Proceedings of the 1st international Symposium on AI and Fluid Mechanics
Paper No S1 P1

Active flow control using Neuroevolution guided Deep Reinforcement Learning

T Singh[1], **L Cordier**[1], and **R Fablet**[2]

[1] Institut Pprime, CNRS – Université de Poitiers – ISAE-ENSMA, UPR 3346, 86962 Futuroscope, France.
[2] IMT Atlantique, UMR CNRS Lab-STICC, Brest, France.

Keywords: Active Flow Control (AFC), Deep Reinforcement Learning (DRL), Evolutionary RL (ERL).

Abstract. This study investigates model-free Deep Reinforcement Learning (DRL) for active flow control applications. While recent proof-of-concept studies underscore the potential advantages of using DRL for flow control, key challenges such as poor sample efficiency, limited exploration, learning instability, and partial observability continue to limit the application of DRL in practical flow control applications. In order to address these challenges, this study explores hybrid/evolutionary reinforcement learning approaches, specifically the Collaborative Evolutionary Reinforcement Learning (CERL) framework, which combines neuroevolution with policy-gradient methods. To benchmark the performance and highlight fundamental differences in learning, the widely-used Proximal Policy Optimization (PPO) algorithm is used as a systematic baseline for comparison. The training efficiency and control performance of these learning approaches is evaluated on canonical flow control problems, involving drag reduction in case of rotary cylinders, with one and three-cylinder (fluidic pinball) configurations. We consider test cases with increasing levels of complexity, across a range of flow dynamics, ranging from periodic to chaotic. The results suggest that with proper environment definition and exploration mechanisms in place, model-free DRL, which inherits the efficiency and flexibility of a nonlinear controller, can be effectively used to determine flow control policies that outperform known optimal control strategies. When the control problem is sufficiently complex, using hybrid learning offers a significantly better alternative to on-policy DRL approach, tackling several existing challenges concurrently and optimising policy search for flow control applications.

1 Introduction

Engineering fluid flows are high-dimensional, nonlinear, and often chaotic, posing significant challenges for designing efficient model-based flow control strategies. Deep Neural Networks (DNNs) offer a promising solution due to their potential of serving as nonlinear controllers by mapping high-dimensional states to optimal actions. Driven by the computational and algorithmic advancements in recent years, Deep Reinforcement Learning (DRL) has emerged as an attractive framework for optimising these mappings through feedback-based learning, eliminating the need for explicit models. The application of DRL for controlling fluid flows, though recent, is a promising and rapidly growing area of research [1, 2, 3]. While demonstrating the feasibility and potential of using DRL for flow control, these studies highlight several obstacles that need to be addressed, in order to tackle more complex problems and eventually transition towards practical flow control applications. One of the major challenges is posed by the large sample requirement, generally associated with model-free DRL algorithms, which exclusively rely on iterative sampling in order to learn. Since obtaining samples remains costly, both in experimental and especially in numerical fluid mechanics setups, this large sample requirement remains a major bottleneck

for practical flow control applications. In addition, this training sample requirement usually scales up with the flow complexity, as reported by the handful of studies performed for relatively complex flows [4, 5]. Another challenge is DRL's limited exploration and sensitivity to hyperparameters [6]. Unlike standard machine learning, RL involves interactive learning, making optimization more complex and often unstable. In the context of flow control applications, most studies have predominantly employed on-policy learning algorithms, specifically Proximal Policy Optimization [PPO; 7] algorithm. While the rationale behind this preference is often unaddressed [1], this trend is likely driven by the availability of open-source implementations and PPO's relative stability during training. However, this stability comes at the expense of large sample requirement, and restrictive exploration of the state-action space. Therefore, alternative optimisation algorithms remain crucial but unexplored for flow control applications. Off-policy learning algorithms, with the ability to learn from a diverse set of experiences and reuse past experiences using experience replay, provides an attractive alternative for more efficient learning. However, off-policy learning algorithms are notoriously unstable, and often require tedious hyperparameter tuning and/or active management of the replay buffer to maintain learning stability [8]. Given these challenges, alternative learning approaches that can combine the sample efficiency and exploration properties of off-policy learning with the stability of on-policy learning are ideal. As the flow complexity increases, the aforementioned challenges become even more pronounced, posing significant difficulties in scaling DRL for practical flow control applications. This work explores the potential of hybrid or evolutionary reinforcement learning algorithms, which combine neuroevolution with standard gradient-based methods for DRL, exploiting the complementary advantages of evolutionary algorithms and gradient-based RL, while overcoming individual weaknesses.

