Journal of Engineering and Applied Science 1 (1) : 24-29, 2006

Gradient Descent Adjusting Takagi-sugeno Controller For a Navigation of Robot Manipulator

S. Kermiche, M.L. Saidi and H.A. Abbassi
Asutomatic and signal Laboratory, Faculty of Engineering,
Annaba University, B.P.12 Annaba, Algeria

Abstract: This paper presents a solution for the problem of learning and controlling a 2R-plan robot manipulator in the presence of fixed obstacle. The objective is to move the arm from an initial position (source) to a final position (target) without collision. Potential field methods are rapidly gaining popularity in obstacle avoidance applications for mobile robots and manipulators. The idea of imaginary forces acting on a robot has been suggested by Andrews, Hogen and Khatib. Thus, we propose an approach based on potential fields principle, we define the target as an attractive pole (given as a vector directly calculated from the target position) and the obstacle as a repulsive pole (a vector derived by using fuzzy logic techniques). The linguistic rules, the linguistic variables and the membership functions, are the parameters to be determined for the fuzzy controller conception. A learning method based on gradient descent for the self tuning of these parameters is introduced. Therefore, it is necessary to have an expert person for moving the arm manually. During this operation of teaching, the arm moves and memorizes the data (inputs and outputs). This operation is used to find the controller parameters in order to reach the desired outputs for given inputs.

Key words: Robot manipulator, fuzzy logic control, obstacle avoidance method

INTRODUCTION

Yesterday's teleoperator movements were quite easily controllable by a human operator, but in nowadays the accuracy and complexities of the positioning of robots may be better achieved by supplementing human capabilities with computer power in order to generate these complex trajectories and to control the robot manipulator accordingly.

Robot manipulator is designed to perform efficiently very complex tasks in cluttered environments. In particular, they are required to move in the presence of fixed or even mobile obstacles, tracking a prescribed path without any collision. Some methods for generating collision-free paths are adapted from mobile robots[1-5]. Robot serial manipulator need to avoid both the end-effectors and the links. For this reason, their accessible workspace is rather limited unless their number of joints increases.

The manipulator moves in a field of forces where the goal position is an attractive pole and where obstacles and kinematics joint limits are repulsive forces (Fig. 1)[2].

These two forces determine the arm's orientation, the attractive force is calculated from the goal position and for the repulsive force a fuzzy technique is used.

The arm, the obstacle and the target (goal) can take any position inside the workspace.

The fuzzy controller using the obstacle avoidance is able to evaluate the repulsive force corresponding to the obstacle's relative position.

The learning method allows the automatic adjustment of the parameters. During manual training the controller memorizes the data.

The following method uses an adjustable fuzzy controller for the parameters determination (the number of memberships functions, the linguistic variables, the rules etc..)

MODELLING

The modelling consists to represent the arm behaviour by algebraic equations, here geometric model is used.

The parameters of the arm model are joints and operational positions. The first parameters permit to modify its geometry and the secondary determine the position and the orientation of the end-effectors[6-9].

The direct geometric model is described by the following equations :

$$X_2 = L_1 \mathrm{Cos} q_1 + L_2 \mathrm{Cos}(q_1 + q_2) \quad (1)$$

Corresponding Author: H.A. Abbassi, Asutomatic and signal Laboratory, Faculty of Engineering, Annaba University, B.P.12 Annaba, Algeria

