2020 Australian & New Zealand Control Conference (ANZCC)
26-27 November 2020. Gold Coast, Australia

Autonomous Boundary of Human-Machine Collaboration System Based on Reinforcement Learning

Qianqian Zhang[1], Yun-Bo Zhao[2], Yu Kang[*]

2020 Australian and New Zealand Control Conference (ANZCC) | 978-1-7281-9992-4/20/$31.00 ©2020 IEEE | DOI: 10.1109/ANZCC50923.2020.9318326

Abstract— **This paper provides a human-machine collaborative control framework, including artificial intelligence decision systems, human-level control, arbiter judgment, and learning of autonomous boundary, so that human suggestions are incorporated into the training process of decisions, assisting agents to learn quickly control decision tasks. Based on the model-free deep reinforcement learning algorithm HITL-AC, the human feedback (reward or punishment) is connected with the reward of the agent, so that the agent continuously tries to find a better boundary during the system's operation, avoiding defects of pre-fixed boundary. This formulation improves the data efficiency of reinforcement learning and plays a guiding role in seeking human intervention when the agent is in an uncertain environmental state during the test use phase. The fourth section of the paper gives a training demonstration of a real-time environment (bipedal walker). Compared with existing standard reinforcement learning methods that do not consider boundary concepts, the method with boundary information mentioned in this article can accelerate the process of agent reinforcement learning during the training phase, and seek human help when guiding the dangerous state of the agent during the test phase. And the real-time optimization algorithm (HITL-AC) for the boundary is better than the fixed value algorithm (HITL-FIX). This is beneficial for solving real-world problems, further proving the feasibility and effectiveness of the proposed framework and method.**

I. INTRODUCTION

Artificial Intelligence (AI), as a cutting-edge technology that is most likely to change the future world, its development has made machines have certain intelligent and autonomous capabilities. How to combine machine intelligence with human intelligence has been a hot research topic for many researchers on human-machine fusion control systems [1] [2] [3] [4]. Thanks in part to advances in intelligent systems such as the internet of things and cyber physical systems (CPS) [5], human-machine integration [6] [7] has evolved to varying degrees, such as measuring physical indicators and assisting human with disabilities in sports, flexible wearable devices, unmanned systems in the driving field, and intelligent weapon systems in the military field, etc. [8] [9] [10] [11]. Literature [12] puts human in the loop, and proposes to use POMDP as a unified framework to handle the alert service of the driving system. And literature [13] discusses the shared autonomy of human-machine interaction under the framework of optimal control theory, and further considers the user-driven system as a constrained nonlinear optimization problem. Literature [14] proposes a comprehensive learning framework including two stages: learning to grasp and learning to move, realizing the task of a humanoid robot capable of moving a table in cooperation with people. Interestingly, Lex [15] proposed an arguing machine framework based on the idea of integration that the primary and secondary systems solve the same control task at the same time, and use the differences between them to decide whether human intervention as a supervisor is needed. Lin [16] explores probability, credibility check, and consistency check through arbitration, and decides whether the system adopts the agent's decision or the suggestion given by the human database, and gives the verification of the 3D game world. Different from [16], Siddharth [13] proposes a model-based deep reinforcement learning method for a class of machines where the control target representation, dynamic transfer model and user's policy cannot be known. The above-mentioned human-machine collaboration system attempts to determine the decision-making authority of human partners and intelligent machines through the arbitration mechanism, without considering a clear boundary perspective. In this paper we will seriously consider the handover boundary problem of the human-computer collaboration system.

Since both humans and machines have a certain degree of intelligence, it is necessary to consider the control range of the their respective decisions. Based on the randomness and unpredictability of artificial intelligence, we study the boundary problem of decision theory of human-machine systems based on artificial intelligence. This research will contribute to the in-depth development of human-machine integration. The boundary of the machine in the traditional two-layer human-machine fusion system is relatively clear, and the fusion only includes the two layers of automatic execution of the machine and human control. In comparison, the three-layer human-machine cooperative control system considered in this paper includes automatic execution of the machine, intelligent autonomy of the machine, and human control. For example, human-machine cooperative systems widely existing in actual engineering and daily life: autonomous weapon systems in the military field, semi-autonomous driving, remotely controlled surgery, smart homes, wearable de-

