INTECH OPEN ACCESS PUBLISHER

Mobile robot Navigation Based on Q-Learning Technique

Lazhar Khriji, Farid Touati, Kamel Benhmed and Amur Al-Yahmedi

Electrical and Computer Engineering Department, Sultan Qaboos University

Abstract This paper shows how Q-learning approach can be used in a successful way to deal with the problem of mobile robot navigation. In real situations where a large number of obstacles are involved, normal Q-learning approach would encounter two major problems due to excessively large state space. First, learning the Q-values in tabular form may be infeasible because of the excessive amount of memory needed to store the table. Second, rewards in the state space may be so sparse that with random exploration they will only be discovered extremely slowly. In this paper, we propose a navigation approach for mobile robot, in which the prior knowledge is used within Q-learning. We address the issue of individual behavior design using fuzzy logic. The strategy of behaviors based navigation reduces the complexity of the navigation problem by dividing them in small actions easier for design and implementation. The Q-Learning algorithm is applied to coordinate between these behaviors, which make a great reduction in learning convergence times. Simulation and experimental results confirm the convergence to the desired results in terms of saved time and computational resources.

Keywords Mobile robot, behaviors based navigation, fuzzy logic and reinforcement learning.

1. Introduction

The existence of robots in various types (walkers, manipulators, mobiles…) became very significant in the industrial sector and especially in the service sector (Youcef, Z., 2004). Due to the growing interest of the service robots, they can achieve their mission in an environment which contains several obstacles (Ulrich, I. & Borensstein, J., 2001) (i.e. in factories, hospitals, museums and even in our houses). The mobile robots, known also as wheeled robots, have the advantage of the simplicity of manufacturing and mobility in complex environments. The capacity to move without collision in such environment is one of the fundamental questions to be solved in autonomous robot-like problems. The robot should avoid the undesirable and potentially dangerous objects. These possibilities have much interest of the subject of robot-like research.

A Behavior Based Navigation (BBN) system is developed using fuzzy logic. The main idea is to decompose the task of navigation into simple tasks (Parasuraman, S.; Ganpathy, V. & Shiainzadeh, B., 2005). With this type of navigation system we should identify a behavior for each situation defined by the sensory inputs of the robot (Fatmi, A.; Al-Yahmedi, A.; Khriji, L. & Masmoudi, N., 2006). All actions (behaviors) are mixed or fused to produce only one complex behavior. However, a number of problems with regard to this type of navigation system are always in study.

In this paper, we study the use of the reinforcement learning algorithm, Q-learning, to answer the question of coordination between the behaviors. The motivation of this study is as follows: Q-learning is a simulation- based stochastic technique which provides a way to relate state, action and reward through Q-value in the look up table. In real application, a controller simply searches the look up table and chooses the best-valued decision, no need to perform a complex on-line computation. Hence, the real time requirement of decision can be met in this way.

However, the dynamic coordination between behaviors is a large space Markov decision process (MDP) when the practical application involves a large number of behaviors; this is due to the exponential increase in the number of admissible states with the number of behaviors. In this situation, Q-learning will encounter two main problems. First, learning the Q-value in tabular form may be infeasible because of the excessive amount of memory needed to store the look up table. Second, because the Q-value only converges after each state has been visited multiple times, random exploration policy of Q-learning will result in excessive slowness in convergence. In this study, we solve the above large space Q-learning problems by taking advantage of fuzzy logic techniques.

The remainder of this paper is organized as follows: related researches on Fuzzy Behavior Based Navigation are reviewed in Section 2. Section 3 gives the basic principle of Reinforcement Learning (RL) (i.e. Q-learning). Simulation and experimental results are given in Section 4. Section 5 concludes the paper.

2. Fuzzy Behavior Based Navigation

Fuzzy logic is very much used for the control in the robotics field (Das, T. & Kar, I., 2006; Antonelli, G.C. & Fusco, S.G., 2007; Dongbing, G. & Huosheng, H.,2007). The basic idea in the fuzzy logic is to imitate the capacity of reasoning and decision making from the human been, using uncertainties and/or unclear information. In fuzzy logic the modeling of a system is ensured by linguistic rules (or fuzzy rules) between the input and output variables (Fatmi, A.; Al-Yahmedi, A.; Khriji, L. & Masmoudi, N., 2006; Aguirre, E. & Gonzalez, A., 2000). A fuzzy rule can be described by:

> R1: If *goal* is far then *speed* is large and *rotation* is zero.
> R2: If *goal* is right then *speed* is medium and *rotation* is right.
> …
> Rn: If *goal* is near then *speed* is zero and *rotation* is zero.

