DEEP REINFORCEMENT LEARNING FOR THE REDUCTION OF THE DRAG IN THE FLOW PAST BLUFF BODIES

P. Mudiyanselage & F. Gueniat*
*Author for correspondence
Department of Mechanical Engineering,
Birmingham City University,
Birmingham,UK
E-mail: florimond.gueniat@bcu.ac.uk

NOMENCLATURE

Q	Q-learning value
π	Policy (control strategy)
a	action value
x	state, derived from measurement
Ω	support of the states x
$\mathcal{A}$	support of the actions a
γ	discount rate
R	Reward value
C_d	drag coefficient
p	non dimensional pressure
L	diameter of the cylinder
u	non dimensional velocity
u_i	component of u
ρ	density
ν	kinematic viscosity
t	non dimensional time
Re	Reynolds number

Subscripts

i, j	discrete time steps or Cartesian direction i
s	given state

Abstract

Closed-loop control of engineering flows remains challenging, even after decades of efforts from the community, and most of the success are related to passive designs. Additionally, most of the successfully closed-loop control rely on the unrealistic situations such as when full information about the system is available, and heavily rely on either the knowledge of the dynamics or on an accurate reduced order model. In this work, we present a model free, fully-data driven methodology that allows to identify a near optimal control strategy. Noteworthy, this methodology relies only on scarce sensors such as pressure transducers.

Keywords: Computational Fluid dynamics, Reinforcement Learning, Drag Reduction

1 Introduction

While the accuracy of the design of nonlinear, large dimensional and complex systems have significantly improved over the last decades, the closed-loop control of such systems still remains challenging. It is desired, in order to improve performance, to increase robustness to non-modeled perturbations. That includes the cases of ill or partially modelled plants. While flow manipulation and open-loop control are common practice, much fewer successful closed-loop control efforts are reported in the literature. Additionally, in the context of flow control, closed-loop control suffers from severe limitations, effectively preventing its use: a typical turbulent flow involves both a large range of spatial scales and exhibits a rich and fast dynamics. As a consequence, controllers have to be able to react very quickly to changes in the environment. An implication is that practical implementation needs reduced-order models. Further, many of them rely on unrealistic assumptions. For example, Model Predictive Control (MPC) approaches require to solve the governing equations in real-time. If a Reduced-Order Model (ROM) is employed, as is common practice to alleviate the CPU burden, deriving the ROM means using the velocity or pressure fields with, e.g., proper orthogonal decomposition, see, for instance, [1; 2; 3].

Hence, flow control with this classes of approaches is restricted to numerical simulations or experiments in a wind tunnel, equipped with sophisticated visualization tools such as particle image velocimetry. Many closed-loop control efforts of the literature rely on the unrealistic situation of full information about the system. Even when a reduced-order model (ROM) is employed, full information on the system is required to evaluate and tune the time-dependent coefficients associated with the modes of the ROM, see, e.g., [1], among many others. Many approaches from the flow control literature also rely on strong assumptions such as linearity of the governing equations of the system.

The present work relies on a change of paradigm started by the author, [4]: we want to derive a general nonlinear closed-loop flow control methodology suitable for actual configurations and as realistic as possible. Contrary to MPC, no a priori model, nor even a model structure, describing the dynamics of the system is required to be available. The approach proposed is data-driven only, with the sole information about the system given by scarce and spatially-constrained sensors. The method then exploits statistical learning methods. It is achieved in two steps. We consider that the information yon the system at hand is limited and comes from a few sensors (located at the boundary of the fluid domain,

e.g., on solid surfaces). The resulting information takes the form of short time-dependent vectors. The first step is to build a state-observer, based on this information. It allows to correctly reconstruct the information needed to estimate the state of the system or, more generally, the quantity of interest (QoI) that is needed for deriving the control law, for instance the instantaneous drag forces on a body.

The second step is the use the recent advances in deep reinforcement learning to derive the control law. As will be seen in the application examples, the resulting control strategy is data-driven only, intrinsically robust against perturbations in the flow and does not require significant computational resources nor prior knowledge of the flow once the models have been trained, making it desirable in practical situations.

Both steps are illustrated sucessfully on the case of the flow past a cylinder, where the QoI is the drag coefficient.

Sec. 2 describes the numerical setup and the methodology. Results are presented and discussed Sec. 3. The manuscript ends on concluding remarks in Sec. 4.

