

Parameter Tuning of Fuzzy Subsets Inertia Influence in Navigation Case

Y. Dahmani and ¹A. Benyettou

Signals, Images and Speech Laboratory, B.P. 78 C.P. 14000, Ibn Khaldoun University Tiaret, Algeria

¹University of Sciences and Technology of Oran, B.P. 1505 M'Naouer 31000 Oran Algeria

Abstract: This study deals with avoider obstacle behaviour of a mobile robot in unknown environment. The use of fuzzy logic allows to provide universal rules for avoiding obstacles, and on other hand it gives assurance for a predicted and reliable behaviour of the robot. By lack of reference trajectory, it was urged to use one of reinforcement learning method, fuzzy Q-learning which allows to take into account continual state spaces and actions. In this study, an adaptation method was proposed of fuzzy linguistic rules: after the stage of rules extraction from a fuzzy inference system of fixed structure, we provide a methodology for parameter tuning of the fuzzy sets in terms of the robot inertia without affecting the conclusion rules.

Key words: Fuzzy logic, reinforcement learning, fuzzy Q-learning, parameter tuning

INTRODUCTION

The reactive navigation, for a mobile robot, is based on modifying dynamically a behaviour according to unpredicted changes in the environment. More precisely the robot must be able to adapt continuously its speed and direction in order to avoid obstacles that it encounters in expectedly. This approach differs from planning (planification) which is based on a complete knowledge of the environment and location of the robot at every instant of time.

Here, the environment may be unknown and/or variable so that the robot must be provided with generic rules of behaviour in order to adapt itself accordingly.

Fuzzy logic allows to express knowledge in natural language by means of vague and ambiguous words. This knowledge may be expressed in symbolic way in terms of rules that we can translate immediately.

A behaviour of avoiding obstacles may be easily modeled by rules like "If there is an obstacle on the left, and there is nothing ahead and nothing on the right, then turn slightly to the right". In their symbolic form, these rules are universal i.e. they can be apply them to any robot, but during a practical application a given robot must translate them i.e. move from the qualitative to the quantitative and give a precise meaning to the terms "fuzzy" of its basic rules : this is the role of learning.

The approach followed in this study consists of 3 stages:

- Modelling of the state of a robot with respect to obstacles and writing the basic symbolic rules

- Dynamic adaptation of rules conclusions through the fuzzy Q-learning
- Modification of membership functions with respect to the variable load of the robot

The first two steps are not original: the contribution is mainly in the third step, a methodology was proposed for parameters tuning of the fuzzy subsets.

Main rules for reactive navigation: The robot used in our study is an aspirator-robot, which is supposed to clean the inside of a room in a random way while avoiding obstacles.

However this method can be extended to any similar robot. The basic robot behaviour is defined by means of fuzzy inference system which is optimised by learning for a given load. One of the problem is to maintain an optimal behaviour with different loads while the load is altered i.e. a modified inertia.

The considered aspirator-robot is provided with three ultra-sound radars, one in the left, one in front of it and one in the right. This robot is also provide with steering wheel controlled in relation to the sensed obstacles. The minimum distances between the robot and the obstacles sensed by the left, front and right sensor are denoted l, f, r. $\Delta\theta$ is the change in direction be under taken (with the convention : $\Delta\theta > 0 \Rightarrow$ turn to the left, $\Delta\theta < 0 \Rightarrow$ turn to the right), while the speed of travelling is denoted by v (Fig. 1).

Two fuzzy modes for each input variable, Near and Far were assigned. The predicate "l is Near" relate to a near obstacle on the left and so on. The membership



Fig. 1: Block diagram of the robot

functions on the distances domain are shown in Fig. 2. For each input, two parameters to be tuned:

- Parameter a refers to the minimum distance between the robot and an obstacle
- Parameter b denotes the longest braking distance : over this distance, the robot has no constraints on its speed.

