NAO robot obstacle avoidance based on fuzzy Q-learning

***Abstract*—This paper proposes a novel active SLAM framework to realize obstacle avoidance and finish the autonomous navigation in indoor environment. The improved Fuzzy optimized Q-Learning (FOQL) algorithm is used to solve the obstacles avoidance problem of the robot in the environment. To reduce the motion deviation of the robot, fractional PI controller is designed. The localization of the robot is based on FastSLAM algorithm. Simulation results of avoiding obstacles using traditional Q-learning algorithm, optimized Q-learning algorithm and FOQL algorithm are compared. The simulation results show that the improved FOQL algorithm has a faster learning speed than other two algorithms. To verify the simulation result, the FOQL algorithm is implemented on a NAO robot and the experimental results demonstrate that the improved Fuzzy optimized Q-Learning obstacle avoidance algorithm is feasible and effective.**

***Index Terms*—FASTSLAM, fuzzy logic control, fractional PI, obstacle avoidance, Q-learning**

1. Introduction

Humanoid robot has attracted many researchers in recent years [1-4], and autonomous navigation ability of humanoid robot has been a hot research [5-7]. Autonomous navigation integrates environment perception, dynamic decision and planning etc. Autonomous navigation ability can avoid the space constraints of the robot, which can make the robot widely used in many areas [8-10].

Obstacle avoidance is an important ability to autonomous navigation robot [11-13]. In recent years, many collision avoidance algorithms have been studied. Some of these popular methods for path planning include Artificial Potential field Method(APM) [14, 15], Vector Field Histogram method(VFH) [16] and, fuzzy logic control method [17], neural network method [18], genetic algorithm [19] and so on. APM simplifies all the information of obstacles and the destination to a resultant force. Thus, the local details of obstacles will be lost. VFH selects the robot candidate heading by quantifying the obstacle strength value of each angle of the robot, which cannot make the robot go through narrow channels. The obstacle avoidance method based on fuzzy logic control forms certain rules according to the prior knowledge. Fuzzy logic control approach demonstrates good robustness property, real-time performance and less dependence on the environment. But there exists the phenomenon of symmetry which cannot be determined. The obstacle avoidance method based on neural network designs controller according to the position of obstacles. But it needs a lot of time to train the network in offline to find a global optimal path. The obstacle avoidance method based on genetic algorithm divides the robot movement environment into some equal parts and searches an optimal path. Path planning using genetic algorithm is a simple and effective method. But genetic algorithm tends to arise the local-trap problem and premature convergence problem [17-19].

Traditional obstacle avoidance algorithms, such as APM and VFH, are invalid when the information of the obstacle is incomplete or completely unknown. However, reinforcement learning (RL) has a better generalization ability when the obstacle is unknown.

RL is a machine learning method that regards the feedback of the environment as an input and adapts the environment. RL decides and optimizes the chosen action by interaction between the agent and the environment. RL has been successfully applied to path planning [20, 21]. Q-learning is one of the most popular algorithms in the RL algorithm [22-24]. However, traditional Q-learning converges slowly. C. Deng and J. E. Meng [25] optimized traditional Q-learning by selecting the action based on probability, which speeds up the convergence rate of the algorithm. The improvement of convergence rate will contribute to improve the speed and real-time performance of Q-learning algorithm. However, the optimized Q-learning still cannot further meet the fast requirement of obstacle avoidance. In practical applications, being real-time is very important for obstacle avoidance, which requires fast and effective avoidance obstacle algorithms. In this paper, a new avoidance obstacle algorithm based on an improved fuzzy Q-learning is proposed. An optimal fuzzy Q-Learning can further improve the convergence rate by integrating daily experience into algorithm. The effectiveness of obstacle avoidance will increase by learning based on the experience of the robot.

Ref. [26] obtains the state space of Q-learning by fuzzy inference system. However, the state space of Q-learning is built based on checkerboard path planning model, which is simple and effective. In this paper, we propose an improved FQL algorithm which initializes the table entry Q based on Fuzzy Inference System (FIS) instead of random initialization. The fuzzy rules of FIS are based on daily experience. In addition, to solve the problem of walking deviation of the robot, fractional $\boldsymbol{PI}$ controller proposed by our previous work [27] is used in this paper. FastSLAM algorithm [28] is used to localize the position of the robot.

The rest of this paper is organized as follows. Section 2 gives coordinate system, robot mobility model and robot observation model. Section 3 introduces traditional Q-learning algorithm, optimized Q-learning algorithm and fuzzy Q-learning. We also analyze the convergence of fuzzy Q-learning and design fractional $\boldsymbol{PI}$ controller. Section 4 shows the results of the simulation. Experimental results are presented in Section 5 and conclusions are shown in Section 6.

2. System model

System model includes moving model and observation model of the robot. The moving model is used to describe mobile control of the robot. And in the fuzzy Q-learning obstacle avoidance algorithm, the design of action elements is based on the robot moving model. The observation model is used to describe the environment perception of the robot, especially the obstacles and landmarks. In order to describe moving model and observation model of the robot clearly, a unified coordinate system is required to build.

2.1 Global coordinate system and local coordinate system

An appropriate coordinate system is needed to be established to describe the position of the robot and the obstacles for avoiding obstacles. In this paper, two two–dimensional rectangular coordinates, global coordinate system and local coordinate system are built.

Global coordinate system is a fixed coordinate system which is used to mark the position of the robot and the objects in the environment. Global coordinate system is established according to the initial pose of the robot. The initial position of the robot is regarded as the origin of the global coordinate system. The robot faces the positive direction of X axis, and the direction of robot's left side is the positive direction of Y axis. Local coordinate system, that is, the coordinate system of the robot, is described for the position of the observed object relative to the robot. The local coordinate system is built according to the current position of the robot. The positive direction of X axis in the local coordinate system is the same as the robot's direction, and the positive direction of Y axis is the robot's left side. Fig. 1 shows the global coordinate system and local coordinate system. The global coordinate system is described by solid line, and the local coordinate system is described by dotted line. θ is the angle between the robot and the X axis of global coordinate, Ø is the angle between robot and the object and d is the distance between the robot and the object.

