ICWE 16
August 27-31, 2023
FLORENCE - ITALY
16th International Conference on Wind Engineering

DRLinFluids: An open-source Python platform of coupling deep reinforcement learning and OpenFOAM

Qiulei Wang [1], Lei Yan [1], Gang Hu [1,2,3]

[1] *School of Civil and Environmental Engineering , Harbin Institute of Technology ,Shenzhen 518055, China*
[2]*Shenzhen Key Laboratory of Intelligent Structure System in Civil Engineering , Harbin Institute of Technology , Shenzhen 518055,China*
[3] *Guangdong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and Engineering Applications , Harbin Institute of Technology ,Shenzhen518055,China,*
hugang@hit.edu.cn

SUMMARY:
We propose an open-source Python platform for applications of deep reinforcement learning (DRL) in fluid mechanics. DRL has been widely used in optimizing decision making in nonlinear and high-dimensional problems. Here, an agent maximizes a cumulative reward by learning a feedback policy by acting in an environment. In control theory terms, the cumulative reward would correspond to the cost function, the agent to the actuator, the environment to the measured signals, and the learned policy to the feedback law. Thus, DRL assumes an interactive environment or, equivalently, a control plant. The setup of a numerical simulation plant with DRL is challenging and time-consuming. In this work, a novel Python platform, namely DRLinFluids, is developed for this purpose, with DRL for flow control and optimization problems in fluid mechanics. The simulations employ OpenFOAM as a popular, flexible Navier–Stokes solver in industry and academia, and Tensorforce or Tianshou as widely used versatile DRL packages. The reliability and efficiency of DRLinFluids are demonstrated for two wake stabilization benchmark problems. DRLinFluids significantly reduces the application effort of DRL in fluid mechanics, and it is expected to greatly accelerate academic and industrial applications.

Keywords: Deep Reinforcement Learning, OpenFOAM, DRLinFluids

1. INTRODUCTION

DRL has been extensively and successfully applied to various fluid mechanics problems, and considerable efforts are now being devoted to coupling state-of-the-art CFD solvers and DRL control algorithms, and deploying these effectively on high-performance computing (HPC) systems. In the previous studies, the interactive CFD environments almost relied on self-programing partial differential equation (PDE) solvers based on the lattice Boltzmann method or Navier−Stokes equation. The employed in-house solvers require significant effort and experience to achieve the efficiency and reliability of well-developed open-source CFD solvers such as OpenFOAM, particularly for complex flow fields. By contrast, OpenFOAM has been developed for more than ten years, and its accuracy and robustness have been demonstrated by substantial research. Many computational fluid mechanics courses employ OpenFOAM nowadays.

Moreover, OpenFOAM is also increasingly used in the industry.

Currently, there is no general and mature platform for simplifying the application of DRL in OpenFOAM simulations. This paper proposes an open-source Python platform, DRLinFluids, for coupling DRL and OpenFOAM based on reliable DRL packages, including Tensorforce(A. Kuhnle 2017) and Tianshou(J. Weng 2021). DRLinFluids is a flexible platform to utilize DRL in the field of computational fluid mechanics, even in continuum mechanics. Two case studies of active flow control of bluff bodies are conducted to demonstrate the feasibility and reliability of DRLinFluids successfully. We release DRLinFluids under an open-source license on GitHub (https://github.com/venturi123/DRLinFluids), and we expect that this open-source platform will greatly simplify and accelerate the relevant research and application of DRL in fluid mechanics, bluff-body aerodynamics, and wind engineering.

2. METHODOLOGY

2.1. Deep reinforcement learning

Modern reinforcement learning algorithms are mainly based on the Markov decision process (MDP), which can be roughly divided into two topics, including model-based and model-free. Since the model-free method neither cares about the environment model nor needs to learn the model itself, there is no problem of inaccurate environment fitting, which means that it is relatively easier to implement and train.

At present, there are many well-known deep reinforcement learning frameworks, such as DeeR, Dopamine, OpenAI base-lines, Coach, and Acme. Most of them are developed based on Python, with PyTorch and TensorFlow as automatic gradient solvers. Tensorforce(A. Kuhnle 2017) and Tianshou(J. Weng 2021) are used as an alternative ready-to-use backends to provide a wide range of reinforcement learning algorithms in the DRLinFluids platform, including DQN, A3C, PPO, TD3, SAC, and GAE. In addition, custom DRL algorithms can also be easily integrated into the DRLinFluids by inheriting the built-in DRLinFluids' DRL class, which means all the model-free DRL algorithms are sup-ported by DRLinFluids theoretically.

