International Journal of Hybrid Information Technology
Vol.8, No.11 (2015), pp.297-306
http://dx.doi.org/10.14257/ijhit.2015.8.11.25

Planning Flying Robot Navigation in a Three-dimensional Space by Optimaztion Combining Q-learning and Monte Carlo Algorithms

Sima Vosoghi Asl[1], Zohreh Davarzani[2] and Soheila Staji[3]

[1, 3]Technical and vocational university of Sabzevar academy, Iran
[2]Department of computer engineering, Payame Noor University, 19395-4697, Tehran, Iran
Sima.vosoghi@yahoo.com, Davarzani.z@pnurazavi.ac.ir, soheila_staji@yahoo.com

Abstract

This article examines navigation of a flying robot inside a building environment in three dimensional spaces in which the size and location of some obstacles are not determined and other obstacles and target can be moving. This article suggests a new method by combining Q-learning algorithm and Monte Carlo algorithm on optimal navigation by the flying robot. The rewards are intended to be maximized when the robot flies in the right route; moreover, the maximum performance power would be measured according to the future predictions and the well-doing of that action would be also measured. Here, this method has been implemented with Webots simulator, and simulated data are analyzed by MATLAB. The simulation results show that control of the policy obtained from Q-learning and Monte Carlo methods is more efficient compared to traditional methods in controlling flying robot navigation.

Keywords: *Q-learning, navigation, dynamic environment, Monte Carlo, obstacles, flying robot*

1. Introduction

Robot navigation is an essential issue in the discussion on robot guidance. Therefore, there has been significant attention to this issue in the recent years. Robot navigation is that the robot passes a route with no collision with the obstacle and reaches the target in the shortest route and shortest time. Several methods have tried to solve this problem including Artificial potential field method, genetics algorithm, Particle Swarm Optimization, visibility graph method, fuzzy logic and Neural network. Route planning for robots is categorized based on different situations. Route planning for robots is divided into two groups in terms of environment: Route planning for robots in static environment, where the obstacles in the map are static, and route planning for robots in dynamic environment which includes both static and dynamic (moving) obstacles. This article performs the robot navigation inside a building in dynamic environment and obstacles.

2. Literature Review

Mobile robot navigation was suggested first in 1991 by Latombe as route planning by optimal searching for a free route from the start point to the end point which has the desired consent as the shortest route and minimum of time and energy [21].

Optimal or approximately optimal ways was proposed by Griffin and Alexopoulos in 1992 using visibility graph method [2]. The deficiency of this method lies in its low algorithm efficiency. By creating a complete visibility graph, it can be concluded that this

ISSN: 1738-9968 IJHIT

visibility graph needs time period equal to (N_3) O, which is too long. Furthermore, the route obtained from graph method is often very close to obstacles and may lead to the robot destruction. By increasing obstacles, the probability of collision increases. Artificial potential field method for route planning algorithms was introduced in 1992 by Barragard, *et al.,* [5]. Based on receiving sensor information from environment. Although it is easy to implement, and it acts powerful in producing movements, the main problem is that the robot may be trapped in a local trap before reaching the target. The techniques to escape local traps should be examined. Recently, artificial intelligence methods in environment are used for sensor-based route planning. Applying fuzzy logic on route planning methods can perform the movement rules in order to prevent the obstacles without using a precise model of environment, which is studied in 2005 by Yang et al. and in 2009 by Hachour [32, 15]. However, it is difficult to develop fuzzy rules for complex environments and their generalization power is limited.

The neural network for methods based on route planning to control robot movements in order to select the best route and prevent collision with obstacles, was introduced by ALTaharva, *et al.,* in 2008, Youn, *et al.,* in 2010 and Tsai, *et al.,* in 2011. Learning how to work with neural network is too time intensive [3, 34, 28].

Also, Catillo, *et al.,* introduced a multi-purpose genetic algorithm as offline and point to point for mobile independent robot in 2007 [8]. Particle Swarm Optimization was introduce in 2009 by Garcia, *et al.,* and Ant Colony Optimization was used by Hwang et al. in 2011 for route planning. The performance of these methods is highly dependent on parameters [11, 17]. This limitation leads to unfavorable performance.

