Journal of Applied Sciences 4 (3): 388-392, 2004
ISSN 1607-8926

Fuzzy Reinforcement Rectilinear Trajectory Learning

[1]Y. Dahmani and [2]A. Benyettou
[1]Signals Images and Speech Laboratory, Ibn Khaldoun University, B.P. 78 C.P. 14000 Tiaret, Algeria
[2]Department of Data Processing, University of Sciences and Technology of Oran,
B.P. 1505 M'Naouer 31000 Oran, Algeria

Abstract: The objective of this work tries to answer the question, in what the reinforcement learning applied to fuzzy logic can be of interest in the field of the reactive navigation of a mobile robot. In the first instance we have established an algorithm applying the reinforcement learning to fuzzy limited lexicon. We have applied it to a robot for the training of the follow-up of a rectilinear trajectory of a starting point "D" at a point of unspecified arrival "A", while avoiding with the robot butting against a possible obstacle.

Key words: Fuzzy logic, fuzzy Q-learning, reinforcement learning, fuzzy inference system, mobile robot

INTRODUCTION

Now-a-days robots need more sense, decision and technology[1]. Among the points most difficult and required by the current world, is certainly the navigation of the robots in mediums generally not structured[2].

It is in this way that we have tried the robot learn how to follow. Firstly to make the robot learn how to follow a straight line trajectory aiming to make the follow behaviour perfectly an object which constitutes one of the modules in the navigation of a mobile robot while considering the behaviour-based architecture[3].

Robot architecture: A rather standard architecture was used[4], the robot considered is circular having three sensors one in front and one on each side. The angle of sensors orientation chooses are of 45° on both sides of the frontal axis of the robot (Fig. 1).

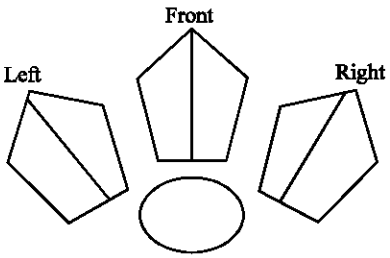


Fig. 1: Structure and position of the sensors

The robot must move along a straight line trajectory, from a starting point "D" to any objective point "A". It must thus learn to follow this trajectory. By these three sensors the robot calculates the length "l" compared to an eventual obstacle, its orientation θ and the angle θ' with respect to the objective (Fig. 2).

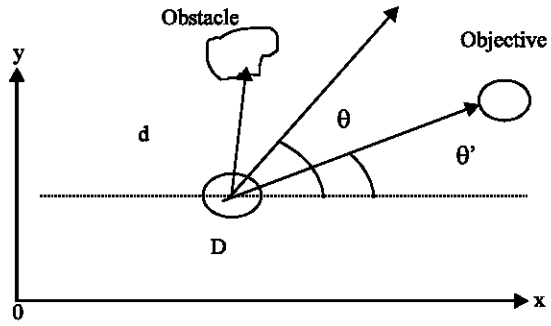


Fig. 2: Kinematic model

Navigation: Our aim is to allow the robot, initially to orient its angle directly towards the objective point. Then it must learn how to move along this trajectory by holding its straight line.

The use of fuzzy logic seems to give good results in this kind of problems such navigation without an analytical model of the environment. It remains to notice, as soon as the environment becomes complex, two problems emerge to knowing:

- The difficulty of the construction of the rule basis
- Refinement of this rule basis

In order to remedy to these problems, we proposed a model using fuzzy logic and the reinforcement learning. The main concepts in the reinforcement learning, are the

Corresponding Author: Dahmani Youcef, Cite des 50 logements universitaires, A-3 Zaaroura Tiaret 14000, Algeria
Tel: (213) 46 43 88 62 Fax: (213) 46 42 47 10 E-mail: damani_y@yahoo.fr

agent and environment[5-7]. The agent has a number of possible actions, the agent improves some actions in the environment which is modeled by a set of states. For some states, the agent receives a signal of the environment called the reward. The task of the reinforcement learning is to find the action which gives the greatest value of the discounted reward called the Q-value. The step passes by two stages:

- A phase of Exploration
- The second phase of Exploitation

The reasoning process: The well known method by the most popular reinforcement learning is Q-learning where an agent updated successively the quantity $Q_t(x,a)$ which represents the quality of the selected action "a" for the state "x". Within this framework, we used an alternative of Q-learning associated with fuzzy logic in order to as well as possible use its properties such the formulation of human knowledge in the form of fuzzy rules and the use of imprecise and vague data. Its principle consists in proposing several conclusions for each rule and to associate each potential solution a quality function[8,9].