All input and output variables (i.e. *goal*, *speed*, *and rotation*) have degree of memberships in their membership functions (ex: *goal*: far, right… *speed*: large, medium, zero… and *rotation*: right, zero…). For the behavior based navigation, the problem is decomposed into independent and simpler behaviors (go to goal, avoid obstacles…). Each behavior is designed using fuzzy logic.

Let us mention for example the behavior *go to goal*; the objective of this behavior is to guide the robot to reach a desired target point. For testing purposes we considered an environment without obstacles. To reach the goal, the robot turns right then left with a high, medium or null speed depending on the inputs which are;

- *Distance*: [meter] is the distance between the robot and the goal.
- *Θ-error*: [degree] is the difference between the desired angle and the current angle of the robot.

Figs. 1-2 show the fuzzy membership functions of the input distances and the input *θ-error*, respectively.

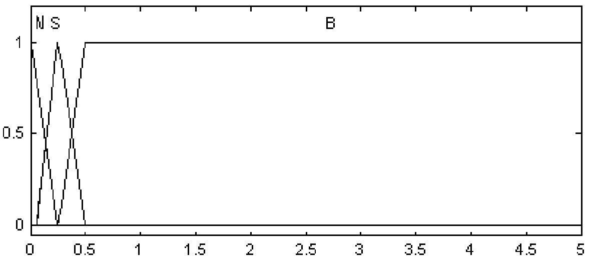


Figure 1. Membership functions of the input distances

(N: Near, S: Small, B: Big)

The two outputs of this behavior, which are also identical for the other behaviors, are:

- *Steering*: [degree] is the desired angle for the next step.
- *Velocity*: [meter/sec] is the speed of the robot.

Figs. 3-4 show the fuzzy membership functions of the output steering and the output velocity, respectively.

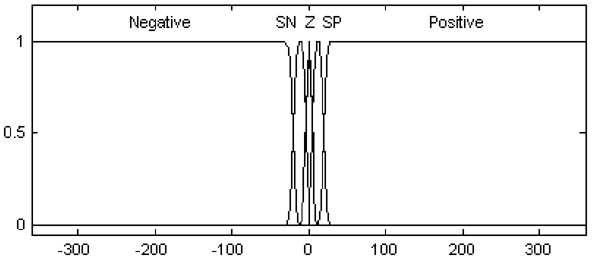


Figure 2. Membership functions of the input *θ-error*.

(SN: Negative Small, Z: zero, SP: Positive Small)

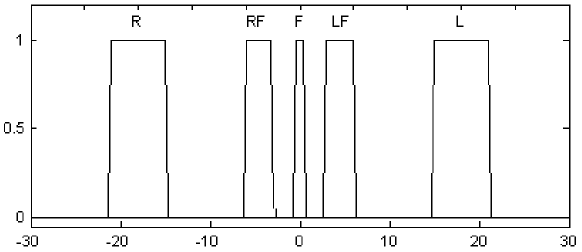


Figure 3. Membership function of the output steering

(R: Right, FR: Right Forward, F: Forward, FL: Left Forward, L: Left)

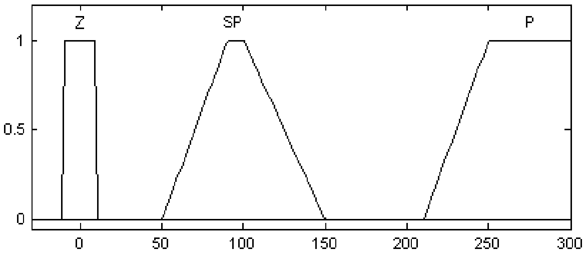


Figure 4. Membership function of the output velocity

(Z: zero, SP: Positive Small, P: Positive)

Inputs			**Outputs**	
Distance	*Θ-error*		Steering	Velocity
Small	Z		F	P
	SN		RF	SP
	N		R	SP
	SP		FL	SP
	P		L	SP
Big	Z		F	P
	SN		RF	SP
	N		R	SP
	SP		FL	SP
	P		L	SP
Near	Z		F	Z
	SN		RF	Z
	N		R	Z
	SP		FL	Z
	P		L	Z

Table 1. Inference rules of the behavior go to goal

The *goal reaching* behavior is expected to align the robot with the direction of the goal. To achieve this behavior the rules, shown in Table I, were devised. By choosing different start points, Fig. 4 shows that in simulation the robot reaches the desired goal based on the minimization of the way to be taken.

