Obstacle Avoidance of Hexapod Robots Using Fuzzy Q-Learning

Jun Hong, Kaiqiang Tang, Chunlin Chen
Department of Control and Systems Engineering
School of Management and Engineering
Nanjing University, Nanjing 210093, China
Email: clchen@nju.edu.cn

***Abstract*—Safe and autonomous obstacle avoidance plays an important role in the navigation control of hexapod robots. In this paper, we combine the method of reinforcement learning with fuzzy control to achieve the autonomous obstacle avoidance for a hexapod robot in complex environments. A fuzzy Q-learning algorithm is first presented and an obstacle avoidance approach is proposed using the Fuzzy Q-learning algorithm regarding the specific requirements of the hexapod robot. Then, the proposed approach is implemented for a real hexapod robot system that uses ultrasonic sensors to detect the obstacles in an unknown environment and learns an optimal policy to avoid the obstacles. Several groups of experiments are carried out to verify the performance of the proposed approach.**

***Index Terms*—Fuzzy Q-learning, Hexapod robots, Obstacle avoidance, Reinforcement learning**

I. Introduction

Non-structural road complex environmental information limits the use of traditional wheeled or crawler vehicles [1], [2]. In contrast to wheeled or crawler robots, hexapod robots [3]-[10] are more maneuverable and have the ability to move on tough environment because of its rich degrees of freedom [11], so it has a wide range of application prospects in touch works such as forest harvesting, mine development, disaster relief. Therefore, the study of hexapod robot has received great attention and how to improve the capability of obstacle avoidance for hexapod robot has become a key to hexapod robot research. Consequently, the major challenge is the realization of real-time obstacle avoidance control strategy [12] because the obstacles are dynamic as the hexapod robot is moving. The two main issues in the obstacle avoidance for hexapod robots are obtaining information of the obstacles and controlling the movement of the hexapod robot at the current state. Many positive results on this topic have been reported [13]-[15].

However, in the field of obstacle avoidance for hexapod robots, the obstacle information is complex and unknown [2] which can not be easily obtained in a tough environment, therefore, we can not solve this type of problem by establishing an accurate mathematical model. Fuzzy control [16] is a good way to solve this type of ambiguous system, it uses the fuzzy variable to represent the obstacle distance of the robot, and then the fuzzy variable is transformed into the precise control variable, which can achieve better control performances. This kind of control approach focuses on the establishment of knowledge base, which is mostly based on a large number of expert experience and test results, and individual differences may lead to a great difference, which has some limitations in the implementation of the process, a good approach is through learning algorithms. Reinforcement Learning (RL) [17] is one of the main machine learning domains which allows autonomous agents to learn and improve their performance through the obtained experiences while interacting with an unknown environment [18]. Reinforcing learning maximizes the cumulative reward earned by an action from the environment through the interaction between the environment and the agent. This method is different from the supervision of learning technology which is through the case and the counter example to tell what kind of behavior to take, but through trial and error to find the optimal strategy. The main feature of reinforcement learning is that only the optimal environmental reward can be used to discover the optimal behavior strategy leading to slow convergence time and long consumption time in the learning process. During the learning process, the next state of the robot can be used to measure the dimension of the environment reward, this can speed up the convergence rate of the algorithm to a certain extent.

The difficulty of fuzzy control is to establish the knowledge base which is always based on expert experience, this problem can be solved by way of learning. Also, many practical problems can not be regarded as Markov decisions which limits the application of reinforcement learning. Many problems can be discretized and be transferred into a limited set of states by fuzzy logic, this can create conditions for the use of reinforcement learning, so the combination of fuzzy control and reinforcement learning may be interesting in the field of obstacle avoidance for hexapod robots.

In this paper, a Fuzzy Q-Learning algorithm of obstacle avoidance is proposed. We divide the obstacle avoidance for hexapod robots into two parts: determining the current state and choosing the appropriate action. This main idea of the algorithm is as follows: First, we represent the distance of obstacles with membership functions, then, we can use rule base to make sure which rule will be taken, then the Q function is established and updated by Q-learning. Lastly, we can obtain the best strategy of obstacle avoidance based on the Q function as the learning process is over. The knowledge base of the fuzzy control is gradually formed based on the results of the learning process [19]. As the role of fuzzy control is to discretize the problem, so the convergence of Fuzzy Q-Learning algorithm is consistent with typical reinforcement learning.