2.2. OpenFOAM and deep reinforcement learning coupling schemes

To ensure the flexibility of DRLinFluids, a low-level RL class, and high-level training and evaluation platform are defined separately. The object-oriented programing (OOP) paradigm is adopted to design the RL class to improve code reusability, which organizes platform design around data and objects.

2.2.1. Low-level DRL class

As shown in Figure 1, the class includes four parts, which are wrap-per, runner, logger, and extractor. The extractor function can handle many types of CFD quantities, not only the basic flow information like velocity, pressure, turbulence kinetic energy, etc., obtained from the probe as you have mentioned but also the lift force (or coefficient), drag force (or coefficient), or moment (or coefficient) or quantities derived from the basic flow information, such as vorticity and y+. In addition, other types of quantities are also considered according to flexible user-defined RegExps.

2.2.2. High-level platform

To further simplify the application of DRL coupled with OpenFOAM based on the DRL class, we introduce Tensorforce(A. Kuhnle 2017) and Tianshou(J. Weng 2021) as the backend algorithm and builds a high-level reinforcement learning training and evaluation platform.

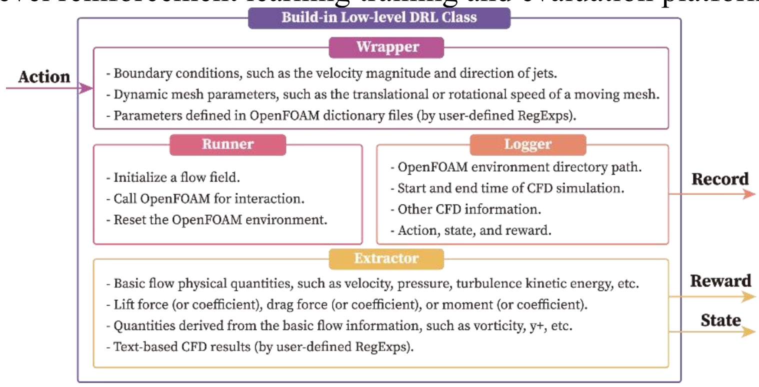

Figure 1. Architecture of the build-in low-level DRL class in DRLinFluids that comprises of four parts, including wrapper, runner, logger, and extractor.

3. CASE STUDIES

To demonstrate the feasibility and reliability of DRLinFluids, two case studies of flow control of bluff bodies are tested based on DRLinFluids. The first case is an adaptation of a circular cylinder flow control conducted by Rabault *et al*(J. Rabault 2019). The second case is a square cylinder flow controlled by jet flow at the trailing edges of two sides.

3.1. Cases: Active flow control of a 2D circular cylinder and square cylinder

The use of DRLinFluids successfully carries out a jet actuation control policy, which remarkably reduces the drag and lift force acting on the circular and square cylinders, and suppresses vortex shedding in their wake region. To find the optimal feedback control, the policy network is continuously updated during interaction to maximize the reward function, which takes both drag and lift coefficients into consideration. As shown in Figure 2 and Figure 3, a significant drag reduction by up to about 8% and 13.6% occurs for circular cylinder case and square cylinder case, respectively. In addition, the lift fluctuations are also suppressed to a very low level, and the regular vortex shedding has been destroyed by the jet flow.

It is noteworthy that similar control strategies are obtained despite two different reinforcement learning algorithms being adopted in the two cases (PPO for cylinder and SAC for square). The jet velocity starts at a relatively large jet flow, fluctuates weakly, and eventually reaches a very low level. The corresponding ratio of the jet velocity to the incoming wind velocity is only 0.58% and 3.443% for the circular cylinder and square cylinder cases, respectively. Apparently, the DRL agent has found an effective and efficient solution to maintain a small drag and a very weak lift for the cylinders. It shows that under a low Reynolds number (Re=100), active flow control developed based on DRLinFluids can effectively eliminate the adverse effect of vortex shedding on drag and lift of both circular and square cylinders with only a small amount of hyper-parameter tuning required.