Reinforcement learning learns through interaction with environment, and is suitable for mobile dependant robots. Q learning algorithm can be used by policy control. It works by the maximum reward receiving from environment using the test and error process and follows the route, which is introduced by Dayan and Watkins in 1992 [30]. Q learning algorithm is used widely for its simplicity. However, the integration trend for traditional Q learning algorithm is old fashion. Developed Q learning algorithm deals with the responsibility of delayed reward, developed Q learning algorithm learning is the multi-stage incremental of Q learning algorithm.

For local optimization, greedy action selection strategy can be used only based on the current value. Greedy action selection strategy (ϵ <0.1) was introduced by Sutton in 1998 [27]. The higher the ϵ, the probability to select a non-optimal action in current situation is increased. There should be a balance between discovery and exploitation. Annealing simulation to make balance between discovery and exploitation was introduced in 2004 by Guo, *et al.,* [14]. in 2007, in order to guide the discovery and accelerate the reinforced learning, Framling introduced a method. Q learning by changed the initial value which was introduced in 1996 by Simmons and Koening accelerated Q method, but for complex environments where the number of modes is too high, the time and space of this storage is difficult [19]. Lampton and Valasek in 2009 could decrease some of the modes through comparison method. Senda, *et al.,* [20, 25]. Limited some of the modes by new definition of the mode space. Although the above methods accelerated the learning, still many ways to increase integration speed remains.

Flying robots inside the building do maneuvers for many applications in real world for their high power in GIS. Small flying robots can be used in order to seek injured people inside a building. They are also used in stores and factories in order to monitor the chemical and radioactive substances which are dangerous for human to contact. Swarms used the flying robot inside building for efficient performance, which is a powerful method for the redundancy of the flying robots. In Swarms method, the robots do their work in parallel with each other. The space for coordination among robots and works is very important in order to let robots do team works and pass their desired routes to do their works. These parallel algorithms are recently developed, and assume that the relative or absolute information of robot locations is available for all robots. Such an algorithm

prevents robots to collide each other which were introduced by Hoffmann and Tomlin in 2008 [16]. The method to use GPS which was introduced in 2008 by Rudol and is used for inside of buildings is weak, and the location and position they give may be unreliable [24].

In order to solve problems, the GIS should be done for routes and obstacles. Robot flight inside buildings such as official buildings is too challenging, for the presence of obstacles including walls, furniture, people, *etc.*, in the environment. So measuring the proximity to these obstacles is necessary.

The narrow corridors and doors inside buildings make some limitations. Sounararaj, *et al.*, in 2009 and Grzonka, *et al.*, in 2009 examined this limitation. Also Valenity, *et al.*, examined the endurance in short flights in 2007 [26, 12, 29]. Based on problems a flying robot has, Lupashin, *et al.*, in 2010 and Kirchner and Furukawa in 2005 introduced sensors for route navigation to reach the target and spatial coordination in flight guidance. They obtained good results, but the results are not suitable for applications in unknown real environments [22, 18].

Some other researchers used laser scanner. In 2009, Achtelik, *et al.*, and in 2008, Guenard, *et al.*, extracted movement data using specific navigation algorithms [1, 13]. And Bachrach in ٢٠٠٩and Blosch in 2010 introduced Simultaneous imaging and mapping of location [4, 6]. This method is time intensive, since it should fine the location and information of each robot for every moment and has heavy and complicated calculations, so it requires a very fast processor. Moreover, like all other methods mentioned, this method is for two-dimensional environments and is not suitable for indoor environments.

3. Kinematics and Robot Model

In order to evaluate Q-learning and Monte Carlo algorithms, a small sized Blimp was selected for implementation of self-control navigation system. This balloon has 1.4 meters length and 0.75 meters diameter. The gondola of the Blimp is put under the main body of its coverage. In both sides of gondola, two drive propellers are installed as the main driving force. These two drive propellers are run with 2 Dc motors which are appropriate for inside test flights. The server associated with main DC propeller can produce the driving force around the horizontal axis. The main structure of this Blimp is depicted in Figure 1.