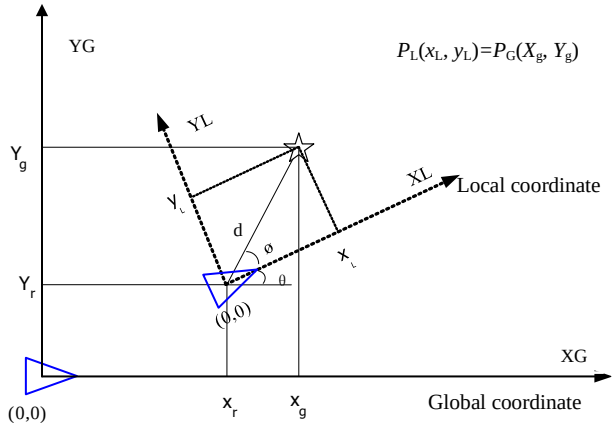


Fig. 1. Coordinate system.

2.2 Robot moving model

The moving model of the robot demonstrates the transition relationship between the control variable and the current pose of the robot. The robot pose at $(t-1)$ is expressed as $R_{t-1}=[x_r, y_r, \theta_r]_{t-1}^T$, and the control variable at t is expressed as $u_t=[\Delta x, \Delta y, \Delta\theta]_t^T$. The robot pose at t can be obtained from moving function $f(R_{t-1}, u_t, n_t)$ when R_{t-1} and u_t are known, and the formula is

$$R_t = f(R_{t-1}, u_t, n_t) = R_{t-1} + u_t + n_t \tag{1}$$

Because there exists noise when the robot moves, Gaussian noise n_t is added to moving function. The mean value of n_t is 0 and the variance of n_t is q^2. That is $n_t = N(0, q^2)$.

2.3 Robot observation model

The observation model can provide the position of the observed object relative to the robot. In Fig. 1, star represents the observed object, and the observed value $z=[d, \phi]^T$ is obtained from the laser scanner on the robot's head. The coordinates of the observed object in the robot coordinate system are

$$P_L = \begin{bmatrix} x_L \\ y_L \end{bmatrix} = \begin{bmatrix} d\cos\phi \\ d\sin\phi \end{bmatrix} \tag{2}$$

The current pose of the robot is $R_t=[x_r, y_r, \theta]_t^T$, and the coordinates of the observed object in the global coordinate system can be calculated by the following equation.

$$P_G = \begin{bmatrix} X_g \\ Y_g \end{bmatrix} = \begin{bmatrix} x_r + d\cos(\theta+\phi) \\ y_r + d\sin(\theta+\phi) \end{bmatrix} \tag{3}$$

3. Obstacle avoidance algorithm

3.1. Reinforcement learning

RL is an online machine learning method, which regards the environment feedback as the input and can adapt the environment. In reinforcement learning, the environment is described as a set of states S and the action of an agent is described as a set of actions A. The learning purpose of the agent is to generate an appropriate action policy which can select an action sequence and result in an optimal result. The r_t is the reward of the a_t in the current environment s_t.

Fig. 2 shows the structure of RL. The agent can find the optimal policy by trial and error method. In any state s_t, the agent performs an action a_t which can make the state of environment become s_{t+1} and get a reward r_t to evaluate the action. r_t is the immediate reward of performing action a_t when the environment state is s_t. The agent repeats these steps until it satisfied termination condition or becomes target state. The action sequence performed by the agent from the initial state to the termination state is called an episode or a trial.

The whole process of reinforcement learning is that the agent determines and optimizes the choice of an action by interacting with the environment. And the interaction between the agent and environment is modelled based on Markov Decision Process (MDP)[29].

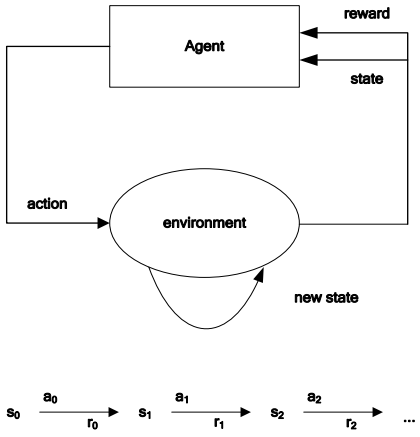


Fig. 2. Reinforcement learning schematic diagram

3.2. Q-learning algorithm

Q-learning is one of the most popular algorithms in the reinforcement learning algorithm [30]. Traditional Q-learning is valid for path planning but convergence rate is slow [31]. Some researchers improved the convergence rate by optimizing the action selection strategy of traditional Q-learning. In this paper, the proposed fuzzy Q-learning algorithm can further improve the convergence rate comparing with the optimized Q-learning.

3.2.1. Traditional Q-learning

The purpose of traditional Q-learning is to directly learn the evaluation $Q(s,a)$ of each state-action pair (s_t, a_t). a_t is one of the possible actions chosen in state s_t. The value of $Q(s,a)$ denotes the cumulative reward after the agent takes action a in state s. Traditional Q-learning is given by [29].

$$Q(s_t, a_t) = r_t + \gamma \max Q(s_{t+1}, a) \tag{4}$$

where r_t denotes the immediate reward received from the environment by taking action a_t in state s_t, and γ denotes the discount factor. γ reflects the degree that future action influences current action. The larger γ is, the larger the weight value considering future action is. Here $\gamma \in (0,1)$. Action a_t is chosen randomly for traditional Q-learning algorithm. The steps of traditional Q-learning algorithm are given in Table 1.

Table 1
Traditional Q-learning algorithm

Algorithm 1. Traditional Q-learning algorithm

Initialize arbitrarily all $Q(s,a)$ values

Observe the current state s_t

Repeat (for each episode):

Choose an action a_i randomly and execute a_t

Receive immediate reward r_t

Observe new state s_{t+1}

Update $Q(s_t,a_t)$

$Q(s_t,a_t) \leftarrow r + \gamma \max_a (s_{t+1},a)$

$s_t \leftarrow s_{t+1}$

3.2.2. Optimized Q-learning

For traditional Q-learning algorithm, the agent randomly chooses an action from actions space. All actions chosen by equal probability from action space will result in slow convergence rate. So traditional Q-learning algorithm is optimized by choosing an action based on different probability. Then the optimized Q-learning can improve convergence rate on the basic of satisfying convergence condition [25-30,32].

$$P(a_i \mid s) = \frac{k^{Q(s,a_i)}}{\sum_j k^{Q(s,a_j)}} \tag{5}$$

where $P(a_i \mid s)$ represents the chosen probability of action a_i by the agent. k is a constant and $k > 0$. Thus, an action which has a high Q value can have a high probability. The probability of all actions is nonzero. Therefore, the optimized Q-learning algorithm has a higher learning efficiency than traditional Q-learning algorithm. The value of k reflects how strongly the selection favors actions with high Q value. k demonstrates the importance of Q value when an agent chooses the action. The larger k is, the higher probability of chosen action above average Q. In contrast, the smaller k is, the higher probability of chosen other actions below average Q. So the agent will choose the action whose Q value is not very large. The steps of the optimal Q-learning algorithm are given in Table II.

