Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Motion Optimization for a Robotic Fish Based on Adversarial Structured Control

Shuaizheng Yan[1,2], Jian Wang[1,2], Zhengxing Wu[1,2], Junzhi Yu[1,3], and Min Tan[1,2]
[1]*State Key Lab Management and Control for Complex Systems, Institute of Automation, CAS, Beijing 100190, China*
[2]*University of Chinese Academy of Sciences, Beijing 100049, China*
[3]*Dept. Mech. Eng. Sci., BIC-ESAT, College of Engineering, Peking University, Beijing 100871, China*
{*yanshuaizheng2018, wangjian2016, zhengxing.wu, junzhi.yu, min.tan*}*@ia.ac.cn*

Abstract— **This paper proposes a task-based control optimization method for the robotic fish. It is essentially an adversarial structured control consisting of a global control module and a local compensation control module. In detail, the global control module emulates an optimized central pattern generator with Evolutionary Strategy, while the local control module produces targeted compensation control signals with Soft Actor-Critic. The linear summation of two control laws works for the final robotic fish control. Considering that the evolutionary computation optimization algorithms generally have the defect of falling into the local optimum, we propose a method of antagonistic training to improve the optimization performance. The effectiveness of the designed controller is validated by simulation on agents in Mujoco. Noticeably, the simulation results demonstrate that the proposed method teaches the agent fish to move to any target point with a low energy consumption, which lays a good foundation for application of reinforcement learning in real robotic fish control.**

Index Terms— **Robotic fish, Adversarial training, Structured control**

I. Introduction

As a typical underwater robot, bionic robotic fish is playing an increasingly important role in education, hydrological monitoring, biological motion analysis and other fields. Basically, effective locomotion control can help the robot fish to realize fast, stable and energy-saving swimming and complete complex tasks better [1]. Recently, the research results on the motion optimization methods of bionic robotic fish are emerging one after another. However, many control theory studies focus solely on improving one aspect of its capabilities, such as high performance or low power consumption. It is difficult to directly theoretically derive a controller satisfying with both high performance and low power consumption. Fortunately, Deep Reinforcement Learning (DRL) seems to provide a good solution to the problem about multi-goal optimization of high-dimensional continuous control, but its feasibility and accuracy has been questioned.

In this work, we provide an effective method based on adversarial structured control (ASC) for a robotic fish aiming at improving control performance while reducing energy consumption. The ASC consists of two independent control modules, namely global control module and local compensation control module. In detail, the control law of the global control module is the rhythmic signal generated by the central pattern generators (CPGs), while the local module offers an extra compensation signal through DRL. On the one hand, we can optimize a finite number of CPGs parameters by evolutionary calculation. On the other hand, the compensation control law is generated according to the real pose of the robot fish to coordinate the global control, so as to achieve a higher control accuracy. Besides, thanks to the assumption of two modules independence, various optimization approaches can be employed, such as state-of-the-art DRL and evolutionary computation methods.

Compared with the previous works, the main contributions of this paper lie in two aspects. First, we design a simulation agent fish based on the physical parameters of a real robotic fish, and train the agent in Mujoco physical simulation platform with simple hydrodynamic model [2]; Second, we tested the control effect of ASC with the agent fish, proving that our approach can effectively optimize the control of robotic fish to achieve the unification of high performance and low power consumption. Thus, the proposed method provides a valuable paradigm for using DRL to control robotic fish.

II. Related Work

Recently, DRL has been an important topic in the research of complex continuous control. There are some great teams working on machine learning in robotics. Skydio and Wayve have produced excellent results in UAV control industry and driverless vehicle field, respectively, but they are more based on the combination of deep learning and computer vision technology. Their representative achievements conclude deep stereo regression [3], SegNet [4], and PoseNet [5]. Besides, Robotic AI & Learning Lab of Berkeley has made outstanding contributions to create more DRL theoretical research and innovative learning methods, such as the hot Soft Actor-Critic method [6].

