Autonomous Navigation with Improved Hierarchical Neural Network Based on Deep Reinforcement Learning

Haiying Zhang[1,2], Tenghai Qiu[1,2], Shuxiao Li[1,2], Chengfei Zhu[1,2], Xiaosong Lan[1,2], Hongxing Chang[1,2]

1. Institute of Automation, Chinese Academy of Sciences, Beijing 100190
2. School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049
E-mail: {zhanghaiying2017, tenghai.qiu, shuxiao.li, chengfei.zhu, lanxiaosong2012, hongxing.chang}@ia.ac.cn

Abstract: Compared with traditional navigation strategies in normal environments, the unmanned vehicles in battlefield environments require better navigation strategies. This research formulates the autonomous navigation in battlefield environments as a markov decision process (MDP) and introduces deep deterministic policy gradient (DDPG) to obtain the continuous control signal. Meanwhile, the curriculum learning is employed to increase utilization of samples in this research. Inspired by the biological mechanism, an improved hierarchical neural network is proposed to refine the input information, which plays a better role in coordinating the choice of agent's behavior. Experimental results show that the models we proposed are able to acquire effective navigation strategies without knowing the whole information of environment. At the same time, it is proved that the hierarchical neural network and the curriculum learning are effective for improving efficiency of learning and generalization capability of models.

Key Words: Autonomous Navigation, DDPG, Improved Hierarchical Neural Network, Curriculum Learning

1 Introduction

Technology is constantly changing the way people live. The unmanned combat system, a masterpiece of the new era, has gradually became the focus of military research in most countries due to the low casualty rate and low cost. The unmanned ground combat system is proposed later than the unmanned aerial combat system, and the research process is also slower in complex and dynamic environments or in emergencies. Nevertheless, the unmanned ground combat system has gained more attention and the investment has surged in recent years. As a fundamental problem of the unmanned combat system, autonomous navigation has been the most basic and important issue in military research. Autonomous vehicle navigation in battled environments is more complicated than in other environments. The vehicle not only needs to autonomously arrive the destination safely without colliding obstacles, but also to comply with dynamic constraints. In addition, it needs to response quickly when environments changed.

Generally, navigation can be divided into two parts: global navigation and local navigation. The former uses completed environmental information to plan the optimal path, while the latter is hard to guarantee the planned path to be the optimal one owing to the incomplete information and dynamic environments. The commonly used algorithms of global navigation contain A*[1], Probabilistic Roadmaps (PRM)[2], Rapid-exploration Random Tree (RRT)[3], Genetic Algorithm (GA)[4]. Although the algorithms of global navigation can plan the optimal path of maps, they are not able to response to dynamic environments. Compared with global navigation, local navigation is more efficient and flexible in dynamic environments, which can avoid obstacles in real time and get to the destination even in new and unseen maps. Article Potential Field Method[5] and Fuzzy Logic Method[6] are two famous methods of local navigation with advantages of simple and low calculation, but they still exist some problems, such as local oscillation, local optimum and inaccessibility of targets.

Reinforcement learning (RL) is the problem that an agent learns behavior through trial-and-error interactions in a dynamic environment, so it provides a new approach for autonomous navigation in battled environments. RL is able to learn successful navigation policies and output end-to-end control sequences of vehicles by formulating the process of navigation as a Markov Decision Process (MDP). In recent years, deep learning (DL) has triggered the upsurge of artificial intelligence and promoted the development of RL. DL provides a rich representation with deep neural network enabling RL algorithms to break out the storage limit of Q-tables. As a result, the combination of DL and RL which calls deep reinforcement learning (DRL) enables the model to learn an effective navigation strategy in a complex environment. Many works about DRL in autonomous navigation have been studied. By utilizing the deep Q-network (DQN) algorithm, Minh *et al*. first achieved human-level success in many computer games and some of these games are related to navigation[7]. Target driven navigation[8] and successor features[9] are also added to RL to solve navigation problems. Chen *et al*. train agents to learn obstacles avoidance policies and path planning policies through off-line training[10].