2 Methodology

2.1 Flow configuration and numerical methods

We consider a canonical flow control configuration, i.e., the flow over a 2D cylinder with rotary oscillations, as depicted in figure 1. The geometry and boundary conditions (except the cylinder actuation) are similar as in Ren et al. [4], with a non-dimensional cylinder diameter, $D = 1$, and parabolic inlet velocity profile such that mean velocity magnitude, $u_{\text{mean}} = 1$. The flow solution is obtained by Direct Numerical Simulation (DNS). The spatial discretization is performed using the finite element based FEniCS package [9], employing an unstructured grid. The flow solution is calculated by the Incremental Pressure Correction Scheme (IPCS). For more details on the numerical scheme, refer to Appendix B in Rabault et al. [10].

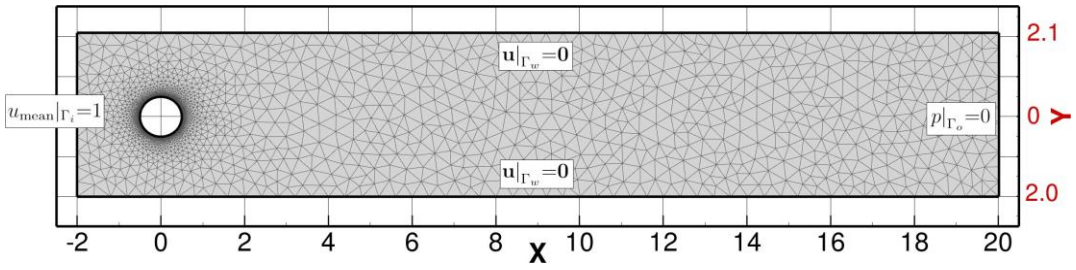
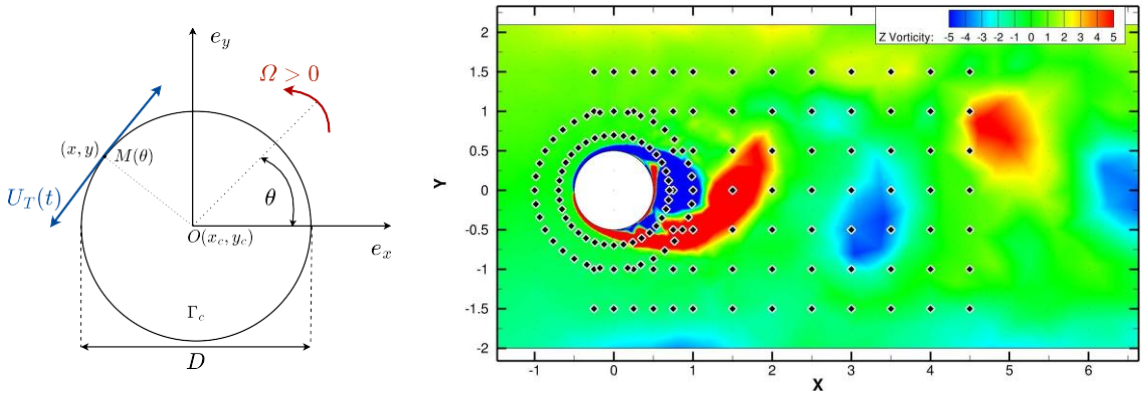


Figure 1: Computational domain for the cylinder wake configuration. The inlet, outlet and wall boundaries are denoted by Γ_i, Γ_o, Γ_w, respectively. The boundary conditions, similar to Schäfer et al. [11], are shown in the figure.

The rotary actuation is introduced by the non-dimensional instantaneous rotation amplitude $\Omega(t)$, defined as:



(a) (b)

Figure 2: (a) Schematic of the cylinder undergoing rotary oscillations. (b) Probe locations used for observing the flow field, marked by black squares. The contours depict instantaneous vorticity for the uncontrolled state at Re = 1000.

$$\Omega(t) = \frac{U_T(t)}{\overline{U}} = \frac{\dot{\theta}(t) \times D}{2\overline{U}} \tag{1}$$

where $U_T(t)$ is the tangential velocity applied to the cylinder boundary and $\dot{\theta}(t)$ is the instantaneous rotation rate. Note that positive actuation ($\Omega > 0$) corresponds to counter-clockwise cylinder rotation, as shown in figure 2a.

The uncontrolled or baseline flow, *i.e.*, flow over a stationary cylinder, becomes unstable at Re≈ 47, as a consequence of a Hopf bifurcation [12]. The flow ceases to be stationary and vortex shedding appears in the wake. This shedding pattern, popularly known as Kármán vortex street, intensifies with increasing Re, transitioning from exhibiting periodic to chaotic dynamics, as the Reynolds number is increased. As a result, Re can be considered as a proxy for the complexity of the control problem. Therefore, in order to cover sufficient range of problem complexity, we study Re $\in \{100, 200, 300, 1000\}$.

