Reinforcement learning for bluff body active flow control in experiments and simulations

Dixia Fan[a,b,1], Liu Yang[c,1], Zhicheng Wang[c,1], Michael S. Triantafyllou[a,b,2], and George Em Karniadakis[c]

[a]Department of Mechanical Engineering, Massachusetts Institute Technology, Cambridge, MA 02139; [b]Sea Grant College Program, Massachusetts Institute of Technology, Cambridge, MA 02139; and [c]Division of Applied Mathematics, Brown University, Providence, RI 02912

Edited by Katepalli R. Sreenivasan, New York University, New York, NY, and approved August 31, 2020 (received for review March 15, 2020)

We have demonstrated the effectiveness of reinforcement learning (RL) in bluff body flow control problems both in experiments and simulations by automatically discovering active control strategies for drag reduction in turbulent flow. Specifically, we aimed to maximize the power gain efficiency by properly selecting the rotational speed of two small cylinders, located parallel to and downstream of the main cylinder. By properly defining rewards and designing noise reduction techniques, and after an automatic sequence of tens of towing experiments, the RL agent was shown to discover a control strategy that is comparable to the optimal strategy found through lengthy systematically planned control experiments. Subsequently, these results were verified by simulations that enabled us to gain insight into the physical mechanisms of the drag reduction process. While RL has been used effectively previously in idealized computer flow simulation studies, this study demonstrates its effectiveness in experimental fluid mechanics and verifies it by simulations, potentially paving the way for efficient exploration of additional active flow control strategies in other complex fluid mechanics applications.

reinforcement learning | experimental fluid mechanics | bluff body | drag reduction | accelerated discovery

The classical paradigm for designing fluid control strategies consists of a first stage devoted to exploring and understanding the physics of the problem over a wide parametric range, followed by careful modeling and developing specially designed control strategies to exploit the gained understanding, culminating in an optimal tuning of the control parameters (1). This process involves careful computational or experimental investigation, guided by intuition obtained through the investigation and resulting in heuristically derived control techniques. Hence, this procedure is typically quite slow to yield effective results. In recent years, machine learning has received increasing attention for fluid control problems (2) because it could provide a more efficient pathway to achieving effective solutions. For example, in complex flow problems without full understanding of the underlying physics, we could directly optimize the control strategy and hence reduce substantially human involvement in the modeling and design of the control strategies.

Among the machine-learning tools, reinforcement learning (RL) offers intriguing opportunities for quick progress, as it has demonstrated its potential for achieving "superhuman" performance in board games (3, 4) and a capability for tackling complex, high-dimensional continuous control tasks (5). Recent explorations of RL for computational fluid mechanics problems include bio-locomotion of single and multiple fishes (6, 7), motion and path planning for aerial/aquatic vehicles (8–10), active flow control for bluff bodies (11–13), and foil shape optimization (14).

To our best knowledge, RL applications in fluid problems are to date limited to computer simulations. Experimental testing requires different methodologies, due to limitations in sensing and actuation, noise in measured signals, and delay in data transmissions. Hence, the feasibility of RL in experimental fluid mechanics has remained an open question. To address this question, in this work we consider the problem of flow control in a bluff cylindrical body in cross-flow, using two small rotating control cylinders; this system was first introduced in ref. 15. We demonstrate that with a properly designed reward function and noise reduction methodology, the RL agent can discover a flow control strategy that is close to the optimal control found from a laborious nonautomated, systematic grid search. To maximize the system power gain efficiency, the RL agent can also find the optimal strategy to achieve the balance between drag reduction and power loss from friction. These are the two main experimental tasks we pursue in the current study, but in addition we verify our experimental findings with high-fidelity simulations emulating the experimental conditions, hence providing the missing link to the underlying modified flow patterns and associated physical mechanisms of drag reduction. Taken together, our results suggest that RL can be applied to other complex experimental fluid mechanics applications that may be difficult to tackle with classical methods.

Problem Description

Turbulent flow past bluff bodies is encountered in a wide range of natural and manmade systems that operate within a fluid. In particular, the flow past a bluff cylinder has been characterized as a "kaleidoscope" of interesting fluid mechanics phenomena (16). Over a wide range of Reynolds numbers, the flow around a cylinder is characterized by the formation of a downstream

Significance

Reinforcement learning (RL) has been applied effectively in games and robotic manipulation. We demonstrate the effectiveness of RL in experimental fluid mechanics by applying it to reduce the drag of circular cylinders in turbulent flow, a canonical fluid–structure interaction problem. Although physics agnostic, RL managed to reduce the drag by 30% or reach another specified optimum point very quickly. Following this discovery, we used high-fidelity simulations to probe the underlying physical mechanisms so that the discovered control techniques can be generalized to other similar flow problems. More broadly, RL-guided active control can lead to efficient exploration of additional flow-control strategies in experimental fluid mechanics, potentially paving the way for accelerating scientific discovery and different designs in flow-related engineering problems.

Author contributions: D.F., L.Y., M.S.T., and G.E.K. designed research; D.F., L.Y., and Z.W. performed research; D.F. and Z.W. contributed new reagents/analytic tools; D.F., L.Y., and Z.W. analyzed data; and D.F., L.Y., Z.W., M.S.T., and G.E.K. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1]D.F., L.Y., and Z.W. contributed equally to this work.

[2]To whom correspondence may be addressed. Email: mistetri@mit.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2004939117/-/DCSupplemental.

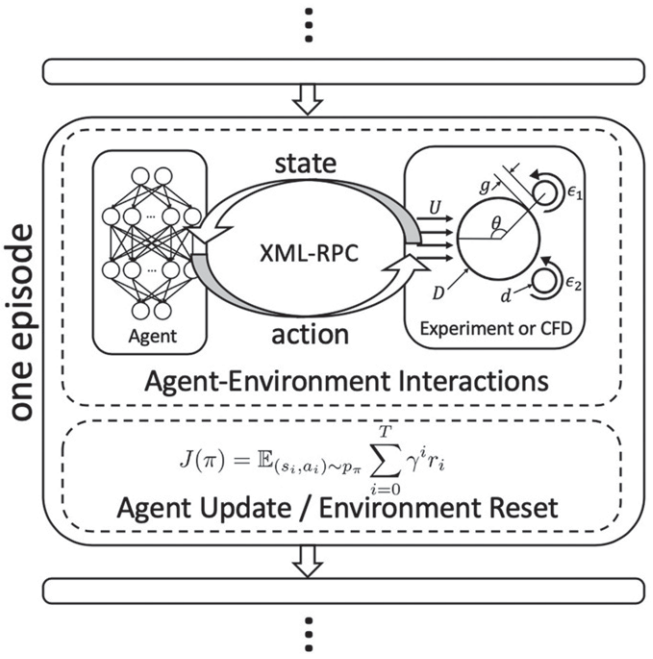

Fig. 1. Sketch of the reinforcement-learning process in both experimental and simulation environments for one episode. Generally, each episode consists of two stages (the dashed blocks): In the first stage the agent interacts with the experimental or simulation environment via the XML-RPC protocol at a fixed frequency. Each interaction consists of a state inquiry and an action decision; in the second stage the agent updates its policy based on the experience collected while waiting for the reset of the environment.