The low efficiency of sample utilization and the low generalization capabilities of models have always been the main problems in DRL field. The former leads to the slow learning process, and the latter brings about the failure of navigation in new environments. The former can be solved with curriculum learning which gradually increases the complexity of the learning task by choosing more and more difficult examples for the training model[11]. In our task, curriculum learning is employed as our training method. As for the latter, many content works have been done to improve the generalization ability of models. The intrinsic curiosity

*This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61573350.

module (ICM) is employed to measure the novelty of the states by predicting the consequences of its own actions, which strengthens exploration in new environments[12]. The value iteration networks (VIN) learns to plan and predict outcomes and generalizes better to new and unseen domains[13]. Everett *et al.* add LSTM to predict other agents' behavior and take single line laser as input[14]. Tai *et al*. take sparse laser range readings, the velocity of the robot and the relative target position as input, and applies DRL in mapless navigation successfully[15]. Similarly, the simple and essential state information is taken as input in our models. Compared with regarding the image as input, the single line laser uses less and intuitive information to describe the obstacles space, which means that the generalization ability can be improved with less training. Furthermore, the single line laser is easier to be acquired and the cost is lower.

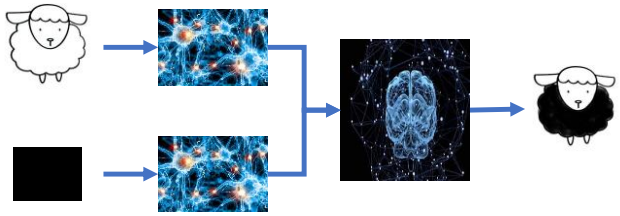


Fig.1: The visual analysis process

For the navigation problem, giving priority to avoiding obstacles or to driving to the target directly is the choice that the unmanned vehicle needs to make at every moment. How to balance these two behaviors is a key issue, because the unmanned vehicle needs to be secured safety and reaches the destination with a shorter path or a lower loss. Most works treat them as a whole, and some try to deal with them separately. Wang Z *et al*. used dueling network to process these two behaviors by splitting the Q-value of action into the value of state and the advantage of action[16]. In the field of biology, people commonly assumes that the process of object vision is in a single processing way. Contrary to this, Konen *et al*. found that the basic object information such as shape, size and viewpoint are represented sparately in two parallel and hierarchically organized neural systems[17]. As shown in Fig. 1, the color and shape of a sheep are identified by two sub-pathways. It is a good example to demonstrate that the visual system consists of many separate sub-pathways which analyze different aspects of the same retinal image. Although the final perception is a unified visual scene, it is normally done through a delicate coordination of a series of pathways in the visual system. Inspired by this, an improved hierarchical neural network is introduced to promote learning navigation strategies by further refining the input information in this paper.

In the algorithm family of DRL, the original DQN is only used in tasks with a discrete action space. Deep deterministic policy gradient (DDPG) which bases on actor-critic architecture is proposed to solve continuous control problems[18]. Later, Minh *et al*. provides Asynchronous Advantage Actor-Critic (A3C) which optimizing the DRL with asynchronous gradient descent from parallel on-policy actor-learners[19]. Compared with DDPG, A3C needs several parallel simulation environments, which limits its extension to some specific simulation engine. Thus, we choose DDPG as our training algorithm to realize end-to-end control of agents.

In this paper, the autonomous navigation in battlefield environments is formulated as an MDP and DDPG is introduced to obtain continuous control commands. Meanwhile, an improved hierarchical neural network inspired by a biological mechanism is proposed for aiding DRL agents to learn successful navigation policies in challenging environments and curriculum learning is applied to increase utilization of samples. Through a series of experiments in simulated environments, the results show that our models can learn continuous navigation strategies more effectively and has better generalization capabilities in unseen environments.