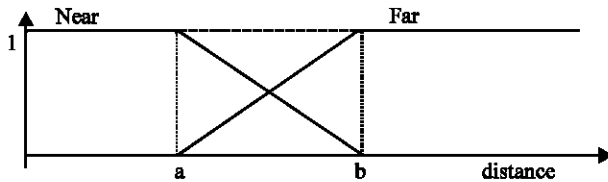


Fig. 2: Membership functions for distances

With this choice of fuzzy subsets, there are only eight cases to be taken into consideration, hence we can deduce immediately eight rules for speed and eight other rules for angular deviations. These rules are presented in Table 1, with customary conventions : PB for Positive Big, NS for Negative Small and so on. These rules have only to translate the natural behaviour of any conductor. For example rule 2 may be written in a symbolic way as :

If l is Near and f is Near and r is Far
Then $\Delta\theta$ is NB , $V = 0$.

The robot is in a left corner : it has to stop ($V=0$) and turn strongly to the right ($\Delta\theta$ is NB). For the speed we introduce a reduction coefficient $C < 1$, for the rule 3, which corresponds to corridor. This coefficient can be deduced from the corridor width.

Table 1: Basic rules

	l	f	r	$\Delta\theta$	v
R1	Near	Near	Near	PG	0
R2	Near	Near	Far	NG	0
R3	Near	Far	Near	ZR	$C * V_{max}$
R4	Near	Far	Far	NP	V_{max}
R5	Far	Near	Near	PG	0
R6	Far	Near	Far	NG	0
R7	Far	Far	Near	PP	V_{max}
R8	Far	Far	Far	ZR	V_{max}

These rules are then universal. The difference between two robots comes from the interpretation that they assign for every conclusion term: Positive Small may be translated as 10% for one and 7% for the other. During

the learning process that each robot would refine its basic rules.

Reinforcement learning: The reinforcement learning considers an agent in interaction with its environment. The agent and its environment form a dynamic system^[1,2] and their interaction can be modelled as a markovien decision problem. The markovien hypothesis is often very strong, but allows a rigorous formulation^[3].

In every state, the agent have a certain number of possible actions. When it chooses an action, it passes in another new state and receive a scalar signal from its environment, called the reinforcement, this signal is perceived as reward or as a punishment. The task of the agent is to choose the action corresponding to optimal policy, i.e. the policy which will optimise the future positive reinforcements. The reinforcement function is important in the design of learning process because it must reflect the agent progression towards the task achievement.

In reactive navigation, there is no reference trajectory, so that we can't apply a supervised learning^[3,4]. The only feedback that the agent can receive on its actions is the reinforcement that it obtain from the environment.

The Q-learning: The most popular reinforcement learning method is the Q-learning^[5]. As a matter of fact, it differs from the other methods as follows:

- It allows to find the optimal control policy, without a pre-established model
- It is the simplest technic and its convergence has been proved.

At every step in time, an agent observes the state vector x_t , chooses and apply an action u_t belonging to U_{x_t} set of possible actions in this state. The system passes then to the state x_{t+1} and the agent receives a reinforcement $r(x_t, u_t)$. A quality value denoted $Q(x, u)$, measures the need of choosing action u when we are at state x . The calculation of $Q(x, u)$ is independent from the followed policy, it is given at every step in time by the formula:

$$Q(x_t, u_t) - Q(x_t, u_t) + \beta \left[r(x_t, u_t) + \gamma \max_{v \in U_{x_{t+1}}} Q(x_{t+1}, v) - Q(x_t, u_t) \right]$$

where γ is the discount factor $0 < \gamma < 1$.

This function Q , is initialised with zero and updated incrementally by the precedent formula as a function of the received reinforcement and the value of the following state. The proof of the formula can be found for example in Glorionec^[2], Sutton and Barto^[6].