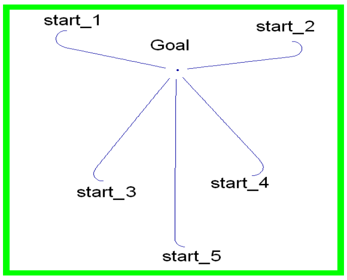


Figure 4. Simulation results of the behavior *goal reaching*.

The other behaviors are:

Obstacle avoidance: It tends to steer the robot in such a way as to avoid collision with obstacles that happens to be in the vicinity of the robot. This behavior is the most realistic in the mobile robot navigation; several works are interested to conceive it by different techniques: for example the potential field technique (Borenstein, J. & Koren, Y., 1991) hybrid learning approach (Joo, M. & Deng, C., 2005).

Wall Following: The motivation to conceive such behavior explains when the robot uses only the *obstacle avoidance* behavior; it will also consider the wall as obstacle. Therefore, the objective of this behavior is to keep the robot at a safe close distance to the wall and to keep it in line with it. The simulation results of Fig. 5 show the effect of adding the *Wall Following* behavior to the design. It confirms clearly our objective.

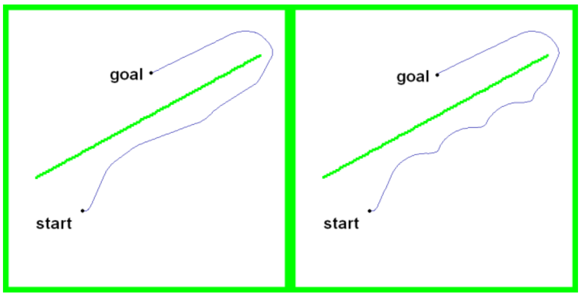


Figure 5. Simulation results with and without *Wall Following* behavior.

Emergency situations: This behavior tends to drive the robot away from U-traps and similar obstacles. Fig. 6 shows the simulation result of the *emergency situation* behavior. It is clear that the robot gets out from traps and delicate situations and reaches the goal safely.

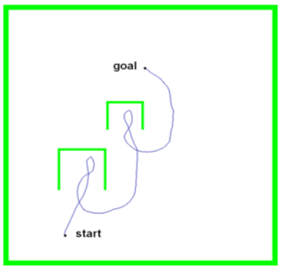


Figure 6. Simulation results of the *emergency situation* behavior

The inputs of the supervision layer are the distances to obstacles as measured by the different sonar's fixed on the robot as well as Drg and *θerror*. The supervision layer is made up based on fuzzy rules as follows,

IF *context* THEN *behavior*

For example a rule could be

R$^{(i)}$: IF *RU is F and FR is F and FL is F and LU is F* THEN *Goal reaching*

Where RU, FR, FL and LU are the Right up, Front right, Front Left and Left up respectively-IR sensors readings as defined in Fig.7. F is far and Goal Reaching is the *goal reaching* behavior.

These three behaviors need inputs to read information about the navigation environment. The Pekee robot is equipped by 15 infrared sensors integrated on its body. For reason of simplification of the problem, this work considers clustered sensors into 6 groups. Each group informs the robot on the nearest obstacle detected. Fig. 8 shows the membership functions of the distance between the robot and the obstacle i, *D_roi* (mm).

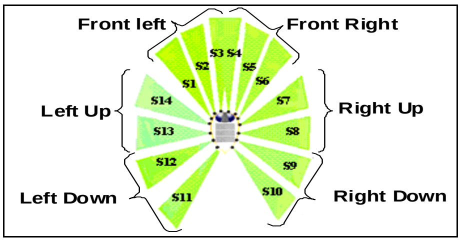


Figure 7. Clustered sensors

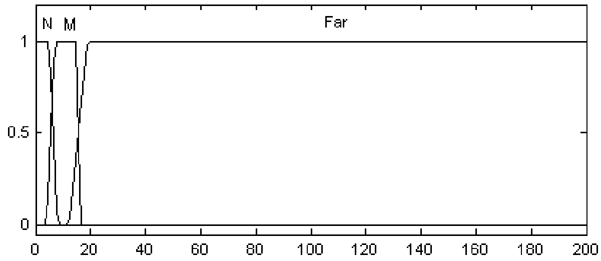


Figure 8. Membership functions of the input D_ro. (N: Near, M: Medium, F: Far)