2.2 Learning algorithm

2.2.1 Collaborative Evolutionary Reinforcement Learning

Collaborative Evolutionary Reinforcement Learning (CERL) is a hybrid learning algorithm, proposed by Khadka et al. [13]. Broadly speaking, the framework binds collaborative, off-policy learning (Policy Gradient learners) with neuroevolution (NE) in order to concurrently tackle most of the challenges discussed in section 1. The schematic of the CERL algorithm is depicted in figure 3.

2.3 Learning agent and environment description

The agent is a feed forward neural network with two hidden layers of size 512 neurons each. The agent interacts with the environment in discrete time steps $t \in [0, n_a]$ and observes the environment state using 151 pressure probes, as shown in figure 2b. The probe locations are similar as in Ren et al. [4]. Based on these input observations or state (s_t), the agent outputs an action (a_t) using its policy $\pi(s_t) = a_t$. The action proposed is directly equal to the control parameter Ω and is continuous in the range $[-1, 1]$. In response to this action, considering that the control objective is drag reduction, the reward obtained by the agent at each DRL time step (t) is defined as:

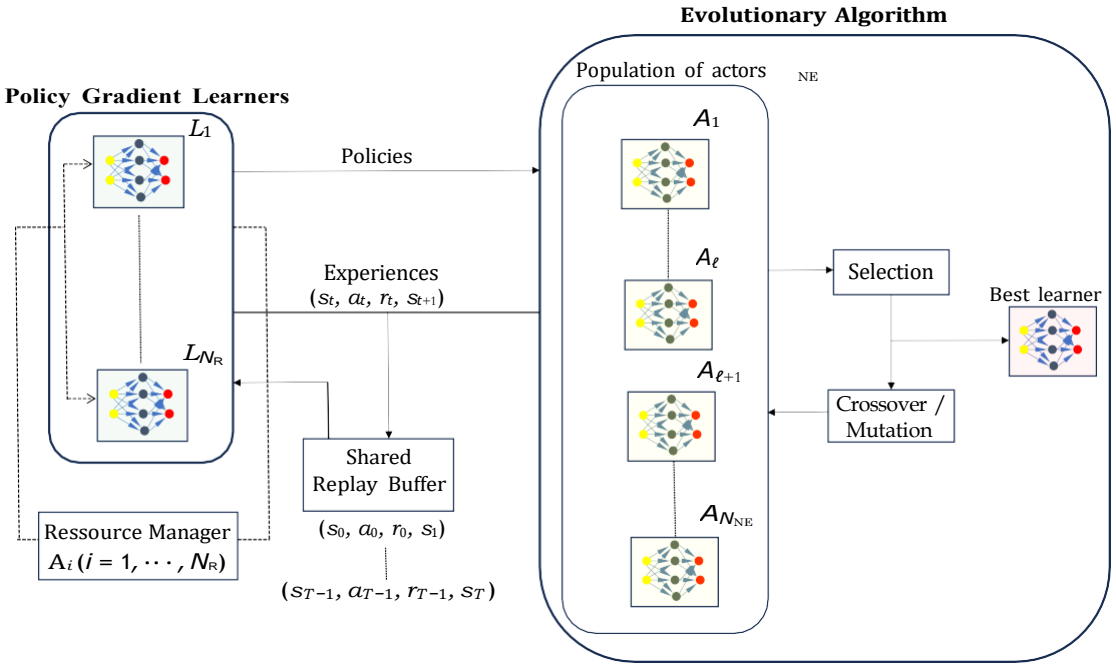


Figure 3: Schematic of the CERL framework. Two optimisation mechanisms, one based on policy gradient learning, the other on an evolutionary algorithm, work in collaborative policy search. The population P_{PG}, of dimension N_{PG}, is optimised by an off-policy algorithm, each policy being able to use a different algorithm and/or different hyperparameters. A resource manager manages the available computing resources by allocating the N_R possible CPU workers to the most promising learners. The evolutionary population P_{NE}, of dimension N_{NE}, evolves according to standard evolutionary operators (selection, mutation, crossover). The weakest individuals in the evolutionary population are regularly replaced by individuals from the gradient-based population. Figure adapted from Khadka et al. [13].

$$r^t = C_{D_0} - \langle C_D \rangle_{T_a} - \alpha_L \langle C_L \rangle_{T_a} \tag{2}$$

where C_{D_0} denotes the mean drag coefficient of the uncontrolled system, $\langle . \rangle_{T_a}$ represents the average over the action duration and α_L is the lift penalty coefficient. These parameters change with the considered Re and are as specified in table 1.