This paper is organized as follows. Section II introduces the problem formulation of obstacle avoidance and the mechanical

978-1-5386-2726-6/17/$31.00 ©2017 IEEE

structure of the hexapod robot. Section III introduces the fuzzy Q-Learning algorithm for the obstacle avoidance control of hexapod robots. Section IV gives the details of the experiments. Section V concludes the paper.

II. Problem Formulation

In this paper, a CR-6 hexapod robot as shown in Fig. 1 is used as an experimental platform, which is equipped with three ultrasonic sensors. This kind of robots has 18 degrees of freedom and can easily measures the distances for the obstacles. It has a variety of gaits and is able to complete all kinds of actions.

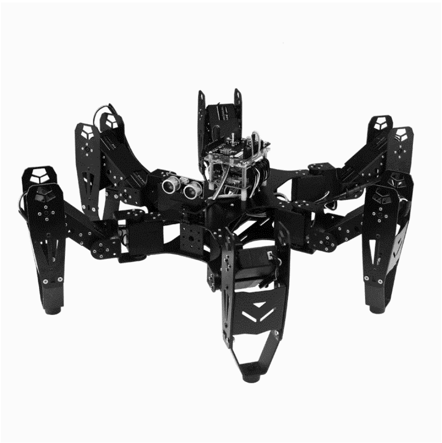

Fig. 1: The CR-6 hexapod robot platform.

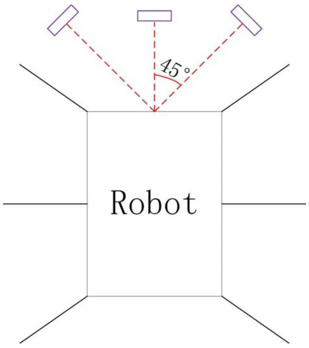

Fig. 2: Arrangement of ultrasonic sensors of the hexapod robot.

The ability to avoid obstacles is one of the most important capabilities of hexapod robots. In order to achieve obstacle avoidance for the robot, the main task is to prevent the collision with obstacles. So we need to develop a real-time obstacle avoidance algorithm. The adoption of control strategy should be conducted simultaneously with the measurement of obstacles as the obstacles are dynamic when the robot is moving. The process of obstacle avoidance can be divided into two parts: conversing obstacle information into a finite set of states and taking a proper action at the current state.

1) Detection of Obstacles: As is shown in Fig. 2, we can measure the obstacle information in three directions ahead of the robot by its sensors, these distance information will be processed with fuzzy sets and then the obstacle information can be conversed into a finite set of states.

2) Control Design: As the information of obstacles is transferred into a current state, the robot needs to decide which action to take at a certain state according to the learned control strategy.

III. Reinforcement Learning and Fuzzy Q-Learning

In this section, we will first introduce reinforcement learning and fuzzy control, then, a fuzzy Q-learning algorithm of obstacle avoidance for hexapod robots is presented.

A. Reinforcement Learning and Fuzzy Control

For typical reinforcement learning algorithms, the agents interact with the environment so as to maximize the reward from the environment. This kind of interaction process can be modelled as the Markov Decision Process (MDP) [20] which consists of four elements (S,A,R,T). In this collection, S is a finite state space and A is a finite action space, T is the state transition function, which describes the probability that the robot will take the action a at the state s, and R describes the reward of the hexapod robot moving from state s to the next state. The aim of reinforcement learning is to find an optimal strategy $\pi : S \rightarrow A$, so that the agent at a certain point in time, given a state s, selects the corresponding action a, and gets the maximum reward from environment.

Reinforcement learning focuses on how the agent obtains the optimal strategy when the T function and the R function are unknown. First, the definition of the value function under the optimal strategy is given as:

$$V^*(s) = \max(\gamma \sum T(s,a,s')(r(s,a,s' + V^*(s')))). \tag{1}$$

The Bellman equation is usually used to find the optimal action value function:

$$Q^*(s_t,a_t) = E[r(s_t,a_t) + \gamma \max_{a'_{t+1}} Q^*(s_{t+1},a'_{t+1})]. \tag{2}$$

The value function approximated by Bellman iteration under the optimal strategy is:

$$V^*(s) = (1-\alpha)V^*(s) + \alpha(r(s,a,s' + \gamma V^*(S'))). \tag{3}$$

In this formula, $\gamma \in (0,1)$ is the discount factor, which can affect the learning speed and convergence rate to a certain extent. α is the learning rate.