2 Methods

2.1 Background

In this paper, the autonomous navigation in the battlefield environments is dealt in simulated environments, and the DDPG is used to train our models. The process of navigation is formulated as an MDP. At each step, the agent receives an observation of its current state s_t , takes a corresponding action a_t , receives a reward r_t , and transits the current state s_t to the next state s_{t+1} following the dynamic transition $p(s_{t+1} \mid s_t, a_t)$.

2.2 DDPG

To train the navigation policies, the DDPG algorithm which can output continuous control commands of agents is introduced. It is based on actor-critic architecture. The actor part selects action a_t according to the input state s_t with a policy network $a_t = u(s_t \mid \theta^u)$. The critic part takes the action a_t and the state s_t as input to evaluate the quality of the action a_t with a value network $Q(s_t, a_t \mid \theta^Q)$.

The estimation value y_i at each step is calculated by a discount factor γ according to the following equation:

$$y_i = r_i + \gamma Q'(s_{i+1}, u'(s_{i+1} \mid \theta^{u'}) \mid \theta^{Q'}) \quad (1)$$

where Q' and u' denote the previous values of Q and u .

Then the critic is updated by minimizing the loss L in equation (2) and the actor policy is updated by using sampled policy gradient as equation (3):

$$L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i \mid \theta^Q))^2 \quad (2)$$

$$\nabla_{\theta^u} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a \mid \theta^Q)\mid_{s=s_i, a=u(s_i)} \nabla_{\theta^u} u(s \mid \theta^u) \mid s_i \quad (3)$$

The target network is updated as follows:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \quad (4)$$

$$\theta^{u'} \leftarrow \tau\theta^u + (1-\tau)\theta^{u'} \quad (5)$$

where θ^Q and θ^u denote the network parameters of the value network and the policy network.

2.3 Improved Hierarchical Neural Network

In order to improve the generalization ability of models, the improved hierarchical neural network is applied to the actor part to refine the input information. The input information is divided into two sub-modules and then concatenated and fed into fully connected lays for further processing. The two sub-modules are the obstacle avoidance module and the goal processing module. As shown in Fig. 2, the readings of laser s_t^l and the agent's direction s_t^d are fed into the obstacle avoidance module. The goal information s_t^g and the direction information s_t^d are fed into the goal processing module. The outputs of the two models are then concatenated and fed into a fully connected network to output the continuous action sequence. Moreover, the laser readings s_t^l and the relative target position s_t^g differ greatly in dimension, which has a negative influence on behavior learning of agents. To reduce the influence of the input information, the number of output nodes of these two models is identical for making the model easier to learn.

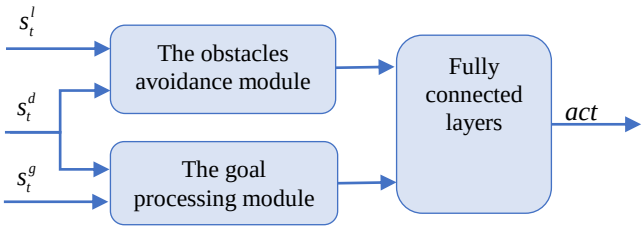


Fig. 2: Actor part network

The obstacles avoidance module has two fully connected layers with 300 and 150 units respectively, each followed by ReLU nonlinearities. The goal processing module also has two fully connected layers with 200, 150 units respectively, each followed by ReLU nonlinearities. The module of fully connected layers contains 5 fully connected layers with 300, 400, 300, 200, 200 units, each also followed by ReLU nonlinearities, and the last fully connected layer is followed by a tanh function.

In the critic part, as shown in Fig. 3, the laser readings s_t^l, the goal information s_t^g, the agent's direction s_t^d, the agent's action *act* and a constant *b* are merged together as an input vector. After 7 fully-connected neural network layers with 300, 400, 400, 400, 400, 300, 10 units with ReLU nonlinearities, the input vector is transferred to *Q* value.