Among them, DRL algorithms for real-world robots have provided potential prototype for practical applications. Levine *et al.* created a case of hand-eye robot training

[*] This work was supported in part by the National Natural Science Foundation of China under Grant 61603388, Grant 61973303, Grant 61633017, and Grant 61836015, in part by the Beijing Natural Science Foundation under Grant 4192060, in part by Youth Innovation Promotion Association CAS (2019138), and in part by Pre-Research Fund of Equipments of China under Grant 61403120108.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

with large-scale data collection [7]; Ebert *et al*. focused on teaching robots new skills with self-supervised model-based approach [8]; Pong *et al*. presented the temporal difference models with high learning efficiency and asymptotic performance thanks to combination of model-based and model-free training methods [9]; Srouji *et al*. made efforts to explore adding inductive bias for improving sampling efficiency using structured control network [10].

However, for robotic fish focusing on the bionic mechanism, the lack of data, visual feedback and the limitation of computing resource makes these large-scale data collection methods unable to put into full play. Hence, the locomotion control of bionic robotic fish adopts traditional methods, such as backstepping sliding control, fuzzy control, etc. Moreover, CPGs are increasingly used to control the rhythmic movements, especially swimming in robotic fish. For instance, Yu *et al*. provided a lot of outstanding work in the field of traditional control of bionic robotic fish combined with CPGs [11], [12]. In order to reduce the dependence of DRL methods on high computing resource and improve the effect of intelligent control, we proposed that the optimized CPGs were simply used to generate the rhythmic signal as the global control, and the compensation control produced by DRL was added to explore a better control law for enhancing its robustness and stability.

III. Adversarial Structured Control Architecture

In this section, we develop an architecture for policy net π_θ based on the prior information of robot motion and attitude feedback. The whole control system is split into two parts, namely optimized global control module with transcendental knowledge and compensation control module, as shown in Fig. 1.

Intuitively, according to the prior knowledge of the rhythmic movement of the robotic fish, an optimized rhythmic signal is what we essentially demand. Hence, the global control module optimizes CPGs parameters with evolutionary strategies (ES), while the residual compensation module stabilizes the local dynamics around global control using Soft Actor-Critic (SAC) algorithm. Finally, the summation of two signals is exactly the final control law, as shown in following formula:

$$a_t = u_t^g + u_t^l, \tag{1}$$

where a_t, u_t^g, u_t^l are the final control law, global control signal and local compensation signal at each moment, respectively. It is worth noting that decoupling of control method is inspired by structured control net (SNC) [10]. However, the main differences between our method and the principle of SCN are the division and fitting of actions, which is given by:

$$\begin{cases} a_t = u_t^s + u_t^e = u_t^n(s_t, s_t^d) + K \cdot s_t, & SCN \\ a_t = u_t^s + u_t^e = u_t^g(s_t, s_t^d) + (s_t - s_t^d), & ASC \end{cases} \tag{2}$$

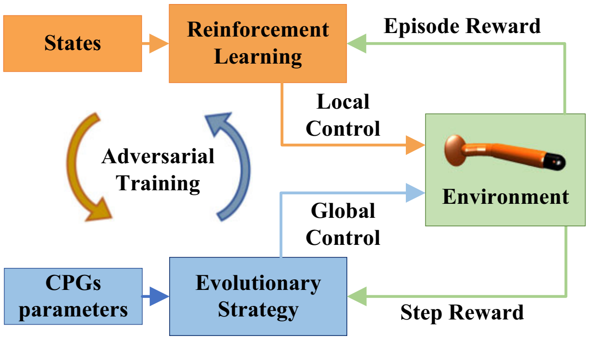


Fig. 1. Adversarial structured control architechture.

where $u_t^s(s_t)$ is a nonlinear term presented by a sequence of CPGs parameters. SCN uses the summation of 16 trig functions to represent the Fourier series of CPGs. It trains a multilayer perceptron, input and output of which are agent states and trig function parameters, respectively. Furthermore, the linear network is designed by applying the nonlinear network as the bias of the last layer node. Although this structure seems intuitive, it is difficult to train a sufficiently robust nonlinear control network for real robots. Therefore, our approach weakens the dependence of the nonlinear network on the real-time state, and only provides a general control strategy. The deviation caused by this change is compensated by local compensation control, which improves the feasibility of this method in real world robot applications. That is, for global control module, the control signal decoded from a set of CPGs parameters directly applies to the servo angles. With regard to the linear control module, we replace the linear product term with the random action strategy obtained by SAC algorithm.