The environments were partially adapted from the publicly available code of Holm [14], which is in turn an extension of the original implementation by Rabault et al. [10]. The choices made when defining the environment are crucial and in many ways similar to defining the control problem. Separate environments were created for each value of Reynolds number considered for control. While the structure of each environment is kept similar, the parameters used within the environments vary, and are summarized in table 1. Detailed parametric study was performed to determine these parameters. Here, we omit the details of the parametric study to maintain brevity. However, we note that for the high Reynolds number flow, shorter episode length (T_e) can be used for improving training efficiency. Two factors likely contribute to the effectiveness of shorter episodes at higher Reynolds numbers. First, the agent can act at a higher frequency since the flow convects relatively faster, which can provide the agent to extract enough experience from shorter episode lengths. Second, for more complex flows, relatively shorter time series might already cover a large portion of the system dynamics [15].

Reynolds number, Re	100	200	300	1000
Drag coefficient (baseline), C_{D_0}	3.20	3.14	3.15	3.52
Strouhal number (baseline), St_n	0.300	0.317	0.333	0.350
Shedding period (baseline), $T_s = 1/\mathrm{St}_n$	3.33	3.15	3.00	2.86
Action duration, T_a	0.25	0.25	0.25	0.20
Episode length, T_e	20.0	20.0	20.0	10.0
Actions per episode, $N_{a/e} = T_e/T_a$	80	80	80	50
Actions per shedding period (T_s/T_a)	≈ 13	≈ 13	12	≈ 14
Ratio T_e/T_s	≈ 6	≈ 6.3	≈ 6.7	≈ 3.5
Numerical time step, δt $(\times 10^{-3})$	5.00	2.00	1.00	0.25
Number of time steps per action, $N_{t/a} = T_a/\delta t$	50	125	250	800
Lift penalty coefficient, α_L	0.20	0.20	0.20	0.20

Table 1: Parameters used to define the learning environment(s) for the single cylinder test case. All specified values are non-dimensional. C_{D_0} corresponds to the mean drag value for the uncontrolled flow. All variables being without dimension, the natural shedding frequency (f_{s_n}) is equal to the natural Strouhal number (St_n).

3 Results

This section presents some of the results obtained using the CERL and PPO agents for learning optimal control policies for the cylinder wake flow configuration. While studies were conducted across different Reynolds numbers, only Re = 1000 results are presented here for brevity. In order to highlight the advantages and limitations of each learning approach, the results are presented in a comparative way. The training samples, *i.e.* the number of agent-environment interactions, hereby denoted as S, acts as a common metric for comparing training performance. Unless stated otherwise, the range of actuation (action) is capped to $\Omega \in [-1, 1]$, *i.e.* $\Omega_{\max} = 1$. The aim is to see whether learning remains effective as complexity increases, focusing on the control performance and training efficiency.

The uncontrolled flow at Re = 1000 can be categorized as weakly turbulent, providing a challenging control problem. Figure 4 depicts the natural and controlled state obtained by the PPO and CERL agent. Even though the control policy learnt by the PPO agent manages to reduce the mean drag coefficient by 31.82%, fluctuations in drag are noticeable. Concerning the lift, there are asymmetric fluctuations of large magnitude, that sometimes surpass even the baseline case. In comparison, the control policy learnt by the CERL agent not only achieves a 45.56% mean drag reduction, but drastically reduces the lift fluctuations, while maintaining symmetry, leading to a notable stabilisation of the wake. It is evident from figure 4, that the actuation magnitude and frequency for both the control policies are fundamentally different. The actuation observed in the PPO control are lower than that in the CERL control, both in magnitude and frequency. In order to better explain and understand these differences, we compare our results with detailed parametric studies by Lu and Sato [16] and Cheng et al. [17]. These studies are largely similar to ours, except they consider a larger domain size and the forcing is sinusoidal, *i.e.*, of the form $U_T(t) = \Omega_c \sin(2\pi f_c t)$. Despite these differences, comparisons can be drawn in the frequency domain. Frequency analysis reported in figure 4 show that in the PPO control, there is a resonance or *"lock-in"* where the cylinder oscillation frequency (f_c) matches the vortex shedding frequency (f_{s_c}). In addition, the frequency ratio (f_c/f_{s_n}), where f_{s_n} is the natural shedding frequency was found to be equal to 1. In Lu and Sato [16] and Cheng et al. [17], large fluctuations in lift and drag were observed, when the frequency ratio f_c/f_{s_n} was close or equal to 1. Moreover, these fluctuations tends to increase as the value