Table 2
Optimized Q-learning algorithm

Algorithm 2. Optimized Q-learning algorithm

Initialize arbitrarily all $Q(s,a)$ values

Observe the current state s_t

Repeat (for each episode):

Choose an action a_t according to Eq. (5) and execute a_t

Receive immediate reward r_t

Observe new state s_{t+1}

Update $Q(s_t,a_t)$

$Q(s_t,a_t) \leftarrow r + \gamma \max_a (s_{t+1},a)$

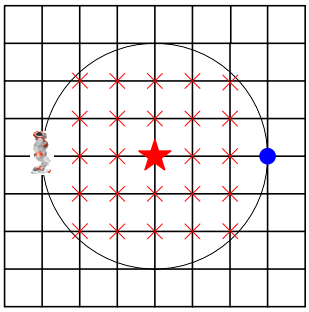
$s_t \leftarrow s_{t+1}$

3.3. Fuzzy Q-learning obstacle avoidance algorithm

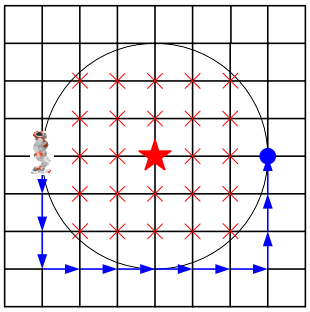
$Q(s,a)$ is generally initialized to 0 or other random value for traditional Q-learning and the optimized Q-learning, which makes the agent try many times in every episode. It will influence learning speed. Convergence rate will be improved if $Q(s,a)$ is initialized an appropriate value. In this paper, an optimized fuzzy Q-learning is proposed to avoid obstacles. The proposed algorithm initializes $Q(s,a)$ using fuzzy rules based on the optimized Q-learning, which can further improve convergence rate.

3.3.1. Checkerboard path planning model

The obstacle avoidance is considered as a path planning which can find an optimal path to reach the target point. To realize obstacle avoidance, checkerboard path planning model is established in Fig. 3.



(a) Checkerboard model



(b) An optimal obstacle avoidance path from the origin to the destination

Fig. 3. Checkerboard path planning model

In Fig. 3. (a), the length of each small square is 0.1 meter. The red star represents the obstacle and the blue point is the target location. The size of checkerboard path planning model is $(h_{\max}+0.8)\times(l_{\max}+0.8)$. $h_{\max}$ denotes the maximum transverse distance between obstacles and $l_{\max}$ denotes the maximum longitudinal distance between obstacles. The safe distance between the robot and the obstacle is set as 0.3 meter in consideration of the outline of the robot. So the red cross denotes danger area. The robot will collide with the obstacle if the robot is in danger area. For the checkerboard path planning model, the robot can move from an intersection to another adjoining intersection. Those intersections constitute the state set S. Fig. 3. (b) shows an optimal obstacle avoidance path that the robot moves from current position to target position.

3.3.2. Action set and Reward function

According to the checkerboard model, action set A consists of four action elements which are forward, backward, left and right. The corresponding control input is $u_{forward}=[0.1,0,0]^T$, $u_{backward}=[-0.1,0,0]^T$, $u_{left}=[0,0.1,0]^T$ and $u_{right}=[0,-0.1,0]^T$. Although these actions are simple, but it can help the robot compete the task of obstacle avoidance effectively. The size of action space A plays an important role in the operational efficiency of Q-learning algorithm. Complex action set will increase the amount of computation.

Reward function, which can provide immediate reward value by evaluating the interaction result between the robot and the environment, plays a key role in learning speed. State set of the environment includes three parts: danger area s_d, safe area s_s and target point location s_e. If the robot reaches the danger area, the safe area or the target point, the immediate reward is a negative value, zero or a positive value, respectively. The current robot position is S and reward function is

$$r=\begin{cases}100 & s=s_e\\ 0 & s\in s_s \text{ and } s\neq s_e\\ -100 & s\in s_d\end{cases} \tag{6}$$

3.3.3. Initialize $Q(s,a)$ based on FIS

For the improved fuzzy Q-learning obstacle avoidance algorithm, $Q(s,a)$ is initialized based on Fuzzy Inference System (FIS) instead of a random value. That is, each state-action pair is evaluated by FIS. Thus $Q(s,a)$ will have a relatively appropriate initial value which can improve learning speed.

$Q(s_i,a_i)$ is initialized by the following equation.

$$Q(s_i,a_i)=\begin{cases}-10 & s_i^{'}\in s_d\\ FIS & s_i^{'}\in s_s\end{cases} \tag{7}$$

$$s_i \xrightarrow{a_i} s_i^{'}$$

where $s_i\in S, s_i^{'}\in S, a_i\in A$. The robot changes from state s_i to state $s_i^{'}$ after taking action a_i. $Q(s_i,a_i)$ value is obtained from FIS if $s_i^{'}\in s_s$. The fuzzy rules are as follows.

If D is P and DR is NR, Then Y is B
If D is P and DR is NR, Then Y is M
If D is N and DR is NR, Then Y is S
If D is N and DR is NR, Then Y is VS

where D is the distance between the robot and the obstacle. DR denotes the difference between d_1 and d_2. d_1 is the distance between the robot in state s_i and the target location. d_2 is the distance between the robot in state $s_i^{'}$ and the target location. Y is the output of FIS. The fuzzy set of D is $Z=\{P,N\}$ which denotes 'positive' and 'negative', respectively. The range of D is

$[0.3,1.0]$ .The fuzzy set of DR is $W=\{NR,FR\}$ which denotes 'near' and 'far', respectively. The range of DR is $[-0.1,0.1]$. The fuzzy set of Y is $C=\{VS,S,M,B\}$ which denotes 'very small', 'small' , 'middle' and 'big', respectively. The Y is the output of FIS and its domain is $\{1,4,5,10\}$. Fig. 4 shows the degree of membership function of D , DR and Y . Minimum value method is used to deal with the "and" logic in the rule base. The fuzzification adopts membership value method and the defuzzification adopts centroid method. The advantage of centroid method is that it makes full use of the information of the result of inference and it is robust. The formula of centroid method is

$$y^*=\frac{\sum_i (y_i * \mu(y_i))}{\sum_i \mu(y_i)} \tag{8}$$

where y_i is fuzzy single-point value and $\mu(y_i)$ is the membership degree of y_i . The output surface is shown in Fig. 5.