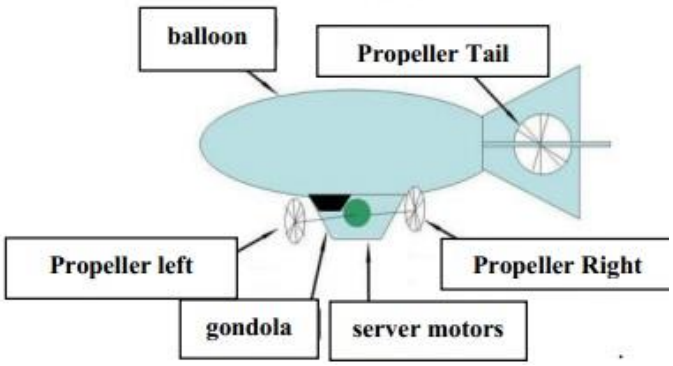

Figure 1. Small Blimp and c and Server Motors Installed on the Gondola of Blimp [33]

All information about the Blimp includes the position of air balloon, its direction, *etc.* It is given in the environment coordination reference system, while the information on the mode of Blimp including dimensional speed and angular acceleration is measured through processor coordination reference system, which is called body coordinates. In every step of navigation, the direction of Blimp must be toward the target.

Figure 2 shows that θ is the angle between the current position of Blimp and the direction of x axis in the context of environment, and α is the angle between target position and the direction of x axis is the coordinator.

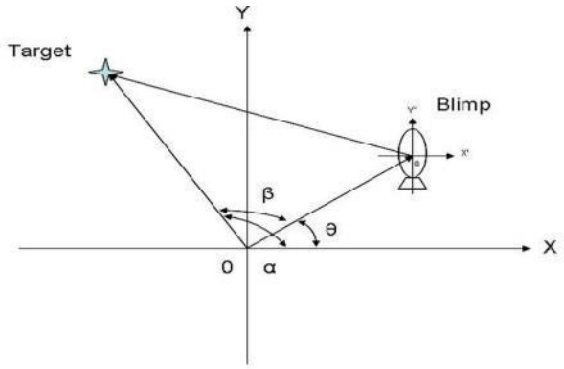

Figure 2. The Coordinates for Body of Blimp [33]

As the angular difference between the current position of Blimp and target position β=α-θ is shown, it is the angle which is required by the Blimp in the context of environment to be reduced. This can be transformed from β by the transformation function coordinating between the environment and body.

4. Hybrid Algorithm of Q-learning and Monte Carlo

4.1. Q-learning [31]

A method for reinforced learning is Q learning method. In this method, the possible modes are determined in a distinct time and the possible actions for the robot are determined. Then, we consider a reward or punishment for every action of the robot, and based on formula 1, every pair (mode, action) will have a Q value. In the learning stage, robot fills the table of Qs, and uses this table in practice stage, so that in passing from every mode to the other, selects the action which has the highest value of Q.

$$Q(\text{state, action}) = R(\text{state, action}) + \Upsilon \max[Q(\text{next state, all action})] \quad (1)$$

In this formula, R is the award for every mode for a given action in the current time. Υ is discount factor equal to 0.7, and is multiplied by the maximum Q the robot can obtain. The description of algorithm is as follows:

1. For every a, q consider the values of input table as.
2. Observe the current position of s.
3. Do it until infinity.
    a. Select an action and do it.
    b. Receive the r reward.
    c. Observe the new position of s.
4. Update the input table for (s,a).

4.2. Mont Carlo Method [9]

The learning is about the running policy. It is tried to evaluate the applied policy for decision making and improve it. This algorithm works based on experiment in the environment, and performs according to taking the average from estimated recursive form State Action Value. According to the relatively large environment, a large number of episodes are required to find the optimal route. For the better integration of algorithm, the start modes are selected randomly, and these start modes have been effective in calculating Action Values.