It is well known that in the field of mobile robots, the coordination between behaviors is a major behavior based navigation problem (Fatmi, A.; Al-Yahmedi, A.; Khriji, L. & Masmoudi, N., 2006; Aguirre, E. & Gonzalez, A., 2000). Several strategies have been proposed in literature to overcome this problem. Among them we can site for instance:

- A first strategy based on the weighted combination between the behaviors. The velocity will be the weighted sum of the velocities of the behaviors (Cang, Y. & Danwei, W., 2001; Althaus, P. & Christensen, H.I., 2002). This strategy can be applied using the hierarchical fuzzy logic (Hasegawa, Y.; Tanahashi, H. & Fukuda, T., 2001; Hagars, H.A., 2004 ; Ziying, Z.; Rubo, Z. & Xin, L., 2008).
- A second strategy consists on the activation of, only, one behavior in each situation (Fatmi, A.; Al-Yahmedi, A.; Khriji, L. & Masmoudi, N., 2006). As a result time and computational resources are saved.

We have followed the second strategy in our work. The reinforcement learning technique used to improve the suitable choice of behavior by exploiting the acquired experimental learning of the robot during its navigation.

3. Reinforcement Learning

Reinforcement Learning (RL) is a machine learning paradigm (Lanzi, P.L., 2008). It makes possible the solution of the problem in a finite time based on its own experimental learned knowledge. The basic idea in reinforcement learning is that an agent (robot) is placed in an environment and can observe the results of its own actions. It can see the environment and concludes its current state (s_t). Then the algorithm selects a decision (d_t). This decision can change the environment and the state of the agent becomes (s_{t+1}). The agent receives a reward r for the decision d_t. While processing, the agent keeps trying to maximize this reward.

3.1. Q-Learning Algorithm

Q-learning is an artificial intelligent technique that has been successfully utilized for solving complex MDPs that model realistic systems.

In the Q-learning paradigm, an agent interacts with the environment and executes a set of actions. The environment is then modified and the agent perceives the new state through its sensors. Furthermore, at each epoch the agent receives an external reward signal. In this learning strategy, an objective is defined and the learning process takes place through trial and error interactions in a dynamic environment. The agent is rewarded or punished on the basis of the actions it carries out. Let s denote a state and d denote a decision, the objective of this learning strategy is teaching agent the optimal control policy, s → d, to maximize the amount of reward received in the long term. Over the learning process, Q-value of every state-decision pair, Q(s,d), is stored and updated. The Q-value represents the usefulness of executing decision d when the environment is in state s. Q-learning directly approaches the optimal decision-value function, independently of the policy currently being followed. Its updating rule is (Akira, N.; Hiroyuki W.; Katsuhiro H. & Hidetomo I., 2008),

$$Q(s_t,d_t) = (1\text{-}\alpha).Q(s_t,d_t) + \alpha.(r + \gamma.\max_{d_{t+1}} Q(s_{t+1},d_{t+1}) \quad (1)$$

Where r_t denotes the reward received at epoch t, $0 \prec \gamma \prec 1$ denotes the discount factor and α denotes the learning rate. Let Q*(s,d) denotes the optimal expected Q-value of state-decision pair (s,d). If every decision is executed in each state infinitely, Q(s,d) will converge to Q*(s,d).

The Q-Learning algorithm can be written:

1. Obtain the current state s_t.
2. Choose a decision d_t, and execute it.
3. Obtain the new state s_{t+1} and the immediate reward r.
4. Update the matrix $Q(s_t, d_t)$ with the equation:

$$Q(s_t, d_t) = (1\text{-}\alpha).Q(s_t, d_t) + \alpha.(r + \gamma.\max_{d_{t+1}} Q(s_{t+1}, d_{t+1}))$$

5. Assign $s_t = s_{t+1}$
6. While $s_t \neq s_{optimal}$ return to 2.

where α and γ are the training rate and the discount factor, respectively. Both parameters belong to the interval $[0,1]$.

3.2 Application of the Q-Learning algorithm

The Q-Learning algorithm is exploited to coordinate between the fuzzy behaviors. The different parameters are:

- ***States***: They are taken from all possible combination between variables:
  - *D_roi*, i = 1...4: The distance between the robot and the obstacle i. (*D_roi* ∈ [0, 0.3]m or *D_roi* > 0.3m)
  - *Distance*: The distance between the robot and the goal (*Distance* ∈ [0, 0.3]m or *Distance* > 0.3m)
  - *Θ-error*: The difference between the desired angle and the current angle of the robot
  - (Absolute value (*Θ-error*) <5deg. else if *Θ-error* > 0 else *Θ-error* < 0).