The fuzzy rules will produce such result that the initial value of $Q(s_i,a_i)$ will be big if the robot moves along the edge of an obstacle and is close to the destination. The fuzzy rules will make the robot bypass the obstacles with fewer steps according to daily experience. The degree of membership function D expresses the level which the robot moves along the edge of an obstacle. The degree of membership of DR expresses the level which the robot is close to the destination.

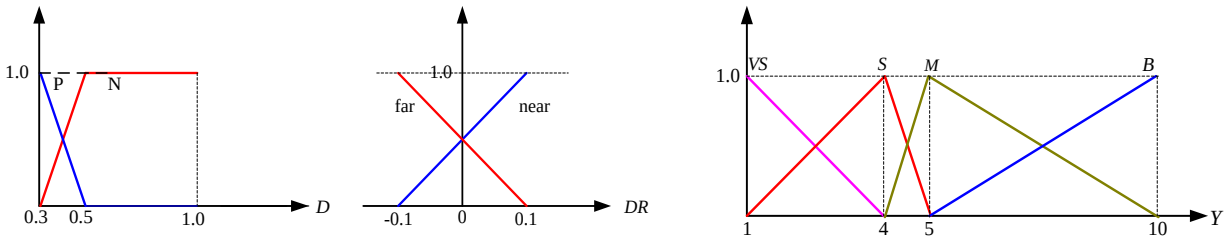


Fig. 4. The D , DR and Y membership functions, respectively

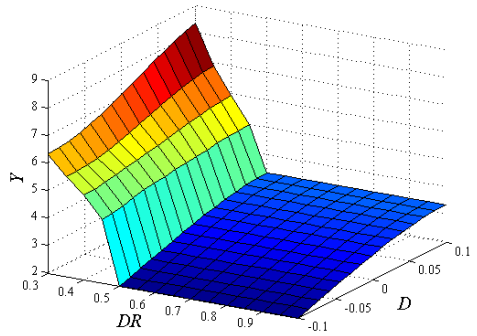


Fig. 5. The output surface of FIS

3.3.4. Fuzzy Q-learning

In order to further improve the convergence rate, fuzzy Q-learning is proposed to avoid obstacles on the basic of the optimized Q-learning. Daily experience is integrated into Q-learning by initializing $Q(s,a)$ using FIS. The efficiency of obstacle avoidance will increase by learning based on its experience.

3.4. Convergence analysis

The fuzzy Q-learning algorithm is convergent under certain conditions. In this paper, Q-learning is a deterministic Markov Decision Process (MDP). According to the convergence theorem of Q-learning MDP [33], the algorithm is convergent if three conditions are satisfied. Firstly, the immediate reward values are bounded. That is, there exists a positive constant C which make all state-action pair satisfy $|r(s,a)|<c$. Secondly, the initial value of $Q(s,a)$ is finite values and the discount factor γ meets the condition $0\le\gamma<1$. Thirdly, each state-action pair should be usually visited infinitely. $\gamma=0.9$ and immediate reward satisfies $|r(s,a)|\le 100$ in this paper. In the learning process of fuzzy Q-learning, the robot will try enough times so that the third condition can be satisfied. So the fuzzy Q-learning obstacle avoidance algorithm is convergent.

Table 3
Fuzzy Q-learning algorithm

Algorithm 3. Fuzzy Q-learning algorithm
Initialize all $Q(s,a)$ values based on FIS
Observe the current state s_t
Repeat (for each episode):
Choose an action a_t according to Eq. (5) and execute a_t
Receive immediate reward r_t
Observe new state s_{t+1}
Update $Q(s_t,a_t)$
$Q(s_t,a_t) \leftarrow r + \gamma \max_a(s_{t+1},a)$
$s_t \leftarrow s_{t+1}$

3.5. Fractional PI controller

The control noise can lead to the movement deviation of the robot when the robot walks. The deviation denotes the difference between the desired pose of the robot and the actual pose of the robot. In order to reduce the influence of control noise, fractional PI (PI^{α}) controller is designed, which can give a reasonable control input of correcting deviation. The transfer function of PI^{α} is [26]

$$G(s) = k_p + \frac{k_i}{s^{\alpha}} \tag{9}$$

We use trial and error method to get the values of k_p and k_i in the fractional PI controller. where $k_p = 0.9$, $k_i = 0.1$ and $\alpha = 0.1$. PI^{α} controller can consider the current and previous deviation, which can give a reasonable control input of correcting deviation. Fig. 6 shows the diagram of PI^{α} controller. x_R is the real pose of the robot and x_D is the desired pose of the robot. ee is the difference between x_D and x_R. u is the deviation correction control input.