The fuzzy Q-learning: The fuzzy Q-learning is an extension of the Q-learning allowing the in-line optimisation of conclusions of a fuzzy inference system. The rule premises correspond to the states and the conclusions to the actions. The principle consists in proposing several actions for each rule and associate to them a quality function which will be updated at each step and which will allow to select the optimal conclusion. The initial rule basis has then the following form :

$$R_i : \text{If } x_i \text{ est } A_1^i \text{ and } \dots \text{ and } x_n \text{ est } A_n^i \\ \text{Then } y \text{ is } u_1^i \text{ with } q_1^i = 0 \text{ or} \\ y \text{ is } u_2^i \text{ with } q_2^i = 0 \text{ or} \\ \dots \\ y \text{ is } u_N^i \text{ with } q_N^i = 0$$

where $u_j^i, I=1..n, j=1..N$ are the potential conclusions in which the quality is initialised to zero. The initial values of conclusions, u_j^i , depend on extent of knowledge.

The FQL algorithm of FQL is given in Fig. 3. The updating equations are given in Glouennec^[2].

1. Observe situation u
2. Choose action u according to an exploring policy
3. Apply action u and observe the new state
4. Take the reinforcement r(x,u)
5. Estimate the value of the new state
6. Update Q(x,u)
7. Update the qualities q_j^i

Fig. 3: Fuzzy Q-Learning algorithm

Application to the mobile robot: A fuzzy inference system of Takagi-Sugeno of order 0 was used, to optimise the steering angles. For this, three different numerical interpretations for the terms NB, NS, ZR, PS and PB, were proposed.

Rule R_1 for example will be:
 R_1 : If g is Near and f is Near and d is Near
 Then $\Delta\theta_1 = 45^\circ, q_1^1$
 or $\Delta\theta_2 = 40^\circ, q_2^1$
 or $\Delta\theta_3 = 35^\circ, q_3^1$

with q_j^1 corresponding to action j quality in the rule I with $I = 1..8$ et $j=1..3$. $\Delta\theta_1, \Delta\theta_2$ et $\Delta\theta_3$ are then three coherent interpretations of "Positive Big". In this situation, the introduction of knowledge allows to undergo a local exploration, which limits the risk of aberrant behaviours.

A safe distance between the robot and an obstacle was allocated and the reinforcement signal was defined by:

$$r = \begin{cases} -1 & \text{if it is very near form the obstacle} \\ 0 & \text{otherwise} \end{cases}$$

The learning consists then in evolving the robot through an unknown environment in order to update the q_j^i values, in the way to choose among the three propositions, the one which will maximise the rewards.

Fuzzy subsets parameters readjustment

Influence on the dynamic: The fuzzy Q-learning is well adapted to rules conclusions optimisation: After the learning, for every rule the conclusion of maximum quality was kept. We hope optimise the parameters a and b of the fuzzy subsets Near and Far defined in the inputs domains. The first experience consists to change the value of b while keeping a constant. We can make the following remarks (Fig. 4).

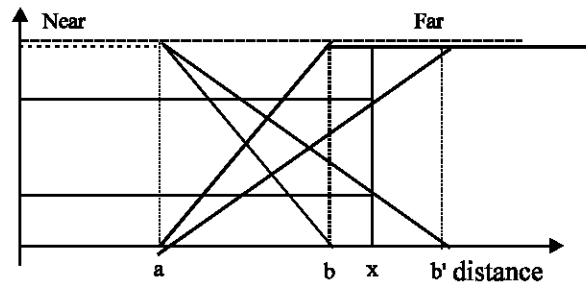


Fig. 4: Effect of the change of the parameter "b" on dynamics

Let b et b' be two modal values for the fuzzy subset Far, with $b < b'$. We denote $Far(b)$ and $Far(b')$ the corresponding fuzzy subsets. Let x be an observation of the distance for the concerned direction, with $b < x < b'$. For this distance, x, the extent of belonging to $Far(b)$ will be equal to 1: If there are no other obstacles, the robot will then travel at its maximum speed. Contrarily, the extent of belonging to $Far(b')$ will be less than 1 and in the same conditions as previous, the robot will already started to decelerate.

Hence we may conclude the following:

- Increasing b makes the robot softer, because it reacts very late
- Decreasing b makes it more reactive.

The difference $b-a$ corresponds to the distance in which the robot modulate its speed: the more it is shortened, the more the orders are likely to be brutal.