R_i: If x_1 is A_1^I and ..... and x_n is A_n^I Then
y is u[i,1] with q[i,1]=0
or
y is u[i,2] with q[i,2]=0
.....
or
y is u[i,N] with q[i,N]=0.

Where $(u[i,j])_{j=1}^{N}$ are potential solutions whose quality is initialized arbitrarily.
The inferred output is given by the formula:

$$U(x) = \frac{\sum_i \alpha_i(x) u[i,PEE(i)]}{\sum_i \alpha_i(x)}$$

The quality of this action is:

$$Q(x,U) = \frac{\sum_i \alpha_i(x) q[i,PEE(i)]}{\sum_i \alpha_i(x)}$$

The approach, is similar in its principle to that previously quoted except that the set of the suggested actions are not crisp values but fuzzy subsets; because in practice, we can be in the presence of case where the set of the actions to be chosen is not given in known actual values but rather in the form of linguistic terms such to have the choice between turning slightly or midway. Hence the rules which we will use will take the following form:

R_i: If x_1 is A_1^I and ..... and x_n is A_n^I Then
y is B[i,1] with q[i,1]=0
or
y is B[i,2] with q[i,2]=0...
.....
or
y is B[i,N] with q[i,N]=0.

Where B[i,j] represents the fuzzy subset associated with the rule I and the conclusion j.

Fuzzification of inputs and outputs: In present case, it was considered that fuzzy linguistic rules with two inputs, $\Delta\theta$ which is the difference between the course of the robot i.e. its orientation and the objective, ($\Delta\theta = \theta - \theta'$), as for the second entry "d", it represents the distance to an obstacle, while the output is the orientation α which the robot must take.

Hence the following fuzzy subsets with their fuzzification was obtained (Fig. 3).

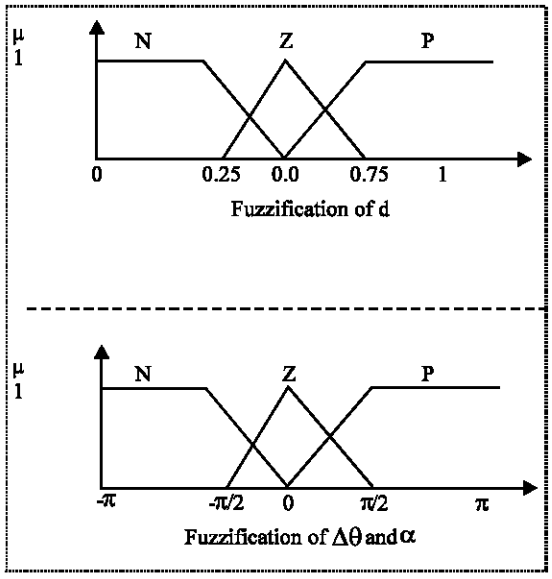


Fig. 3: Fuzzification of inputs and outputs

Construction of the basic rules: The basic rules of simulator are given by:

R_1: If $\Delta\theta$ est N and d is N Then
α is N with q_{1N}
or
α is Z with q_{1Z}
or
α is P with q_{1P}

In total, we have nine rules:

R_i: If $\Delta\theta$ is A and d is B Then
α is N with q_{iN}
or
α is Z with q_{iZ}
or
α is P with q_{iP}

with A et B are subsets whose linguistic terms can be N (negative), Z (zero), or P (positive).

Exploration phase: The first step in reinforcement learning is the exploration, which consists in choosing the best actions progressively. We proposed the following diagram block (Fig. 4) and we must follow the following algorithm:

1. Initiate the different qualities q_{iA} by number 0
2. Repeat for a given number of period "n"
3. Calculation of the degrees of membership of each input to the various fuzzy subsets:
   $\mu_{Aj}^{I}(x_j)$ for j=1 to n and I =1 to N.
4. Calculation of the truth value of each rule, for I=1 to N:
   $\alpha_i(x) = \min_j(\mu_{Aj}^{I}(x_j))$ for j = 1 to n
5. Choose an action by the pseudo-stochastic method which is summarized by:

- The action with better value of q_{iA} has a probability P of being selected.
- Otherwise, an action is selected randomly amongst all the other possible actions in a given state

