Discovering Active Flow Control strategies using Neural Networks and Deep Reinforcement Learning

article preprint: https://arxiv.org/abs/1808.07664

Jean Rabault, Miroslav Kuchta & Atle Jensen

jean.rblt@gmail.com

University of Oslo, Department of Mathematics

contact

paper

Artificial Neural Networks (ANNs) trained through Deep Reinforcement Learning (DRL) have recently proven efficient at solving a wide range of learning tasks where no good solution is known otherwise. In particular, tasks such as optimal control are satisfactorily solved by ANNs/DRL, as was shown initially through the examples of playing Atari games and the game of Go [1, 2]. The DRL paradigm allows the ANN to learn directly from interacting with the problem to solve, through trial and error, and to explore the phase space so as to find an efficient control strategy. This is appealing, as no prior knowledge nor training set about the problem to solve are needed, and the network generates its own training data underway. In a recent work, we showed that this methodology can also be used to control complex systems arising from Fluid Mechanics problem [3]. This makes ANNs/DRL an appealing tool for solving many control problems in classical physics.

Introduction

Active Flow Control is known to be a difficult problem, as a consequence of the combination of non-linearity, high dimensionality, and time-dependence that is implied by the Navier-Stokes equations [4]. The combination of these three factors is the underlying mechanism of many complex problems across a wide range of research fields, and traditional methods usually fail to tackle such situations. However, progresses have been recently achieved on problems presenting such complexity, using a technique called Deep Reinforcement Learning [5].

Deep Reinforcement Learning (DRL) consists in using an Artificial Neural Network (ANN) to learn a control strategy that maximizes the reward obtained when interacting with the system to control through three channels: an observation of the state of the system S_t, a reward depending on the quality of the control R_t, and the active control set by the ANN A_t, see Fig. 1.

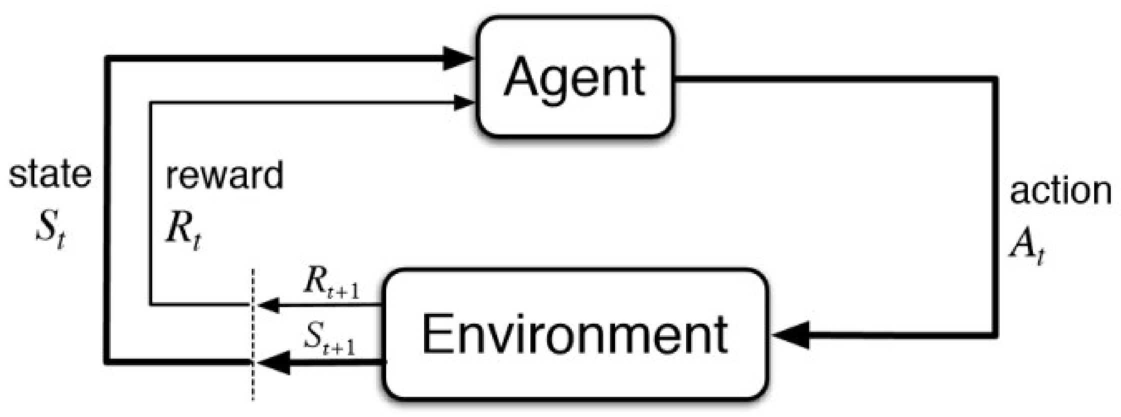

Figure 1: The framework for performing Reinforcement Learning. The Agent is usually an ANN, while the environment is the system to control (simulation or real world).

Simulation environment, ANN and DRL training

In order to investigate the ability of ANNs to perform active flow control, we design a simple 2D simulation of the Karman street behind a cylinder at moderate Reynolds number ($Re = \bar{U}D/\nu = 100$), following a well-known benchmark [6]. The configuration of the simulation is visible in Fig. 2. A parabolic velocity profile is used as the inflow (left wall), while a no slip boundary condition is applied at the cylinder walls and the upper and lower bounds of the domain. A free outflow condition is applied at the right wall. At this Reynolds number, a clear vortex shedding pattern takes place. Pressure probes (black dots) are used to provide the state observation S_t, while two jets normal to the cylinder wall are located at the top and bottom of the cylinder (red dots), which provides the action A_t. Finally, the drag coefficient $C_D = F_D/(0.5\rho\bar{U}^2D)$ is used as the reward R_t.