[*]Corresponding author. This work was supported by the National Natural Science Foundation of China under Grants (61673361, 61725304, 61673350), National Key Research and Development Projects of China (2018AAA0100800, 2018YFE0106800), and the Major Science and Technology Projects of Anhui Province under Grant (912198698036)

[1]Qianqian Zhang is with the Department of Automation, University of Science and Technology of China, Hefei, 230027, China `ZQQ789@mail.ustc.edu.cn`

[2]Yun-Bo Zhao is with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310023 China `ybzhao@ieee.org`

[*]Yu kang is with the Department of Automation, the State Key Laboratory of Fire Science, University of Science and Technology of China, and Institute of Advanced Technology, Hefei 230027, China `kangduyu@ustc.edu.cn`)

vices, etc, which all involve more complex boundary issues. How to deal with the conflicts between the autonomous decision-making ability of artificial intelligence technology and human decision-making, in other words, can we obtain a dynamic boundary that changes with the environment and complete the division of decision-making authority between artificial intelligence and human intelligence?

To solve the problems mentioned above, the main contributions made by this article include: we propose a human-machine fusion intelligent framework and a model-free deep reinforcement learning algorithm to connect human feedback (reward or punishment) with the reward of the agent. In addition, the framework allows the agent to constantly try and find better boundary during the operation of the system. Finally, the proposed method can guide the agent to accelerate the training speed during the training phase, and improve the defects of the simple reinforcement learning method with low data efficiency (requires a large amount of episodes training), and can guide the agent to seek human's intervention in the test phase. We achieve the goal of cooperation between intelligent agent decision-making and human control to complete control tasks. The section II of the paper introduces relevant background knowledge, based on which, the main modeling and learning methods of this paper are proposed (section III). The simulation experiment in the section IV of the paper proves the effectiveness of the proposed method.

II. PRELIMINARY

Markov Decision Process (MDP) [17] is a model for describing latent stochastic sequential decision problems. The third section of this paper introduces the system's Markov Decision Process (MDP) modeling and specific solutions. Before this, this section first introduces the basic background knowledge of Markov decision process and reinforcement learning.

A. Markov Decision Process

The Markov decision process is based on the Markov stochastic process as a theoretical basis and considers the quadruple (S, A, P, R), where S represents the state set in the decision process. A represents the set of actions in the decision process. P is called the state transition probability between states. R is the reward function after taking the action to reach the next state. MDP studies how to find the optimal strategy, that is, the strategy when the maximum return is expected. MDP is often used in the modeling of reinforcement learning problems in machine learning. Through the use of dynamic programming, random sampling, and other related methods, MDP can solve the strategy that maximizes the cumulative reward, which is then applied in the control decision system.

In order to simplify the model, the Markov property of state transition is generally assumed, that is, the probability of transition to the next state is only related to the state at the current moment, and has nothing to do with the previous historical state.

$$P(s_{t+1} = s'|s_t = s, s_g) = P(s_{t+1} = s'|s_t = s) \quad (1)$$

$0 \leq g \leq t-1$. Similarly, the Markov hypothesis is made for the action (strategy) function and the action value function:

$$\pi(a_t = a'|s_t = s, s_g) = \pi(a_t = a'|s_t = s) \quad (2)$$

$$q(s_t = s, a_t = a|s_g, a_g) = Q(s_t = s, a_t = a) \quad (3)$$

Dynamic programming (DP) is a type of method solved by decomposing the original problem into subproblems. The problem applicable to dynamic programming needs to satisfy two properties: 1) constructing the global optimal solution (optimal substructure) from the optimal solution of the subproblem; 2) the subproblem repeats many times, and the result of the subproblem can be stored for reuse (overlapping sub-problems). Therefore, DP has become a general method for solving MDP problems.

B. Reinforcement Learning

Reinforcement learning focuses on how agents take different actions in the environment to maximize the cumulative rewards. In addition, the reinforcement learning task is learned and solved based on MDP. In other words, the actual system can be analyzed to obtain the corresponding MDP model representation, and then the reinforcement learning task is established to solve the control decision problem in the actual process.