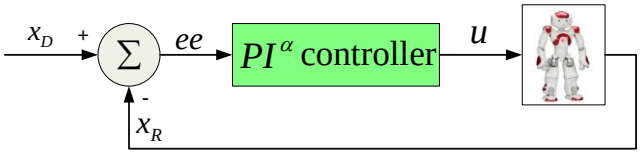
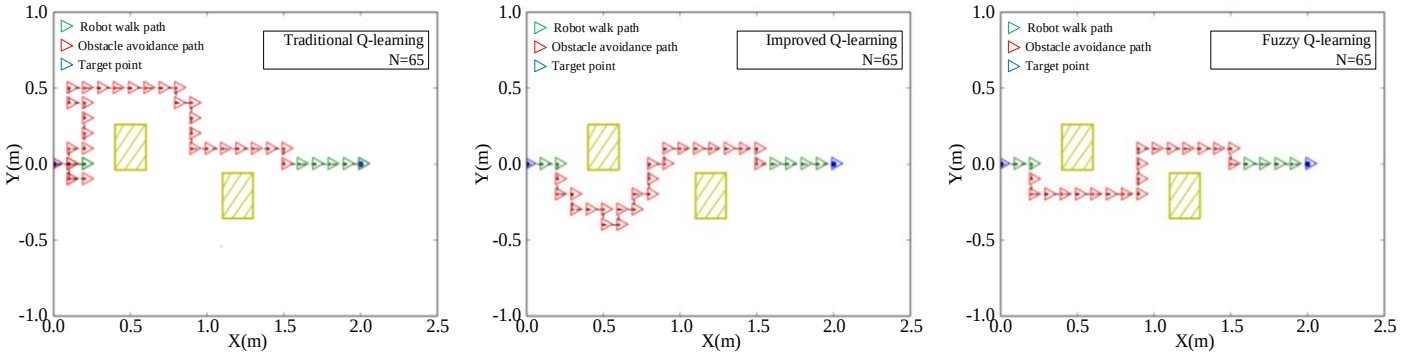
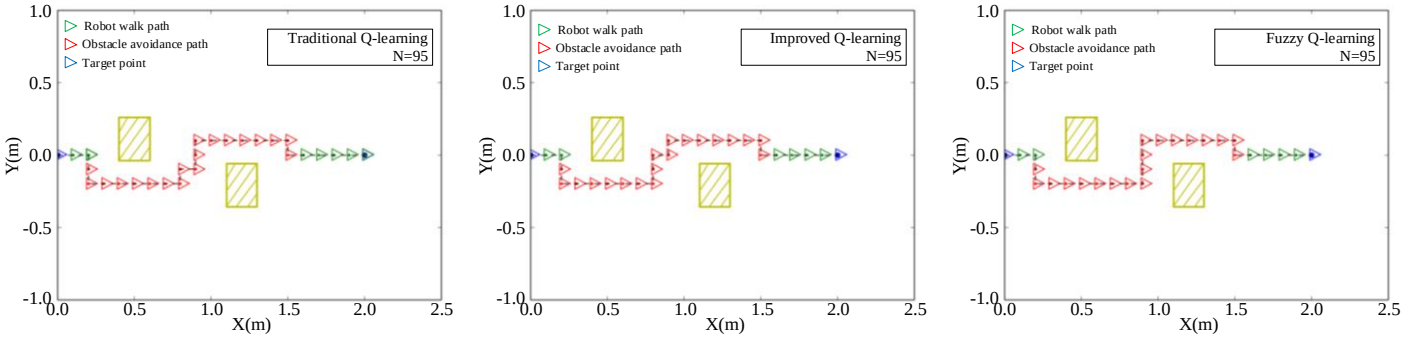


Fig. 6 The diagram of fractional PI controller



(a) The result of traditional Q-learning (b) The result of improved Q-learning (c) The result of fuzzy Q-learning

Fig. 7. The results of path planning at learning times $N = 65$



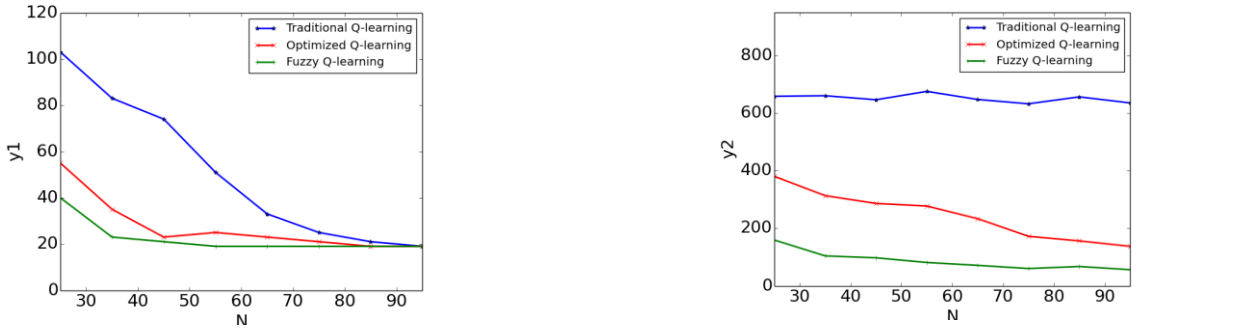
(a) The result of traditional Q-learning (b) The result of improved Q-learning (c) The result of fuzzy Q-learning

Fig.8. The results of path planning at learning times $N = 95$

4. Simulation results

The simulation results of obstacle avoidance are provided in this section. Firstly, the learning speed of the traditional Q-learning, the optimized Q-learning and the fuzzy optimized Q-learning is compared, and convergence rate of the fuzzy Q-learning is the fastest. Then control noise is added to the movement of the robot to test the performance of the fractional PI controller. In simulation environment, the robot moves from original point $(0,0)$ to the target point $(2,0)$. Checkerboard path planning model is established once the robot encounters obstacles. Then the robot will plan an obstacle avoidance path according to the proposed algorithm. The k in Eq. (5) is 2.0 ($k = 2.0$) and the discount factor of the traditional Q-learning, the optimized Q-learning and the proposed fuzzy Q-learning is 0.9 ($\gamma = 0.9$) in this paper.

Fig. 7 and Fig. 8 show the results of path planning using the traditional Q-learning [17], the optimized Q-learning [18,19] and the fuzzy Q-learning at learning times $N = 65$ and $N = 95$, respectively, when the number of the obstacle is two. In these two figures, the triangle represents the robot and the rectangle with slash is the obstacle. The red path (from the point $(0.2,0)$ to the point $(1.5,0)$) in Fig. 7 and Fig. 8 is the obstacle avoidance path obtained by the traditional Q-learning, the optimized Q-learning and the fuzzy Q-learning, respectively. The robot will perform obstacle avoidance when $d_{ro} \le 0.35m$. d_{ro} is the minimum distance between the robot and obstacle. The size of the checkerboard model built by the robot is $1.2m \times 1.5m$. Fuzzy Q-learning can find the optimal obstacle avoidance path at $N = 65$, while the traditional Q-learning and the optimized Q-learning can't find it at $N = 65$. The traditional Q-learning, the optimized Q-learning and the fuzzy Q-learning can find an optimal obstacle avoidance path when $N = 95$.