wake, resulting in a significant low pressure region in the rear of the cylinder and the periodical shedding of vortices due to a flow instability (17–21). As a result, a bluff body in cross-flow experiences a large mean plus unsteady drag force mainly due to pressure drag (22) and a large oscillatory lift force that may result in serious fatigue and subsequent structural damage (21). Therefore, bluff body flow control is a main research topic for fluid mechanics applications, using either passive or active control methods (23). To use active control to either alter the boundary layer or directly modify the wake, a flow control study typically requires first a good understanding of the flow physics, then a construction of a conceptual simplified model, and finally a manual tuning of the control outputs (24). This process can be quite lengthy and sometimes intractable for truly complex flow problems.

For these reasons, we selected a bluff body flow control problem to demonstrate the feasibility of applying RL to experimental fluid mechanics problems. The model, shown in Fig. 1, is chosen to use two fast-rotating smaller control cylinders to alter the flow pattern around a main cylinder placed upstream within a uniform flow, aiming to reduce the effective system drag or maximize the system power gain. Similar fluid problems have been studied both experimentally (15, 25) and computationally (26), investigating the effects of 1) the small to main cylinder diameter ratio d/D, 2) gap ratio g/D, 3) the smaller control cylinder configuration, and 4) the rotation rate $\epsilon = \frac{\omega d}{2U}$ on the fluid forces and flow patterns. Here, D is the diameter of the main cylinder, d is the diameter of the smaller control cylinder, g is the gap between the main and each of the smaller cylinders, ω is the rotation speed of the smaller cylinders, and U is the incoming velocity. Past results showed that the counterrotating cylinder pair could effectively reduce the main cylinder drag force as well as diminish the oscillatory lift force by suppressing the vortex shedding in the wake (15, 27). The physics behind this phenomenon are that when the control cylinders are placed at appropriate locations and rotate at a sufficiently fast speed, they are able to interact with the main cylinder separating boundary layers, causing them to reattach and form a narrower wake behind the cylinder, hence significantly reducing the pressure drag.

Technical Approach

Experimental Model and Procedure. The sketch in Fig. 1 outlines the experimental procedure and highlights one episode of the learning process, corresponding to one towing experiment lasting 40 s (the towing speed is 0.2 m/s and the entire towing length is 8 m). At the beginning of each experiment, the control cylinders are held still for 4 s to ensure a fully developed wake. Then, the RL agent starts interacting with the environment via a state inquiry and an action decision at 10 Hz ($\delta t = 0.1$ s). The states in the current experiment are the drag and lift coefficients C_d and C_l on the three cylinders altogether, which are calculated as

$$C_d = \frac{F_d}{0.5\rho U^2 DL}, C_l = \frac{F_l}{0.5\rho U^2 DL}, \quad [1]$$

where ρ is the fluid density, and F_d and F_l are the average drag and lift forces, respectively, over δt. After completing an experiment, the carriage is brought back to the starting point at a homing speed of 0.2 m/s. Then, the policy of the RL agent is updated based on the experience learned from all of the previous experiments up to that time, while the environment is reset and prepared for the next experiment. A 2-min pause is followed between towing experiments to avoid cross-contamination of results between successive experiments (28). The total wall clock time for one episode is 3 to 4 min.

The policy of the RL agent in the current work is updated only between experiments, instead of at every agent–environment interaction, to reduce delays due to the limitation of our hardware. The control and data collection interfaces are developed in $C\#$ language, and the RL agent is implemented in Python language based on the deep-learning package TensorFlow (29). The Extensible Markup Language-Remote Procedure Call (XML-RPC) protocol is then applied for data communication between the cross-language platforms, which allows us to take advantage of the machine-learning tools developed in Python, as well as the established experimental and computational platforms in other languages with minimum effort.

We present a description of the experimental hardware and a pseudocode of the RL procedure in *Materials and Methods*. Further details of the experiment setup and procedure are given in *SI Appendix*, section S1.

High-Fidelity Numerical Simulation. In addition to experiments, to explain the flow physics and visualize the wake patterns, we also apply the same RL algorithm to the entropy–viscosity-based large eddy simulation (LES) (30), which was implemented in the framework of the high-order spectral element method (31).

The computational domain has a size of $[-7.5\,D, 20\,D] \times [-10\,D, 10\,D] \times [0\,D, 4\,D]$ in the x, y, z direction, respectively. The details of the setup, including mesh resolution and boundary condition, can be found in *SI Appendix*, section S4. The Reynolds number is $Re_D = 10{,}160$, which is the same as that of the experiment. In the simulation, a dimensionless time step 2×10^{-4} is used, and the state inquiry and action decision are made every 600 time steps. It is worth noting that the RL-guided LES starts from the fully turbulent flow, which is the result of previous simulation of flow in the same geometry configuration with the small cylinders held still. The configuration parameters for the simulation are presented in Table 1.

Table 1. Experimental and simulation parameters

Parameter	Experiment	Simulation
D	5.08 cm	1
d/D	0.125	0.125
g/D	0.05	0.05
L/D	9	4
θ	$2\pi/3$	$2\pi/3$
Re_D	10,160	10,160
δt	0.1 s	—
$\delta tD/U$	0.395	0.12
ϵ^{max}	3.66	3.66

Results

Experimental Task One: Drag Reduction. We have tested three cases to demonstrate the importance of an appropriately designed RL reward and the application of a Kalman filter (KF) (32) for noise reduction when the agent inquires the states. The results of the $\overline{C_d}$ and $\overline{C_l}$ as well as $\overline{\epsilon_{1,2}}/\epsilon^{max}$ for the first 200 episodes are plotted in Fig. 2, while the setups of the reward and the filter in the three cases are as follows:

1) Case I: $r=-sgn(C_d)C_d^2-0.1{C_l}^2$ with KF;
2) Case II: $r=-sgn(C_d)C_d^2-0.1{C_l}^2$ without KF;
3) Case III: $r=-sgn(C_d)C_d^2$ with KF.