6. Calculation of the contribution of each chosen rule by the pseudo-stochastic method:
   $\mu(\alpha)=\min(\alpha_i(x), \mu_B^{I}(\alpha))$
7. Aggregation of rules:
   $\mu(\alpha)=\max_I(\mu_B^{I}(\alpha))$
8. Defuzzification of the output variable:

$$\alpha = \frac{\int u\mu(u)du}{\int \mu(u)du}$$

9. Calculate the new orientation of the robot:
   $\theta = \theta + \alpha$
10. Move the robot and compute the variation
    $\Delta\theta = \theta - \theta'$.
11. Calculate the reinforcement:

$$r = \begin{cases} +1 & \text{if } d\_ac > d\_an \\ -1 & \text{otherwise} \end{cases}$$

where d_ac is the current distance with respect to the objective following the displacement of the robot, as for d_an is the old distance compared to the objective i.e. the state before the displacement of the robot.

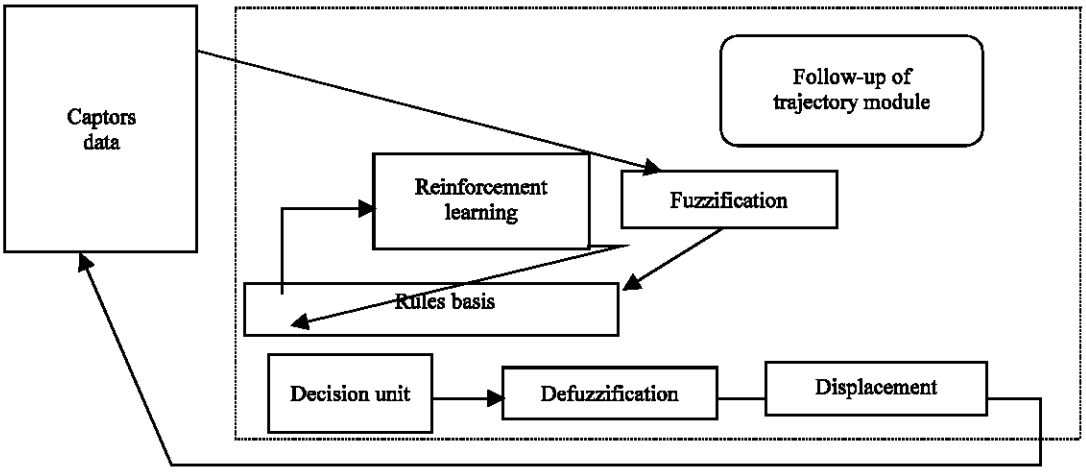


Fig. 4: Follow-up of corridor module

12. Update qualities of the rules which contributed to the variation of the angle α[10,11]:

$$q_{iA} = (1-\beta)\, q_{iA} + \alpha_i^* r^* \gamma$$

β: learning rate , γ: delay factor , α_i: truth value of rule I

13. If "n" is reached or d_ac is small, we stop the learning process

Exploitation phase: The optimal policy is obtained by choosing the action which, in each state, maximizes the quality function:

$$u = \arg \max_{u \in U_x} Q^*(x,u)$$

This policy is called "greedy" . However, at the beginning of the learning, the values Q(x,u) are not significant and the greedy policy is not applicable.

For our case, the robot is set to its starting point "A" and for each displacement, it follows the following steps:

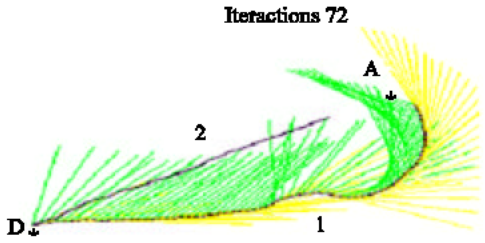
- Direct the robot towards its arrival point
- Repeat
- Calculate the distance "d" compared to possible obstacles
- Move by choosing the action of best quality using the fuzzy controller
- Until reaching the goal

RESULTS AND DISCUSSION

During present work, the following coefficients were choose:
The probability P = 0.9, γ = 0.9 and β = 0.9
Number of passes n = 1000
The basic rules are given by Fig. 5.