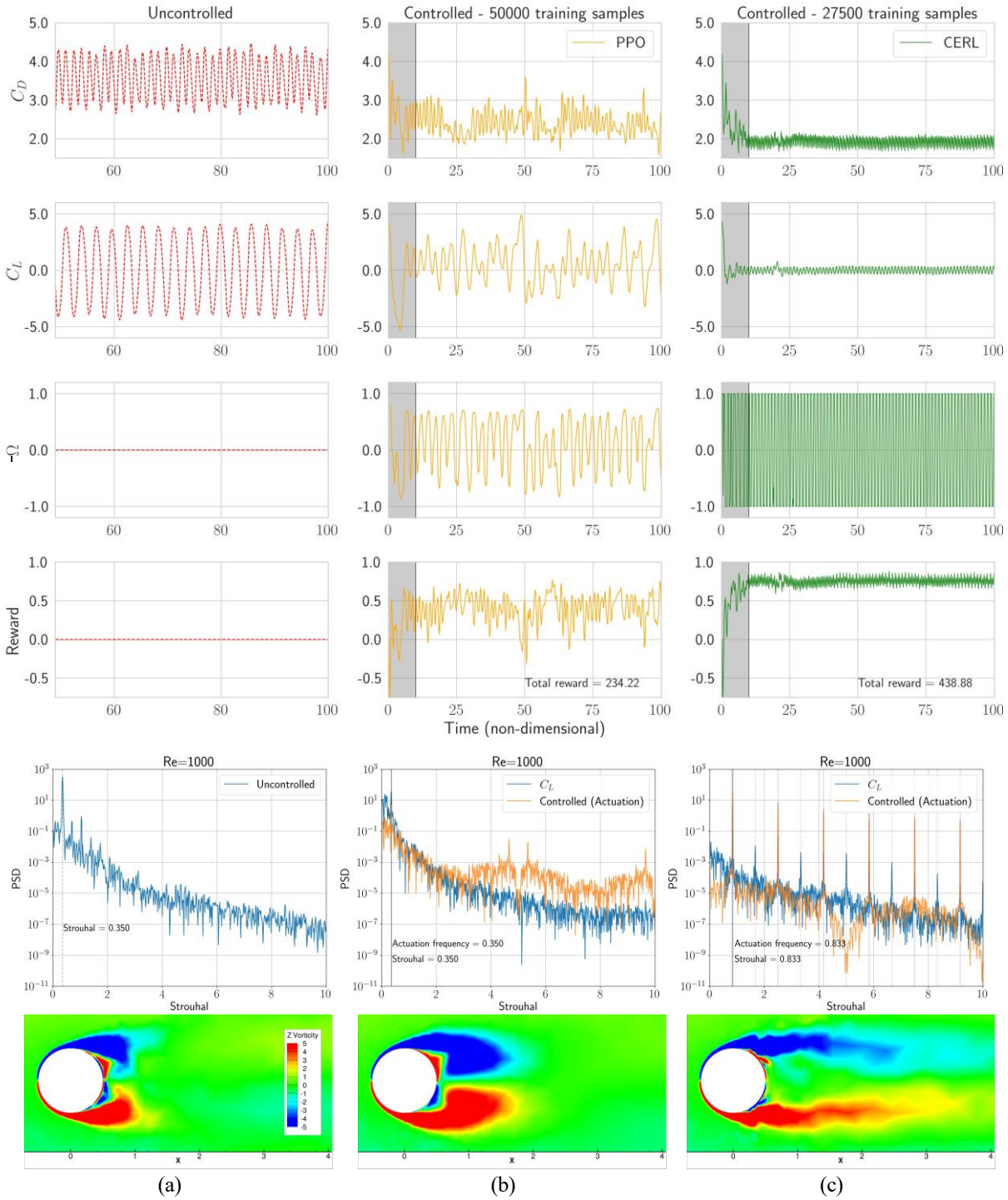


Figure 4: Uncontrolled state (a) and controlled state obtained using the PPO agent (b) and CERL agent (c), respectively, for the Re = 1000 case ($\Omega_{\max} = 1.0$). The control starts from $t = 0$ and is applied for 10 episode length. From top to bottom: drag coefficient (C_D), lift coefficient (C_L), rotation amplitude (Ω), Reward, PSD plot and time-averaged vorticity contours. The frequency ratio (f_c / f_{s_n}) is equal to 1 and 2.38 for the PPO and CERL control, respectively.