The purpose of testing these three cases is to demonstrate the importance of implementation of a real-time filter (case I vs. case II) and a properly designed reward (case I vs. case III) for the application of RL in fluid mechanics experiments. Compared to the reward of case III, we augment the reward of case I with a weighted squared lift coefficient and intend to inform the RL agent to reduce the oscillating lift force by minimizing the drag force through preventing the alternating shedding of vortices in the cylinder wake.

The result of case I is plotted in Fig. 2*A*; it shows that $\overline{C_d}$ drops quickly and converges to approximately $\overline{C_d^*}$ in about 10 episodes, i.e., about 0.5 h in wall clock time, where $\overline{C_d^*}$ is the minimum value found in the reference experiment of the control cylinders rotating at $\epsilon_2=-\epsilon_1=3.66$ (the result of the reference experiment can be found in *SI Appendix*, section S1, together with a comparison with the experimental result from ref. 25). The learning curve of actions shows that the agent learns to rotate the two cylinders in the opposite directions with near-maximum speeds. We observe that the $\overline{C_d}$ increases in the first few episodes before decreasing and converging to an asymptotic value, which is a result of the agent's random exploration in the early stage of learning.

The time traces of C_d and actions of the four different episodes in case I (highlighted with cross markers in Fig. 2*A*) are displayed in Fig. 2*D*. A comparison between the raw data (blue) and the filtered data (red) reveals that the KF manages to remove the high-frequency oscillations in C_d. The first episode in Fig. 2*D* 1) shows that when the learning process just begins, the RL agent fails to make any informed decision. In the fifth episode shown in Fig. 2*D*, 2) the RL agent explores the rotation of the first control cylinder at its maximum speed in the counterclockwise direction, which results in an increase of $\overline{C_d}$. After tens of policy updates, at the 50th episode shown in Fig. 2*D*, 3) when the active control is turned on, the RL agent manages to make the correct decision to rotate the control cylinders in the right direction and at the right speed and, therefore, reduce the $\overline{C_d}$. Comparing the actions of the 150th episode in Fig. 2*D* 4) to those of the 50th episode, we see that the actions are more stable with less variation. Additionally, two repeated experiments have been performed for case I, and the results can be found in *SI Appendix*, section S2.

The result of case II is plotted in Fig. 2*B* and shows that after 200 episodes, the RL agent fails to reduce the $\overline{C_d}$ as effectively as in case I where KF is employed. The learning curve of actions indicates that the RL agent is not able to learn an appropriate policy for the second rotational cylinder, resulting in large $\overline{C_d}$ and $\overline{C_l}$. The comparison between cases I and II clearly shows the necessity of noise reduction when applying RL techniques in experimental environments and real-world applications.

The results of case III are plotted in Fig. 2*C* and show that the $\overline{C_d}$ is reduced slowly over the number of episodes. Between the 55th and the 150th episodes, the $\overline{C_d}$ reaches a relatively constant value of about 0.83, which is higher than the $\overline{C_d^*}=0.72$, and the $\overline{C_l}$ is as large as 0.77. With the increase of episodes, we observe that around the 155th episode, the $\overline{C_d}$ drops suddenly and converges to the $\overline{C_d^*}$, while the magnitude of $\overline{C_l}$ decreases to a value close to zero.

To explain such a drastic change of the hydrodynamic coefficients at around the 155th episode, in Fig. 2 we visualize the policy evolution between the 151st and the 160th episodes: The RL agent policy is initially stuck at a local minimum but then manages to escape due to exploration. During the entire learning process, the values of C_d are mostly concentrated in the interval $[0.5, 1.5]$, while the values of C_l are mostly concentrated in the interval $[0, 1]$. We highlight the two intervals by the black squares in Fig. 2*E*. Note that inside the highlighted region, the policy for ϵ_2 gradually approaches the strategy of rotating with maximum speed, showing the process of learning in the time interval. In addition, the policy could be far from optimal outside the highlighted region, as the agent learns from the experience collected and can hardly generalize the policy for outlier states.

Experimental Task Two: Maximization of the System Power Gain Efficiency. We define the system power gain efficiency as $\eta=\Delta P/(0.5\rho U^3 DL)$, which increases as the drag force is reduced, $\overline{C_{d0}}-C_d$, and decreases as the power loss due to the friction of the control cylinder rotation, $C_f\frac{\pi d}{D}(|\epsilon_1|^3+|\epsilon_2|^3)$, increases. We restrict $\epsilon_1=-\epsilon_2$ in this task, $\overline{C_{d0}}$ is the average drag coefficient when $\epsilon_1=\epsilon_2=0$, and the frictional coefficient is calculated as $C_f=0.027/Re_d^{(1/7)}=0.0097$ (33). Our goal is to maximize the average system power gain efficiency $\overline{\eta}$ over one episode. Therefore, we construct the reward function as follows:

$$r=\eta=\frac{\Delta P}{0.5\rho U^3 DL}=[\overline{C_{d0}}-C_d]-\left[C_f\frac{\pi d}{D}(|\epsilon_1|^3+|\epsilon_2|^3)\right]. \quad [2]$$

Due to the trade-off between drag reduction and the power loss due to the cylinder rotation, for the static control, the maximum of $\overline{\eta}$ is achieved at $\epsilon_2/\epsilon^{max}\approx 0.8$, shown in Fig. 3*A* by the black solid line as reference. The dots in Fig. 3*A* represent $\overline{\eta}$ estimated in each episode and are shown to be concentrated near the peak of the reference line for episodes with well-trained RL agent. In fact, the optimal $\overline{\eta}$ from the RL experiment is found to be higher than the maximum from the static control, which could be explained by the fact that the control strategy designed by the agent is dynamic instead of static. We also plot the $\overline{C_d}$ and the $\overline{\epsilon_2}/\epsilon^{max}$ for each episode in Fig 3*B*.