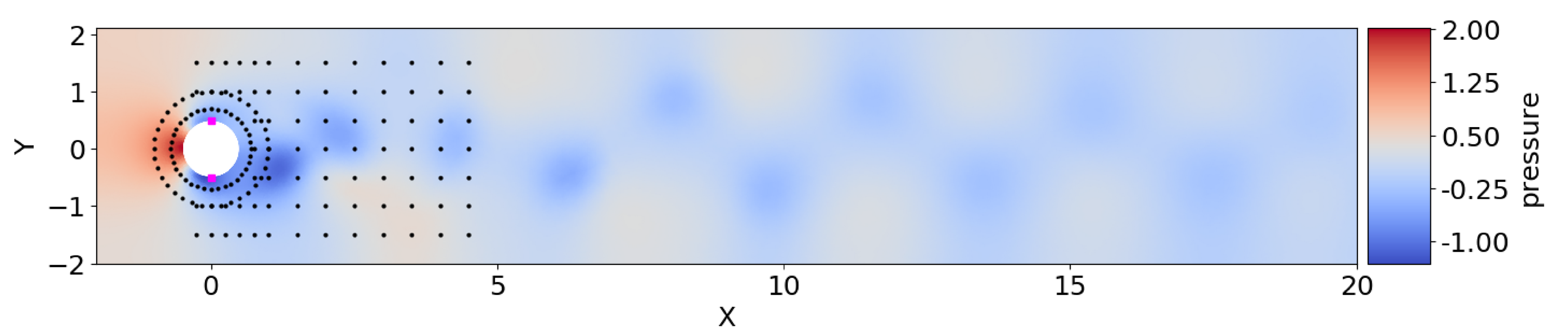

Figure 2: Configuration of the 2D flow simulation. The Karman alley is clearly visible. Black dots indicate pressure probes (state S_t), red dots the jets (action A_t), and the drag coefficient is used as the reward ($R : t$).

The ANN used is a simple fully connected network with two layers of 512 neurons each, using a ReLU activation function. We use the PPO algorith [1, 2, 5], which belongs to the Policy Gradient family. Training takes place in around 200 epochs, which takes around 24 hours on a modern desktop computer using one single core. Over 95% of the computation time is spent in the fluid dynamics simulation, rather than the training of the ANN itself.

Results

After training, the control strategy is able to efficiently reduce drag. As visible in Fig. 3, the wake of the cylinder is strongly modified by the active control, which results in a drag reduction of around 8%.