Case	$\Omega_{\max}$	S	$\overline{C_D}$	Drag reduction (%)	$\overline{r_t}$	PSR
Baseline	-	-	3.52	-	-	-
Controlled - PPO	1.0	50000	2.40	31.82	0.409	1.33
Controlled - CERL	1.0	27500	1.92	45.56	0.758	0.25

Table 2: Control performance, Re = 1000. S denotes the training samples required and $\overline{r_t}$ is the average reward gathered by the agent at each time step.

of Ω_c increases. This might explain why the control in the PPO case does not reach the actuation limits, since that would be sub-optimal, unless the frequency ratio is increased. We argue that discovering this requires diligent exploration on the part of the agent, which seems to be lacking in the case of PPO. In addition, as reported in Lu and Sato [16] and Cheng et al. [17], irrespective of the value of Ω_c, when the forcing is sinusoidal and frequency ratio $f_c/\ f_{s_n}$ is 1, the mean drag for the controlled state remains greater than that of the uncontrolled state. In contrast, despite similar fluctuations, a significant mean drag reduction is obtained by the policy learnt by the PPO agent. This can be directly attributed to the superiority and flexibility of using a feedback-based, nonlinear controller. Taken together, these results indicate that the PPO control converges towards a solution that might be considered locally optimal but globally sub-optimal. In the CERL control (figure 4), we observe similar lock-in phenomenon but in this case, the high frequency actuations increase the frequency ratio $f_c/\ f_{s_n}$ to a value of 2.38. This combination of frequency mismatch between f_c and f_{s_n}, combined with high amplitude oscillations, was found to be the optimal strategy for drag reduction, up to a certain optimal frequency ratio [17]. Through a detailed parametric study, the authors found that when the forcing is sinusoidal, an optimal frequency ratio of 4 and $\Omega_c = 3$ should achieve a mean drag reduction of 40%. For comparison, the CERL's control policy manages to achieve a drag reduction of around 45% with a frequency ratio of around 2.38 and $\Omega_{\max} = 1$, which is not only better but also much more efficient in terms of actuation power required. These results also implicitly show the advantage of using a nonlinear controller over a simpler form of forcing represented as a sinusoidal function. Note that the agents require 50000 and 27500 training samples to learn the PPO and CERL policies, shown respectively in column (b) and (c) of figure 4. The latter is not only 45% more sample efficient but due to its superior exploration properties, achieves twice as much reward as the former.

Figure 5 shows the vortex shedding sequence for the controlled flow, obtained by the CERL agent. It can be seen that the control transforms the chaotic vortex shedding of the uncontrolled system into a highly periodic one, with a forcing period, $T_c = 1.2$. We deduce from the observation of the shedding sequence that the controller maintains a cylinder rotation that is similar (co-rotating) to the vortex being formed. This rotation creates a circulation that is opposite to the forming vortex, inhibiting the shear layer interaction, and thereby, the strength of the vortex being formed. At the same time, the rotation encourages the formation of a vortex in the opposite shear layer, effectively helping synchronize the shedding of vortices in the wake. This physical mechanism, termed as *"vorticity control"* has been reported in multiple studies using open-loop [18, 19] and closed-loop [20] control strategies, where the authors observed a similar "destructive" cylinder rotation, when globally optimal drag reduction parameters were selected. Interestingly, this control mechanism, discovered by the CERL agent through evolution guided reinforcement learning, is also observed in marine animals for drag reduction and maximum propulsion efficiency [21, 22]. It is satisfying to see that combining ideas from natural evolution and biology, even though in highly simplified forms, leads us back to solutions perfected by nature over the course of billions of years.