Simulation Task. In the simulation task, we use the same reward as in cases I and II of the experimental task one but we do not apply the Kalman filter on the original states. Considering the computational cost, the RL agent is designed to interact with the environment more frequently, as shown in Table 1. Moreover, each episode consists of a smaller number of state inquiries than that in the experiments; the former consists of 10 interactions,

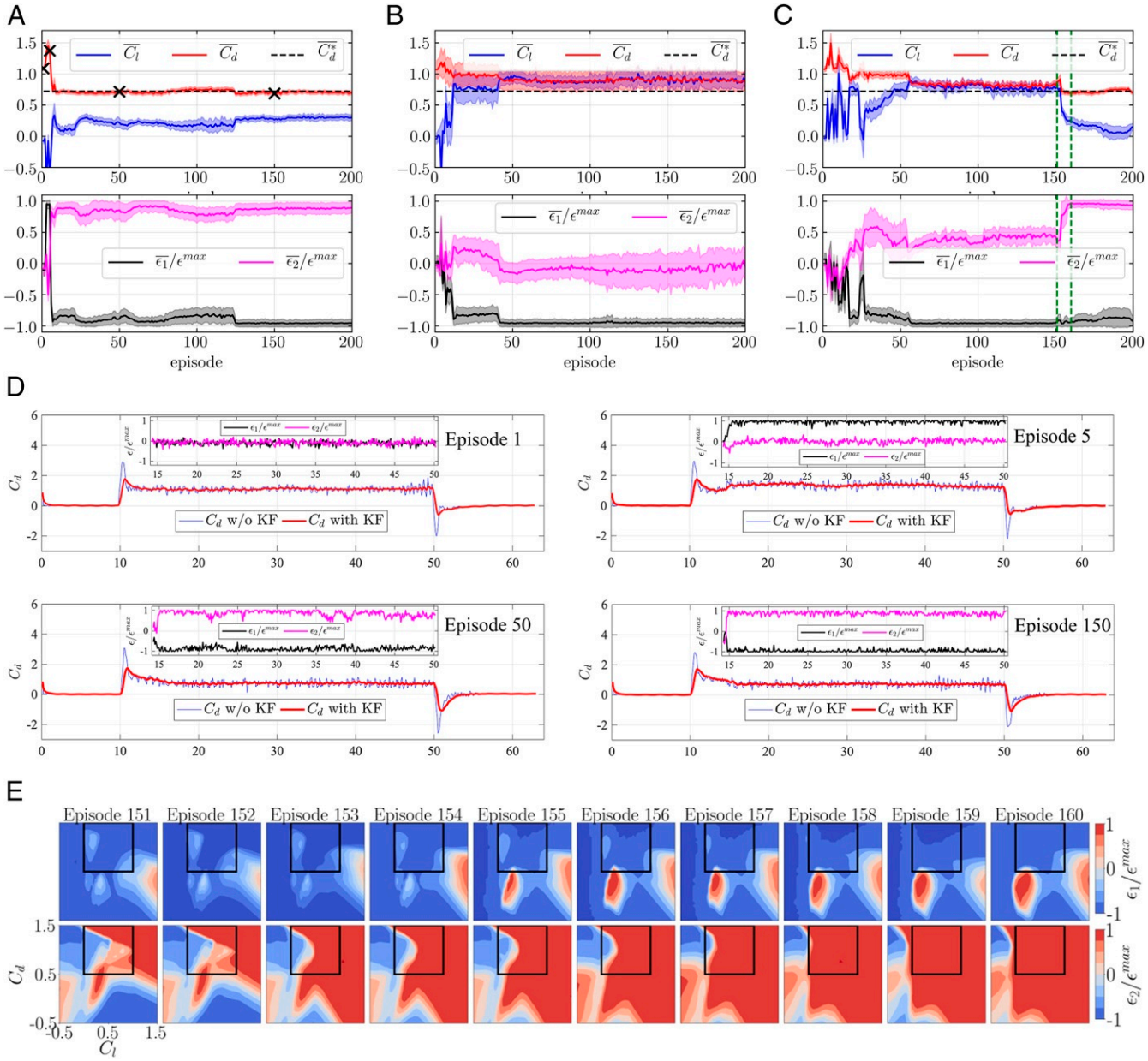

Fig. 2. Training process in experimental task one: (*A*) case I, (*B*) case II, and (*C*) case III. In *A–C*, *Top* row shows the hydrodynamic coefficients over 200 episodes, and *Bottom* row shows actions over 200 episodes. The solid lines and the shaded areas represent the mean value and 1 SD over each episode (with the first 50 and last 50 interactions dropped), respectively. The black dashed lines represent $\overline{C_d^*}$. (*D*) Time trace of drag coefficient and actions (*Insets*) for episodes 1, 5, 50, and 150 in case I. The carriage moves at the 10th s and stops at the 50th s. The active control is switched on at the 14th s. (*E*) Visualization of the policy evolution from episodes 151 to 160 in case III, corresponding to the region between the two green dashed lines in *C*. In *E*, *Top* row shows $\epsilon_1/\epsilon^{max}$ and *Bottom* row shows $\epsilon_2/\epsilon^{max}$ in terms of C_d and C_l.

while the latter consists of 360 interactions, corresponding to about 0.24 and 28 vortex shedding periods, respectively. In Fig. 4, we present the drag coefficient as well as the actions during the training, where we can clearly see that the agent learned to rotate both cylinders at about the maximum speed, consistent with that in the experimental task one. Consequently, the drag coefficient is reduced from about 1.0 to about 0.7. The lift coefficient did not converge to zero, which could be attributed to the fact that the rotations of the two cylinders are not completely symmetric, just as in the experiments.

Having the three-dimensional (3D) simulation results available, we can now explain the changing flow topology and quantify the emerging dynamics. The instantaneous 3D vortical wake and $\overline{\omega_z}$ (the z component of vorticity averaged both in time and in the spanwise direction), at three different stages, namely before training, after the 100th episode, and after the 500th episode, are shown in Fig. 5 *A–C*, respectively. When the two small cylinders are held still, due to the small gap between the large and small cylinders, a jet is formed within the gap, shown in the local velocity field around the small cylinder in Fig. 5*A*. As observed in the mean vorticity field, such a jet results in the vortex formation about $1D$ behind the rear of the main cylinder. With the increase of the rotation speed of the small cylinders, the shed vortices appear to move closer to the main cylinder aft part, and when the maximum rotation rate is reached (Fig. 5*C*), a pair of elongated vortices is formed right behind the main cylinder.