2 Methodology

2.1 Deep reinforcement learning

To derive a control strategy, one needs to determine a policy π. It simply consists in determining the best control action a w.r.t the current measurements, or state x, at hand. Rewards $R := R(x,a)$ are associated with both the action (if it is costly) and the effect, desired or not, it has on the system. The value of the reward will help chose the correct action to take. For instance, actions associated to the highest rewards are giving the best short term results, it is known as the greedy policy. However the best policy aims at maximizing the rewards on the long term, via the expected value of the rewards:

$$\phi = \sum_{i}^{N} \gamma^i R_i \tag{1}$$

The subscript denotes the discrete times. γ is the discount factor, essentially more weight is given to most recent state compared to older ones.

Under actions, the system is evolving, hence the states are changing. When the probabilities of transition from a state to another are known, the optimal policy may be identified by means of a dynamic programming algorithm, [5; 6]. Usually, the transition probabilities are extremely difficult to identify as both the control policy and the transition probabilities (due to memory effect) can evolve during the learning stage. In this situation, *Reinforcement Learning* is a suitable class of method, [7]. In particular, the *Q-learning* approach consists in relying on the estimation of the *Q-factors*, or *action-values*, Q^π which evaluate the expected reward of a state-action combination when following the policy π:

$$Q^\pi(x_i,a_i) := \left\langle R_{i+1} + \gamma R_{i+2} + \gamma^2 R_{i+3} + \ldots | x_i, a_i \right\rangle, \tag{2}$$

where $\langle .|x,a\rangle$ is the expected sum of the discounted cumulative reward knowing the state x and taken action a.

As stated previously, the transition probabilities from x_i to x_{i+1} can not be accurately estimated. It means that calculating the expected sum is impossible. However, an iterative estimation of the Q-factors can be derived, [7; 8]. Letting the initial Q-factors be given, the Q-factor associated with a state x_s and an action a_s can be updated following the Bellman equation:

$$Q(x_s,a_s) \leftarrow \underbrace{Q(x_s,a_s)}_{\text{old value}} + \alpha_s \left(R(x_s,a_s) + \gamma \underbrace{\max_{\widetilde{a}\in\mathcal{A}} Q(x_{s+1},\widetilde{a})}_{\text{“best” value}} - \underbrace{Q(x_s,a_s)}_{\text{old value}} \right), \tag{3}$$

where $\alpha_s > 0$ is a learning factor.

The action-value $Q(x_s,a_s)$ will increase when the reward associated with the 2-uplet (x_s,a_s) is good, and decrease otherwise.

To learn a good policy π, the system, in different states x_s is stimulated with different actions a_s to estimate the Q-factors. Once converged to Q^π, the control policy is simply taking the action which is associated with the largest Q-factor, [7]:

$$a(x) = \underset{\widetilde{a}\in\mathcal{A}}{\operatorname{argmin}}\, Q(x,\widetilde{a}) \tag{4}$$

Q-factors are easily identified when the system is discrete, as $Q(x,a)$ is a table. When considering continuous states and actions, the generalization of the previous method, known as deep reinforcement learning, relies on function approximators: a neural network replaces the Q-factors, as illustrated Fig. 1. The network is composed of a state estimator (feature extraction) network followed by an actor network composed of two fully connected layers. The learning still follows the Bellman equation, following Eq. (3).

One of the main advantages of this approximation is the inherent ability of neural network to work well in the case of previously unseen inputs. It means an intrisic robustness to noise and uncertainties. In this preliminary work, the TRPO algorithm has been used, [9], using an out-of-the-shelf implementation [1].

2.2 Two-dimensional numerical flow

To further illustrate the methodology discussed above, we consider a 2-D laminar flow around a circular cylinder.