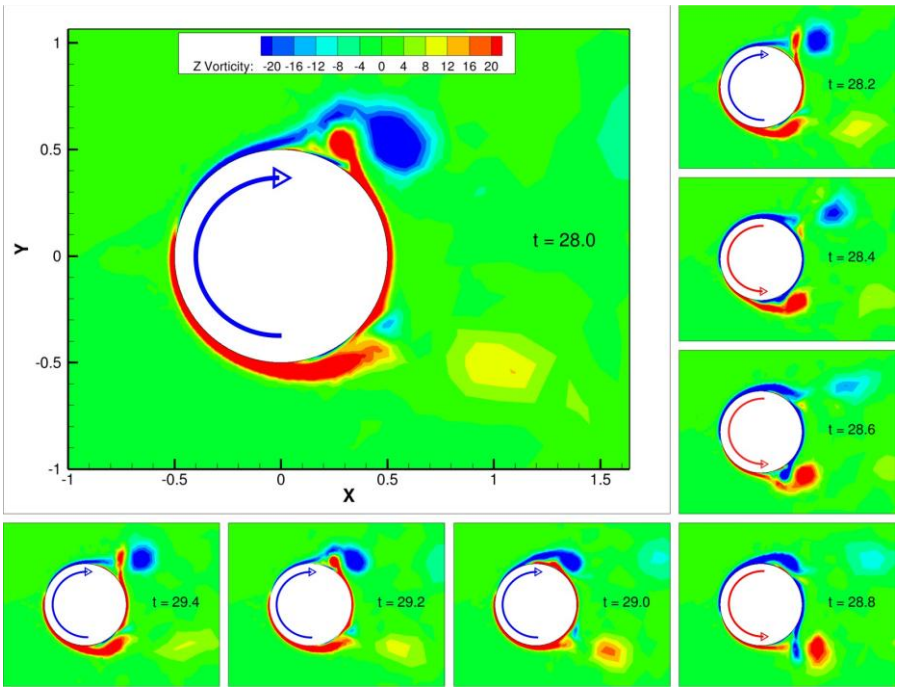
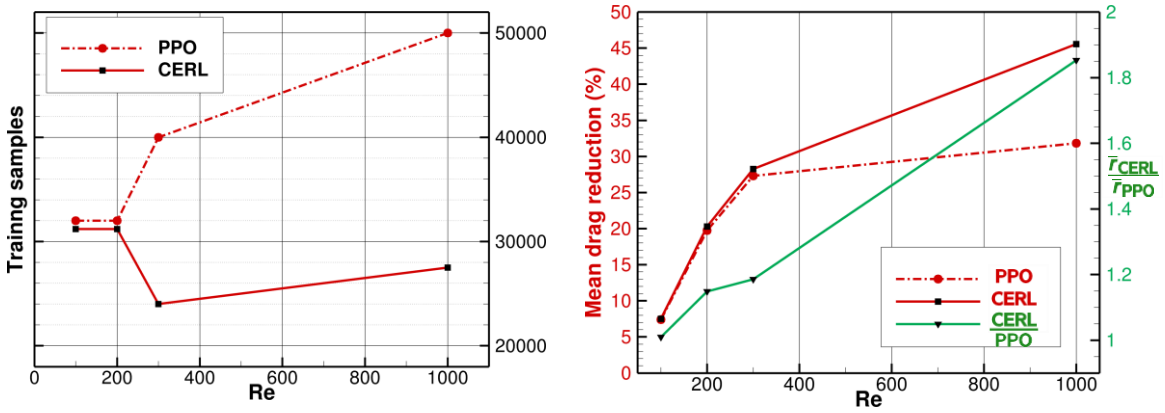


Figure 5: Vortex shedding sequence for the controlled case. The forcing period, $T_f = 1.2$ time units and the sequence has a Δt of $T_f/6$. The arrows denote the cylinder rotation, red being positive (anti-clockwise).

4 Conclusions

Figure 6 summarizes the findings, comparing the average performance of the two learning agents, for the different test cases considered for control. While both the DRL agents were able to achieve the control objective (drag reduction), the relative performance was found to be largely a function of the problem complexity. For the test cases with very low Reynolds number (Re = 100), no significant advantages were observed when using CERL for policy search. The performance of the best CERL agent was found to be similar to the corresponding PPO agent, both in terms of sample requirement and the average performance of the learnt control policies. This observation can possibly be attributed to the relatively lower problem complexity. In such cases, where the flow dynamics are more predictable, the range of actions yielding high reward is narrow, and the reward variability is low. Consequently, the need for extensive exploration is minimized and the key exploration capabilities of CERL provide no significant advantage over the PPO-based learning agent.

The benefits of using hybrid learning approach starts becoming more apparent as the problem complexity is increased. As the Reynolds number increases, the flow becomes more unstable and eventually chaotic. This complexity increases the variability in the reward landscape, making the problem harder to solve. In addition, as the flow dynamics becomes more nonlinear and stochastic, the sensitivity to control actions increases. This combinatorial growth of state-action possibilities makes thorough exploration even more critical to cover the relevant parts of the search space. For instance, considering the weakly turbulent test case at Re = 1000, the best performing CERL agent not only achieves nearly twice as much reward, but requires almost 50% less training samples to train, when compared to the corresponding PPO agent. In addition, the CERL agent's control policy learns to exploit vorticity control with frequency mismatch, but possibly at a lower actuation cost than existing linear optimal control strategies, highlighting benefits of using ANNs as a nonlinear controller. As the role of exploration increases with



(a) (b)

Figure 6: (a) Training sample requirement and (b) corresponding control performance (mean reward and drag reduction), as a function of the Reynolds number. The best-performing CERL and PPO agent are denoted by the solid and dashed lines, respectively. The reward curve in (b) is normalized by the mean reward obtained by the PPO agent ($\bar{r}_{\text{PPO}}$).

the problem complexity, the best-performing CERL agent not only outperforms the corresponding PPO agent, but the performance gains also consistently improves as a function of flow Re (see figure 6b).