(a) Statistics of the number of steps of obstacle avoidance of three Q-learning (b) Statistics of the average value of tentative steps of three Q-learning

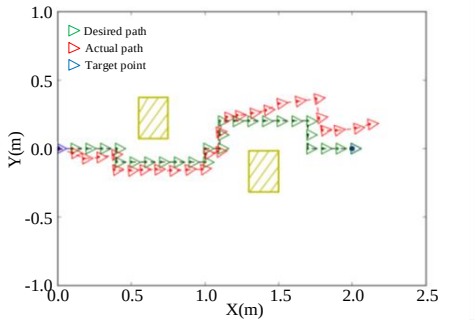
Fig. 9. The results of three Q-learning algorithm at different learning times

Fig. 9. (a) shows the number of steps of obstacle avoidance of the traditional Q-learning, the optimized Q-learning and the proposed fuzzy Q-learning algorithm at different learning times N when there are two obstacles. Fig. 9. (b) shows the average value of tentative steps of the traditional Q-learning, the optimized Q-learning and the fuzzy Q-learning after learning N times. Fig. 9 demonstrates that the proposed fuzzy Q-learning algorithm can find an optimal obstacle avoidance path more quickly than the traditional Q-learning and the optimized Q-learning algorithm.

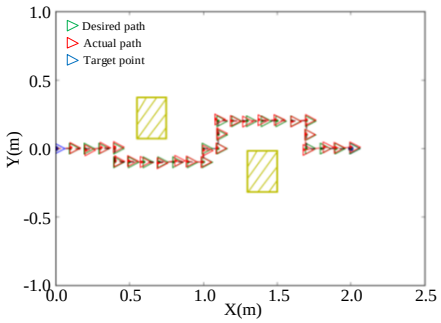
The simulation results show that the proposed fuzzy optimized Q-learning algorithm has higher learning speed and is more effective than the traditional Q-learning and the optimized Q-learning.

Gaussian noise $n = N(0, q^2)$ is added to the motion control and $q = 0.02$. PI^{α} controller is used to restrain the noise influence.

In Fig. 10, the green path is a desired path and the red path is the actual path. Fig. 10. (a) shows the result of obstacle avoidance under noise. Fig. 10. (b) shows the result after the correction of PI^{α} controller. Fig. 11. (a) shows the position error after the correction of PI^{α} controller. Fig. 11. (b) shows the error of the robot angle $\boldsymbol{\theta}$ after the correction of PI^{α} controller. It is obvious that the fractional $\boldsymbol{PI}$ controller can reduce motion deviation effectively.

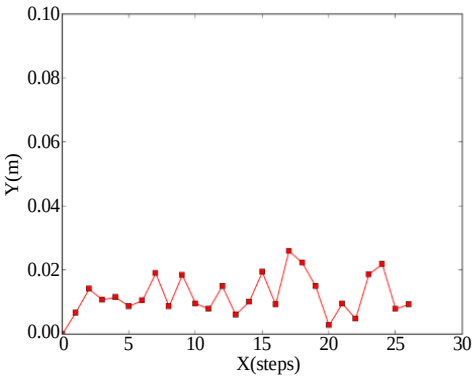


(a) The walking trajectory of obstacle avoidance without PI^{α} controller

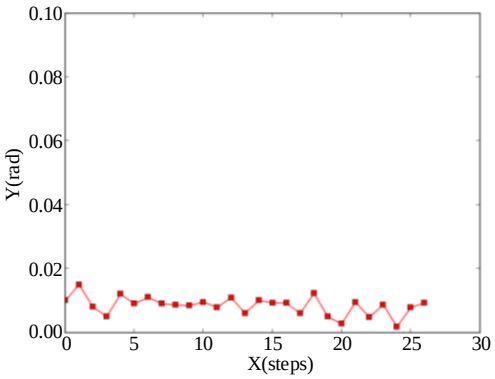


(b) The walking trajectory of obstacle avoidance with PI^{α} controller

Fig. 10. The result of the correction of the noise by the PI^{α} controller



(a) The position error after the correction of PI^{α} controller



(b) The error of the robot angle $\boldsymbol{\theta}$ after the correction of PI^{α} controller

Fig. 11. The error after the correction of PI^{α} controller

From the simulation results, we can see that our approach has 3 advantages: 1. The fuzzy optimized Q-learning (FOQL) algorithm is first used to avoid obstacles and has faster learning speed and less steps than the traditional Q-learning and the optimized Q-learning. 2. Active SLAM framework combining FOQL avoidance obstacles algorithm with FastSLAM is proposed to improve the ability of autonomous navigation. 3. The Fractional PI^{α} controller can reduce motion deviation effectively and restrain the influence of Gaussian noise.

5. Experiment

NAO[34] robot shown in Fig. 12 is a humanoid robot with biped walking. There is a laser scanner at the head of NAO robot. Laser scanner is used for observing object position in unknown environment. Fig. 13 is the detecting result of laser scanner, and the black points are the two obstacles. The laser range is from 20mm to 5600mm, 240° area scanning range with 0.36° angular resolution. The laser can provide the distance between the obstacle and robot.

Fig. 12. The NAO robot

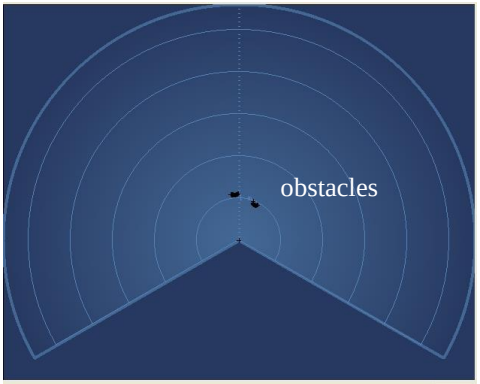


Fig.13 Target's laser scan result, the black points are the reflect of obstacle

The experiment scene of obstacle avoidance is shown in Fig. 14. The NAO robot walks in the environment with obstacles. The task of the NAO robot is to reach the destination bypassing obstacles. The obstacle avoidance of fuzzy optimized Q-learning algorithm with the same setup as the simulation is used in the experiment. The purpose of the experiment is to demonstrate the feasibility of the fuzzy Q-learning obstacle avoidance algorithm. In the process of walking to the destination, the NAO robot localization is based on FastSLAM algorithm. There are three landmarks in the experiment. Obstacle 1 and obstacle 2 are not only the obstacles in the experiment, but also the landmarks in FastSLAM algorithm. Fractional $\boldsymbol{PI}$ controller rectifies the deviation according to the estimation position of the NAO robot by FastSLAM algorithm.

Fig. 14. Experiment scene

In the beginning of the experiment, the NAO robot observes the surrounding environment. If the minimum distance between the robot and the obstacle $d_{ro} \le 0.35$, the NAO robot starts to avoid obstacle by fuzzy Q-learning obstacle avoidance algorithm. Otherwise, the NAO robot will walk until it reaches the destination. The flow chart of the experiment is shown in Fig. 15.