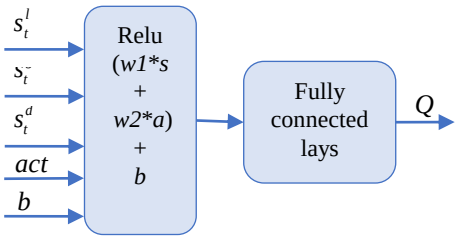


Fig. 3: Critic part network

2.4 Reward Setting

In addition to the appropriate algorithm and the network structure, the setting of reward function is also crucial for DRL. The relative target distance is easy to be measured via the GPS system or the base station positioning technology, so it is selected as a part of rewards. Our reward function is stated in equation (6). If the relative target distance $(d_t - d_{t-1})$ is smaller than last time, the agent gets a positive reward, otherwise, the agent gets a large negative reward. For avoiding minimum training in each episode, the agent gets a negative reward $r_{penalty}$ if the relative target distance is bigger than last time and the number of steps n_{step} is much larger than what the agent needs n_{need}. A positive reward r_{reach} is arranged when the agent arrives at the target, and a negative reward $r_{collision}$ is arranged when the agent collides with an obstacle.

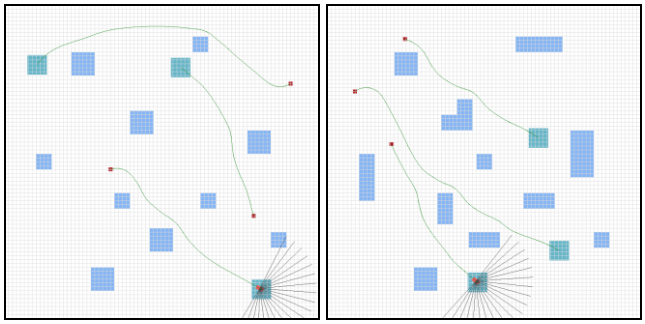
$$r = \begin{cases} \lambda_1 \dfrac{d_t - d_{t-1}}{v * dt}, & if\ (d_t - d_{t-1}) > 0 \\ \lambda_2 \dfrac{d_t - d_{t-1}}{v * dt}, & if\ (d_t - d_{t-1}) < 0 \\ r_{penalty}, & if\ (d_t - d_{t-1}) > 0\ and\ n_{step} > 2 * n_{need} \\ r_{collision}, & if\ collides \\ r_{reach}, & if\ arrives\ target \end{cases} \tag{6}$$

Where d_t is the current relative target distance, and d_{t-1} is the last moment relative target distance. v is the velocity of the agent and dt is time interval. λ_1 and λ_2 are hyper-parameters.

3 Experiments

3.1 Environments Setup

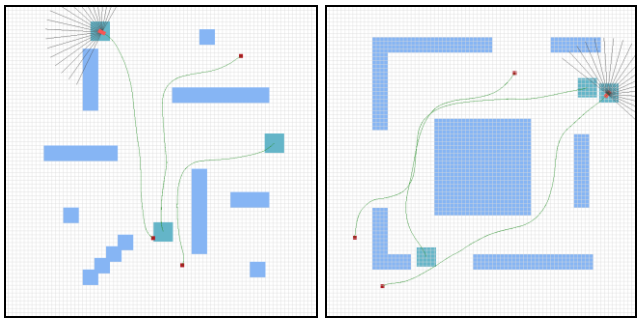
The training procedure of our model is implemented in virtual environments. As shown in Fig. 4, two simulation environments with a size of 800*800 are constructed to show the influence of the training environment on the motion planner. Obstacles in *Env-2* are more complicated and varied than *Env-1*. A red rectangular object with a few black lines denotes the agent equipped with a laser range sensor. The target is denoted by a green square object and can't be rendered by the laser sensor. The red little square denotes the starting position of the agent. At the beginning of each episode, the pose and position of the agent, and the position of the target are initialized randomly in the whole map so that a collision-free path is guaranteed to exist between them. An episode is terminated when the agent either reaching the target, colliding with an obstacle, or after a maximum of 500 steps during training and testing.