Implementation details and results from a follow-up study on controlling the three-cylinder (fluidic pinball) flow configuration, which introduces greater complexity with chaotic flow and limited observation probes, as well as the effects of memory, will be presented in a full-length article to follow.

References

[1] Jonathan Viquerat, Philippe Meliga, A Larcher, and E Hachem. A review on deep reinforcement learning for fluid mechanics: An update. *Physics of Fluids*, 34(11), 2022.

[2] Fangfang Xie, Changdong Zheng, Tingwei Ji, Xinshuai Zhang, Ran Bi, Hongjie Zhou, and Yao Zheng. Deep reinforcement learning: A new beacon for intelligent active flow control. *Aerospace Research Communications*, 1:11130, 2023.

[3] Colin Vignon, Jean Rabault, and Ricardo Vinuesa. Recent advances in applying deep reinforcement learning for flow control: Perspectives and future directions. *Physics of fluids*, 35(3), 2023.

[4] Feng Ren, Jean Rabault, and Hui Tang. Applying deep reinforcement learning to active flow control in weakly turbulent conditions. *Physics of Fluids*, 33(3), 2021.

[5] Pau Varela, Pol Suárez, Francisco Alcántara-Ávila, Arnau Miró, Jean Rabault, Bernat Font, Luis Miguel García-Cuevas, Oriol Lehmkuhl, and Ricardo Vinuesa. Deep reinforcement learning for flow control exploits different physics for increasing reynolds number regimes. *Actuators*, 11(12), 2022. ISSN 2076-0825. doi: 10.3390/act11120359. URL https://www.mdpi.com/2076-0825/11/12/359.

[6] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

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

[8] Guido Novati and Petros Koumoutsakos. Remember and forget for experience replay. In *International Conference on Machine Learning*, pages 4851–4860. PMLR, 2019.

[9] Anders Logg, Kent-Andre Mardal, Garth N. Wells, et al. *Automated Solution of Differential Equations by the Finite Element Method*. Springer, 2012. doi: 10.1007/978-3-642-23099-8.

[10] Jean Rabault, Miroslav Kuchta, Atle Jensen, Ulysse Réglade, and Nicolas Cerardi. Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *Journal of fluid mechanics*, 865:281–302, 2019.

[11] Michael Schäfer, Stefan Turek, Franz Durst, Egon Krause, and Rolf Rannacher. *Benchmark computations of laminar flow around a cylinder*. Springer, 1996.

[12] Charles HK Williamson. Vortex dynamics in the cylinder wake. *Annual Review of Fluid Mechanics*, 28(1):477–539, 1996.

[13] Shauharda Khadka, Somdeb Majumdar, Tarek Nassar, Zach Dwiel, Evren Tumer, Santiago Miret, Yinyin Liu, and Kagan Tumer. Collaborative Evolutionary Reinforcement Learning. In *International Conference on Machine Learning*, pages 3341–3350. PMLR, 2019. URL https://github.com/IntelAI/cerl.

[14] Marius Holm. Using deep reinforcement learning for active flow control. Master's thesis, University of Oslo, 2020.

[15] Katharina Bieker, Sebastian Peitz, Steven L Brunton, J Nathan Kutz, and Michael Dellnitz. Deep model predictive flow control with limited sensor data and online learning. *Theoretical and computational fluid dynamics*, 34(4):577–591, 2020.

[16] X-Y Lu and J Sato. A numerical study of flow past a rotationally oscillating circular cylinder. *Journal of Fluids and Structures*, 10(8):829–849, 1996.

[17] M Cheng, YT Chew, and SC Luo. Numerical investigation of a rotationally oscillating cylinder in mean flow. *Journal of Fluids and Structures*, 15(7):981–1007, 2001.

[18] B Thiria, S Goujon-Durand, and JE Wesfreid. The wake of a cylinder performing rotary oscillations. *Journal of Fluid Mechanics*, 560:123–147, 2006.

[19] M. Bergmann, L. Cordier, and J.-P. Brancher. On the generation of a reverse Von Kármán street for the controlled cylinder wake in the laminar regime. *Phys. Fluids*, 18(2):028101:1–4, 2006.

[20] Thibault LB Flinois and Tim Colonius. Optimal control of circular cylinder wakes using long control horizons. *Physics of Fluids*, 27(8), 2015.

[21] Michael S Triantafyllou, AH Techet, Q Zhu, DN Beal, FS Hover, and DKP Yue. Vorticity control in fish-like propulsion and maneuvering. *Integrative and comparative biology*, 42(5):1026–1031, 2002.

[22] Q Zhu, MJ Wolfgang, DKP Yue, and MS Triantafyllou. Three-dimensional flow structures and vorticity control in fish-like swimming. *Journal of Fluid Mechanics*, 468:1–28, 2002.