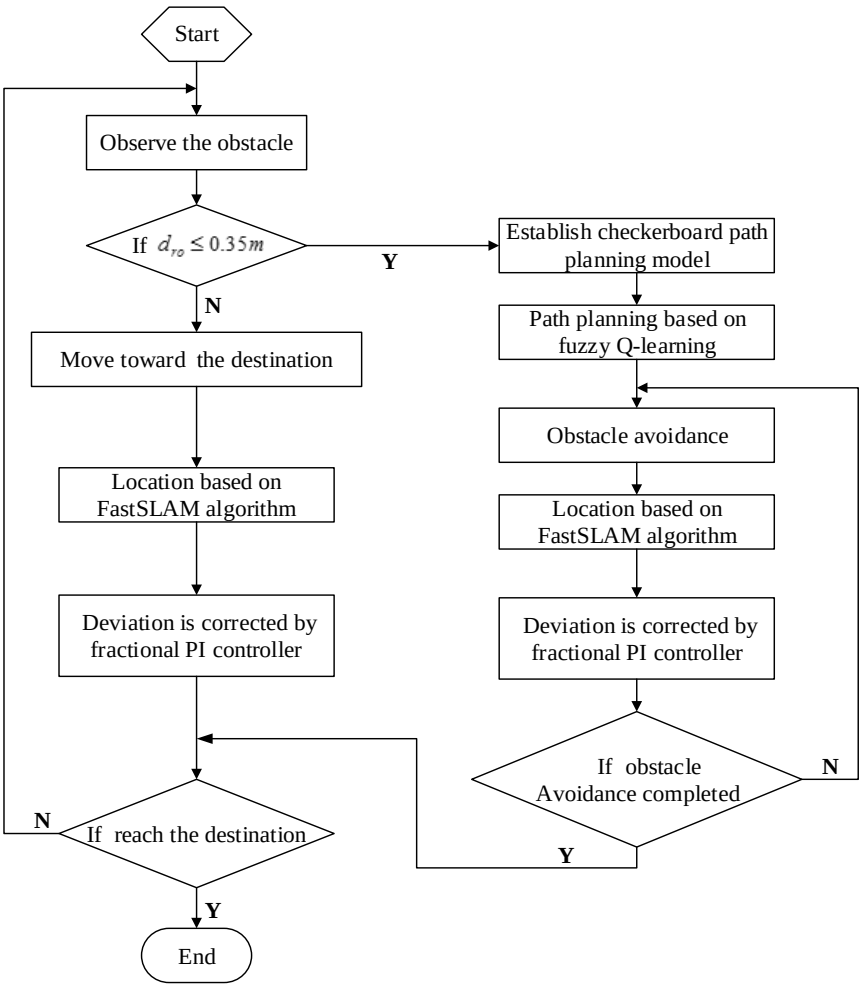


Fig. 15. The flow chart of the system

(a). view 1

(b). view 2

Fig. 16. The process of the obstacle avoidance of NAO robot

Fig. 16 shows the process of the obstacle avoidance of the NAO robot. Fig. 17 shows the result of the optimal obstacle avoidance. The green path is a desired walking path of the NAO robot. The red path is an estimated path by FastSLAM algorithm. The red plus sign is the estimated position of the landmarks. The experiment demonstrates that the NAO robot can successfully avoid obstacles by using the proposed method in this paper. There is some slight difference between the estimated path and the desired path because of external interference and the estimated error.

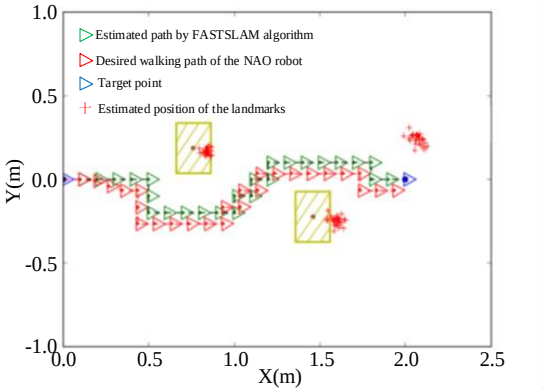


Fig. 17. The result of the obstacle avoidance experiment of NAO robot

6. Conclusion

In this paper, a fuzzy optimized Q-learning (FOQL) combining with FastSLAM is proposed to plan an optimal path without colliding any obstacles in the environment. It is successfully applied to autonomous walking of the NAO robot in unknown environment. The simulation results show that FOQL avoidance obstacles algorithm proposed in this paper has faster convergence rate than the traditional Q-learning and the optimized Q-learning. To reduce the motion deviation of the robot, fractional PI controller is used when the robot walks. Finally, the experiment demonstrates that FOQL algorithm fused with FastSLAM proposed in this paper can effectively avoid obstacles and locate in the autonomous navigation.

Our future works will focus on obstacle avoidance of the robot in dynamic environment, which includes not only fixed obstacles but also dynamics obstacles, and more complex environment should be further studied.

References

[1] Lee, S. H., and Goswami, A. (2012). "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground". *Autonomous Robots*, Vol. *33* No. 4, pp. 399-414.

[2] Radford, N. A., Strawser, et al. (2015). "Valkyrie: Nasa's first bipedal humanoid robot". *Journal of Field Robotics*, Vol. *32* No. 3, pp. 397-419.

[3] Liu, Z., Chen, C., and Zhang, Y. (2014). "Decentralized robust fuzzy adaptive control of humanoid robot manipulation with unknown actuator backlash". *IEEE Transactions on Fuzzy Systems*, Vol. *23* No. 3, pp. 605-616.

[4] Lim, J., and Oh, J. H. (2016). "Backward ladder climbing locomotion of humanoid robot with gain overriding method on position control". *Journal of Field Robotics*, Vol. *33* No. 5, pp. 687-705.

[5] Kuindersma, S., Deits, R., Fallon, M., et al. (2016). "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot". *Autonomous Robots*, Vol. *40* No. 3, pp. 429-455.

[6] Wen, S., Chen, X., Ma, C., et al. (2015). "The Q-learning obstacle avoidance algorithm based on EKF-SLAM for NAO autonomous walking under unknown environments". *Robotics and Autonomous Systems*, Vol. *72*, pp. 29-36.