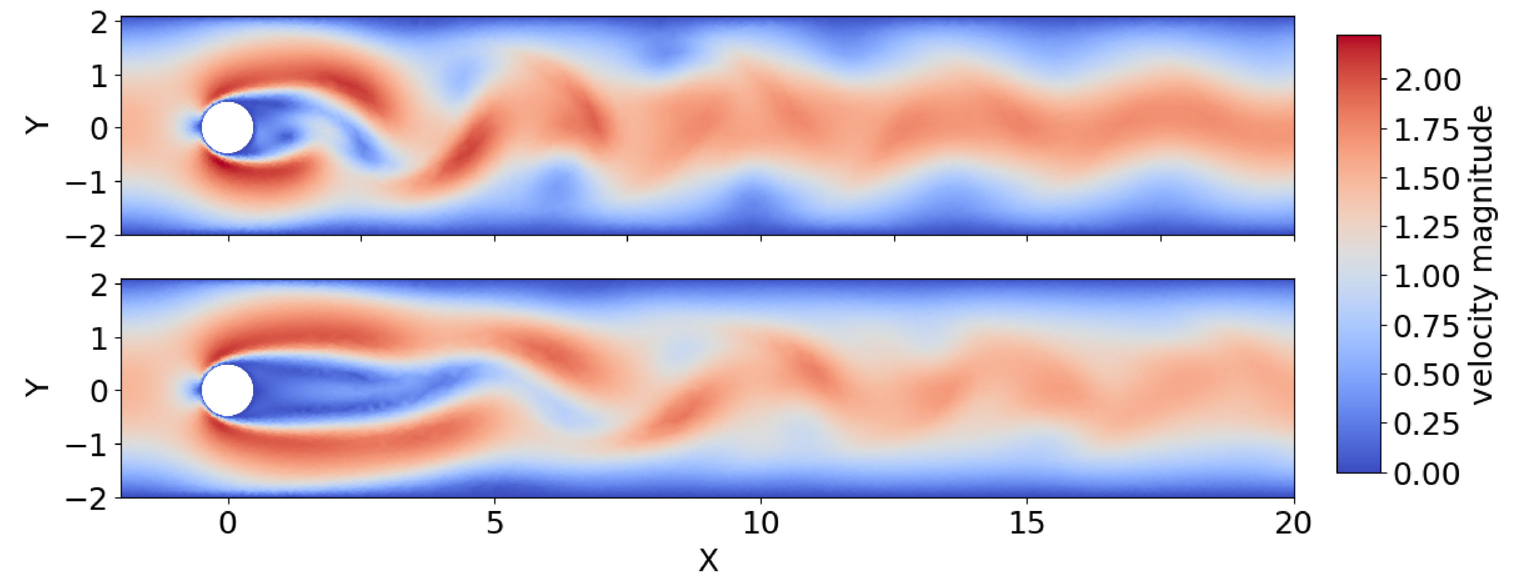

Figure 3: Snapshot of the velocity field in the baseline case (no actuation, top) and with active flow control (bottom). The wake configuration is clearly changed by the active control.

More details about the time dependent drag coefficient and control signal are presented in Fig, 4. As visible there, a transient regime is first encountered at the beginning of the actuation. During this transient regime, a relatively strong actuation is used to change the flow configuration. Following this phase, the network performs only small corrections to keep the flow in this new configuration.

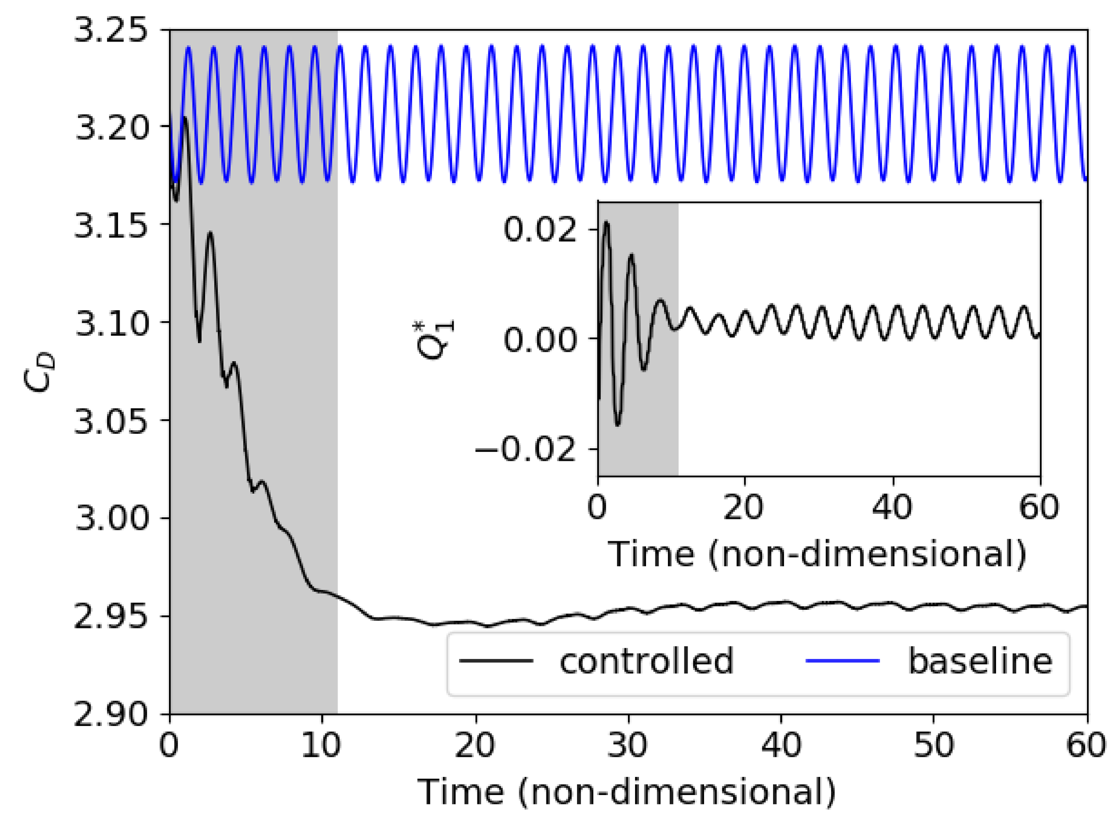

Figure 4: Time series of the drag coefficient C_D and normalized control rate Q_1^*. Two phases are clearly visible: first an initial transient takes place, before a pseudo periodic regime is reached.

More insight into the flow configuration can be obtained by looking at the mean and RMS of the velocity, as presented in Fig. 5. As visible there, the active control reduces the velocity fluctuations and elongates the recirculation bubble, in a similar fashion to what would be observed with boat tailing.