Solving the reinforcement learning problem is actually solving the optimal strategy. The optimal strategy can be obtained by solving the optimal value function, and the solution of the optimal value function is to optimize the Bellman equation. In short, the solution of reinforcement learning finally evolves into optimization Bellman equation.

$$Q(s_t, a_t) = R(s_t, a_t) + \gamma \sum_{s_{t+1}} P(s_t, a_t, s_{t+1}) Q(s_{t+1}, a_{t+1}) \quad (4)$$

Because the interaction between the agent and the environment is similar to that between humans and the environment, it can be considered that reinforcement learning is a general learning framework that can be used to solve a series of artificial intelligence decision-making problems. The basic components of reinforcement learning include:

- State: data information of the environment in which the agent is located. The state set is a set of all possible states;
- Action: The action made by the agent according to the state, the action set is a set of all the actions of the kennel;
- Reward: The reward signal obtained after the agent acts the action;
- Strategy: The mapping relationship between the state and action learned by the agent is called strategy;
- Goal: The goal of reinforcement learning is to find strategies to maximize long-term cumulative rewards.

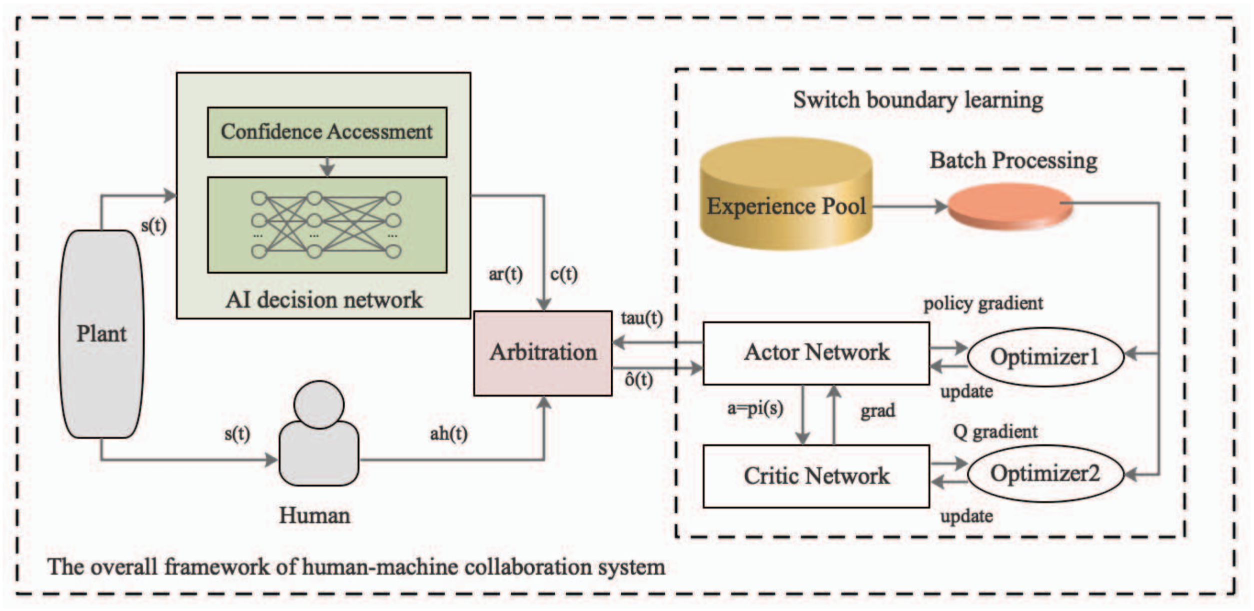

Fig. 1: The overall framework of human-machine collaboration system

The literature [18] and [19] proved that the result of reinforcement learning iterative algorithm is convergent (strategy iteration converges to π^*; value iteration converges to V^*). The contraction mapping theorem is described as :

Lemma 1: The operator F on the normed vector space $\mathcal{X}$ is contracted by γ, for $0 < \gamma < 1$, $x, y \in \mathcal{X}$, if

$$\| F(x) - F(y) \| \leq \| x - y \| \tag{5}$$

This section provides basic theoretical support for the method proposed and applied in the third section. It supports the relationship between MDP modeling, reinforcement learning.

III. Methodology

The model-free reinforcement learning method provides us with convenience. Since there is no need to build a model, all agent decisions are obtained through interaction with the environment, which is suitable for scenarios that are difficult or impossible to model. Observation of the environment and human guidance are used as input, so that the agent and human share the control of the plant. Not only can the human knowledge be used to solve the dilemma that the agent cannot deal with the determination of uncertain situations, but also the rapid decision of the agent can be used to solve the sudden situation where the human has no time to respond. And due to the integration of human knowledge, the learning process of the agent is accelerated. Fig. 1 shows the block diagram of the human-machine cooperative control system.