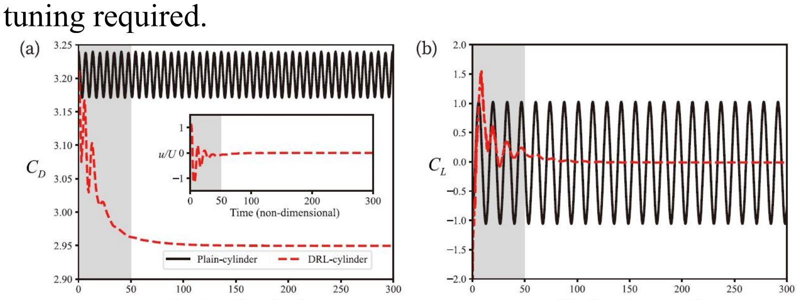

Figure 2. (a) and (b) Temporal variations in drag coefficient C_D and lift coefficient C_L for the cylinder without (plain-cylinder) and with (DRL-cylinder) active flow control, and normalized velocity flow rate of the jet flow.

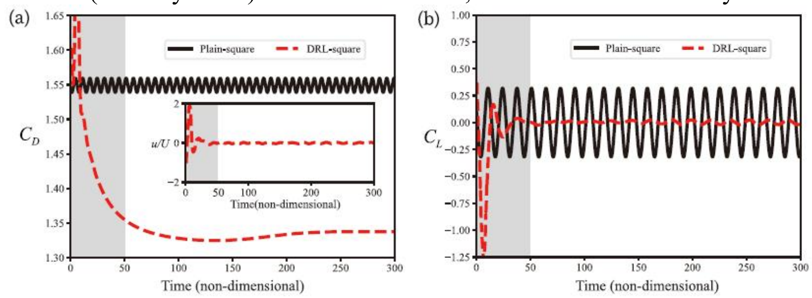

Figure 3. (a) and (b) Temporal variations in drag coefficient C_D and lift coefficient C_L for the cylinder without (plain-square) and with (DRL- square) active flow control, and normalized velocity flow rate of the jet flow.

4. CONCLUSIONS AND FUTURE WORK

This study has developed a flexible and user-friendly platform, DRLinFluids, for connecting reinforcement learning with fluid mechanics, and two case studies have been tested for validation. DRLinFluids uses one of the most used open-source CFD software, OpenFOAM, as the environment to interact with the agent in different reinforcement learning algorithms from a built-in or a custom DRL backend.

These two validation cases demonstrate the effectiveness and reliability of DRLinFluids in DRL-based active flow control. More generally, since OpenFOAM has powerful application functions in CFD, we believe that DRLinFluids have a much broader application outlook in the field of fluid mechanics, which can bridge the gap between cutting-edge DRL algorithms and traditional fluid mechanics research or rapid implementation of DRL applications in the relevant industries. DRLinFluids is, therefore, an important step toward the application of DRL control to full-scale, realistic engineering configurations. The next steps in the development of DRLinFluids include the development of 3D active flow control benchmarks, testing on large-scale supercomputers, and the development of automated methodologies to define invariants and symmetry-aware strategies that take advantage of the structure of the dynamic system to control, which are the key elements needed to apply DRL on increasingly complex flow control problems.

ACKNOWLEDGEMENTS

This study was supported by the National Key R&D Program of China (No. 2021YFC3100702), National Natural Science Foundation of China (Nos. 52108451, 12172109, and 12172111), Shenzhen Science and Technology Innovation Commission (No. GXWD2020123015542700320200823230021001), Shenzhen Key Laboratory Launching Project (No. ZDSYS20200810113601005), and Guangdong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and Engineering Applications (No. 2020B1212030001) and by Natural Science and Engineering Grant No. GD-2022A151501149 of Guangdong Province, China.

REFERENCES

A. Kuhnle, et al. “Tensorforce: A TensorFlow library for applied reinforcement learning,” Github. https://github.com/tensorforce/tensorforce.(2017)

J. Rabault, et al, “Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control,” J. Fluid Mech. 865, 281–302 (2019).

J. Weng, et al, ”Tianshou: A highly modularized deep reinforcement learning library,” e-print arXiv:2107.14171 (2021).