A. Net Architecture

Biological research indicates that the rhythmic swimming of fish is controlled by CPGs. In general, CPGs refer to the neural circuits in the central nervous system. Through the mutual suppression of neurons, it achieves self-excited oscillation and generates stable periodic signals. Therefore, CPGs are widely used in the field of multiple degree-of-freedom (multi-DOF) robot locomotion control [13]. Since the control object is a robotic fish, we choose Hopfield nonlinear oscillator with more intuitive and simple parameters to construct CPGs. The global control module adopts ES algorithm [14] to optimize CPGs parameters, including amplitude, frequency, phase difference, and bias.

For a procession of robotic fish control, each step of its action has a lasting impact, and it is hard to design a suitable value function estimation, so the Evolutionary Strategy is a better choice than the strategy gradient. Generally, ES algorithms set the goal function with the following estimator:

$$\mathbb{E}_{\theta \sim p_\psi} F(\theta) = \mathbb{E}_{\varepsilon \sim N(0,I)} F(\theta + \sigma\varepsilon), \tag{3}$$

where $F(\cdot)$ denotes the reward estimated from an environment, θ indicates the parameters of a deterministic policy π_θ in environment, which is exactly CPGs parameters for the robotic fish. Therefore, we optimize θ using stochastic gradient ascent with the following function estimator:

$$\nabla_{\theta}\mathbb{E}_{\varepsilon\sim N(0,I)}F(\theta+\sigma\varepsilon)=\frac{1}{\sigma}\mathbb{E}_{\varepsilon\sim N(0,I)}\left\{F\left(\theta+\sigma\varepsilon\right)\varepsilon\right\}, \quad (4)$$

which is approximated by sampling different gradients randomly. In order to accelerate the optimization of ES, we also employ a trick of mirror sampling [15].

As mentioned above, the residual compensation network provides stability and complements in complex tasks. Studies have shown that stochastic policy commonly works better for real robot control. Meanwhile, SAC algorithm is considered to be a relatively effective solution to the continuous control problem because of incorporating the maximum entropy theory to ensure that advantageous actions preserved in policy. In particular, SAC algorithm is employed in this method, and optimization objective is designed as follows:

$$J(\pi)=\sum_{t=0}^{T}\mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t,a_t)+\alpha\mathscr{H}(\pi(\cdot|s_t))\right], \quad (5)$$

where $\mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t,a_t)\right]$ denotes standard expected sum of rewards in reinforcement learning (RL), α is defined as the temperature parameter determining the relative importance of the entropy term against the reward, and $\mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\mathscr{H}(\pi(\cdot|s_t))\right]$ represents the stochasticity of the optimal policy.

According to SAC, we designed a policy net and a critic net. The policy network outputs the action corresponding to each moment state, and then evaluates the action by the critic network. Particularly, we should astrict the output of the policy network, since its job is to generate a small amount of control signals to calibrate and improve the motion,

$$a_l=K_l\cdot tanh(\mu_t+\sigma_t\cdot\mathscr{D}_{sample}), \quad (6)$$

where $\mathscr{D}_{sample}$ represents the data sampled from a normalization, μ_t and σ_t are parameters of compensation action distribution, and K_l is designed as a weight ranged $0.01\sim 0.03$ to restrict output which can ensure that the steering gear does not exceed the rated speed.

B. Training Methods

We train the proposed ASC with several state-of-the-art training methods, i.e., ES for global control module, Deep Deterministic Policy Gradient (DDPG) and SAC for the compensation control module. Specifically, there are some effective tricks in training.