This paper provides a learning method for the switching boundary of the human-machine cooperative control system through reinforcement learning. To this end, we model the nine-tuple as follows:

$$(s_t, a_t, r_t, s_{t+1}, \tau_t, \tau_{t+1}, r_t^{'}, u_t, h_t) \tag{6}$$

Among them, s_t and s_{t+1} are the input state at the time of t and $t+1$, respectively, which can be numerical value, voice, image, etc. a_t is the action signal corresponding to the input state s_t at the time of t, and r_t is the reward generated by the state action for (s_t, a_t). Corresponding to the input state, the agent gives actions, and the human gives the state security level h_t. u_t represents the control signal of the human, which determines whether the plant is controlled by the human (1 represents human intervention, and 0 represents human non-intervention). τ_t, τ_{t+1} is the switching boundary of the human-machine collaborative system at the time of t and $t+1$, and it determines the timing of human-machine switching. $r_t^{'}$ represents the reward or penalty value generated by the state boundary for (s_t, τ_t).

After the modeling is completed, we seek the actor-critic algorithm [20] to solve the problem we proposed, that is, to complete the task efficiently, and to obtain the dynamic boundary of human-machine switching through training. The actor-critic algorithm combines the solution idea of value function approximation, which is divided into two parts: actor and critic. The actor is responsible for updating the strategy, and the critic is responsible for updating the action value function. Fig. 2 shows a schematic diagram of the actor-critic algorithm architecture. From Fig. 2 we can see that the algorithm combines two methods of strategy gradient and value function approximation. The actor selects an action based on probability, and the critic evaluates the action based on the action selected by the actor and gives a reward. The actor modifies the probability of subsequent selection actions based on the critic's evaluation. After the actor selects the action, the environment performs the action selected by the agent, and output feedback signals (reward and state) to the agent.

We give the algorithm flow as following algorithm 1.The

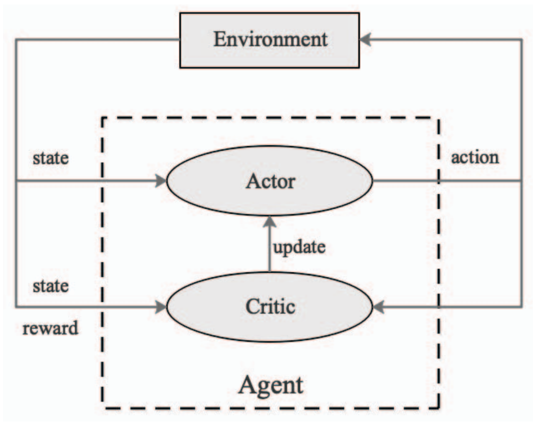

Fig. 2: A schematic diagram of the actor-critic algorithm architecture

input of the algorithm $\hat{s}_t$ is the output of the intelligent decision-making system, including the input state, action, credibility, and human acceptable judgment signal of the state. Taking autonomous driving as an example, h is the judgment of a person about the safety of the car at this moment (1 represents safety, 0 represents danger). The output of the algorithm is the switching boundary τ. This learning algorithm is based on the actor-critic algorithm framework. First, the weight parameters of the strategy network (actor) and value network (critic) are initialized, and the experience pool D is initialized (step 2). For each input tuple (s_t, s_{t+1}, h_t), the strategy network calculates the boundary τ_t (step 6). Comparing the size of the input c_t and τ_t, combined with the human judgment of the state s_t, the algorithm gives the control signal u_t whether humans need to intervene, and boundary estimate τ_{t+1} at next time step(Step 8), where c_t refers specifically to the state s_t and gives indicators of safety information, such as position, angle, etc. Step 9 stores the converted experience samples $(s_t, s_{t+1}, r_t, \tau_t, \tau_{t+1})$ to the experience pool. Randomly sample a small batch of N conversion experience samples from the experience pool, and calculate the target value and current value of i at each moment in each batch of experience samples (step 10). Then we update the value network Q_ω according to the time difference error calculated in step 10 (step 11). Based on the current value function $Q_\omega(s_i, \tau_i)$, the stochastic gradient ascent method is used to update the strategy network π_θ. The Δ in algorithm 1 represents the rate of boundary learning, which is preset to 0.001 here. The reward value is obtained by the state, action, and human control signals. The reward function here is in the following form:

$$r(s_t, a_t, u_t) = \begin{cases} r(s_t, a_t), & u_t = 0 \\ r(s_t, a_t) + r(s_t, a_t^h) & u_t = 1 \end{cases} \tag{7}$$