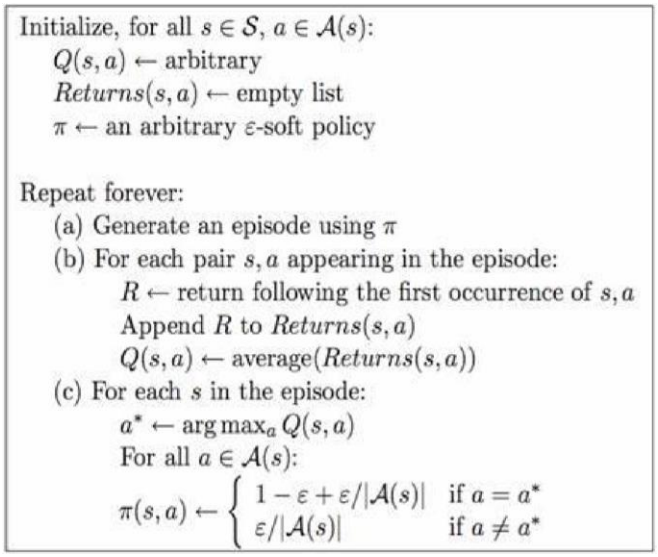

Figure 3. Algorithm Monte Carlo [9]

5. Hybrid Algorithm

In this method, which is in fact the combination of two previous methods, the speed and accuracy of the agent (flying robot) is increased. In this method, first, the feedbacks of mode and action and table of Q values are empty, and system works as greedy. First, a mode is selected in random, and select an action can be done in that mode. Then the related reward would be received and it is going to the next mode. In this moment the Q value table should be filled. In fact, the formula 2, which is a combination of Q learning and Mont Carlo is calculated. In the next stage, according to the more selected values and in the forward step, the action selection policy is updated. It should be considered that updating this policy is because an action with higher value cannot always lead to the optimal route in the next steps.

This method, which is in fact the combination of Q learning and Mont Carlo, has the capabilities of both of them. It evaluates the applied policy in every step and improves it, and Q table values are filled according to the desired algorithm for every step. This method examines the most difficult type of navigation which is the navigation in unknown dynamic indoor environment, and the performance is compared to other methods. The combined algorithm of Q learning and Mont Carlo is as the following:

1. Consider the (mode, action) feedbacks as an empty list.
2. Consider π as an arbitrary? For the procedure.
3. For every a, q consider the values of input table as 0.
4. Observe the current position of s.
5. Do it until infinity.
    i. Select an action and do it.
    ii. Receive the r reward.
    iii. Observe the new position of s
6. Update the input table for (s,a)

$$Q(state, action) = Avrage(Return(R(state, action)) + \Upsilon \\ Max[Q(next\ state, all\ action)]) \quad (2)$$

7. For every a € A(s), put the value of argmaxa Q (s,a) in a*.

$$\pi(s,a) = \begin{cases} 1-\varepsilon+\varepsilon/|A(s)| & , \quad a = a^* \\ \varepsilon/|A(s)| & , \quad a \neq a^* \end{cases}$$

The variable under examination in this method is in fact the value of Q, the value of every action for every mode, and under the function of algorithm, the maximum Q in every step is evaluated as the following:

$a^* \leftarrow \text{argmax}_a Q(s,a)$ (4)

6. Results of Combined Learning Simulation in Webots

6.1. Implementation of Webots Simulation

Webots provides two windows to show the simulation results. The first one is artificial world window, through which the movements of automatic Blimp flight can be seen easily. This can help researchers for judgment about automatic Blimp in the artificial world, as shown in Figure 4.

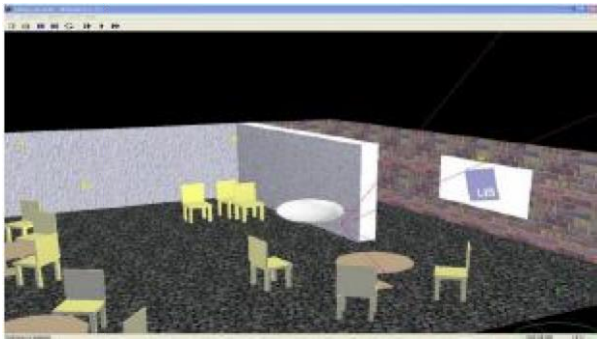

Figure 4. Environment Simulator Webots

The second output window provides specific information on the Blimp flight during the simulation. The information displayed by Webots is updated in every stage of simulation and is saved as a text file in Webots. The information display window is shown in Figure 5.