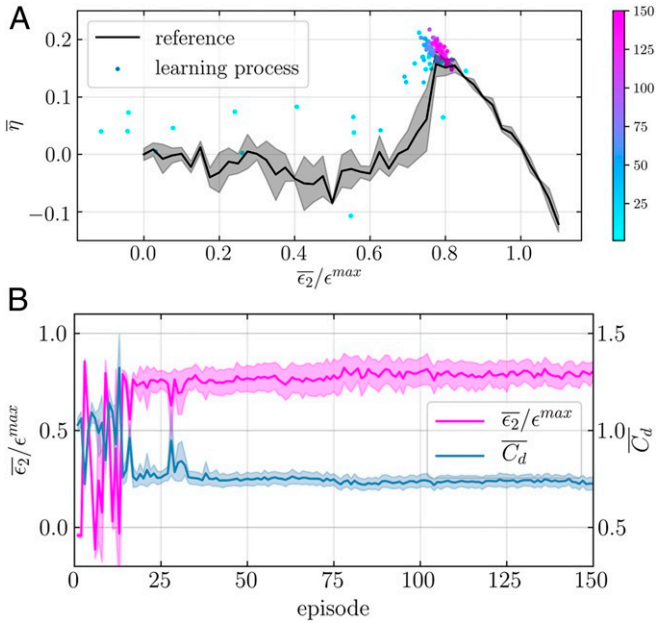

Fig. 3. Training process in experimental task two. (*A*) Evolution of average power gain efficiency $\overline{\eta}$ during the training process. The black solid line and the shaded area represent the mean and 1 SD of $\overline{\eta}$ with constant $\epsilon_1 = -\epsilon_2$ over three independent experiments. The dots represent $\overline{\eta}$ in each episode, while the different colors represent different episode indexes. (*B*) $\overline{\epsilon_2}$ and $\overline{C_d}$ in each episode. The solid lines represent the mean value over each episode while the shaded areas represent 1 SD over each episode.

Such an evolution of the flow morphology is also reflected by the narrowing of the streamlines around the cylinders (the streamlines in Fig. 5 start from points ($-1.5D$, $\pm 0.2D$) from the center of the main cylinder). Furthermore, the pressure distribution around the cylinder circumference reveals that the fast-rotating small cylinders help to reattach the flow at the rear of the main cylinder. In Fig. 5 *A–C*, *Top*, the red line indicates the pressure distribution around the main cylinder, with influence of the small control cylinders, while the dashed black line represents that of the single cylinder at the same Re. In Fig. 5 *A–C*, *Middle*, red color represents positive pressure values while green color is for negative pressure values. As a consequence, the pressure on the rear cylinder surface recovers to a negative value with a smaller magnitude, compared to the nonrotating case, which subsequently leads to a significant pressure drag reduction.

From the instantaneous 3D wake pattern (vortices are represented by the isosurface of $\lambda_2 = -1.5$ and colored by the ω_z), we observe that when the two small cylinders are held still, spanwise coherent vortex tubes are shed from the back of the main cylinder, accompanied by sparse streamwise vortices. When the small cylinder starts to rotate, many streamwise vortices of smaller size are generated, suggesting turbulence intensification. In addition, we observe that at the front half of the cylinder characterized by favorable pressure gradient, with the two small cylinders held still, the isosurface of λ_2 is smooth along the span; however, when the small cylinders are rotating, the isosurface of λ_2 in the front half of the cylinder becomes wavy along the span.

The simulation takes about 0.65 s wall time for each time step on 64 cores of Intel E5-2670, i.e., about 1.1 h for each episode. In total, 500 episodes take more than 3 wk. As an easier case for readers to reproduce, we also conducted RL in a similar problem with $Re_D = 500$, where it takes about 0.75 h for each episode on 12 cores of Xeon(R) Gold 6252 and about 100 episodes, i.e., 3 to 4 d, for full convergence. We present this case in *SI Appendix*, section S6.

Discussion

We demonstrated the feasibility of applying reinforcement learning to discover effective active control strategies in complex experiment and simulation environments for turbulent flow past a circular cylinder.

We selected this problem as representative of many complex flow–structure interaction problems, because despite the apparent simplicity of the system, consisting of a long circular cylinder in cross-flow with two smaller rotating cylinders for control, the mechanics of flow instability that cause vortex formation are known to be very complex. Despite decades of research, additional phenomena are being discovered in vortex-induced oscillations (34), such as multivortex shedding and high harmonics in fluid forces. When control action is used experimentally in such problems, the inherent complexity of their response makes human-guided selection difficult, and only systematic search can yield eventually an optimal solution. For a problem with few parameters, such as the current one, a systematic search is feasible, albeit very laborious. For realistic applications in marine and air vehicle control, for example, a multitude of parameters make a systematic parametric search of a high-dimensional problem virtually impossible; this is where reinforcement learning holds

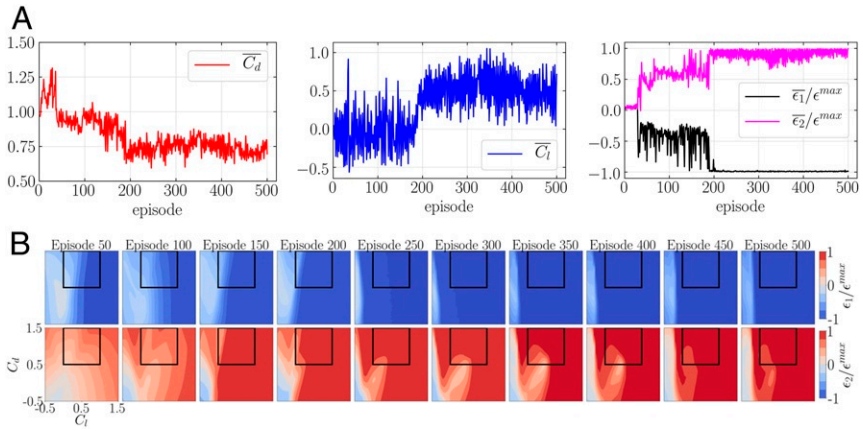

Fig. 4. Training process in the simulation task. (*A*) Drag and lift coefficients and actions over 500 episodes. The lines represent the mean value over each episode. Note that one episode corresponds to 0.24 vortex shedding periods. (*B*) Visualization of policy evolution after 50th to 500th episodes. *A*, *Left* to *Right* shows $\epsilon_1/\epsilon^{max}$ while *B*, *Left* to *Right* shows $\epsilon_2/\epsilon^{max}$ in terms of C_d and C_l.