When there is no human intervention in the system, the reward function is the inherent reward function of reinforcement learning. For example, for the experimental bipedal walker in this paper, the agent has a reward for the walker to move forward and a penalty for falling. In addition, according to the motor torque applied in the whole process, a small amount of points should be subtracted. When $u_t = 1$, it means that human beings have objections to the decision made by the agent, so insert intervention to pull the agent back to a normal running state. At this time, the reward function includes two parts: the reward of the algorithm itself based on the state action pair, and the reward based on the current state and human action, where a_t^h is the representation of human intervention action. $r(s_t, a_t^h)$ can be given a feedback reward (or penalty) at each time step, or it can be given a feedback reward (or punishment) after each episode.

algorithm 1 Human-in-the-Loop-Actor-Critic(HITL-AC)

```
Initialization:
 1: Network input tuple: $\hat{s}_t = (s_t, s_{t+1}, h_t)$
 2: Randomly initialize the network parameters of the strategy network (actor) $\pi_\theta$ and value network (critic) $Q_\omega$, and their respective learning steps $\alpha^\theta, \alpha^\omega$; Randomly initialize the experience pool $D$.
Output: Human-machine switching system boundary $\tau_t$.
 3: repeat
 4:     input tuple $\hat{s}_t$
 5:     repeat
 6:         The strategy network $\pi_\theta$ calculates the threshold $\tau_t$ of the current time step.
 7:         $C = \{c_t \geq |\tau_t| \ or \ c_t \leq -|\tau_t|\}$
 8:         $\tau_{t+1} = \begin{cases} \tau_t - \Delta, u_t = 1 & C \text{ and } h_t = 0 \\ \tau_t + \Delta, u_t = 0 & C \text{ and } h_t = 1 \\ \tau_t, r'_t = 0, & \text{Otherwise} \end{cases}$
 9:         Store tuple $(s_t, s_{t+1}, r_t, \tau_t, \tau_{t+1})$ to experience pool $D$.
10:         Randomly sample small batches of $N$ experience samples $(s_i, s_{i+1}, r_i, \tau_i, \tau_{i+1})$from the experience pool to calculate the target value $y_i^{td} = r_i + \gamma Q_\omega(s_{i+1}, \tau_{i+1})$ and current value $y_i = Q_\omega(s_i, \tau_i)$.
11:         Update the parameters of the value network (critic) $Q_\omega$ according to the stochastic gradient descent method:$CLoss = \frac{1}{N}\sum_i (y_i - y_i^{td})^2$
12:         Update the parameters of the strategy network (actor) $\pi_\theta$ according to the stochastic gradient ascent method: $ALoss = \frac{1}{N}\sum_i Q_\omega(s_i, \tau_i)$
13:     until Reaching the end state or maximum time step.
14: until The reward function reaches convergence.
```

IV. Experimental simulation

We use the bipedal walker in OpenAI Gym as a demonstration experiment, as shown in Fig. 3. Bipedal walker provides a simulation environment that can establish background environment, terrain, bipedal walkers and angle, speed information. In addition, a python-based library PyTorch [1] is also required, which provides a flexible deep learning

[1]https://pytorch.org and https://github.com/pytorch/pytorch

development platform , and laid the foundation for us to quickly verify the experimental effect.

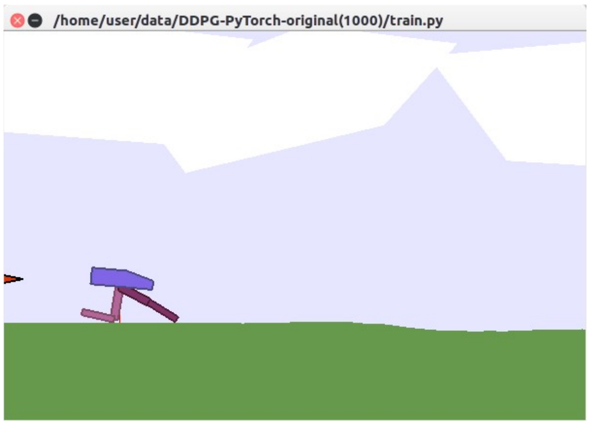

Fig. 3: Example of a figure caption.