1) From the view of network structure, the DRL based on the Actor-Critic architecture is similar to Generative Adversarial Network (GAN) [16]. Hence, researchers used to focus on the similarities and differences between two kinds of model training. In the process of training, we find that in order to improve the efficiency and implementation feasibility, the accuracy of global control module is reduced to some extent. The output result, after only one round optimization of two control modules, is less satisfactory. Therefore, we draw on the advantages of GAN training strategy, and train the two control modules by turns. Namely, we first train the global control module by off-line to obtain a group of better CPGs parameters providing the initial control law for the robotic fish, and then alternately fix one module and optimize the parameters of the other module for over 3 rounds, as shown in Fig. 2.

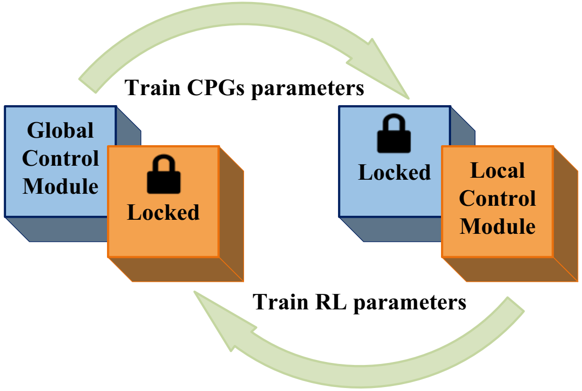


Fig. 2. Adversarial training strategy.

2) The training principle of the model is intelligible. In detail, optimizations of both global and local compensation control parameters are performed alternately for the same objective function. The only difference lies in that RL usually uses the single-step reward R to calculate the gradient, while the gradient of ES is obtained from the total reward G of each episode.
In the training, our optimization goal is to improve the endurance ability on the premise of ensuring the robotic fish accomplish the swimming tasks. Consequently, we design a suitable evaluation function as follow:

$$\begin{aligned}\mathbf{max}\quad & J_\Psi=\cos(\theta_e)\cdot v_m-\beta\cdot\tau\times\dot{\theta}_j, & (7)\\ \text{s.t.}\quad & v_m\leq v_o, & (8)\end{aligned}$$

where Ψ denotes the parameter of CPGs. θ_e represents the yaw angle. v_m and v_o are current and optimized speed of the robotic fish, respectively. τ presents the torque vector of the steering gear. $\dot{\theta}_j$ means the angular velocity vector. Besides, the updating strategy improves the speed $\cos(\theta_e)\cdot v_m$ towards the target point, and reduces the energy consumption from the formula $P=\tau\times\dot{\theta}_j$. In order to keep a balance between impact of speed enhancement reward and energy cost, we restrictively set an upper speed limitation, that is, the speed in the optimization process will not exceed the initial speed corresponding to the optimized parameters. In addition, β is designed to determine the relative importance of the consumption term against the reward.

3) Compared with single DRL applying multilayer perceptrons (MLPs) control method, our approach has fewer parameters. Moreover, the policy net and critic net are two fully-connected MLPs with two hidden layers each, and each of which consists of 32 units as well as *tanh*

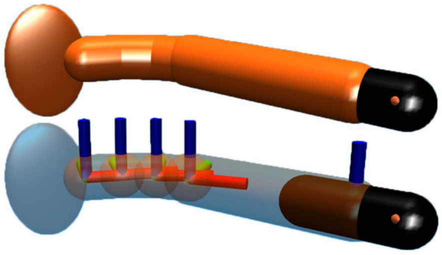


Fig. 3. Robotic fish model in Mujoco.

TABLE I. Physical Parameters of the Robotic Fish.