Q-Learning is a model-independent reinforcement learning algorithm proposed by Watikns, it iterates through the state-action reward, which takes $Q(s_t,a_t)$ as the reward function. The basic form of Q-Learning algorithm is:

$$\begin{aligned} &Q(s_t,a_t) = Q(s_t,a_t) + \\ &\alpha(r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)). \end{aligned} \tag{4}$$

In this formula, r_{t+1} is the instant reward when taking action a at the state s_t, and $\max_a Q(s_{t+1},a)$ is the maximum reward function when taking action a at the state s_{t+1}.

The first step of Q-Learning is to initialize Q function, next the agent determines to take action a at the state s using $\varepsilon - greedy$ strategy, then we can get empirical knowledge and training experience $(s_t, s_t, s_{t+1}, r_{t+1})$, the Q function is updated according to the formula (4). When the agent accesses the target state, the algorithm terminates an iteration loop, and the algorithm continues to start a new iteration loop from the initial state until the end of the learning.

Many traditional control methods are based on the precise mathematical model of the controlled object. However, with the improvement of the system complexity, many variables and problems can not be described precisely, and it is almost impossible to solve such problems in a typical way. Based on this, we can use fuzzy logic to describe things that can not be accurately described and use fuzzy control to control them. Although this is an inaccurate representation, we can establish membership functions to indicate them. Fuzzy control can be divided into four steps: blurring the input of the system, building a knowledge base, fuzzy reasoning and converting fuzzy variables to exact variables. Firstly, the input is converted to the value of the domain in the appropriate proportion, the process of measuring the physical quantity is described by the colloquial variable, and the relative membership degree of the value is obtained according to the appropriate linguistic value. This colloquial variable is called the fuzzy subsets. Secondly, we need to establish a knowledge base which consists of data base and rule base where the data base provides a standard of handling fuzzy data, and the rule base describes the control objectives and strategies by a set of control rules. Then, the system needs to imitate the fuzzy concept of human judgment, use fuzzy logic and fuzzy inference method to deduce and get fuzzy control signal. The last step is to convert fuzzy control signal into exact control variables. The core idea of fuzzy control is to qualitatively blur variables that can not be described exactly, and then establishing membership functions for them.

B. Fuzzy Q-Learning Algorithm of Obstacle Avoidance

In this subsection, a Fuzzy Q-learning algorithm for obstacle avoidance is presented by combining fuzzy logic and reinforcement learning. Fuzzy logic is used to transfer the distance of obstacles into a finite set of states so that the obstacle avoidance can be regarded as a Markov process. Q-Learning algorithm is used to obtain the control rules which is the most important for fuzzy control. The hexapod robot is equipped with three ultrasonic sensors to measure the distance of obstacles during operation, then we use fuzzy logic to blur the measurement, and the distance is separated into near middle and far three ambiguities shown in Fig. 3. In this way, the obstacle information will be transferred into 27 states, and we divide the obstacle avoidance of hexapod robot into two parts: confirming current state and taking corresponding action. As the robot will meet several different states at a time, so we need to establish a fusion mechanism and then ensure the next action. After a small amount of learning episodes, the hexapod robot can take the best action at a time which can let it avoid obstacles safely and efficiently.

Firstly, we have to establish membership function for each input. As the input is separated into three ambiguities, the membership function for each input is set as formula (5), the image of the membership function is shown in Fig. 3. Then, there are 27 states for the robot, and we define three actions for each state: Going forward, Turning Left and Turing Righe, so there are 81 rules in total. There may be several rules that will be touched off for the robot at a time, but not every rule is correct, so the aim of learning is to find the most suitable action for each state. After the learning process, we can get the best strategy for the robot. As the robot will meet different states at a time, we need to make a fusion for all the states. All the states are independent, so we calculate the product of the reward function and its membership value for each state, and choose the maximum reward function of them as the standard of action taking.

$$F(x) = \begin{cases} \exp(\frac{-x^2}{1500}), & x \in (0,100); \\ \\ \exp(\frac{-(x-125)^2}{1500}), & x \in (25,225); \\ \\ \exp(\frac{-(x-250)^2}{1500}), & x \in (150,250); \end{cases} \tag{5}$$

Where x is the distance of obstacles and $F(x)$ is the membership.