The agent needs to learn a strategy so that the walker can walk on the randomly generated terrain without falling. The input state involves 24 inputs, including the hull angle, angular speed, forward speed, the angle of the two legs and their speed , whether it touched the ground, and 10 lidar sensor parameters. The action space includes 4 continuous values, which control the torque of four motors. The Pacers get rewards for going forward, and will get a penalty score of -100 when they fall, and a small amount of points will be consumed by applying motor torque. A better strategy will get a higher score, and a total of 300+ points will be obtained at most.

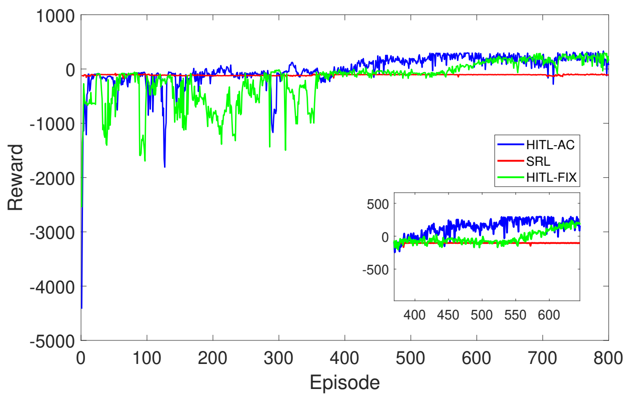

Fig. 4: Comparison of training effects of three methods

This paper uses the actor-critic algorithm to train pedestrians, and adds human suggestions to achieve a better and faster training process, as shown in Fig. 4. In the training process of the first 800 episodes, SRL (Standard Reinforcement Learning) training is very poor. In contrast, HITL-FIX (Human in the Loop Fix) can achieve an overall improvement in reward after the 500th episodes. Compared with HITL-FIX, HITL-AC (Human in the Loop Actor Critic) can improve the training effect faster (near the 400th episodes). Among them, the red curve SRL represents the reward evolution obtained by the agent training 1000 episodes on the SRL algorithm. The green curve HITL-FIX represents the training process after adding the suggestion of the human (the person thinks that a certain state is dangerous and needs to be reset), this part corresponds to the judgment of h_t in algorithm 1. The blue curve HITL-AC modifies the timing of human intervention (with a flexible switching boundary) based on the fixed switching boundary of the HITL-FIX algorithm, and then incorporates human suggestions into the learning algorithm process.

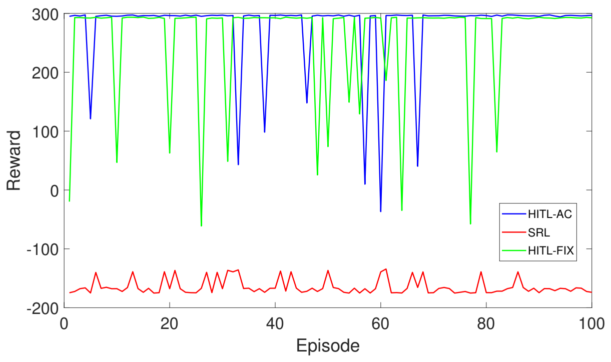

Fig. 5: Test performance comparison

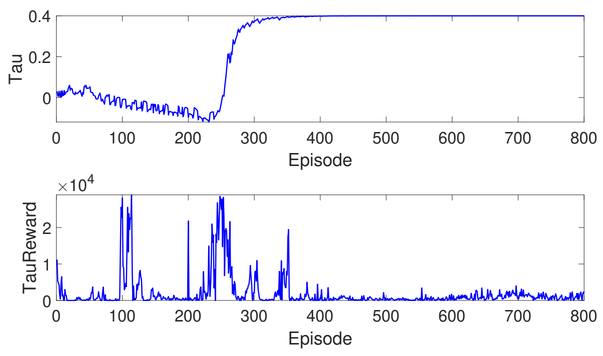

Fig. 6: Boundary learning trends and performance

Fig. 5 gives the test results of 100 episodes of the agent after training. Like the Fig. 4, the red SRL reward is the smallest. Green HITL-FIX and blue HITL-AC have a significant improvement, and HITL-AC has a greater average reward $(279.7164 > 260.4995)$ than HITL-FIX, indicating that compared to standard reinforcement learning algorithms, HITL-AC can train the agent more quickly to achieve better control effect. Fig. 6 shows the curve of the switching boundary learning process corresponding to algorithm 1 and the reward size of the corresponding training process.

