SCIENCE CHINA
Information Sciences

• LETTER •

November 2018, Vol. 61 119204:1–119204:3
https://doi.org/10.1007/s11432-018-9463-x

Missile aerodynamic design using reinforcement learning and transfer learning

Xinghui YAN[1,2], Jihong ZHU[1,2*], Minchi KUANG[1,2] & Xiangyang WANG[1,2]

[1]*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China;*
[2]*State Key Laboratory of Intelligent Technology and System, Tsinghua University, Beijing* 100084*, China*

Received 28 February 2018/Accepted 3 May 2018/Published online 17 September 2018

Citation Yan X H, Zhu J H, Kuang M C, et al. Missile aerodynamic design using reinforcement learning and transfer learning. Sci China Inf Sci, 2018, 61(11): 119204, https://doi.org/10.1007/s11432-018-9463-x

Dear editor,

Aerodynamic design is usually a time-consuming process of four steps [1]. First, an initial design profile is obtained with designer's domain knowledge. Second, the design profile is represented as a CAD model using softwares such as Catia or UG. Third, a CAE software, such as ICEM or Hypermesh, is applied to generate corresponding meshes. At last, a computational fluid dynamics (CFD) software, such as Fluent or CFX, is used to calculate the performance. Therefore, only numbered of design configurations could be evaluated. The design efficiency and results need to be improved desperately.

Some researchers focused on offering alternatives to the most time-consuming CFD simulation with surrogate model, such as Kriging and multiple adaptive regression splines (MARS) [2,3]. Kutz [4] used deep neural networks (DNNs) to model complex flows such as turbulent flows. Besides, a developing estimation method based on experimental data and empirical equation is also a good choice, such as Datcom [5]. However, these methods have limitations. Surrogate model is only accurate for limited data field. Using DNNs to model complex flows suffers from lacking training data. Engineering estimation software is suitable for large design space, but the results are only for reference in terms of accuracy.

Reinforcement learning (RL) is gaining more and more attention recently and has been successfully applied to many challenging problems. DNNs can extract a high-level representation of raw data. Thus, RL along with DNNs offers a new way for aerodynamic design. We propose a deep learning approach with shared-layer deep deterministic policy gradient (SL-DDPG) to design aerodynamic shape, combining state-of-the-art RL method [6] and transfer learning (TL) [7]. DDPG is used to extract design rules from a semi-empirical method in continuous action space with high resolution while TL is used to accelerate the learning process with a CFD method. To the best of our knowledge, this study is the first to use RL and TL for aerodynamic design.

Model and methodology. We consider an aerodynamic design problem for a normal layout missile as shown in Figure 1(a). The objective is to maximize the lift-drag ratio coefficient and maintain the position of pressure center. The entire design architecture is illustrated in Figure 1(b). We apply DDPG in two related tasks: aerodynamic design in semi-empirical software environment and CFD environment, which are the source task and the target task, respectively. The source task intends to extract the hidden design rules, which is a novel approach to utilize the data and empirical equations involved in Datcom. In the target task, the DDPG network initialized with transferred features focuses on obtaining high performance de-

* Corresponding author (email: jhzhu@tsinghua.edu.cn)