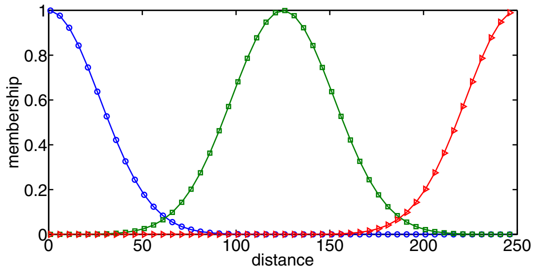

Fig. 3: The membership function for the distance of obstacles.

As the only information available for learning process is the system reward, in order to distinguish the impact of different states on environment reward, we need to blur the reward from environment based on the state. We have to create a proper membership function for the environment reward, there are 27 states, we can encode them from 000 to 222. Usually, we use Gaussian function to represent the membership function of the environment reward which can be indicated by:

$$r(s) = 20 * \exp(\frac{-(s-2)^2}{2}). \tag{6}$$

Where s is defined as followed:

$$s = \min(a,b,c). \tag{7}$$

There are three measurements of the obstacles, and every measurement will be blurred by its own membership function, here we use three variables $< a, b, c >$ to represent them.

Above all, the detailed FQL algorithm is shown in Algorithm 1. It is worth mentioning that knowledge gained from past learning episodes is not necessarily current, and we need to find a balance between exploration and exploitation, so we use ε-greedy strategy as the action-taken strategy. In this way, we can ensure the algorithm will not be localized optimally during the learning process.

IV. Experimental Results

In this section, the strategy of obstacle avoidance is learned and trained in simulation, then, we test it on the CR-6 hexapod robot platform.

A. Learning by Simulation

Firstly, we need to establish a model to describe the obstacle avoidance of hexapod robot. As the the distance of the obstacles will be blurred, we do not need to get accurate value. Here we set the hexapod robot as a particle, and establish an array with a size of 40*40 to indicate the environment in which the hexapod robot is running. An element in the array of 1 indicates that there is an obstacle, and an element in the array of 0 indicates that this area can be pass which is shown in Fig. 4. The information of obstacles can be blurred into 27 states, so we need an array with a size of 27*3 to store state-action Q function, this array will be initialized as zero. In order to find a balance between explore and exploit, we use ε-greedy strategy as action-taken strategy, here we set ε as 0.1, this can make sure that the algorithm will explore new knowledge with certain probability. If the robot hits the obstacles during the learning period, there will be a great negative reward to punish this action.

We randomly set the number and location of obstacles in the environment to eliminate the occasional factors in the learning process. The process of learning can be regarded as a combination of explore and exploit. At the beginning of the learning process, the experience gained has a certain chance of being influenced by the environment, so these knowledge will be selectively accepted, and the focus of learning process is to explore more. As the learning episodes continues to increase, our recognition of the knowledge is increasing, at this stage, the focus of learning process is to exploit more. So the learning rate α and ε is set as:

$$\begin{cases} \alpha = \frac{0.05}{0.01*s+1}. \\ \\ \varepsilon = \frac{0.02}{0.01*s+1}. \end{cases} \tag{8}$$

Where s is the learning steps, α and γ are initialized as 0.05 and 0.02 which will increase as the learning episode is growing. At the start of learning process, we randomly select a position as the initial position of the hexapod robot, and then hexapod robot continues to run on the map established above, the hexapod robot learns through trial and error. After a certain number of learning episodes, we can obtain the complete state-action Q function. Based on this, the best strategy of obstacle avoidance for hexapod robots is formed.

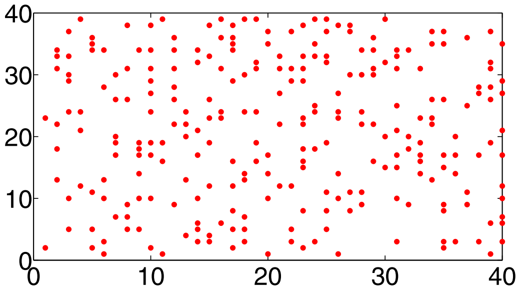

Fig. 4: The simulation environment with clustered obstacles.