Variable	Unit	L_0	L_1	L_2	L_3	L_4
m_i	kg	1.528	0.159	0.159	0.171	0.091
l_i	m	0.291	0.062	0.062	0.062	0.137
c_i	m	0.18	0.044	0.044	0.045	0.037
$I_{i,z}$	kg· m^2($\times 10^{-4}$)	290	1.8	1.8	2.0	1.6

[1] m_i, l_i, c_i, and $I_{i,z}$ represents mass, length, the position on the x-axis in local coordinate, and the moment of inertia with respect to z-axis in local coordinate of each linkage, respectively.

nonlinearities. This network structure applies not only to SAC methods, but also to DDPG. The difference is that DDPG directly outputs the required joint Angle for position servo, while SAC takes the mean value generated by actor network as the compensation control signal. By comparison, we find that owing to the advantage of SAC, the training result is not sensitive to the small change of node numbers and learning rate. Therefore, we employ SAC as the algorithm architecture of compensation control module. However, the output value scaling factor K_t of the strategy network, as well as the exploration parameter σ in ES optimization, should be specially designed with hardware structure of different robots. In addition, it is worth noting that in ES optimization, we use the trick of mirror sampling to improve the efficiency of optimization.

IV. Results and Analyses

In order to verify the effectiveness of the proposed method, dynamic simulations are conducted with an agent fish in Mujoco. After achieving good demonstration of the control optimization results, the speed and power consumption of agent fish under open loop control and ASC are compared and analyzed.

A. Simulation Platform

Fig. 3 is a simulated agent fish designed in Mujoco. To reduce fluid resistance, the robotic fish adopts a slender body shape of the pike. The head contains a yaw direction of freedom, capable of flexible rotation in ±50°. The robotic fish is about 614 mm long and weighs 2.21 kg. It is based on physical parameters of robotic fish, which are shown in the Table I [17].

In simulation, we assume that the robotic fish only produces energy consumption and yaw motion through the tail structure, thereby only four hinge joints equipped with servo motors under position control are designed to simulate the joints motions of the real robotic fish. In addition, the position feedback coefficient K_p in servo control is set to 4.

B. Simulation Results

In the simulation, we designed two different control training tasks:

1) The robotic fish moved from (0, 0) to (4, 0). In addition to this, the task requires the robotic fish to save the energy as far as possible, but the linear speed cannot decrease significantly.
2) The robotic fish moves from (0, 0) to (2.5, 2.5). Meanwhile, the robotic fish should save the energy during the process.

Fig. 4(a)-4(e), 4(k)-4(o) show the motion sequence under an open loop control while the rest are under the closed loop control from the proposed method. Fig. 4(a)- 4(j) correspond to task 1, and the others correspond to task 2. As can be seen, the swimming path and posture of the robotic fish are terrible. Due to poor CPGs parameters resulting in incorrect swimming posture, the robotic fish is usually unable to complete the task, which also causes a lot of unnecessary energy loss. In addition, this set of motion sequences shows that this method has two obvious advantages. First, optimization in our approach is carried out on the basis of achieving the goal; Second, the optimization results are exactly in line with the results of motion control theory research on bionic robotic fish.

In order to verify the generalization ability of our training method, we fine-tune the attitude of the robotic fish in both tasks, and take CPGs parameters as the initial training state of the robotic fish. The training curve is plotted in Fig. 5. Two lines correspond to the training results of the above exercise sequence diagram. The shadow range represents the results of different initial CPGs parameters training.

In training stage, this method has a few training parameters. In the global control module, each steering gear only includes four parameters (i.e., amplitude, frequency, phase difference, and offset). Hence, there are 16 parameters for 4 steering gears to be optimized. Meanwhile, for each task, the global motion module can be directly optimized offline to obtain the satisfactory solution. Therefore, in the simulation, the total number of training for each task does not exceed 1000. However, in the task with higher complexity, the training difficulty increases correspondingly, and multiple confrontational training is required. Consequently, task 1 has basically converged in the third round of alternate training, while task 2 needs more time. In particular, in the confrontation training, random strategy in RL will lead to fluctuations in the original better results, but it will converge again after optimization. Through analysis, we find that generally optimizing the global control law can greatly reduce the energy loss, while learning the local compensation control law can help the robotic fish further improve the completeness and stability of the task. More importantly, as

