019 Pytorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32* (ed. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox & R. Garnett). Curran Associates.

PINO, F., SCHENA, L., RABAULT, J. & MENDEZ, M.A. 2023 Comparative analysis of machine learning methods for active flow control. *J. Fluid Mech.* **958**, A39.

PROTAS, B. 2004 Linear feedback stabilization of laminar vortex shedding based on a point vortex model. *Phys. Fluids* **16** (12), 4473–4488.

RABAULT, J., KUCHTA, M., JENSEN, A., REGLADE, U. & CERARDI, N. 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* **865**, 281–302.

RABAULT, J. & KUHNLE, A. 2019 Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Phys. Fluids* **31** (9), 094105.
RAFFIN, A., HILL, A., GLEAVE, A., KANERVISTO, A., ERNESTUS, M. & DORMANN, N. 2021 Stable-Baselines3: reliable reinforcement learning implementations. *J. Mach. Learn. Res.* **22** (268), 1–8.
REN, F., RABAULT, J. & TANG, H. 2021 Applying deep reinforcement learning to active flow control in weakly turbulent conditions. *Phys. Fluids* **33** (3), 037121.
SCHULMAN, J., LEVINE, S., ABBEEL, P., JORDAN, M. & MORITZ, P. 2015 Trust region policy optimization. In *International Conference on Machine Learning* (ed. F. Bach & D. Blei), pp. 1889–1897. PMLR.
SCHULMAN, J., WOLSKI, F., DHARIWAL, P., RADFORD, A. & KLIMOV, O. 2017 Proximal policy optimization algorithms. arXiv:1707.06347.
SHANNON, C.E. 1948 A mathematical theory of communication. *Bell Syst. Tech. J.* **27** (3), 379–423.
SILVER, D., LEVER, G., HEESS, N., DEGRIS, T., WIERSTRA, D. & RIEDMILLER, M. 2014 Deterministic policy gradient algorithms. In *International Conference on Machine Learning* (ed. E.P. Xing & T. Jebara), pp. 387–395. PMLR.
SINGH, S.P., JAAKKOLA, T. & JORDAN, M.I. 1994 Learning without state-estimation in partially observable Markovian decision processes. In *Machine Learning Proceedings 1994* (ed. W.W. Cohen & H. Hirsh), pp. 284–292. Morgan Kaufmann.
SONODA, T., LIU, Z., ITOH, T. & HASEGAWA, Y. 2023 Reinforcement learning of control strategies for reducing skin friction drag in a fully developed turbulent channel flow. *J. Fluid Mech.* **960**, A30.
SUDIN, M.N., ABDULLAH, M.A., SHAMSUDDIN, S.A., RAMLI, F.R. & TAHIR, M.M. 2014 Review of research on vehicles aerodynamic drag reduction methods. *Intl J. Mech. Mech. Engng* **14** (2), 37–47.
SUTTON, R.S. & BARTO, A.G. 2018 *Reinforcement Learning: An Introduction*. MIT Press.
SUTTON, R.S., MAEI, H.R., PRECUP, D., BHATNAGAR, S., SILVER, D., SZEPESVÁRI, C. & WIEWIORA, E. 2009 Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 993–1000. Association for Computing Machinery.
SUTTON, R.S., SZEPESVÁRI, C. & MAEI, H.R. 2008 A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. *Adv. Neural Inform. Proc. Syst.* **21** (21), 1609–1616.
TAKENS, F. 1981 Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*, pp. 366–381. Springer.
TANG, H., RABAULT, J., KUHNLE, A., WANG, Y. & WANG, T. 2020 Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning. *Phys. Fluids* **32** (5), 053605.
VARELA, P., SUÁREZ, P., ALCÁNTARA-ÁVILA, F., MIRÓ, A., RABAULT, J., FONT, B., GARCÍA-CUEVAS, L.M., LEHMKUHL, O. & VINUESA, R. 2022 Deep reinforcement learning for flow control exploits different physics for increasing Reynolds number regimes. *Actuators* **11**, 359.
VERMA, S., NOVATI, G. & KOUMOUTSAKOS, P. 2018 Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proc. Natl Acad. Sci. USA* **115** (23), 5849–5854.
VIGNON, C., RABAULT, J. & VINUESA, R. 2023 Recent advances in applying deep reinforcement learning for flow control: perspectives and future directions. *Phys. Fluids* **35** (3), 031301.
WANG, Q., YAN, L., HU, G., CHEN, W., RABAULT, J. & NOACK, B.R. 2023 Dynamic feature-based deep reinforcement learning for flow control of circular cylinder with sparse surface pressure sensing. arXiv:2307.01995.
WANG, Z., BAPST, V., HEESS, N., MNIH, V., MUNOS, R., KAVUKCUOGLU, K. & DE FREITAS, N. 2016 Sample efficient actor–critic with experience replay. arXiv:1611.01224.
WHITE, C.C. & SCHERER, W.T. 1994 Finite-memory suboptimal design for partially observed Markov decision processes. *Oper. Res.* **42** (3), 439–455.
XU, D. & ZHANG, M. 2023 Reinforcement-learning-based control of convectively unstable flows. *J. Fluid Mech.* **954**, A37.
YU, H. & BERTSEKAS, D.P. 2008 On near optimality of the set of finite-state controllers for average cost POMDP. *Math. Oper. Res.* **33** (1), 1–11.
ZENG, K. & GRAHAM, M.D. 2021 Symmetry reduction for deep reinforcement learning active control of chaotic spatiotemporal dynamics. *Phys. Rev.* E **104** (1), 014210.
ZIEBART, B.D. 2010 Modeling purposeful adaptive behavior with the principle of maximum causal entropy. Thesis, Carnegie Mellon University, Pittsburgh, PA.
ZIEBART, B.D., MAAS, A., BAGNELL, J.A. & DEY, A.K. 2008 Maximum entropy inverse reinforcement learning. In *Association for the Advancement of Artificial Intelligence*, pp. 1433–1438. AAAI.