Configuration of the test case The present simulations are carried out using OpenFOAM [2] to solve the continuity and incompressible Navier-Stokes equations, solved on the 2D flow domain $\mathcal{D}$. The spatial and temporal coordinates are denoted by x_i and t. All the physical variables are nondimensionalized by

[1] https://github.com/hill-a/stable-baselines/

[2] www.openfoam.org

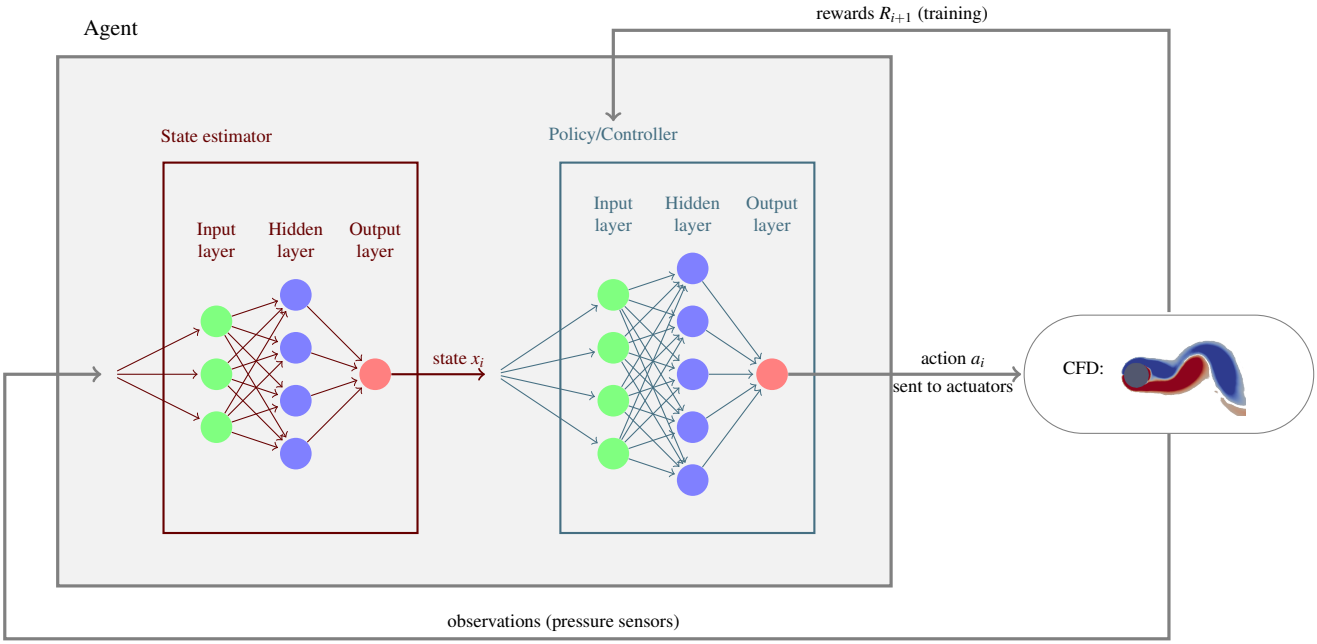


Figure 1: Schematic workflow of the deep reinforcement learning method applied to flow control.

the inlet uniform velocity $u_\infty = 1$ and the diameter of the cylinder $L = 1m$.

$$\begin{aligned} \frac{\partial u_i}{\partial x_i} &= 0, \\ \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} &= -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_i^2}. \end{aligned} \tag{5}$$

The equations are solved by a finite volume method and the PIMPLE algorithm. Turbulence is resolved using the Realizable $k-\varepsilon$ model. The Reynolds number Re was fixed at 200, meaning that the nondimensional kinematic viscosity is fixed at $\nu = 0.05$.

The cylinder is represented by its surface $\partial\mathcal{D}_{cyl}$. The drag force $F_{d,i}$ is calculated by integrating the normal pressure and the tangential viscous contributions on the cylinder surface on the x_1 direction. The drag coefficient is defined as:

$$C_{di} = \frac{F_{d,i}}{1/2\rho A u_\infty^2}, \tag{6}$$

where the inlet velocity, being adimensionate, the front surface is $A = 1m^2$ and F_d is the drag force on the cylinder.

Validation of Numerical Models The dimension of the computational domain, normalized by the cylinder diameter, are $25L \times 10L$, and the distance of the cylinder to the inlet is $5L$.

The numerical resolution of the unstructured mesh was determined after a grid refinement study to ensure the grid-independency of the solution, and verified against the Strouhal number associated with the vortex shedding of the 2D unactuated flow, [10]. The difference is within 2%. Post processed calculation of the y^+ wall coordinates has been done as well, to verify it remained in the valid domain of the turbulence model. The total number of cells used was 149029.