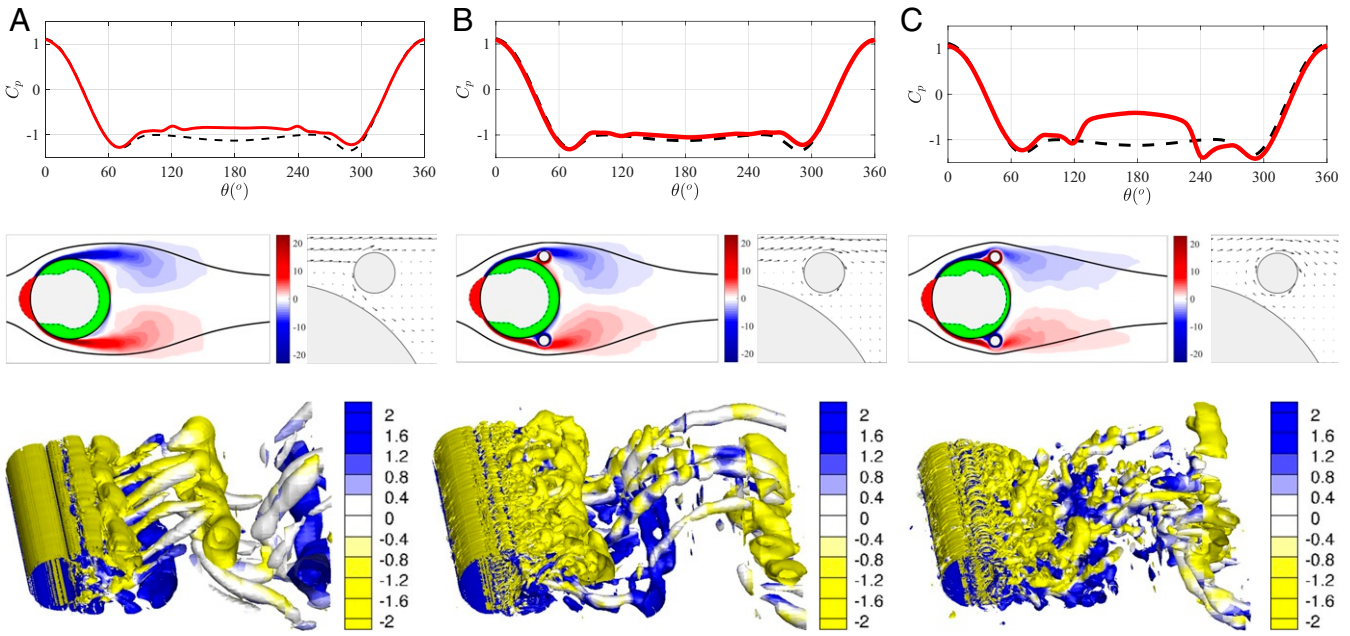

Fig. 5. Visualization of the vortical flow at three different stages of training. (*A*) Before training (both small cylinders are held still). (*B*) After 100 episodes (both small cylinders rotate at an intermediate speed). (*C*) After 500 episodes (both small cylinders rotate at about the maximum speed). (*A–C*, *Top*) Local pressure coefficient on the cylinder surface as a function of angle (θ), with the front stagnation point as the zero degree. The coefficient is shown by the red lines, with black dashed lines representing the reference coefficient of a single cylinder. (*A–C*, *Middle Left*) *z* component of vorticity averaged spanwise and in time with the green/red area indicating the magnitude of negative/positive pressure on the main cylinder. (*A–C*, *Middle Right*) Velocity field near the upper small cylinder. (*A–C*, *Bottom*) Three-dimensional vortices. Note that to plot *B*, we restart the simulation from the flow snapshot saved at episode 100, keep the control cylinders rotating at same speeds as those of episode 100, and continue to run the simulation over two vortex shedding periods; similar procedures are performed to obtain *C*.

great promise. Still, even in this lower-dimension problem there are generic problems that it shares with more complex problems; noise in the measurements provides added difficulty and requires special treatment.

Specifically, we used as an example the problem of reducing the drag force acting on a circular cylinder in cross-flow using as actuators two smaller rotating cylinders and accounting for the energy loss in the cylinder rotation. With a properly designed reward function, the agent was able to learn a control strategy that is comparable to the optimal one found in extensive static control experiments, using only tens of experiments and requiring only several hours of wall-clock time. The RL agent managed to learn the successful control strategy of small cylinder rotation to reattach the flow behind the main cylinder and hence increase the rear pressure recovery, as illustrated in the companion simulation studies. Here we point out that in the simulation task, the current bottleneck of the computation is due to the high-fidelity LES, which takes more than 3 wk for 500 episodes on 64 cores of Intel E5-2670. The training time of the RL agent is negligible compared with the time consumed by LES. Parallelization of multiple simulation environments interacting with the same agent, when there are enough computational resources, can be one of the solutions for the speedup.

We also note that when applying reinforcement learning in the real-world experiment, it is important, as a first step, to identify

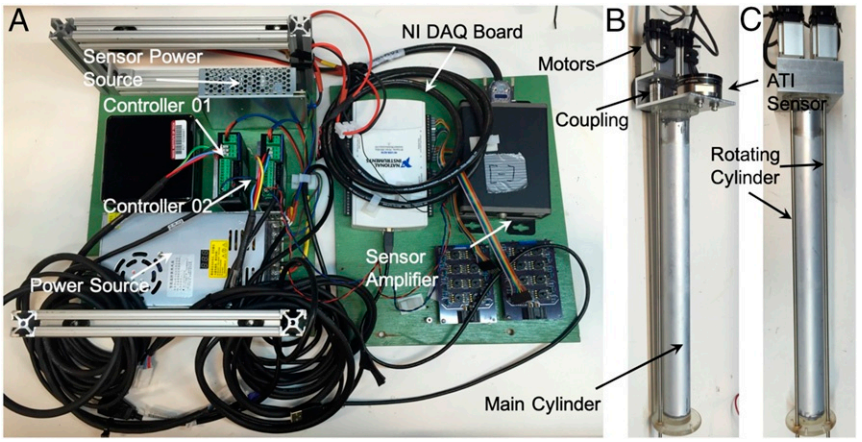

Fig. 6. (*A–C*) Images of the control panels (*A*) and model side view (*B*) and back view (*C*) used in the experiment.

the limitations of the available hardware. Taking our experiment as an example, the constraints of the communication speed and the motor response time limit the frequency of the state inquiry and action decision, which may be essential for other flow control problems, for example, using high-frequency traveling waves to control the flow around an airfoil (35). Additionally, the rotation speed of the control cylinders cannot exceed $\epsilon = 3.8$ due to the structural pin–pin setup; when the motor rotates faster than $\epsilon = 3.8$, the control cylinder starts to vibrate laterally as a flexible cylinder, hence changing the physics of the problem. Furthermore, in the current experiment, we demonstrated the necessity of noise reduction techniques using a Kalman filter, a method which is especially suitable for experimental setups, while in the future, research should be performed on whether and how the neural network itself may regularize and quantify the noise.