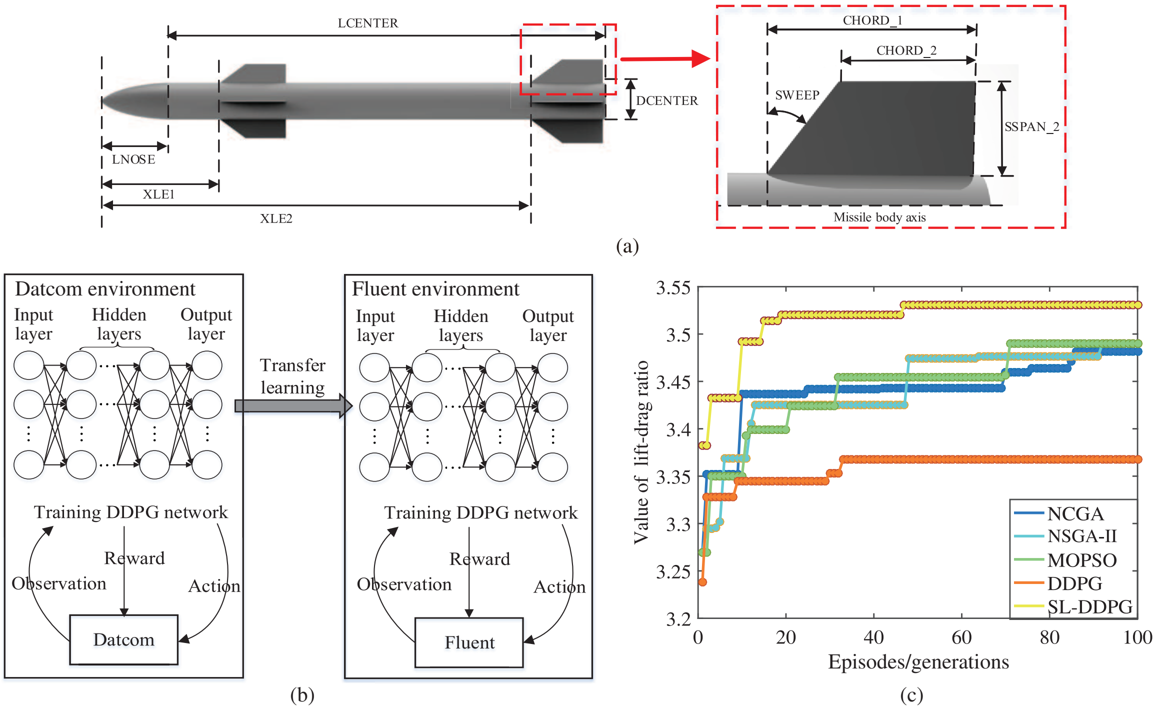

Figure 1 (a) Design parameters of the missile; (b) architecture of the deep learning approach for aerodynamic design; (c) performance comparison of NCGA, NSGA-II, MOPSO, DDPG, and SL-DDPG in the hybrid environment.

sign configurations precisely in Fluent, which is a widely used high-precision CFD software.

For the source task, the iterative interaction between the agent and the Datcom environment is realized by a Python program which writes the input file, executes Datcom code and reads the output file in sequence. We apply DDPG to counter extract the design policy which counts for the improvement of aerodynamic performance by rewarding the increase of lift-drag ratio and penalizing its degradation and pressure center variation. Aerodynamic design process can be essentially regarded as a Markov decision process (MDP) which basically meets RL's requirement. Taking the learning agent as a virtual designer, the t^{th} design configuration as state $s_t \in S$, the change for the t^{th} design configuration as action $a_t \in A$, the final design configuration is achieved by iteratively interacting between the agent and the environment. Then the extracting of design rules is equal to the optimal policy π^*: $S \rightarrow A$ which returns the best action at each step.

For the target task, we aim to eliminate the calculation deviation between Datcom and Fluent and obtain design configurations precisely. In DNNs, the former layers extract common features while the latter ones focus on the specific features of the particular task. Therefore, both the input and hidden layers are shared as a common feature transformation, and the output layers are not shared because the calculation methods are different in Datcom and Fluent. Only the output layer is re-trained on the target task as an adaption of environment switching.

To further improve the design efficiency, a hybrid model of Datcom and Fluent is employed. Although there may be calculation deviation between Datcom and CFD, Datcom could still show the change tendency properly with parameters variating. Hence, the hybrid model takes advantage of Datcom's speed and Fluent's accuracy by setting the calculation results of Fluent as reference value and fitting the change tendency with Datcom. In the hybrid model, the estimation of Fluent result after taking action a_t at state s_t is defined as

$$\begin{aligned}&\tilde{T}_F\left(s_t, a_t, s^{t'}, a^{t'}\right)\\&= T_F\left(s^{t'}, a^{t'}\right) + \varepsilon\left(s_t, a_t, s^{t'}, a^{t'}\right),\end{aligned} \tag{1}$$

where $T_F(s^{t'}, a^{t'})$ denotes the exact result of Fluent after taking action $a^{t'}$ at state $s^{t'}$, which acts as the baseline value. ε is the prediction for the deviation between $\tilde{T}_F(s_t, a_t)$ and $T_F(s^{t'}, a^{t'})$ which can be calculated as

$$\begin{aligned}\varepsilon\left(s_t, a_t, s^{t'}, a^{t'}\right) &= \frac{T_D(s_t, a_t) - T_D(s^{t'}, a^{t'})}{T_D(s^{t'}, a^{t'})}\\&\quad\cdot T_F\left(s^{t'}, a^{t'}\right),\end{aligned} \tag{2}$$

where $T_D(s_t, a_t)$ is the exact result of Datcom after taking action a_t at state s_t. The state-action pair $(s^{t'}, a^{t'})$ is chosen according to the match degree M after traversing the Fluent case base:

$$M\left(s_t, a_t, s^{t'}, a^{t'}\right) = 1 - \sum_{i=1}^{n} \frac{1}{n} \cdot \left| \left[s_t(i) - s^{t'}(i)\right] + \left[a_t(i) - a^{t'}(i)\right] \right| \Big/ \left[s^{t'}(i) + a^{t'}(i)\right]. \quad (3)$$