Actuators Two suction and oscillatory blowing actuators (SaOBAs) are considered, [11]. They correspond to two slots of a length of $l = 0.1745m$ each, on the surface of the cylinder. One is on top of the cylinder, the other one is on the bottom of the cylinder: $a_i = (a_{top,i}, a_{bot,i})$. They represent a suction and blowing wall actuation of the flow, see Fig. 2. The value of the actuation corresponds to the velocity condition imposed on the wall, along the x_2 direction, and is limited for both actuators to the range $[-1,1]$. Note that the actuations are totally decoupled, as seen on Fig. 2, where the top actuation is weakly negative while the bottom actuation is strongly positive. This configuration has been chosen on purpose. In this low Reynolds case, continuously sucking in the boundary layer is the optimal way to inhibit its growth, [12; 13], as long as the cost of actuating the system is neglected. However, in the present case, the amount of fluid removed from the boundary layer (or the amplitude of actuation) cannot be enough to remain below the critical Reynolds and fully relaminarize the flow. It will be considered as the *oracle* in the following to compare the efficiency of the control.

Sensors Observations comes from 18 sensors. The sensors are wall pressure measurements, evenly distributed around the cylinder. They are similar to dynamic pressure transducers.

2.3 Reward formulation

The reward has been constructed as the sum of a few parts corresponding to the several objectives:

$$R_i = R_{C_d,i} + R_{a,i} + R_{reg,i} \tag{7}$$

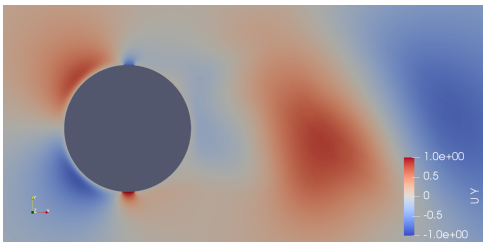


Figure 2: Typical actuation of the flow. Colors encode the u_2 field and are available online. Actuators are visible on the two sides of the cylinder

The first objective is the reduction of the drag coefficient C_d. A low drag is to be associated to large reward, consequently:

$$R_{C_d,i} = -w_d(C_{di} - 1.55) \tag{8}$$

The value 1.55 has been chosen as close to the average drag, to make the reward positive when the drag coefficient is low, and negative when high. w_d is a weight fixed to 1 in the setup. The second objective is the take into account the actuation cost, defined as the intensity of the actuation. A large actuation is penalized as:

$$R_{a,i} = -w_a||a_i||_2. \tag{9}$$

w_a is a weight fixed to 0.1 in the setup. It is hence expected that the actuation identified by the proposed methodology to be around 10% lower that the oracle one.

The last objective is to avoid large fluctuations of the drag, which corresponds to replicating a bang-bang controller, i.e. a controller only activated when the drag is too high. A regularization term is hence introduced, as the sum of difference in the past $N_{reg} = 5$ drag coefficients:

$$R_{reg,i} = -w_{reg} \sum_{k=0}^{N_{reg}} |C_{di-k} - C_{di-k-1}| \tag{10}$$

w_{reg} is a weight fixed to 0.1 in the setup.

3 Results and Discussions

The training has been done on 5000 time steps, for a total of 96h cpu time on a single processor machine. It is already a significant computational burden, in a reasonable numerical setup. However, the deep reinforcement learning has produced the expected outputs. Results are summarized Tab. 1 and illustrated Fig. 3. The average actuation is nearly constant, and is found to be 13% smaller than in the oracle case, which was expected with the definition of the cost function. The baseline, time averaged drag coefficient for the cylinder is 1.531, [10]. The main results is a reduction of the time-averaged coefficient C_d of 29%, compared to the 33% with the oracle case. Notably, the amplitude