As mentioned above, the instant reward has great impact on the convergence rate of the algorithm, so we use a fuzzy variable defined by formula (7) to define the reward from the environment, and the learning results id compared with the fuzzy Q-Learning algorithm without fuzzy reward. As is shown in Fig. 5, the algorithm with fuzzy reward has a faster convergence rate which shows that the algorithm with fuzzy reward has a better convergence speed than the other.

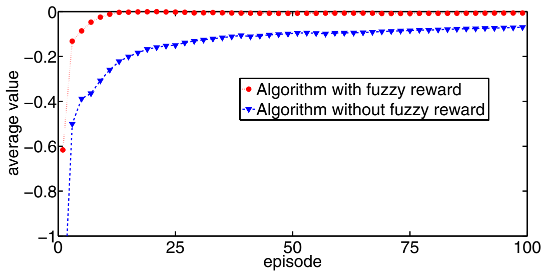

Fig. 5: Performances of fuzzy Q-learning algorithm with and without fuzzy reward.

After obtaining the best strategy for obstacle avoidance of a hexapod robot, we establish a new map to test the results. We randomly choose a start position for the hexapod, then the robot moves in the map using the strategy. As is shown in Fig. 6, the robot can safely move many steps in the map independently.

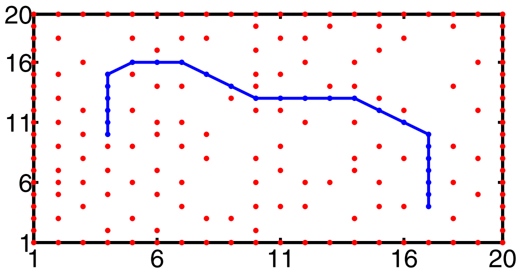

Fig. 6: Trajectory of the hexapod robot.

In addition, we change some parameters of the algorithm

Algorithm 1 Fuzzy Q-Learning algorithm of obstacle avoidance

```
Initialize the map of the learning environment and the testing environment, state-action function $Q(s_t, a_t)$
Initialize learning rate $\alpha$ and discount factor $\gamma$
Initialize the episodes marked as $M$ for the learning period and the steps Marked as $s$ for each episode
for episode = 1, $M$ do
  for steps = 1, $s$ do
    Initialize a start position in the map randomly
    Measure the distances of the obstacles and blur them
    Ensure all the states marked as $s1$ that will meet and use the rule base to obtain the rules that have to be taken
    for state =1, $s1$ do
      Use $\varepsilon$-greedy strategy based on Q function to choose an action
      Take the action, and reach the next state $s_{t+1}$
      Use formula (6) and (7) to obtain $r_{t+1}$ and update the state-action Q function by formula (4)
    end for
    Make a fusion for all the states and decide the best action
    Take the action, and reach the next state $s_{t+1}$
    Update learning rate $\alpha$ and $\varepsilon$ as the steps increase by formula (8)
  end for
end for
Obtain the best strategy for hexapod robots based on the Q function
```

to test whether the FQL algorithm is robust enough to these changes. Two extra experiments are carried out as follows:

(1) Changing learning steps of one episode. The learning steps are set as 100 and 500. As is shown in Fig. 7, the learning process with the steps set as 500 has a faster convergence rate than the other, and also the average reward is bigger. The reason is that as the steps increase, the strategy obtained is with high accuracy, and the algorithm tends to exploit rather than explore because we have already obtain some useful experience. So the learning episodes will be greatly reduced.

(2) Changing learning rate of the algorithm. The learning rate α is set as 0.08, 0.01 and 0.20. As is shown in Fig. 8, the learning process with α set as 0.08 exhibits more robust performance than the learning process with α set as 0.20, but the learning process with α set as 0.01 tends to be a little slow. As we all know, if the learning process learns some wrong experience, there will be a serious punishment, so the experience will be badly influenced if the learning rate is set to be so big, however, if the learning rate is set to be too small, the learning process will be extended. So we need to set a proper value to have a better effect through multiple sets of experiments.

According to the two groups of experiments above, although some parameters may influence the results of the algorithm, we can see that the algorithm will eventually converge though some parameters of the algorithm are changed. It is proved that the FQL algorithm has a great adaptability to the changes of parameters.