The hybrid model is iteratively checked by the maximum match degree and the consistency of the two best matched predictions in the Fluent case base. The consistency is calculated as

$$C\left(s_t, a_t, s^{t'}, a^{t'}, s^{t''}, a^{t''}\right) = 1 - \left|\tilde{T}_F\left(s_t, a_t, s^{t'}, a^{t'}\right) - \tilde{T}_F\left(s_t, a_t, s^{t''}, a^{t''}\right)\right| \Big/ \min\left(\tilde{T}_F\left(s_t, a_t, s^{t'}, a^{t'}\right), \tilde{T}_F\left(s_t, a_t, s^{t''}, a^{t''}\right)\right). \quad (4)$$

When the maximum match degree is below 95% or the consistency is below 99%, a Fluent calculation will be carried out to update the hybrid model with respect to s_t and a_t.

The interaction of the agent and the hybrid model is also realized by a Python program. Previously, a 3D baseline missile model is built using Catia and a corresponding mesh file is generated with ICEM. In the Fluent calculation, mesh morphing technique [8] is introduced to adjust the missile geometry and mesh according to the agent's action without re-modeling.

Experiment and results. The design configurations are evaluated by lift-drag ratio and position variation of pressure center, which are common criteria for aerodynamic design. DDPG and SL-DDPG are applied to search for excellent missile shape configurations along with three widely used optimization algorithms in the aerospace domain, namely NSGA-II, NCGA and MOPSO. The experimental details of the proposed method are described in Appendix A.

In the experiment of target task using hybrid model, NSGA-II and NCGA both had 100 generations of 20 individuals each. MOPSO had 100 iterations and swarm size of 20. DDPG and SL-DDPG both adopt max episodes of 100 and max steps of 20. Thus, the calculation amounts of the above algorithms during the optimization are nearly the same. As illustrated in Figure 1(c), SL-DDPG outperforms other algorithms significantly in both search speed and objective value. The lift-drag ratio of SL-DDPG reaches 3.49 at 10^{th} episode while other algorithms achieve such performance at 71^{th} generation or later. In terms of obtaining similar final solution, SL-DDPG saves up to about 85% calculation time with the knowledge transferred from pre-trained layers.

Conclusion and future work. This study presents a novel approach for challenging missile aerodynamic design problem using reinforcement learning and transfer learning. We use DDPG to learn the optimal design policies from semi-empirical method. Then these policies are transferred to the target task in hybrid environment. The shared layers accelerate the learning process significantly compared with the methods without transfer learning. Our experimental results also demonstrate that SL-DDPG outperforms NCGA, NSGA-II and MOPSO significantly in both convergence speed and search capability which offers great value for time-consuming aerodynamic design.

Further investigation should be carried out in studying the adaptability for more related design tasks, along with the detail influences of DNNs' hyper parameters to the design process.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 61603210) and Aeronautical Science Foundation of China (Grant No. 20160758001).

Supporting information Appendix A. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 van Nguyen N, Tyan M, Jin S, et al. Adaptive multifidelity constraints method for efficient multidisciplinary missile design framework. J Spacecr Rockets, 2016, 53: 184–194

2 Ni A, Zhang Y F, Chen H X. An improvement to NSGA-II algorithm and its application in optimization design of multi-element airfoil. Acta Aerodynamica Sin, 2014, 32: 252–257

3 Giacché D, Xu L, Coupland J. Optimization of bypass outlet guide vane for low interaction noise. AIAA J, 2014, 52: 1145–1158

4 Kutz J N. Deep learning in fluid dynamics. J Fluid Mech, 2017, 814: 1–4

5 Zhang W D, Wang Y B, Liu Y. Aerodynamic study of theater ballistic missile target. Aerosp Sci Tech, 2013, 24: 221–225

6 Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. 2015. ArXiv:1509.02971

7 Oquab M, Bottou L, Laptev I, et al. Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, 2014. 1717–1724

8 Biancolini M E, Costa E, Cella U, et al. Glider fuselage-wing junction optimization using CFD and RBF mesh morphing. Aircraft Eng Aerosp Tech, 2016, 88: 740–752