Figure *5*. Information Display Window

Information display window provides the modes of test flight of Blimp during simulation, including three dimensional positions of Blimp body, the angles of rotation, direction, comparison of Blimp propeller power, *etc*. As shown in figure ٤, the flight

information parameters section is referred to the global coordinates in the artificial environment. The simulation environment provides different artificial sensors on the balloon's body to gather information on flight. Using this capability, the movements of automatic Blimp can be simulated in artificial environments. The model of used Blimp for simulation is provided by Webots simulator package, including a driving force system with three separate propellers of artificial plain, which are designed for moving the Blimp in three directions of x, y, z. The structure of driving force is almost the same for all designed air balloons for competition in UAV remote areas in 2007. Figure 6 shows the definition of global coordinates in artificial world, in which axes x, y, z are the total gathered information by navigation system in this coordinate system.

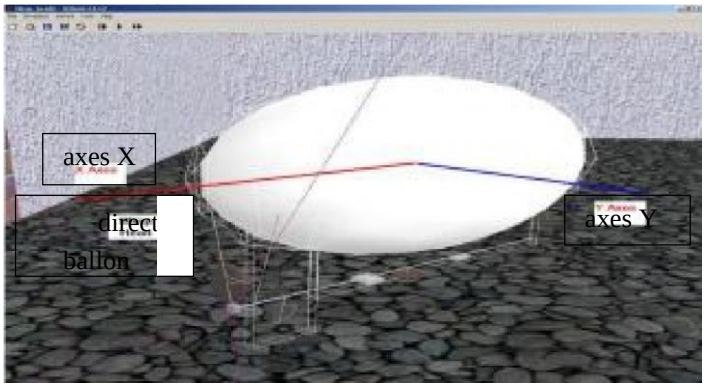

Figure 6. Coordinates of Blimp Body in Artificial Environment

6.2 · Analysis of Hybrid Learning Simulation Results in MATLAB

In this section, the statistical information obtained from the Webots simulator is analyzed in order to investigate the performance of hybrid learning in navigation controls. All flight information including the angular difference, Blimp orientation, the sequence of actions, the sequence of modes and the values of modes (Exploration) were entered into the MATLAB from the Webots data files. Table 1 explains the details of selected variables evaluation. In this table, the variables used in the simulation are presented with their definitions.

Table 1. Evaluated Variables in Matlab

Variables in evaluation	Definitions for analysis
angular difference	*angular difference between the target*
Orientation	*Blimp orientation in flight (With reference to the world frame)*
Q-value	*Q-value represents the mode pairsaction after a certain iteration*
sequence of actions	*Operation changes after a considerable amount of repetitions*
sequence of modes	*Mode changes after a considerable amount of repetitions*

In this section, a particular simulation of Blimp flight tests is introduced in order to show the preliminary results made by MATLAB. In this simulation, the Blimp flights were simulated by 2400 repetitions of Q-learning process through which the Blimp control learned how to turn toward the target position. A large number of Q-value was updated during the process of learning. Figure ۷ shows that the whole mode space was visited by the Q-learning algorithm.

Blimp movement pathways can be visualized in MATLAB by entering the Webots simulation results. Figure 8 shows the movement pathway of the Blimp in a navigation test. Circulating actions can be easily seen in this Figure.