Δθ / d	N	Z	P
N	N, q_{1N}	N, q_{2N}	N, q_{3N}
	Z, q_{1Z}	Z, q_{2Z}	Z, q_{3Z}
	P, q_{1P}	P, q_{2P}	P, q_{3P}
Z	N, q_{4N}	N, q_{5N}	N, q_{6N}
	Z, q_{4Z}	Z, q_{5Z}	Z, q_{6Z}
	P, q_{4P}	P, q_{5P}	P, q_{6P}
P	N, q_{7N}	N, q_{8N}	N, q_{9N}
	Z, q_{7Z}	Z, q_{8Z}	Z, q_{9Z}
	P, q_{7P}	P, q_{8P}	P, q_{9P}

Fig. 5: The basic rules



	N	Z	P
R1	0.00	0.00	0.00
R2	0.00	0.00	0.00
R3	0.00	0.00	0.00
R4	0.00	0.00	0.00
R5	0.00	0.00	0.00
R6	0.00	0.00	0.00
R7	0.23	0.00	0.27
R8	0.44	0.63	0.00
R9	0.00	0.00	0.00

Fig. 6: Exploration phase(1)/exploitation(2)

From Fig. 6, it can be noticed that the trajectory (1) follow-up by the robot at the time of the phase of exploration is different from the trajectory (2) which is carried out at the time of the exploitation phase (after learning).

We also notice according to the above table that only the linguistic rules 7 and 8 which contributed in this example hence the change of the coefficients of qualities of only these two rules.
Another teaching which we can draw from this Table that:

Rule 7: If (θ-θ') is N (negative) and d is P Then α is P (positive) which is favored and has the best quality (0.27).

Rule 8: If (θ-θ') is Z (zero) and d is P Then α is Z (zero) which is favored and has the best quality (0.63).

These two results are completely logical because if the variation tends towards the negative one, it is necessary to apply a positive action to compensate this deviation and if it is null, it is necessary to maintain the action of term zero i.e. to keep the same angle without deviation. It should be also noted that the iteration count that we needed to find these results is 72.

This study has allowed us to implement a fuzzy approach applied to the reinforcement learning. It gave us good results, in particular in the follow-up of rectilinear trajectory. However, the subject is not ready to be completed.

Further work is needed and should be retained:

- Realization of a robot able to learn how to avoid obstacles

- Realization of a robot able to generate actions in conflict
- Integration of the vague concepts, training by reinforcement learning on all the levels of the architecture of the robot and addition of some alternatives
- Realization of a simulator allowing to code and simulate the behavior of a robot in an environment a priori unknown.

REFERENCES

1. Michita, I., K. Hirachi and T. Miyasato, 1999. Physical Constraints on human robot inteaction. Int. Joint Conf. on Artificial Intell., Sweden.
2. Maaref, H., 1999. Imperfect Data treatment in the setting of the Fuzzy Theory: Contribution to the Navigation and the Localization of a Mobile Robot, Memory of authorization to direct research. University of Evry, Paris, France.
3. Joo-Ho, L., A. Guido and H. Hideki, 1999. Physical Agent for Sensored Networked and Thinking Space. Proceedings of the 1998 IEEE. Int. Conf. Robotics and Automation. Leuven, Belgium., pp: 838-843.
4. Uribe-Gutierrez, S. and H. Martinez-Alfaro, 2000. An Application of Behavior-Based Architecture for Mobile Robots Design. Lecture Notes in Artificial Intelligence 1793, MICAI 2000: Adv. Artificial Intelligence. Mexico, pp: 136-147.
5. Glorennec, P.Y., L. Foulloy and A. Titli, 2003. The reinforcement learning, application for fuzzy inference systems. Fuzzy Order 2, Treated IC2, Ed Lavoisier.
6. Hiroshi, I., K. Masatoshi and I. Toru, 1999. State Space Construction by Attention Control. Int. Joint Conf. Artificial Intell., Sweden, pp: 1131-1139.
7. Jacky, B. and L. Yumin, 2000. Path Tracking Control of Non-holomonic Car-Like Robot with Reinforcement Learning. Lecture Notes in Artificial Intelligence 1793 MICAI 2000: Adv. Artificial Intell., Mexico.
8. Glorennec, P.Y., 1998. Algorithms of optimization for fuzzy infernece systems: Application for identification and order. National Institute of Applied Sciences. Rennes, France.
9. Jouffe, L., 1997. Training of fuzzy inference systems by reinforcement methods: Application to the regulation of ambiance in a building of pork raising. Ph.D. Thesis, University of Rennes I, France.
10. Garcia, P., A. Zsigri, A. Guitton, 2003. A Multicast Reinforcement Learning Algorithm for WDM Optical Networks. 7th Intenational Conference on Telecommunications-ConTEL ISBN:953-184-052-0, June 11-13, 2003, Zagreb, Croatia.
11. Mark, D.P., 2000. Reinforcement Learning in Situated Agents: Theoretical Problems and Practical Solutions. Lecture Notes in Artificial Intell. 1812. Berlin, Germany, pp: 84-102.