- ***Decisions***: They are behaviors recognized by fuzzy logic: *goal reaching, Obstacle avoidance, wall following and emergency situations*.

- ***Immediate Reward:*** The immediate reward algorithm is,

```
If distance is near then r = 60000
            Elseif
If the algorithm can decide to about the action
then
   If the best action is chose then r = 1500
   Elseif r = -1500
   Else r = 0
End.
```

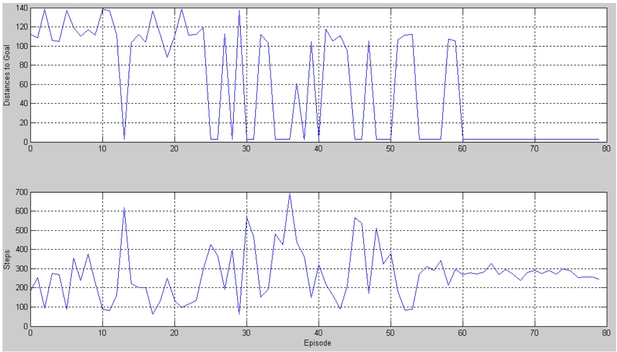


Figure 9. Number of steps and Distance to goal versus Episode

4. Simulation and experimental results

This section will be devoted to the simulation and experimental results after using the Q-Learning approach to coordinate between behaviors based on fuzzy logic. Our experimental procedure comprises two phases: learning and testing. The learning phase is used to obtain convergent Q-values. In each learning step of the learning process, the decision is chosen based on the fuzzy logic results of different behaviors and the Q-value is updated according to Eq. (1). Here, we use the number of learning cycles to measure the learning efficiency as shown in Fig. 9. It shows that the learning process of our Q-learning algorithm becomes stable at around 60 episodes. One learning cycle (episode) is the learning process from the start point to the end of the horizon. The test phase is used to measure the performance of the system and to determine when the learning process has converged. During the test phase, the stored Q-values are loaded and the best-valued decision for current state and event is always selected real robot navigation. In all our simulations and experiments the learning rate is 0.2 and the discount factor is 0.8.

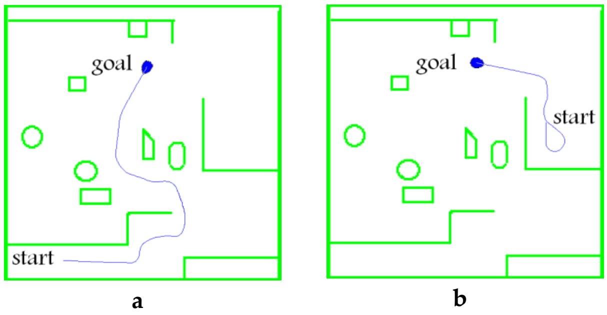
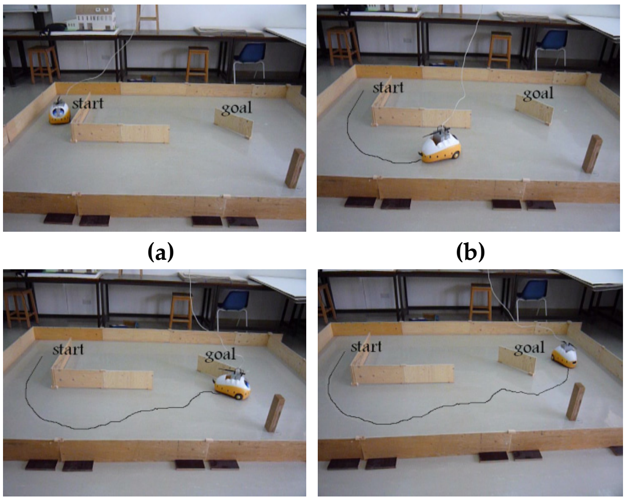


Figure 10. Navigation in a crowded environment after learning.

It is noticed, in both cases of Figs. 10(a-b), that the robot reaches the goal at two different starting points with avoiding obstacles. In Fig. 10(a) the robot follows the walls then it avoids the obstacles and finally it can go to the goal when there are no obstacles. In Fig. 10(b) the robot is putted from the beginning in an emergency situation (3 obstacles in the 3 directions: front, right and left), it could leave this situation and then it avoids the obstacles and finally it goes straight to the goal. The effectiveness of the developed navigation approach was experimentally demonstrated on a Pekee robotic platform. The real experimental results are shown in Fig.11 (a-d) demonstrating the validity of our approach.