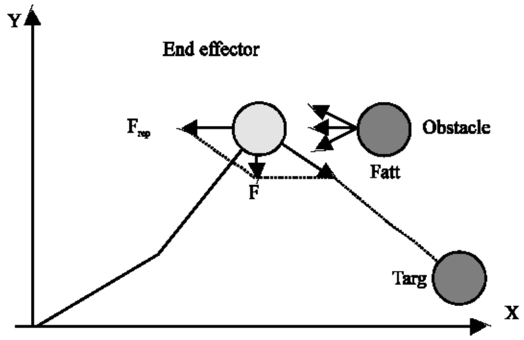

Fig. 1: Coordinate frames for two-link planar robot Collision avoidance strategy

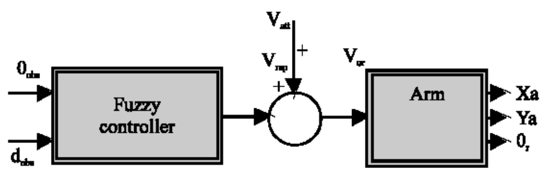

Fig. 2: Controller+Arm

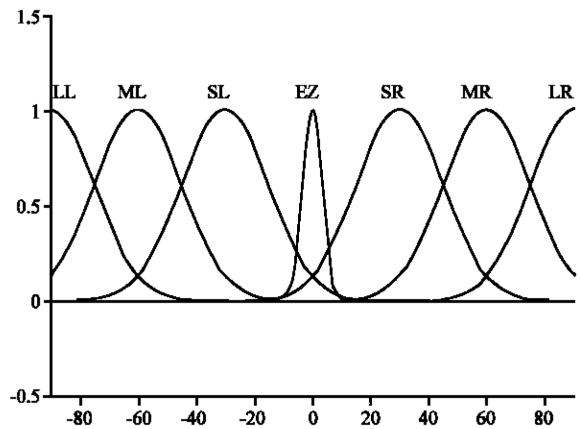

Fig. 3: Membership function for the 1st input variable 0_{obs}

$$Y_2 = L_1 \mathrm{Sin} q_1 + L_2 \mathrm{Sin}(q_1 + q_2) \quad (2)$$

And the inverse geometric model (IGM) by:

$$q_2 = \pm \mathrm{ArcCos}\left\{\left[X_2^2 + Y_2^2 - \left(L_1^2 + L_2^2\right)\right] / 2L_1L_2\right\} \quad (3)$$

$$q_1 = \mathrm{Arctg}\left(\frac{\left(Y_2 / X_2\right)\left(\left[L_1 + L_2 \mathrm{Cos} q_2\right] - X_2 L_2 \mathrm{Sin} q_2\right)}{\left(\left[L_1 + L_2 \mathrm{Cos} q_2\right] + Y_2 L_2 \mathrm{Sin} q_2\right)}\right) \quad (4)$$

COLLISION AVOIDANCE STRATEGY

A fuzzy system is a system based on the concepts of approximate reasoning: linguistic variables, fuzzy propositions, linguistic if-then rules.

The goal is to realize a fuzzy controller able to evaluate the repulsive force (vector) V_{rep} characterizing the actual relative position of the obstacle[6, 10, 2].

The controller has two inputs and one output, the inputs are the observation angle 0_{obs} and the distance d_{obs} towards the obstacle, the output is the repulsive vector V_{rep}. (Fig. 2).

The orientation angle depending on V_{or} is the input of the arm and its outputs are the coordinates (x_a, y_a) and the direction 0_a.

Fuzzification: The fuzzification module performs two tasks:

- Input normalisation , mapping of input values into normalised universes of discourse and
- Transformation of the crisp process state values into fuzzy sets, in order to make them compatible with the antecedent parts of the linguistic rules that will be applied in the fuzzy inference engine.

Fuzzification of the angle 0_{obs}: We suppose that the arm can perceive an obstacle in a direction inside the interval [-90° 90°].

The membership function is represented by seven fuzzy subsets of Gaussian form (Fig. 3):
LL: Large Left; ML: Middle Left; SL: Small Left
ZE: Zero Environment; SR: Small Right; MR: Middle Right
LR: Large Right

Fuzzification of the distance d_{obs}: We admit that the arm can detect an obstacle from a distance of 30 units.

The membership function is expressed by three fuzzy subsets (Fig. 4): S: Short; M: Medium; L: Long

Fuzzification of the repulsive angle 0rep: The membership function of the repulsive angle has a constant form belonging to the interval [-135°135] (Fig. 5).

Interference: Let $x_1, x_2, ..., x_m$ be linguistic variables on the input space $X = X_1 \times X_2 \times ... \times X_m$ and y be a linguistic variable (or a real variable) on the output space Y; then two forms of fuzzy inference rules by the fuzzy "IF ... THEN ..." rule model can be described as follow:
Form (1): Fuzzy Inference Rules by Product-Sum-Gravity Fuzzy Reasoning Method.
The fuzzy inference rules are defined as:
Rule 1: IF x_1 is A_{11} and x_2 is A_{21} and ... and x_m is A_{m1} THEN y is B_1 (5)
Rule 2: IF x_1 is A_{12} and x_2 is A_{22} and ... and x_m is A_{m2} THEN y is B_2 (6)