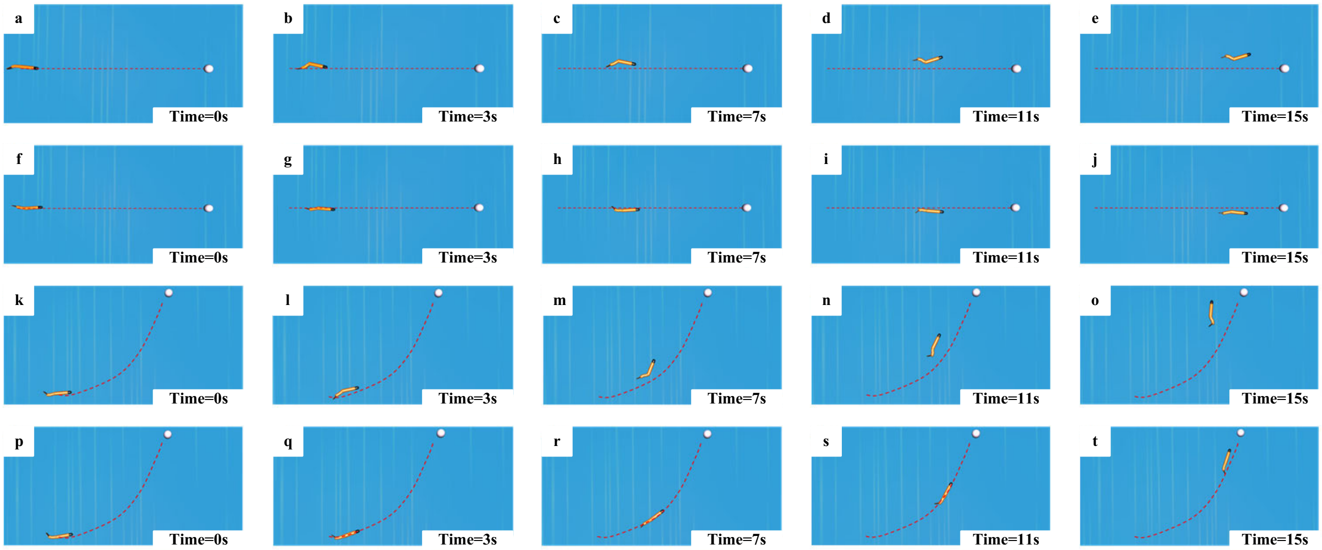


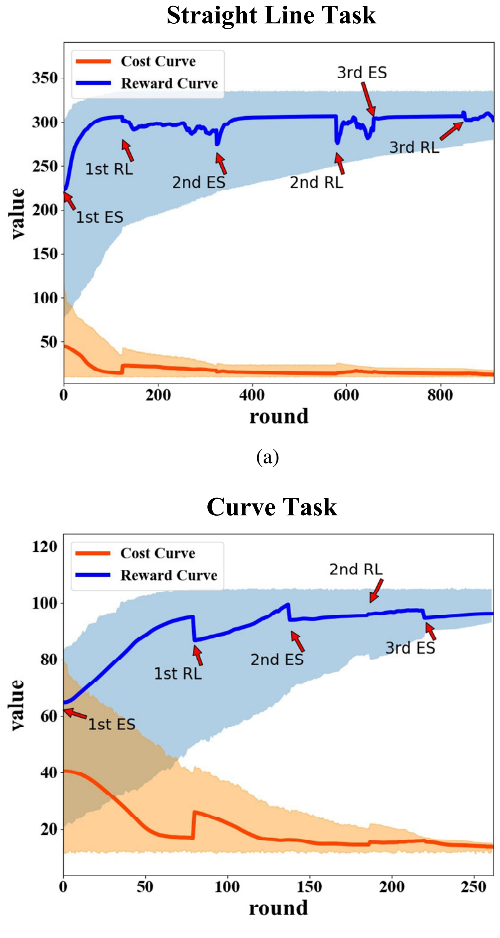
Fig. 4. Task 1 and task 2 motion sequence diagram of robotic fish. (a)-(j) Straight swimming effect of robotic fish under the control before and after optimization. (k)-(t) Curve swimming effect of robotic fish under control before and after optimization.

shown in the Fig. 5, after each exchange of training objects, the score of the objective function would fall first and then rise, which means that our approach helps the original control strategy to escape from the local optimum in search of a better solution.