B. Testing on real robots

With the learned optimal strategy, we test it on the CR-6 hexapod robot platform. We put some obstacles in the environment, and the robot moves independently. As shown in Fig. 9, the hexapod robot can safely avoid the obstacles around it while moving.

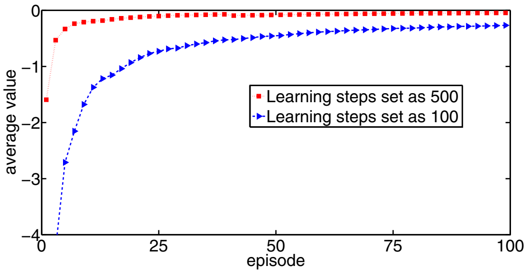

Fig. 7: Performances of the fuzzy Q-learning algorithm with learning steps set as 100 and 500, respectively.

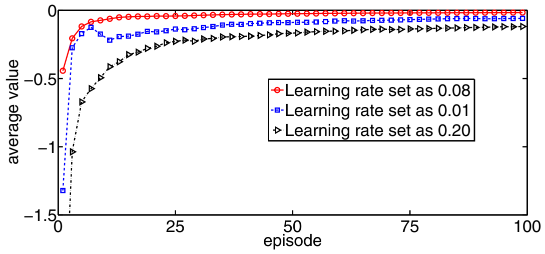

Fig. 8: Performances of the fuzzy Q-learning algorithm with different learning rates.

V. CONCLUSION

In this paper, we combine fuzzy control with reinforcement learning and present the fuzzy Q-Learning algorithm to solve obstacle avoidance for hexapod robots. In addition, the Fuzzy Q-learning algorithm of obstacle avoidance can

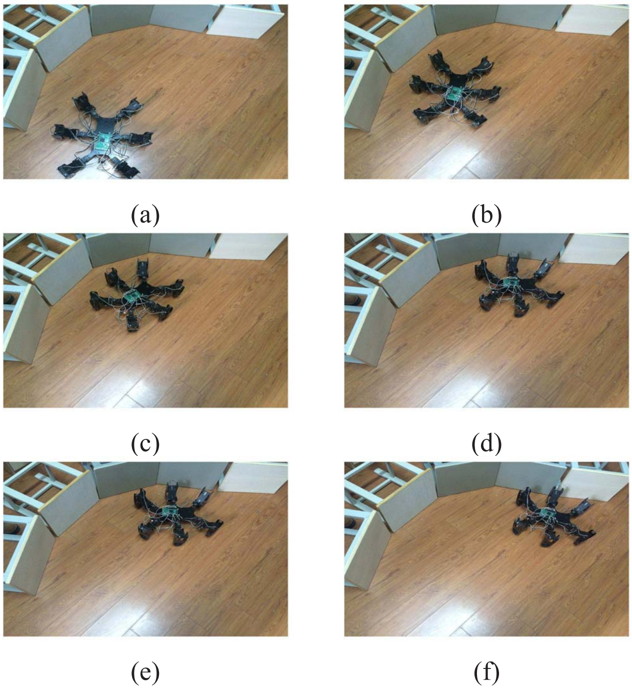
(a) (b)
(c) (d)
(e) (f)

Fig. 9: The learned strategy tested on the CR-6 hexapod robot platform.

achieve good performance and help the hexapod robots avoid obstacles safely and autonomously. However, there remain some problems to be further investigated in our future work. For example, in this paper, we only consider how to avoid obstacles instead of an integrated navigation approach [21], [22] for the hexapod robots. In addition, we will equip the hexapod robot with more sensors to better adapt to unknown complex environments and focus on the learning control [23], [24] approaches accordingly.

Acknowledgment

This work was supported by the National Key Research and Development Program of China (No.2016YFD0702100) and by the National Natural Science Foundation of China (No.61432008).

References

[1] T. Homberger, M. Bjelonic, N. Kottege, et al. "Terrain-dependant Control of Hexapod Robots using Vision." *International Symposium on Experimental Robotics*, 2016.

[2] D. Belter. "Integrated Motion Planning for a Hexapod Robot Walking on Rough Terrain." *World Congress*, pp. 6918-6923, 2016.

[3] A. Manglikl, K. Gupta, S. Bhanoe. "Adaptive Gait Generation for Hexapod Robot using Genetic Algorithm." *IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES-2016)*, 2016.