Influence on the safety: For this part, we will fix the value of b and act on the value of a will be fixed (Fig. 5). On this figure, we consider two modal values a and a', with $a < a'$. We consider the terms $Near(a)$ et $Near(a')$. Lets take a value x such as $a < x < a'$.

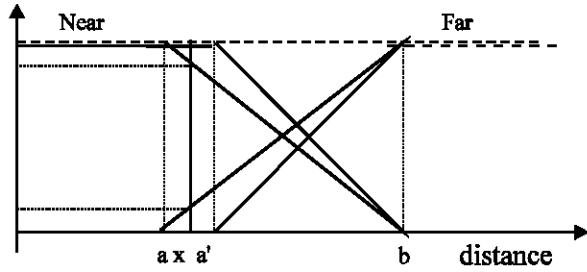


Fig. 5: Effect of the change of the parameter "a" on safety

As previously, dealt with, it could be noticed that the recommended action will be different according to the extend of belonging to Near(a) and Near(a'). For the chosen distance x, the extend of belonging to Near(a') is equal to 1, which will produce a behaviour of prudence. For the same position, the extend of belonging to Near(a) is less than 1 and that of Far is positive: there is a less faster reaction with respect to obstacles.

This translation of parameters has then a great influence on the behaviour of the controller which be more or less nervous according to the performed transformations.

The modification of parameter "a" has an effect on the robot safety and it may be conclude:

- Decreasing the value of a makes the robot to react more tardily in the vicinity of an obstacle
- Increase the value of "a" makes the robot more careful in the vicinity of an obstacle.

However, we have to notice that we should impose a limiting value for a in order to preserve the robot safety. This limiting value is given by the formulae $a + \Delta a \geq a_{min}$ where a_{min} represents the minimal distance to keep in front of an eventual obstacle.

Methodology of tuning: The inertia of the robot increases according to its load, this has an influence on its behaviour:

- More strong the inertia be, more the response time go be long and more it be necessary make the robot more nervous so that it have a good step
- The weaker inertia is, the more the robot reacts to the least request: it should be made softer.

In order to readapt the coefficients of the fuzzy subsets as a function of the inertia, we proposed a table of rules (Table 2), giving the variations to bring to parameters a and b.

Table 2: Table of rules

Load	Light	Heavy
Δa	NS	PS
Δb	PS	NS

The initial values of a and b may be fixed for null load for example. They will be then modified according the loads. If he load becomes heavier, we increase a and decrease b and reciprocally when one returns to a light load.

This study allowed to measure qualitatively the influence of parameters changes of fuzzy subsets on the robot behaviour and to propose a method of simple and intuitive rules.

We are under going works aiming at automating this parametric adaptation. More particularly we are aiming to use the reinforcement learning. In this frame of work, and in order to stay coherent with fuzzy Q-learning, we are trying to characterize the state space: a state may be assimilated to a behaviour. Notion like "Soft" and "Nervous" need to be formalised in a better way.

REFERENCES

1. Baltes, J. and Y. Lin, 2000. Path Tracking Control of Non-holonomic Car-Like Robot with Reinforcement Learning. Lecture Notes in Artificial Intell. 1793 MICAI 2000. Adv. Artificial Intell. Mexico, USA.
2. Glorennec, P.Y., 2003. Apprentissage par renforcement, application aux systèmes d'inférence floue, Commande Floue 2, Foulloy, Galichet, Titli (Eds), Traité IC2, Ed Lavoisier.
3. Imai, M., K. Hirachi and T. Miyasato, 1999. Physical Constraints on human robot interaction. Int. Joint Con. Artificial Intell. Sweden.
4. Joulain, C., P. Gaussier and A. et Revel, 1996. Apprentissage de Catégories Sensori-Motrices par un robot Autonome. Actes de NSI., Marly le Roi, pp: 31-34.
5. Watkins, C., 1989. Learning from delayed rewards. Ph.D. Thesis, Univ. Cambridge, England, UK.
6. Sutton, R. and A. Barto, 1998. Introduction to Reinforcement Learning. MIT Press.