Based on our findings and the experience gained both from the experiments and the simulations, we believe that RL can be applied to a variety of complex flow problems. Several issues still remain open for applying RL in fluid mechanics. For example, we reported that the learning results are sensitive to the selection of the exploration noise magnitude in the RL algorithm. The hyperparameter setups developed for rigid body testbed problems could be unsuitable for fluid mechanics problems. This calls for future studies to design algorithms and select hyperparameters suitable for fluid mechanics. Also, while the learning algorithm we employed is totally model-free, it will be worth exploring the possibility of incorporating some domain knowledge and designing physics-informed reinforcement learning algorithms, to further reduce the time to solution.

Materials and Methods

Experimental Model and Hardware. In Fig. 6 we show the control panels and model used in the experiment. The control panels in Fig. 6*A* have two decks and consist of six major components: two DYN2-series motor controllers, one NI USB-6218 Data Acquisition (DAQ) board, one ATI sensor amplifier, and two power sources. The DAQ board in the upper deck is controlled through the USB communication, and it is in charge of the analog data collection from the sensor amplifier at a sampling rate of 1,000 Hz and sending the signal to the two motor controllers at a feedback rate of 10 Hz. In the lower deck, two independent DYN2-series servo motor controllers for two DST-410 servo motors are powered by a 60-V DC power source.

Images of the experimental model are shown in Fig. 6 *B* and *C* for two views. The main cylinder is made from a hard-anodized aluminum tube to prevent corrosion in the water. The two smaller stainless steel cylinders are connected to the two DST-410 motors via couplings and are supported by the bearings on both ends. Shown in Fig. 6*B*, the ATI-Gamma sensor is installed on top of the model and is used to measure the total lift and drag force on the main and two smaller cylinders altogether.

Reinforcement Learning. Reinforcement learning involves an agent interacting with the environment, aiming to learn the policy that maximizes the expected cumulative reward. At each discrete time step i, the agent makes an observation of the state $s_i \in \mathcal{S}$, selects corresponding actions $a_i \in \mathcal{A}$ with respect to the policy $\pi : \mathcal{S} \to \mathcal{A}$ to interact with the environment, and then receives a reward r_i. The objective is to find the optimal policy π_ϕ parameterized by ϕ which maximizes the expected cumulative reward,

$$J(\pi) = \mathbb{E}_{(s_i,a_i)\sim p_\pi} \sum_{i=0}^{T} \gamma^i r_i, \qquad [3]$$

where $\gamma \in (0, 1]$ is a discount factor and p_π denotes the state-action marginals of the trajectory distribution induced by the policy π.

As mentioned in the previous subsections, in the current work the state is the concatenation of C_l and C_d, while the action is the concatenation of $\epsilon_1/\epsilon^{max}$ and $\epsilon_2/\epsilon^{max}$. The reward received in each time step is induced from the state and action in the subsequent interaction.

The update of the agent follows one of the state-of-the-art deep RL algorithms, viz. the Twin Delayed Deep Deterministic policy gradient algorithm (TD3) (36). We provide the pseudocode in *Algorithm 1*. In this paper, all of the neural networks are feedforward neural networks with two hidden layers, each of width 256. The discount factor γ is set as 0.99. The SD of the policy exploration noise σ is set as 0.1 in the experimental task one of drag reduction, 0.01 in the experimental task two of system power gain efficiency maximization, and 0.025 in the simulation task of drag reduction. We use the Adam optimizer with learning rate 10^{-4} to update θ_i and ϕ. N_i is ∼360 in experiments and 10 in simulation, while N_j is set to 1,000 in experiments and 100 in simulation. The batch size N is set to 512 in experiments and 256 in the simulation. (Note that we skip the training stage if the number of tuples in the buffer is less then N.) The SD of the regularization noise $\tilde{\sigma}$ is set to 0.2, and c is set to 0.5. The actor and target networks are updated every $d = 2$ iterations. The soft updating rate τ is set to 0.005.

Algorithm 1: Reinforcement learning for active fluid control with TD3

initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ with random parameters θ_1, θ_2, ϕ;
initialize target networks $\theta_1' \leftarrow \theta_1$, $\theta_2' \leftarrow \theta_2$, $\phi' \leftarrow \phi$;
initialize replay buffer $\mathcal{B}$;
$t \leftarrow 0$;
for $n_e = 1$ **to** N_e **do**
  for $i = 1$ **to** N_i **do**
    collect the observed state s_i;
    request an action with exploration noise:
    $a_i \leftarrow \text{clip}(\pi_\phi(s_i) + \epsilon_i, -1, 1)$, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$;
    apply a_i in the environment;
  end
  store transition tuples $\{(s_i, a_i, r_i, s_{i+1})\}_{i=1}^{N_i-1}$ into $\mathcal{B}$;
  for $j = 1$ **to** N_j **do**
    sample N transitions (s, a, r, s') from $\mathcal{B}$;
    $\tilde{a} \leftarrow \text{clip}(\pi_{\phi'}(s') + \epsilon, -1, 1)$, $\epsilon \sim \text{clip}(\mathcal{N}(0, \tilde{\sigma}^2), -c, c)$;
    $y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta_i'}(s', \tilde{a})$;
    $L_{\theta_i} \leftarrow N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$, $i = 1, 2$;
    update θ_i with the loss function L_{θ_i}, $i = 1, 2$;
    if t mod d **then**
      $L_\phi \leftarrow -N^{-1} \sum Q_{\theta_1}(s, \pi_\phi(s))$;
      update ϕ with the loss function L_ϕ;
      update the target networks:
      $\theta_i' \leftarrow \tau\theta_i + (1 - \tau)\theta_i'$, $i = 1, 2$;
      $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'$;
    end
    $t \leftarrow t + 1$;
  end
end

Data Availability. All study data are included in this article and *SI Appendix*. In addition, the code for the reinforcement learning agent and simulation environment is shared via GitHub at https://github.com/LiuYangMage/RLFluidControl.