[4] M T. Ibrahim, D. Hanafi, R. Ghoni. "Autonomous Navigation for a Dynamical Hexapod Robot Using Fuzzy Logic Controller." *Procedia Engineering*, pp. 330C341, 2012.

[5] Y. Chou, K. Huang, W. Yu, et al. "Model-Based Development of Leaping in a Hexapod Robot." *IEEE Transactions on Robotics*, pp. 40-54, 2015.

[6] A. Degani, A W. Long, S. Feng, et al. "Design and Open-Loop Control of the ParkourBot, a Dynamic Climbing Robot." *IEEE Transactions on Robotics*, pp. 705-718, 2014.

[7] P. Arena, L. Fortuna, M. Frasca, et al. "An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion." *IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man and Cybernetics Society*, pp. 1823, 2004.

[8] G S. Dordevic, M. Rasic, R. Shadmehr. "Parametric models for motion planning and control in biomimetic robotics." *IEEE Transactions on Robotics*, pp. 80-92, 2005.

[9] P C. Lin, H. Komsuoglu, D E. Kodistchek. "A leg configuration sensory system for dynamical body state estimates in a hexapod robot." *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA IEEE*, vol. 1, pp. 1391-1396, 2003.

[10] J H. Barron-Zambrano, C. Torres-Huitzil, B. Girau. "Perception-driven adaptive CPG-based locomotion for hexapod robots." *Neurocomputing*, pp. 63-78, 2015.

[11] C F. Juang, Y. H. Chen, and Y. H. Jhan. "Wall-Following Control of a Hexapod Robot Using a Data-Driven Fuzzy Controller Learned Through Differential Evolution." *Industrial Electronics IEEE Transactions on*, pp. 611-619, 2015.

[12] Y. Cheng, W. Zhang. "Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels." *Neurocomputing*, 2017.

[13] M. Duguleana, G. Mogan. "Neural networks based reinforcement learning for mobile robots obstacle avoidance." *Expert Systems with Applications*, pp. 104-115, 2016.

[14] I. Gutierrez, and J. Collado. "An LQR controller in the obstacle avoidance of a two-wires hammerhead crane." *Neurocomputing*, 2016.

[15] A T. Nguyen, R. Mrquez, and A. Dequidt. "An augmented system approach for LMI-based control design of constrained Takagi-Sugeno fuzzy systems." *Engineering Applications of Artificial Intelligence*, pp. 96-102, 2017.

[16] M. Akole, B. Tyagi. "Design of Fuzzy Logic Controller for Nonlinear Model of Inverted Pendulum-Cart System." 2008.

[17] B. Pourpeighambar, M. Dehghan, M. Sabaei. "Non-cooperative reinforcement learning based routing in cognitive radio networks." *Computer Communications*, pp. 11-23, 2017.

[18] F. Shoeleh, M. Asadpour. "Graph based skill acquisition and transfer learning for continuous reinforcement learning domains." *Pattern Recognition Letters*,pp. 104-116, 2016.

[19] N. Sahebjamnia, R. Tavakkoli-Moghaddam, N. Ghorbani. "Designing a fuzzy Q-learning multi-agent quality control system for a continuous chemical production line." *Computers and Industrial Engineering*, pp. 215-226, 2016.

[20] E. Nummelin, P. Tuominen. "Geometric ergodicity of Harris recurrent Marcov chains with applications to renewal theory." *Stochastic Processes and Their Applications*, pp. 187-202, 2011.

[21] M. Benzaoui, H. Chekireb, M. Tadjine, et al. "Trajectory tracking with obstacle avoidance of redundant manipulator based on fuzzy inference systems." *Neurocomputing*, pp. 23-30, 2016.

[22] B V. łcaro, I A A. Prado, D A D. Santos, et al. "Trajectory Tracking Control of an Aerial Robot with Obstacle Avoidance." *IFAC-PapersOnLine*, pp. 81-86, 2015.

[23] D. Dong, C. Chen, Z. Chen. "Quantum reinforcement learning." *Natural Computation-ICNC2005, Lecture Notes in Computer Science*, 3611: 686-689, 2005.

[24] C. Chen, D. Dong, H.X. Li, J. Chu, T.J. Tarn. "Fidelity-based Probabilistic Q-learning for Control of Quantum Systems." *IEEE Transactions on Neural Networks and Learning Systems*, 25(5): 920-933, 2014.