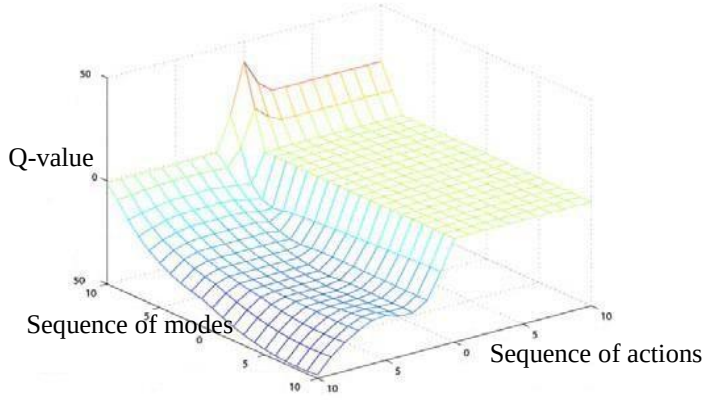

Figure 7. 3D Processed Results of the Hybrid Algorithm in MATLAB

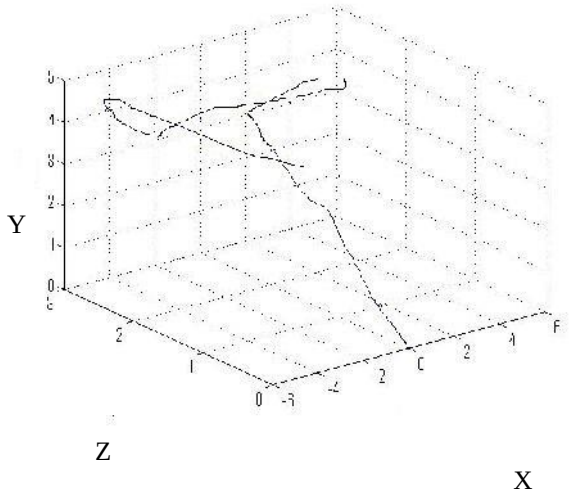

Figure 8. 3D Movement Pathway of Blimp by the Hybrid Algorithm in MATLAB

7. Conclusion

In this paper, the most difficult type of navigation namely the navigation in dynamic environments with moving obstacles were studied and the problems of navigation and obstacle avoidance were solved using the Q-learning and Monte Carlo algorithms. One of the advantages of this method is the high success in majority of cases. In this method, the environment is allowed to changes and navigation is still remained.

References

[1] M. Achtelik, A. Bachrach, S. Prentice and N. Roy, "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments", In Proceedings of unamanned systems technology XI, **(2009)**.

[2] C. Alexopoulos and P. M. Griffin, "Path planning for a mobile robot", IEEE Transactions on Systems, Man and Cybernetics, vol. 22, no. 2, **(1992)**, pp. 318-322.

[3] I. AL-Taharwa, A. Sheta and M. Al-Weshah, "A mobile robot path planning using genetic algorithm in static environment", Journal of Computer Science, vol. 4, no. 4, **(2008)**, pp. 341-344.
[4] A. Bachrach, R. He and N. Roy, "Autonomous flight in un-structured and unknown indoor environments", In Proceedings of the 2009 European micro air vehicle conference and flight competition (EMAV'・9), **(2009)**.
[5] J. Barraquand, B. Langlois and J.-C. Latombe, "Numer-ical potential field techniques for robot path planning", IEEE Transactions on Systems, Man and Cybernetics, **(1997)**.
[6] M. Blösch, S. Weiss, D. Scaramuzza and R. Siegwart, "Vision based mav navigation in unknown and unstructured environments", In 2010 IEEE international conference on robotics and automation (ICRA), **(2010)**, pp. 21–28.
[7] Q. Cao, "An evolutionary artificial potential field algorithm for dynamic path planning of mobile robot", Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, **(2006)**, pp. 3331-3336.
[8] O. Castillo, L. Trujillo and P. Melin, "Multiple objective genetic algorithms for pathplanning optimization in autonomous mobile robots", Soft Computing, vol. 11, no. 3, **(2007)**, pp. 269-279.
[9] R. S. Sutton and A. G. Barto, "Reinforcement Learning", An Introduction MIT Press, **(1998)**.
[10] K. Framling, "Guiding exploration by pre-existing knowledge without modifying reward", Neural Networks, vol. 20, no. 6, **(2007)**, pp. 736-747.
[11] M. A. Garcia, O. Montiel, O. Castillo, R. Sepulveda and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation", Applied Soft Computing, vol. 9, no. 3, **(2009)**, pp. 1102-1110.
[12] S. Grzonka, G. Grisetti and W. Burgard, "towards a navigation system for autonomous indoor flying", In Proceedings of the inter-national conference on robotics and automation (ICRA'・9), Piscataway: IEEE Press, **(2009)**, pp. 2878–2883.
[13] N. Guenard, T. Hamel and R. A. Mahony, "practical visual servo control for a unmanned aerial vehicle", IEEE Transactions on Robotics and Automation, vol. 24, no. 2, **(2008)**, pp. 331–341.
[14] M. Guo, Y. Liu and J. Malec, "A new Q-learning algorithm based on the metropolis criterion", IEEE, **(2004)**.
[15] Hachour, "The proposed fuzzy logic navigation approach of autonomous mobile robots in unknown environments", International Journal of Mathematical Models and Methods in Applied Sciences, vol. 3, no. 3, **(2009)**, pp. 204-218.
[16] G. M. Hoffmann and C. J. Tomlin, "Decentralized cooperative collision avoidance for acceleration constrained vehicles", In Pro-ceedings of the 47th IEEE conference on decision and control, Cancun, Mexico, **(2008)**.
[17] H. J. Hwang, H. H. Viet and T. Q. Chung, "based vec-tor direction for path planning problem of autonomous mobile robots, Lecture Notes in Electrical Engineering: IT Convergence and Services", Springer Netherlands, **(2011)**.
[18] N. Kirchner and T. Furukawa, "Abstract infrared localisation for indoor uavs", In Proceedings of the international conference on sensing technology, **(2005)**, pp. 60–65.
[19] S. Koenig and R. G. Simmons, "The effect of represen-tation and knowledge on goal-directed exploration with reinforcement-learning algorithms", Machine Learning, vol. 22, no. 1, **(1986)**, pp. 227-250.
[20] A. Lampton and J. Valasek, "Multiresolution state-space discretization method for Qlearning", Proceedings of American Control Conference, **(2009)**, pp. 1646-1651.
[21] J. C. Latombe, "Robot motion planning", Kluwer Academic Publishers, **(1991)**.
[22] S. Lupashin, A. Schöllig, M. Sherback and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips", In Proceedings of the international conference on robotics and automation (ICRA'10), Piscataway: IEEE Press, **(2010)**, pp. 642-648.
[23] A. Poty, "Dynamic path planning for mobile robots using fractional potential field", Proceedings of the First International Symposium on Control, Communications and Signal Processing, **(2004)**.
[24] P. Rudol, M. Wzorek, G. Conte and P. Doherty, "Micro un-manned aerial vehicle visual servoing for cooperative indoor ex-ploration", In Proceedings of the aerospace conference, Piscataway: IEEE Press, **(2008)**, pp. 1-10.
[25] K. Senda, S. Mano and S. Fujii, "A reinforcement learning accelerated by state space reduction", SICE 2003 Annual Conference, vol. 2, **(2009)**, pp. 1992-1997.
[26] S. P. Soundararaj, A. K. Sujeeth and A. Saxena, "Autonomous indoor helicopter flight using a single onboard camera", In Proceedings of the 2009 IEEE/RSJ international conference on intelligent robots and systems, **(2009)**.
[27] R. S. Sutton and A. G. Barto, "Reinforcement learning:an introduction", Cambriage, MA: MIT Press.Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 34, no. 5, **(1998)**, pp. 2140-2143.
[28] C. C. Tsai, H. C. Huang and C. K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation", IEEE Transactions on Industrial Electronics, vol. 58, no. 10, **(2011)**, pp. 4813-4821.