ACKNOWLEDGMENTS. D.F. and M.S.T. acknowledge support from the Massachusetts Institute of Technology Riser Digital Twin Consortium and a fellowship provided by Shell International Exploration and Production, Inc; Y.L. and G.E.K. acknowledge support by the Department of Energy Physics Informed Learning Machines (PhILMs) project (DE-SC0019453). The simulation at $Re_D = 10,160$ was performed at The Center for Computation and Visualization, Brown University.

1. S. L. Brunton, B. R. Noack, Closed-loop turbulence control: Progress and challenges. *Appl. Mech. Rev.* **67**, 050801 (2015).
2. S. L. Brunton, B. R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508 (2019).
3. D. Silver *et al.*, Mastering the game of go without human knowledge. *Nature* **550**, 354–359 (2017).
4. D. Silver *et al.*, Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv:1712.01815 (5 December 2017).
5. T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.arXiv:1801.01290 (4 January 2018).

6. M. Gazzola, B. Hejazialhosseini, P. Koumoutsakos, Reinforcement learning and wavelet adapted vortex methods for simulations of self-propelled swimmers. *SIAM J. Sci. Comput.* **36**, B622–B639 (2014).
7. S. Verma, G. Novati, P. Koumoutsakos, Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proc. Natl. Acad. Sci. U.S.A.* **115**, 5849–5854 (2018).
8. G. Reddy, A. Celani, T. J. Sejnowski, M. Vergassola, Learning to soar in turbulent environments. *Proc. Natl. Acad. Sci. U.S.A.* **113**, E4877–E4884 (2016).
9. S. Colabrese, K. Gustavsson, A. Celani, L. Biferale, Flow navigation by smart microswimmers via reinforcement learning. *Phys. Rev. Lett.* **118**, 158004 (2017).
10. G. Novati, L. Mahadevan, P. Koumoutsakos, Controlled gliding and perching through deep-reinforcement-learning. *Phys. Rev. Fluids* **4**, 093902 (2019).
11. P. Ma, Y. Tian, Z. Pan, B. Ren, D. Manocha, Fluid directed rigid body control using deep reinforcement learning. *ACM Trans. Graph.* **37**, 96 (2018).
12. M. A. Bucci *et al.*, Control of chaotic systems by deep reinforcement learning. arXiv:1906.07672 (16 June 2019).
13. J. Rabault, M. Kuchta, A. Jensen, U. Réglade, N. Cerardi, Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* **865**, 281–302 (2019).
14. J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, E. Hachem, Direct shape optimization through deep reinforcement learning. arXiv:1908.09885 (23 August 2019).
15. P. Strykowski, K. Sreenivasan, On the formation and suppression of vortex 'shedding'at low Reynolds numbers. *J. Fluid Mech.* **218**, 71–107 (1990).
16. M. Morkovin, "Flow around circular cylinders-a kaleidoscope of challenging fluid phenomena" in *Proceedings of ASME Symposium on Fully Separated Flows*, A. G. Hansen, Ed. (ASME, Philadelphia, PA, 1964), pp. 102–118.
17. E. Berger, R. Wille, Periodic flow phenomena. *Annu. Rev. Fluid Mech.* **4**, 313–340 (1972).
18. J. T. Lin, Y. H. Pao, Wakes in stratified fluids. *Annu. Rev. Fluid Mech.* **11**, 317–338 (1979).
19. P. W. Bearman, Vortex shedding from oscillating bluff bodies. *Annu. Rev. Fluid Mech.* **16**, 195–222 (1984).
20. H. Oertel Jr., Wakes behind blunt bodies. *Annu. Rev. Fluid Mech.* **22**, 539–562 (1990).
21. C. H. Williamson, Vortex dynamics in the cylinder wake. *Annu. Rev. Fluid Mech.* **28**, 477–539 (1996).
22. S. F. Hoerner, *Fluid Dynamic Drag: Practical Information on Aerodynamic Drag and Hydrodynamic Resistance* (S. F. Hoerner, Midland Park, NJ, 1965), pp. 16–35.
23. H. Choi, W. P. Jeon, J. Kim, Control of flow over a bluff body. *Annu. Rev. Fluid Mech.* **40**, 113–139 (2008).
24. J. Kim, T. R. Bewley, A linear systems approach to flow control. *Annu. Rev. Fluid Mech.* **39**, 383–417 (2007).
25. J. C. Schulmeister, J. M. Dahl, G. D. Weymouth, M. S. Triantafyllou, Flow control with rotating cylinders. *J. Fluid Mech.* **825**, 743–763 (2017).
26. H. Zhu, J. Yao, Y. Ma, H. Zhao, Y. Tang, Simultaneous CFD evaluation of VIV suppression using smaller control cylinders. *J. Fluid Struct.* **57**, 66–80 (2015).
27. J. F. Beaudoin, O. Cadot, J. L. Aider, J. E. Wesfreid, Drag reduction of a bluff body using adaptive control methods. *Phys. Fluids* **18**, 085107 (2006).
28. D. Fan *et al.*, A robotic intelligent towing tank for learning complex fluid-structure dynamics. *Sci. Robot.* **4**, eaay5063 (2019).
29. M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning" in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, K. Keeton, T. Roscoe, Eds. (USENIX Association, Savannah GA, 2016), pp. 265–283.
30. Z. Wang, M. S. Triantafyllou, Y. Constantinides, G. E. Karniadakis, An entropy-viscosity large eddy simulation study of turbulent flow in a flexible pipe. *J. Fluid Mech.* **859**, 691–730 (2019).
31. G. E. Karniadakis, S. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics* (Oxford University Press, Oxford, UK, ed. 2, 2005).
32. P. Zarchan, H. Musoff, *Fundamentals of Kalman Filtering: A Practical Approach* (American Institute of Aeronautics and Astronautics, Inc., 2013).
33. L. Prandtl, "Report on investigation of developed turbulence" (Report NACA-TM-1231, NACA, 1949).
34. D. Fan, Z. Wang, M. S. Triantafyllou, G. E. Karniadakis, Mapping the properties of the vortex-induced vibrations of flexible cylinders in uniform oncoming flow. *J. Fluid Mech.* **881**, 815–858 (2019).
35. S. Calisch, N. Gershenfeld, D. Fan, G. Jodin, M. Triantafyllou, Fabrication and characterization of folded foils supporting streamwise traveling waves. *J. Fluid Struct.* **91**, 102563 (2019).
36. S. Fujimoto, H. Hoof, D. Meger, "Addressing function approximation error in actor-critic methods" in *International Conference on Machine Learning*, J. G. Dy, A. Krause, Eds. (PMLR, Stockholm, Sweden, 2018), pp. 1582–1591.