Env-1 *Env-2*

Fig. 4: Training and testing environments

In addition to regarding the training environments *Env-1* and *Env-2* as a part of the testing environments, two new and unseen environments are constructed to test the generalization capabilities of our models. As shown in Fig. 5, barrier-type obstacles and large-sized obstacles are added separately to *Env-3* and *Env-4*, and none of these obstacles have appeared in the training environments.



Env-3 *Env-4*

Fig. 5: Testing environments

The green lines in Fig. 4 and Fig. 5 are trajectories generated by the model which applies curriculum learning and the improved hierarchical neural network.

3.2 Parameter Setup

The input state s_t consists of 20-dimensional laser range readings s_t^l (with a maximum range of 150), a 2-dimensional relative target distance s_t^g ($x_{target} - x_{agent}$, $y_{target} - y_{agent}$) , and the direction of the agent s_t^d . At each step, the action of the agent is a continuous angle of rotation with a maximum angle of 6 degrees. The velocity v_t of the agent is 50 and the time interval d_t is 0.1.

In the training algorithm, the memory capacity is 7000 and the batch size is 32. the hyper-parameters γ is 0.9 and τ is 0.01. Meanwhile, the critic part and the actor part have an identical learning rate of 0.0003.

The hyper-parameters regarding the reward function (6) are summarized in Table 1.

Table 1: Hyper-Parameters of Reward Function

Hyper-parameters	Value
n_{need}	200
$r_{penalty}$	-0.2
$r_{collicion}$	-1
r_{reach}	1
λ_1	-0.1
λ_2	-0.11

3.3 Experiments and Evaluation

Our models are implemented by a well-known open-source deep learning library Tensorflow and finetuned by the Adam optimizer. Meanwhile, they are trained in a single Intel Core i5-8400 CPU and each training of 20 thousand episodes takes about 3 hours.

In order to test the impact of network structure and training method on the learning ability and the generalization ability of the model, we design several experiments. Experiment1 uses the original network and the simple training method. In the original network, the inputs are fed into fully connected layers directly and are not processed by the obstacle avoidance module and the goal processing module. The simple training method means that the model is trained only in Env-2 with 40 thousand episodes. The starting position and the pose of the agent are initialized randomly in each episode, and the destination is also initialized randomly every 30 episodes; Experiment2 uses the improved hierarchical neural network and the simple training method. The structure of the improved hierarchical neural network is shown in Fig.2, in which the inputs are processed by two sub-models firstly and then are fed into fully connected layers. The simple training method is the same as Experiment1; Experiment3 uses the improved hierarchical neural network as the same as Experiment2, and the training method is curriculum learning that the model is first trained in Env-1 with 20 thousand episodes and then trained in Env-2 with the same number of episodes. The initialization of curriculum learning is the same as the initialization of the simple training method; Experiment4 uses the improved hierarchical neural network as Experiment2, and the training method is curriculum learning liking Experiment3.

After trained, these models are tested in Env-1,2,3,4 with a fixed set of 500 random episodes. The success ratio of them

Table 2: Testing Result

Experiment	Network Structure	Training Method	Env-1(%)	Env-2(%)	Env-3(%)	Env-4(%)
1	Original network	Sample training	0	0	0	0
2	Improved hierarchical neural network	Sample training	89.4	83.8	77.0	74.4
3	Original network	Curriculum learning	0	0	0	0
4	Improved hierarchical neural network	Curriculum learning	**93.1**	**86.8**	**84.5**	**75.3**

are shown in table2. From the test results, the following contents are known:

- Experiment1 and Experiment2, both of which use the original network, have not learned successful navigation strategies, so the agents of them have been spinning somewhere in all testing environments. However, Experiment2 and Experiment4 which use the improved hierarchical neural network have learned successful navigation strategies. This means that compared with the original network, the improved hierarchical neural network greatly enhances the ability of models to learn successful navigation strategies.
- Compared with Experiment2, Experiment4 uses the improved hierarchical neural network while using curriculum learning for training. As a result, Experiment4 has a higher success ratio in all testing environments than Experiment2. Especially in Env-1,2,3, the success ratio has increased by at least three percentage points. This means that the training method of curriculum learning plays a positive role in improving the utilization of training samples and the generalization ability of models.
- Env-3 and Env-4 can test the generalization ability of models more powerfully because they are completely new and unseen to models. The success ratios of the best experiment—Experiment4 are 84.5% and 75.4% in Env-3 and Env-4, which indicates that the model using the improved hierarchical neural network and curriculum learning not only learns successful navigation strategies, but also has a good generalization capability.

To further test the generalization capabilities of the best model, a new environment which contains 20 obstacles with the size of 20*20 is generated randomly at each episode. We collect testing statistics on a fixed set of 1000 random episodes and the success ratio is 81.4%. Two random intercepted scenes are shown in Fig. 6.

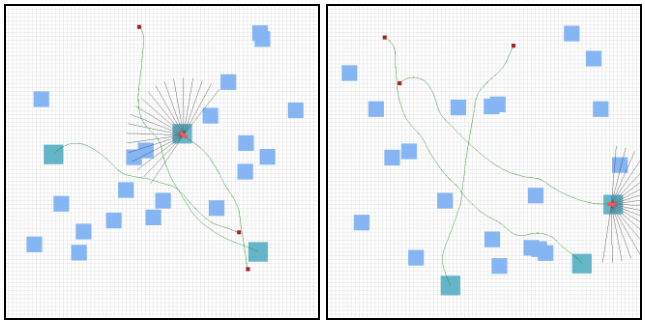


Fig. 6: Two randomly intercepted scenes

The trajectories of the best model are shown as the green lines in Fig. 4,5,7. It is clear that the agent avoids obstacles smoothly and reaches the destination safely. Intuitively, the trajectories are nearly straight except the flexible parts caused by avoiding obstacles. Meanwhile, for the starting direction of agents is random, the primary section of some trajectories is curved where the agent adjusts its direction. At the same time, these trajectories are more in line with the actual driving trajectory of the vehicle owing to the limited steering angle and the continuous control signals. It is proved that the best model applying curriculum learning and the improved hierarchical neural network has learned a high-quality navigation strategy.

4 Conclusions

This paper focuses on autonomous navigation in the battlefield environments where the agent is expected to navigate to the destination without the whole knowledge of map. It takes sparse laser range readings, the direction of the agent and the relative target position as input and uses DDPG to obtain continuous control commands of agents. Meanwhile, the curriculum learning and an improved hierarchical neural network are introduced into the DDPG. The former is to accelerate the training rate and the latter is to improve the learning and generalization ability of models. Our models are trained in two environments and tested in four environments. Meanwhile, 1000 environments generated randomly with 20 obstacles are also used to test our best model, the results of experiments are satisfactory.

As an innovation in this paper, the improved hierarchical neural network is crucial for improving DRL performance in tasks with challenging exploration requirements. The architecture of the network is similar to the neural system of object information in the human visual cortex. It consists of two sub-models, which analyze the different aspects of the same input knowledge. The experimental results show that the models using the network have a high success ratio and the trajectories are smooth while other models with the same learning rate have fallen into the local minimum. It is clear that the agents employing the network are able to reach the target successfully with high-quality trajectory and the improved hierarchical neural network has a better generalization capability in unseen and challenging environments.

Several groups of experiments are also conducted to test the validity of curriculum learning. The models using curriculum learning have higher success ratios than the models without curriculum learning under the same conditions. Although the curriculum learning is less effective than the improved hierarchical neural network for lifting models, it is still a good method to improve the learning and generalization ability of the models.

