Machine Learning For Fluid Dynamics, Workshop
6^{th}–8^{th} March 2024,
Sorbonne University,
Paris, France

Towards practical applications of deep reinforcement learning in computational fluid dynamics

Saeed Salehi[1,2] and Håkan Nilsson[1]
[1]Chalmers University of Technology, Gothenburg, Sweden
[2]Chalmers Industriteknik, Gothenburg, Sweden
{saeed.salehi,hakan.nilsson}@chalmers.se

The latest advancements in artificial intelligence and machine learning have enabled addressing complex challenges in high-dimensional control and decision-making. Deep Reinforcement Learning (DRL), merging deep learning with reinforcement learning, is able to execute highly intricate cognitive tasks at a superhuman level capabilities.

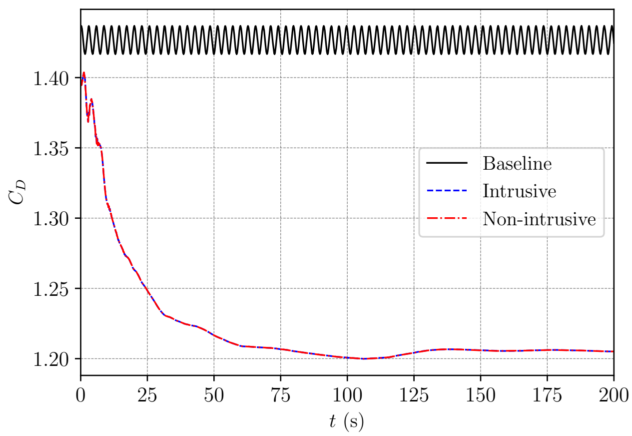
DRL has found application in fluid mechanics, notably in Active Flow Control (AFC) [1], aimed at mitigating adverse effects or amplifying desirable flow features. The optimization of AFC strategies through conventional methods often entails highly nonlinear problem-solving and the design of numerous parameters. In contrast, DRL excels in learning complex AFC strategies, fully harnessing actuator capabilities. Rooted in the reinforcement learning paradigm, DRL navigates the state-action-reward sequence, presenting a robust tool for executing closed-loop feedback control.

Over the last few years, numerous endeavors documented in the literature have aimed to leverage DRL within Computational Fluid Dynamics (CFD), leading to the development of different coupled DRL-CFD frameworks (e.g., Refs. [2, 3]). However, most of these initiatives primarily center on a *non-intrusive* implementation of the algorithm in which the CFD solver is approached as a black box, with the DRL agent communicating information to and from it in an *offline* manner. Typically, such frameworks necessitate the CFD simulation to be stopped after each DRL (control) time step to communicate information (state-action-reward sequence) with the DRL program and restart the CFD after receiving a new action.

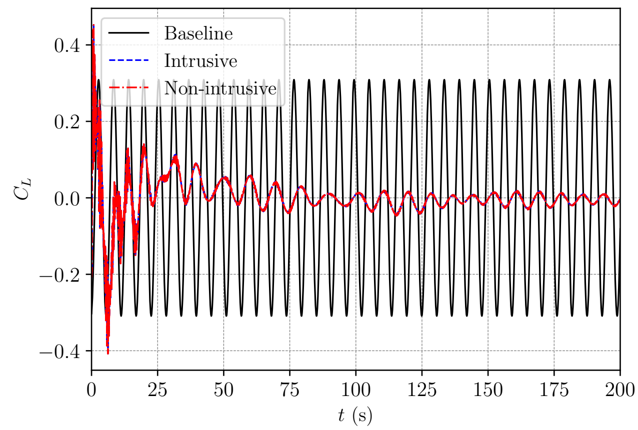
While these frameworks have been demonstrated to be effective and are able to perform complicated control tasks, their efficiencies pose a notable bottleneck, hindering their practical utilization within engineering applications. The frequent interruptions in the CFD simulations slow down the DRL-CFD framework significantly and demand remarkable I/O operations.

The focus of the current work is on extending the capabilities of the DRL algorithm for practical CFD applications through introducing an *intrusive* DRL-CFD algorithm in which the DRL agent is embedded within the general open-source CFD solver OpenFOAM. The implementation allows for executing complete DRL episodes seamlessly, eliminating the need for any extra I/O requirement. Moreover, the parallelization is enabled on both ends of the framework, i.e., DRL and CFD. Essentially, this setup enables not only each CFD simulation to utilize parallel processing through classical domain decomposition techniques, but also allows the DRL algorithm to interact simultaneously with multiple CFD environments.

To evaluate and verify the performance of the developed algorithm, the widely known test case of active flow control of vortex shedding behind a 2D cylinder is revisited. The actuator is a pair of synthetic jets on top and bottom of the cylinder. The reward function is defined as the reduction of drag and the absolute value of lift. Thereby, the DRL agent (which is a deep neural network here) learns to minimize the drag and lift coefficients by applying the optimum jet flow at each time step. The DRL agent was trained through a total of 500 CFD simulations. For the sake of comparison, the DRL-CFD framework is also implemented in a non-intrusive manner in which the CFD solver is treated as a black box. Figure 1 compares the variation of drag and lift coefficients of the cylinder for intrusive, non-intrusive, and baseline (no active



(a) Drag coefficient



(b) Lift coefficient

Figure 1: Variation of (a) drag and (b) lift coefficients for the intrusive, non-intrusive, and baseline (uncontrolled) cases.

flow control) cases. The intrusive DRL-CFD is able to maximize the reward function by minimizing the drag and lift coefficient while presenting results that match the non-intrusive framework.

Acknowledgments

The research presented in this work was carried out as a part of "Swedish Centre for Sustainable Hydropower - SVC". SVC has been established by the Swedish Energy Agency, Energiforsk and Svenska kraftnät together with Luleå University of Technology, Uppsala University, KTH Royal Institute of Technology, Chalmers University of Technology, Karlstad University, Umeå University and Lund University, `www.svc.nu`.

The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS) and the Swedish National Infrastructure for Computing (SNIC) at NSC and C3SE partially funded by the Swedish Research Council through grant agreements no. 2022-06725 and no. 2018-05973.

References

[1] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi, "Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control," *Journal of Fluid Mechanics*, vol. 865, pp. 281–302, 2019.

[2] Q. Wang, L. Yan, G. Hu, C. Li, Y. Xiao, H. Xiong, J. Rabault, and B. R. Noack, "Drlinfluids: An open-source python platform of coupling deep reinforcement learning and openfoam," *Physics of Fluids*, vol. 34, no. 8, p. 081801, 2022.

[3] J. Li and M. Zhang, "Reinforcement-learning-based control of confined cylinder wakes with stability analyses," *Journal of Fluid Mechanics*, vol. 932, p. A44, 2022.