(a) **(b)**

(c) **(d)**

Figure 11. Pekee robot navigates in a real environment with obstacles.

5. Conclusion

In this paper we have shown how Q-learning approach can be used in a successful way of structuring the navigation task in order to deal with the problem of mobile robot navigation. In real mobile robot navigation system there often exists prior knowledge on the task being learnt that can be used to improve the learning process. A navigation approach has been proposed for mobile robot, in which the prior knowledge is used within Q-learning. Issue of individual behavior design were addressed using fuzzy logic (i.e. *go to goal, obstacle avoidance, wall following and emergency situations*). The strategy of behaviors based navigation reduces the complexity of the navigation problem by dividing them in small actions easier for design and implementation. In our strategy, the Q-Learning algorithm is applied to coordinate between these behaviors, which make a great reduction in learning convergence times. Our simulation and experimental results confirmed that a mobile robot with the Q-learning algorithm proposed in this paper is able to learn and choose the best decision in each situation of its dynamic environment.

6. Acknowledgment

The authors wish to thank SQU University for supporting the work and providing the Pekee robot and all licensed software's. SQU grant number "IG/ENG/ECED/08/02".

7. References

Aguirre, E. & Gonzalez, A. (2000). Fuzzy behavior for mobile robot navigation: design, coordination and fusion. International Journal of Approximation Reasoning, Vol.25, pp.255-289.

Akira, N.; Hiroyuki W.; Katsuhiro H. & Hidetomo I. (2008). Cell Division Approach for Search Space in Reinforcement Learning", IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.6.

Althaus, P. & Christensen, H.I. (2002). Behavior coordination for navigation in office environments. IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol.3, pp. 2298-2304.

Antonelli, G.C. & Fusco, S.G. (2007). A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking. IEEE Trans. on Systems, Man and Cybernetics: Systems and Humans, pp. 211-221.

Borenstein, J. & Koren, Y. (1991). The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robot. Transaction on Robotics and Automation, Vol. 7, pp. 278-288.

Cang, Y. & Danwei, W. (2001). A novel behavior fusion method for the navigation of mobile robot. IEEE international conference on System, Man and Cybernetics. pp. 3526-3531, Nashville.

Das, T. & Kar, I. (2006). Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots. IEEE Trans. on Control Systems Technology, pp. 501-510.

Dongbing, G. & Huosheng, H. (2007). Integration of Coordination Architecture and Behavior Fuzzy Learning in Quadruped Walking Robots. IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 37, pp. 670-681.

Fatmi, A.; Al-Yahmedi, A.; Khriji, L. & Masmoudi, N. (2006). A Fuzzy Logic Based Navigation of a Mobile Robot. Inter. Journal of Applied Mathematics and Computer Sciences, Vol. 1, N.2, pp. 87-92.

Hagars, H.A. (2004). A Hierarchical Type-2 Fuzzy Logic Control Architecture for Autonomous Mobile Robots. IEEE trans. on fuzzy systems, Vol. 12, pp. 524-539.

Hasegawa, Y.; Tanahashi, H. & Fukuda, T. (2001). Behavior coordination of brachiation robot based on behavior phase-shift. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 526-531.

Joo, M. & Deng, C. (2005). Obstacle Avoidance of a Mobile Robot Using Hybrid Learning Approach. IEEE transactions on industrial electronics. Vol. 52, 3, pp. 898-905.

Lanzi, P.L. (2008). Learning classifier systems: then and now", Evol. Intel. Springer-Verlag, pp.63–82.

Parasuraman, S.; Ganpathy, V. & Shiainzadeh, B. (2005). Behavior based mobile robot navigation technique using AI system: experimental investigations. ARAS 05 conference, Cairo, Egypt.

Ulrich, I. & Borensstein, J. (2001). The GuideCane-applying mobile robot technologies to assist the visually impaired. IEEE Transaction on System, Man and Cybernetics. Part A: System and Human, Vol. 31, 2, pp. 131-136.

Youcef, Z. (2004). Apprentissage par renforcement et système distribués : application a l'apprentissage de la marche d'un robot hexapode. PhD Thesis. Institut National des Sciences Appliquées de Lyon.

Ziying, Z.; Rubo, Z. & Xin, L. (2008). Research on Hierarchical Fuzzy Behavior Learning of Autonomous Robot. Internet Computing in Science and Engineering, pp. 43-46.