[7] Wen, S., Othman, K. M., Rad, A. B., et al. (2014). "Indoor SLAM using laser and camera with closed-loop controller for NAO humanoid robot". *Abstract and Applied Analysis* Vol. 2014. Hindawi.

[8] Paredes, A. C., Malfaz, M., and Salichs, M. A. (2013). "Signage system for the navigation of autonomous robots in indoor environments". *IEEE Transactions on Industrial Informatics*, Vol. *10* No. 1, pp. 680-688.

[9] Germanos, V., and Secco, E. L. (2016, August). Formal verification of robotics navigation algorithms. In *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES). 2016 Paris:* IEEE. pp. 177-180.

[10] Hossain, M. A., and Ferdous, I. (2015). "Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique". *Robotics and Autonomous Systems*, Vol. *64*, pp. 137-141.

[11] Cherubini, A., Spindler, F., and Chaumette, F. (2014). "Autonomous visual navigation and laser-based moving obstacle avoidance". *IEEE Transactions on Intelligent Transportation Systems*, Vol. *15* No. 5, pp. 2101-2110.

[12] Franzè, G., and Lucia, W. (2015). "An obstacle avoidance model predictive control scheme for mobile robots subject to nonholonomic constraints: A sum-of-squares approach". *Journal of the Franklin Institute*, Vol. *352* No. 6, pp. 2358-2380.

[13] Wen, S., Zheng, W., Zhu, J., Li, X., & Chen, S. (2011). "Elman fuzzy adaptive control for obstacle avoidance of mobile robots using hybrid force/position incorporation". *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. *42* No. 4, pp. 603-608.

[14] Melingui, A., Merzouki, R., Mbede, J. B., et al. (2014). "A novel approach to integrate artificial potential field and fuzzy logic into a common framework for robots autonomous navigation". *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, Vol. *228* No. 10, pp. 787-801.

[15] Yu, Z. Z., Yan, J. H., Zhao, J., Chen, Z. F., et al. (2011). "Mobile robot path planning based on improved artificial potential field method". *Harbin Gongye Daxue Xuebao(Journal of Harbin Institute of Technology)*, Vol. *43* No. 1, pp. 50-55.

[16] Borenstein, J., and Koren, Y. (1991). "The vector field histogram-fast obstacle avoidance for mobile robots". *IEEE transactions on robotics and automation*, Vol. *7* No. 3, pp. 278-288.

[17] Samsudin, K., Ahmad, F. A., and Mashohor, S. (2011). "A highly interpretable fuzzy rule base using ordinal structure for obstacle avoidance of mobile robot". *Applied Soft Computing*, Vol. *11* No. 2, pp. 1631-1637.

[18] Syed, U. A., Kunwar, F., and Iqbal, M. (2014). "Guided Autowave Pulse Coupled Neural Network (GAPCNN) based real time path planning and an obstacle avoidance scheme for mobile robots". *Robotics and autonomous systems*, Vol. *62* No. 4, pp. 474-486.

[19] Karami, A. H., and Hasanzadeh, M. (2015). "An adaptive genetic algorithm for robot motion planning in 2D complex environments". *Computers & Electrical Engineering*, Vol. *43*, pp. 317-329.

[20] Yoo, B., and Kim, J. (2016). "Path optimization for marine vehicles in ocean currents using reinforcement learning". *Journal of Marine Science and Technology*, Vol. *21* No. 2, pp. 334-343.

[21] Zhang, B., Mao, Z., Liu, W., et al. (2015). "Geometric reinforcement learning for path planning of UAVs". *Journal of Intelligent & Robotic Systems*, Vol. *77* No. 2, pp. 391-409.

[22] Kiumarsi, B., Lewis, F. L., Modares, H., et al. (2014). "Reinforcement Q-learning for optimal tracking control of linear discrete-time systems with unknown dynamics". *Automatica*, Vol. *50* No. 4, pp. 1167-1175.

[23] Galindo-Serrano, A., and Giupponi, L. (2010). "Distributed Q-learning for aggregated interference control in cognitive radio networks". *IEEE Transactions on Vehicular Technology*, Vol. *59* No. 4, pp. 1823-1834.

[24] Wen, S., Hu, B., and Lam, H. K. (2015). "Reinforcement learning optimization for base station sleeping strategy in coordinated multipoint (CoMP) communications". *Neurocomputing*, Vol. *167*, pp. 443-450.

[25] Deng, C., and Er, M. J. (2004, July). "Real-time dynamic fuzzy Q-learning and control of mobile robots". In *2004 5th Asian Control Conference (IEEE Cat. No. 04EX904)* 2004, Melbourne: IEEE. Vol. 3, pp. 1568-1576).

[26] Goldberg, Y., and Kosorok, M. R. (2012). "Q-learning with censored data". *Annals of statistics*, Vol. *40* No. 1, pp. 529.

[27] Wen, S., Chen, X., Zhao, Y., et al. (2014). "The study of fractional order controller with SLAM in the humanoid robot". *Advances in Mathematical Physics*, Vol. *2014*.

[28] Fei, W., Jin-Qiang, C. U. I., Ben-Mei, C. H. E. N., et al. (2013). "A comprehensive UAV indoor navigation system based on vision optical flow and laser FastSLAM". *Acta Automatica Sinica*, Vol. *39* No. 11, pp. 1889-1899.

[29] Doltsinis, S., Ferreira, P., and Lohse, N. (2014). "An MDP model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis". *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. *44* No. 9, pp. 1125-1138.

[30] Watkins, C. J., & Dayan, P. (1992). "Q-learning". *Machine learning*, Vol. *8* No. 3-4, pp. 279-292.

[31] Sivananthan, S. (2016). "Multi-penalty regularization in learning theory". *Journal of Complexity*, Vol. *36*, pp. 141-165.

[32] Hwang, K. S., Tan, S. W., and Chen, C. C. (2004). "Cooperative strategy based on adaptive Q-learning for robot soccer systems". *IEEE Transactions on Fuzzy Systems*, Vol. *12* No. 4, pp. 569-576.

[33] Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994). "Machine learning". *Neural and Statistical Classification*, Vol. *13*.

[34] Wen, S., Hu, X., Lv, X., et al. (2019). "Q-learning trajectory planning based on Takagi–Sugeno fuzzy parallel distributed compensation structure of humanoid manipulator". *International Journal of Advanced Robotic Systems*, Vol. *16* No. 1, 1729881419830204.