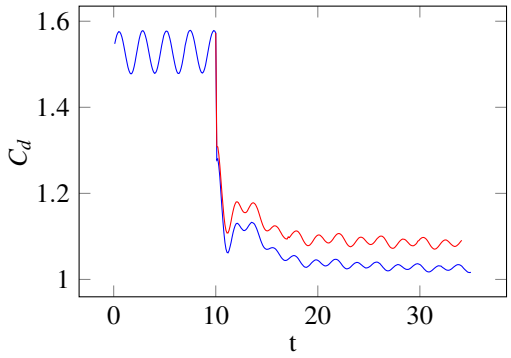
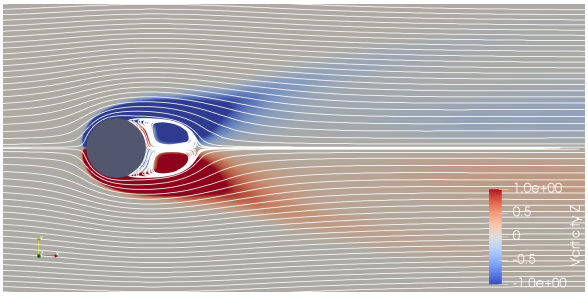
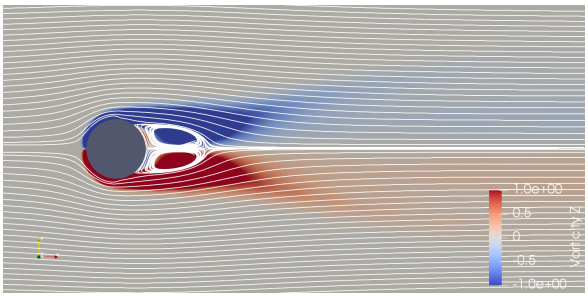


Figure 3: Drag coefficient. At t=10, the controller is turned on.



(a)



(b)

Figure 4: Vorticity field and streamlines averaged in time a: uncontrolled case and b: controlled case. Colors are available online.

of the oscillations are diminished by 77% (82% with the oracle case).

The identify control law corresponds to the near optimal policy, which consists in maximum constant suction on both actuators. The difference with the oracle control law is explained by the costs associated to the actuation in Eq. (9). It is translated by an average reward lower for the oracle case by 12.5%, and hence a suboptimal control w.r.t. to definition of the cost. As the difference on flow fields are indiscernible, for the sake of brevity, only the results obtain with the proposed methodology are reported in Figs. 4 and 5.

Under the action of the controller, the average recirculation bubble length has been increased, while its height has decreased, as seen Fig. 5 and Tab. 1. The separation angle is also reduced, [14]. It mainly explains the gain in drag. The oscillations

Table 1: Main results comparing the baseline, the oracle policy and the identified policy. Results are the drag coefficient C_d, rewards and recirculation bubble dimensions. The height is measured at a distance of 0.3L from the edge of the cylinder.

case	baseline	oracle	present
average reward	0.16	4.64	5.22
average drag coefficient	1.531 ± 0.035	1.028 ± 0.006	1.086 ± 0.008
average actuation	(0,0)	(-1,1)	(-0.79,0.83)
recirculation length	0.85L	1.05L	1.01L
recirculation height	1L	0.76L	0.82L

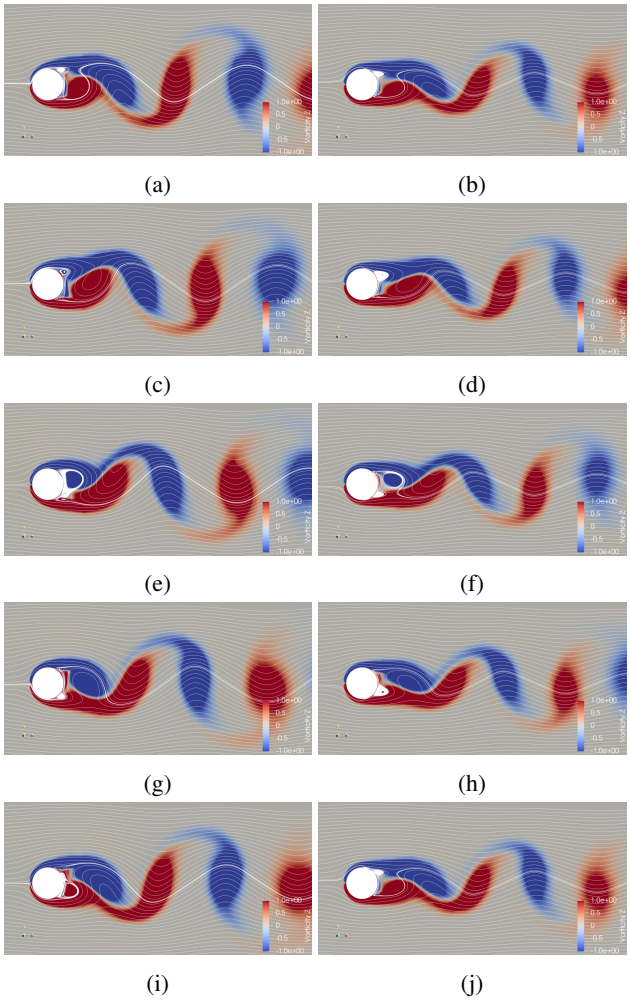


Figure 5: z-vorticity and streamlines during a vortex-shedding cycle. Colors are available online. Left column: baseline case. Right column: present methodology.