V. CONCLUSIONS

In this paper, the autonomous boundary is considered for integrating human suggestions into reinforcement learning methods, humans play the role of guide. During the training process, the agent can gradually learn human operation methods. This is conducive to the liberation of human labor, and when the agent has doubts about the external environment and cannot be determined, it can make full use of the irreplaceable wisdom of the human being and more harmoniously complete the control process in which human and machine cooperate. In addition, reinforcement learning requires a lot of episodes for training to get a better strategy, which makes real-time control of complex and dynamic real-world control problems particularly difficult. Human intervention based on the autonomous boundary (see experimental simulation) can accelerate the process of reinforcement learning, which is beneficial for solving real-world problems.

This paper adds human suggestions to the reinforcement learning training algorithm in a punitive way, and the definition of the boundary is relatively simple and one-sided. Further consideration of stronger generalization capabilities, historical information, and methods based on controversial machine concepts will be reflected in future research work.

References

[1] J. Tjomsland, A. Shafti, and A. A. Faisal, "Human-robot collaboration via deep reinforcement learning of real-world interactions," *arXiv preprint arXiv:1912.01715*, 2019.

[2] Y. Oh, M. Toussaint, and J. Mainprice, "Learning Arbitration for Shared Autonomy by Hindsight Data Aggregation," *ar*, Jun. 2019.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, and T. Sainath, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. p.82–97, 2012.

[4] T. Joo and D. Shin, "Formalizing human–machine interactions for adaptive automation in smart manufacturing," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 6, pp. 529–539, 2019.

[5] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, "Human-machine-interaction in the industry 4.0 era," in *2014 12th IEEE international conference on industrial informatics (INDIN)*. IEEE, 2014, pp. 289–294.

[6] A. Broad, "Learning models for shared control of human-machine systems with unknown dynamics," *ar*, 2018.

[7] D. Gopinath, S. Jain, and B. D. Argall, "Human-in-the-Loop Optimization of Shared Autonomy in Assistive Robotics," *IEEE Robot. Autom. Lett.*, vol. 2, no. 1, pp. 247–254, Aug. 2016.

[8] K. Takei, W. Honda, S. Harada, T. Arie, and S. Akita, "Toward flexible and wearable human-interactive health-monitoring devices," *Advanced healthcare materials*, vol. 4, no. 4, pp. 487–500, 2015.

[9] Y. Ding, M. Kim, S. Kuindersma, and C. J. Walsh, "Human-in-the-loop optimization of hip assistance with a soft exosuit during walking," *Science Robotics*, vol. 3, no. 15, p. eaar5438, 2018.

[10] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," *arXiv preprint arXiv:1811.06817*, 2018.

[11] J. Haner and D. Garcia, "The artificial intelligence arms race: Trends and world leaders in autonomous weapons development," *Global Policy*, vol. 10, no. 3, pp. 331–337, 2019.

[12] C.-P. Lam and S. S. Sastry, "A pomdp framework for human-in-the-loop system," in *53rd IEEE Conference on Decision and Control*. IEEE, 2014, pp. 6031–6036.

[13] S. Reddy, A. D. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," *arXiv preprint arXiv:1802.01744*, 2018.

[14] W. Sheng, A. Thobbi, and Y. Gu, "An integrated framework for human–robot collaborative manipulation," *IEEE transactions on cybernetics*, vol. 45, no. 10, pp. 2030–2041, 2014.

[15] L. Fridman, L. Ding, B. Jenik, and B. Reimer, "Arguing machines: Human supervision of black box ai systems that make life-critical decisions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[16] Z. Lin, B. Harrison, A. Keech, and M. O. Riedl, "Explore, exploit or listen: Combining human feedback and policy model to speed up deep reinforcement learning in 3d worlds," *arXiv preprint arXiv:1709.03969*, 2017.

[17] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[18] V. S. Borkar and S. P. Meyn, "The ode method for convergence of stochastic approximation and reinforcement learning," *SIAM Journal on Control and Optimization*, vol. 38, no. 2, pp. 447–469, 2000.

[19] A. W. Beggs, "On the convergence of reinforcement learning," *Journal of Economic Theory*, vol. 122, no. 1, pp. 1–36, 2005.

[20] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, 2000, pp. 1008–1014.