5 Discussion and Future Works

According to the experimental results, the model with both the improved hierarchical neural network and curriculum learning has the highest success ratio, which has better sample efficiency and better generalization capabilities in unseen and challenging environments. As an innovation, the improved hierarchical neural network we propose is similar to the biological mechanism, which can be used to solve more problems in different fields. However, there are still some flaws in our works. Compared with the real battlefield environments, our simulation environments are too simple, and the navigation strategies learned by the models are not enough to meet the application requirements in the real war world.

In future work, we will do some improvements on our models. On the one hand, we plan to further improve the hierarchical neural network and explore its application in other tasks. On the other hand, a better model needs to be

designed to learn the navigation strategies which are suitable in the real war world.

References

[1] F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek *et al*., Path planning with modified a star algorithm for a mobile robot, *Procedia Engineering*, 96: 59-69, 2014.

[2] X. Yu, Y. Zhao, C. Wang, *et al*., Trajectory planning for robot manipulators considering kinematic constraints using probabilistic roadmap approach, *Journal of Dynamic Systems, Measurement, and Control*, 2017, 139(2): 021001.

[3] Jr. Kuffner, J. James, LaValle, M. Steven, RRT-connect: An efficient approach to single-query path planning, *IEEE Trans. on International Conference on Robotics and Automation*, 2000.

[4] M. Elhoseny, A. Tharwat, A. E. Hassanien, Bezier curve based path planning in a dynamic field using modified genetic algorithm, *Journal of Computational Science*, 2018, 25: 339-350.

[5] Y. Chen, G. Luo, Y. Mei, *et al.*, UAV path planning using artificial potential field method updated by optimal control theory, *International Journal of Systems Science*, 2016, 47(6): 1407-1420.

[6] A. Bakdi, A. Hentout, H. Boutami *et al.* Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control, *Robotics and Autonomous Systems*, 2017, 89: 95-109.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Bellemare, Human-level control through deep reinforcement learning, *Nature*, 518(7540): 529, 2015.

[8] Y. Zhu, R. Mottaghi, E. Kolve, J. Lim, A. Gupta *et al*., Target-driven visual navigation in indoor scenes using deep reinforcement learning, *IEEE Trans. on International Conference on Robotics and Automation*, 2017: 3357-3364.

[9] J. Zhang, J. T. Springenberg, J. Boedecker, W. Burgard, Deep reinforcement learning with successor features for navigation across similar environments, *IEEE Trans. on Intelligent Robots and Systems*, 2017: 2371-2378.

[10] Y. F. Chen, M. Everett, M. Liu, *et al.*, Socially aware motion planning with deep reinforcement learning, in *International Conference on Intelligent Robots and Systems*, 2017: 1343-1350.

[11] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, In *Proceedings of the 26th annual international conference on machine learning*, 2009: 41-48.

[12] D. Pathak, P. Agrawal, A. A. Efros, T. Darrell, Curiosity-driven exploration by self-supervised prediction, *IEEE Trans. on International Conference on Machine Learning*, 2017.

[13] A. Tamar, Y. Wu, G. Thomas, S. Levin, P. Abbeel, Value Iteration Networks, In *Advances in Neural Information Processing Systems*, 2017: 2154-2162.

[14] M. Everett, Y. F. Chen, J. P. How, Motion planning among dynamic, decision-making agents with deep reinforcement learning, In *International Conference on Intelligent Robots and Systems*, 2018: 3052-3059.

[15] L. Tai, G. Paolo, M. Liu, Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation, *IEEE Trans. on Intelligent Robots and Systems*, 2017: 31-36.

[16] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot *et al*., Dueling network architectures for deep reinforcement learning, in *arXiv preprint arXiv*:1511.06581, 2015.

[17] C. S. Konen, S. Kastner, Two hierarchically organized neural systems for object information in human visual cortex, in *Nature Neuroscience*, 11(2): 224, 2008.

[18] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez *et al*., Continuous control with deep reinforcement learning, in *arXiv preprint arXiv*:1509.02971, 2015.

[19] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap *et al*., Asynchronous methods for deep reinforcement learning, in *International conference on machine learning*, 2016: 1928-1937.