Furthermore, we compare our method with ordinary closed-loop control in three ways.

1) In terms of speed, we set the straight swimming speed of the robotic fish to be 0.19 m/s before optimization, while our method resulted in a slight increase to 0.21 m/s. In the training, our reward function sets an upper limit on the speed reward, so as to ensure that the optimization effect of energy saving can not be affected by pursuing high speed, and meanwhile, the linear speed remains stable.
2) Energy consumption is the main item of our optimization objective. In the reward function, when the speed reward reaches the maximum, maximizing the score is equivalent to minimizing the power loss. The energy consumption in the task 1 was reduced by 72.42%. It is illustrated that the overall loss presents a downward trend regardless of the different initial CPGs parameters. What's more, thanks to the adversarial training, the two control modules can escape from the local optimum in the alternate training, and then start to optimize towards the new gradient direction. Therefore, it can be observed in Fig. 5 that the loss value, which tends to be stable after each optimization of ES and RL, can be further reduced.
3) In task 2, the robotic fish generated 14.8% deviation between the final and the target position under the open-loop control, while our approach reduces this to 0.8%. Besides, no matter in the open-loop control or the traditional closed-loop control, the robotic fish cannot achieve the goal and reduce the energy consumption at the same time. Nevertheless, via employing the proposed control optimization method, the robotic fish not only completed the task, but also saved energy. Meanwhile, the robotic fish also generated a characteristic motion track that we can then compare with tracks of real fish and study how to improve the motion efficiency.

Through simulation tests in Mujoco, we have verified the effectiveness of our method. On the one hand, when dealing with different tasks, for any set of CPGs parameters given randomly, our approach can obtain a new control law that enables the robotic fish achieve the complicated optimization goal through adversarial training. Considering the real-time requirements of robotic fish, we discussed the advantages and disadvantages of the proposed method and SCN. In the actual control of robotic fish, nonlinear motion training is difficult but vital to deal with complex tasks. However, the architechture of SCN focuses on the generation of control strategy by a nonlinear network, and the linear term only stabilizes the local dynamics around the residual of global control. Actually, this is to virtually improve the difficulty of training and weaken the feasibility of application on real robot. In allusion to complex problems, our proposed method only provides one or a few sets of CPGs parameters sequences by global control module, and makes full use of the advantage of compensation network to help robotic fish better complete the task. On the other hand, our approach creatively provides a practical and effective solution to the path tracking problem of robotic fish. In detail, the complex path tracking can be divided into subtasks of finite target points tracking. Then, through the independent training of these subtasks, the overall control law should be a combination. At the same time, the effective reduction of energy consumption has become a unique advantage of our



(a)

(b)

Fig. 5. Score and loss curves with different initial states for tasks 1 & 2.

control method.

V. Conclusions and Future Work

In this paper, we have proposed a novel motion optimization method for a robotic fish based on adversarial structured control. As a typical ASC architecture, the proposed method consists of a global control module and a local compensation control module. Specially, the former employs the Evolutionary Strategy to optimize the initial parameters of CPGs, and the latter trains the compensation control law for each subgoal and the current state of the robotic fish through Soft Actor-Critic. The global control module uses the Evolutionary Strategy to optimize the initial parameters of the central pattern generators. The local compensation module trains the compensation control law for each subgoal and the current state of the robotic fish through Soft Actor-Critic. During the training, we alternately lock one module, and undated the other one. Both simulation and experimental results demonstrates the benefits of adversarial training mode: avoiding getting trapped in the local optimum, effective control of real robotic fish, lower energy consumption and more flexible task completion ability. Based on the proposed method, the robotic fish can accomplish the complex motion task and reduce the motion energy loss as far as possible. Furthermore, it also provides a reference control scheme for using DRL to control robotic fish.