are not fully laminar, as the amount of fluid withdrawn does not prevent the critical Reynolds to be reached, even in the oracle case (see oscillations Fig. 3). We expect that development of a better tailored reward function in that respect will help to identify a control law more efficient at fully laminarizing the wake. The exploration is mostly guided toward reducing the drag and the weight given to the actuators is low. As illustrated in Tab. 1 and as expected, the obtained results are sensitive to the definition of the cost function. In the present configuration and objectives, the identified control law is near optimal, [12].

4 Conclusion

The present work has investigated the use of deep reinforcement learning applied to the closed-loop control of engineering flows. The considered system is a cylinder flow, with two blowing/sucking wall actuators. This proof of concept has been done at low Reynolds number. The actuators are fully controlled by the developped controller, and a near optimal control law is identified. Even in this simple configuration, it is shown that the computational effort is quite significant. However, this work shows the promises of such an approach, as it is fully data-driven and equation free, and a solution close to the expected one has been found. It shows an effective drag reduction of 29%. Future works are planned to apply this methodology to an experimental case, where the computational burden will not be a limitation.

References

[1] M. Bergmann and L. Cordier, "Optimal control of the cylinder wake in the laminar regime by trust-region methods and POD reduced-order models," *Journal of Computational Physics*, vol. 227, pp. 7813–7840, aug 2008.

[2] A. Cammilleri, F. Guéniat, J. Carlier, L. Pastur, E. Memin, F. Lusseyran, and G. Artana, "POD-spectral decomposition for fluid flow analysis and model reduction," *Theoritical Computational Fluid Dynamics*, vol. 27, pp. 787–815, feb 2013.

[3] F. Guéniat, L. Pastur, and F. Lusseyran, "Investigating mode competition and three-dimensional features from two-dimensional velocity fields in an open cavity flow by modal decompositions," *Physics of Fluids*, vol. 26, p. 085101, aug 2014.

[4] F. Guéniat, L. Mathelin, and M. Hussaini, "A statistical learning strategy for closed-loop control of fluid flows," *Theoritical Computational Fluid Dynamics*, vol. 30, pp. 497–510, December 2016.

[5] R. Bellman, "On the theory of dynamic programming," *Proceedings of the National Academy of Sciences*, vol. 38, no. 8, pp. 716–719, 1952.

[6] P. Mandl, "Estimation and control in markov chains," *Advances in Applied Probability*, vol. 6, no. 1, pp. 40–60, 1974.

[7] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.

[8] A. Gosavi, "Target-sensitive control of markov and semi-markov processes," *International Journal of Control, Automation and Systems*, vol. 9, pp. 941–951, oct 2011.

[9] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*, pp. 1889–1897, PMLR, 2015.

[10] U. Fey, M. König, and H. Eckelmann, "A new strouhal–reynolds-number relationship for the circular cylinder in the range 47¡ re¡ 2× 10 5," *Physics of Fluids*, vol. 10, no. 7, pp. 1547–1549, 1998.

[11] A. Seifert, O. Stalnov, D. Sperber, G. Arwatz, V. Palei, S. David, D. I, and I. Fono, "Large trucks drag reduction using active flow control," in *The Aerodynamics of Heavy Vehicles II: Trucks, Buses, and Trains*, pp. 115–133, Springer, 2009.

[12] M. Gad-el-Hak, *Flow control: passive, active, and reactive flow management*. Cambridge university press, 2007.

[13] H. Choi, W. Jeon, and J. Kim, "Control of flow over a bluff body," *Annu. Rev. Fluid Mech.*, vol. 40, pp. 113–139, 2008.

[14] G. Chopra and S. Mittal, "Drag coefficient and formation length at the onset of vortex shedding," *Physics of Fluids*, vol. 31, no. 1, p. 013601, 2019.