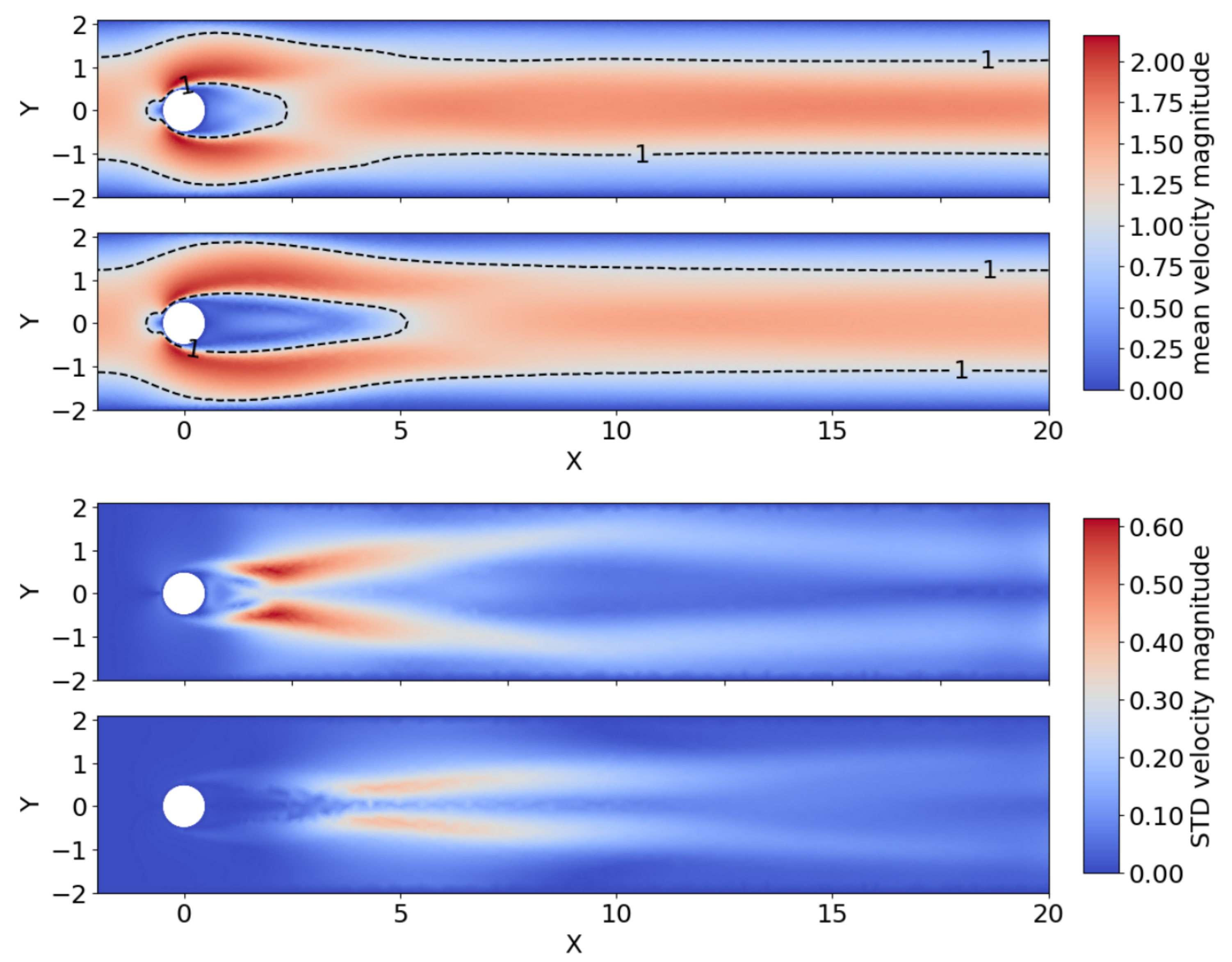

Figure 5: The mean (top double figure) and RMS (bottom double figure) of the velocity magnitude in the case without (upper of each double figure) and with (lower of each double figure) active flow control.

References

[1] Human-level control through deep reinforcement learning, Mnih et. al., Nature (2015).

[2] Mastering the game of Go with deep neural networks and tree search, Silver et. al., Nature (2016).

[3] Artificial Neural Networks trained through Deep Reinforcement Learning discover control strategies for active flow control, Rabault et. al., ArXiv (2018).

[4] Machine Learning Control Taming Nonlinear Dynamics and Turbulence, Duriez et. al., Springer (2017).

[5] Proximal Policy Optimization Algorithms, Schulman et. al., ArXiv (2017).

[6] Benchmark Computations of Laminar Flow Around a Cylinder, Schäfer and Turek, Springer (1996).