For the future work, we plan to complete the validation of the effectiveness of ASC on a real bionic robotic fish first. Then, the real-time information (position, posture and environment) of the robotic fish is extended to the global control module, and the algorithm is optimized to improve the real-time performance of the method, so as to finally accomplish the verification experiment on the robotic fish platform with high-speed computing resource.

References

[1] R. Du, Z. Li, Y. Kamal, and V. Pablo, “Robot fish,” Berlin, Heidelberg: Springer, 2015, vol. 10, no. 9, pp. 3–987.

[2] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” In *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Vilamoura, Algarve, Portugal, Oct. 2012, pp. 5026–5033.

[3] A. Kendall, H. Martirosyan, S. Dasgupta, P. Henry, R. Kennedy, A. Bachrach, and A. Bry, “End-to-end learning of geometry and context for deep stereo regression,” In *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 66–75.

[4] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Trans. Pattern Anal.*, vol. 39, no. 12, pp. 2481–2495, 2017.

[5] A. Kendall, M. Grimes, and R. Cipolla, “Posenet: A convolutional network for real-time 6-dof camera relocalization,” In *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 2938–2946.

[6] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” In *Proc. Int. Conf. Learn. Representations*, Vancouver Canada, Apr. 2018, arXiv:1801.01290.

[7] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection,” *Int. J. Robot. Res.*, vol. 37, no. 4-5, pp. 421–436, 2018.

[8] S. Tian, F. Ebert, D. Jayaraman, M. Mudigonda, C. Finn, R. Calandra, and S. Levine, “Manipulation by feel: Touch-based control with deep predictive models,” In *Proc. IEEE Int. Conf. Robot. Autom.*, Montreal, Canada, May 2019, arXiv:1903.04128.

[9] V. Pong, S. Gu, M. Dalal, and S. Levine, “Temporal difference models: Model-free deep rl for model-based control,” In *Proc. Int. Conf. Learn. Representations*, Vancouver, Canada, Apr. 2018, arXiv:1802.09081.

[10] M. Srouji, J. Zhang, and R. Salakhutdinov, “Structured control nets for deep reinforcement learning,” In *Proc. Int. Conf. Mach. Learn.*, Stockholm, Sweden, Jul. 2018, arXiv:1802.08311.

[11] J. Yu, M. Wang, H. Dong, Y. Zhang, and Z. Wu, “Motion Control and Motion Coordination of Bionic Robotic Fish: A Review,” *J. BIONIC ENG.*, vol. 15, no. 4, pp. 597–598, 2018.

[12] J. Yu, M. Tan, J. Chen, and J. Zhang, “A survey on CPG-inspired control models and system implementation,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 441–456, 2013.

[13] J. Yu, Z. Wu, M. Wang, and M. Tan, “CPG network optimization for a biomimetic robotic fish via PSO,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1962–1968, 2015.

[14] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, “Evolution strategies as a scalable alternative to reinforcement learning,” *arXiv preprint arXiv:1703.03864*, 2017.

[15] D. Brockhoff, A. Auger, N. Hansen, D. V Arnold, and T. Hohm, “Mirrored sampling and sequential selection for evolution strategies,” In *Proc. Int. Conf. Parall. Prob. Solv. Nat.*, Krakow, Poland, Sep. 2010, pp. 11–21.

[16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” In *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, Canada, Dec. 2014, pp. 2672–2680.

[17] Z. Wu, J. Yu, Z. Su, and M. Tan, “An improved multimodal robotic fish modelled after Esox lucíus,” In *Proc. IEEE Int. Conf. Robot. Biomimetics*, Shenzhen, China, Dec. 2013, pp. 516–521.