[29] M. Valenti, B. Bethke, J.-PHow, D.-P Farias and J. Vian, "Embedding health management into mission tasking for UAV teams", In American control conference, Piscat-away: IEEE Press, **(2007)**, pp. 5777-5783.
[30] C. J. C. H. Watkins and P. Dayan, "Q-earning", Machine Learning, vol. 8, no. 3-4, **(1992)**, pp. 279-292.
[31] Y. Zhang, Ming Li and Z. Zhang, "Reinforcement Learning in Robot Path Optimization", Journal Of Software, **(1992)**.
[32] X. Yang, M. Moallem and R. V. Patel, "A layered goal-oriented fuzzy motion planning strategy for mobile robot navigation", IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics, vol. 35, no. 6, **(2005)**, pp. 1214-1224.
[33] Y. Liu, "Autonomous Blimp control with reinforcement learning", Australasian Conference on Robotics and Automation (ACRA), **(2009)** December 2-4, Sydney, Australia.
[34] S. C Yun, V. Ganapathy and L. O Chong, "Improved genetic algorithms based optimum path planning for mobile robot", Proceedings of the 2111th International Conference on Control, Automation, Robotics and Vision, **(2010)**, pp.1565-1570.

Authors

Sima Vosoghi Asl, Technical and vocational university of Sabzevar academy, Iran
sima.vosoghi@yahoo.com

Zohreh Davarzani

Department of computer engineering, Payame Noor University
Davarzani.z@pnurazavi.ac.ir

Soheila Staji

Technical and vocational university of Sabzevar academy, Iran
soheila_staji@yahoo.com