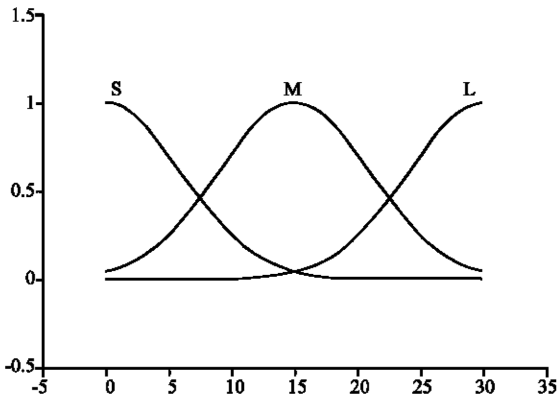

Fig. 4: Membership functions for second input variable

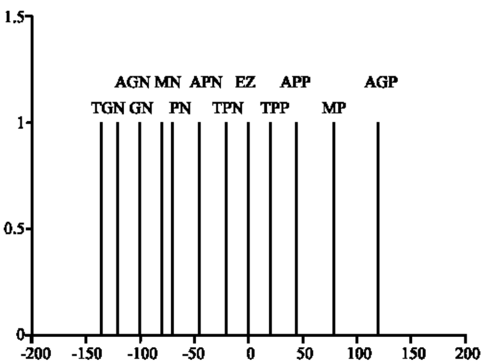

Fig. 5: Membership function of the repulsive angle 0rep

3.2. Inference

Rule n: IF x_1 is A_{1n} and x_2 is A_{2n} and ... and x_m is A_{mn} THEN y is B_n (7)

Where A_{ji} (j = 1, 2, ..., m; i = 1, 2, ..., n) and B_i are fuzzy subsets of X_j and Y, respectively and the subscript i corresponds to the ith fuzzy rule.

Form (2): Fuzzy Inference Rules by Simplified Fuzzy Reasoning Method.

The fuzzy inference rules are defined as:

Rule 1: IF x_1 is A_{11} and x_2 is A_{21} and ... and x_m is A_{m1} THEN y is y_1 (8)

Rule 2: IF x_1 is A_{12} and x_2 is A_{22} and ... and x_m is A_{m2} THEN y is y_2 (9)

Rule n: IF x_1 is A_{1n} and x_2 is A_{2n} and ... and x_m is A_{mn} THEN y is y_n (10)

Where A_{ji} (j = 1, 2, ..., m; i = 1, 2, ..., n) is a fuzzy subsets of X_j and y_i is a real number on Y[2].

For examples the representations of these rules would then be constructed as folloows:

IF (0_{ods} is LL and d_{ods} is S) THEN 0_{rep} is APP OR
IF (0_{ods} is LL and d_{ods} is M) THEN 0_{rep} is TPP OR
IF (0_{ods} is LR and d_{ods} is L) THEN 0_{rep} is EZ.

The rules are summarized in the following Table 1:

Table 1: Fuzzy rule table

		0_{obs}						
0_{rep}		LL	ML	SL	EZ	SR	MR	LR
	S	APP	MP	AGP	TGN	AGN	MN	APN
d_{obs}	M	TPP	APP	MP	GN	MN	APN	TPN
	L	EZ	TPP	APP	PN	APN	TPN	EZ

DEFFUZZIFICATION

The defuzzification module performs the conversion of the union of modified fuzzy sets into a crisp output value followed by the denormalisation of this value.

The height method is the simplest and fastest one because only peak values of the modified fuzzy sets are taken into consideration. The resulting crisp output is the weighted sum of the peak values with respect to the heights of the modified fuzzy sets.

It is interesting to notice that for this type of defuzzification, we do not need to define the widths of the membership functions. It follows that a set of output membership functions can be defined as illustrated in Fig. 5. This type of membership functions is called singletons. This definition corresponds to the special case of Takagi and Sugeno's controller.

ADJUSTABLE FUZZY CONTROLLER

The controller is based on Sugeno inference method and the defuzzification uses height method. The controller's output is function of linguistic variables[11].

Linguistic rules: We suppose that the controller has M+K linguistic variables, M inputs, K outputs and N linguistic rules[9, 1, 2].

Learning method: It is used to determine the controller parameters values in order to reach the desired outputs for given inputs (Fig. 6).

General algorithm: The model is desribed by the following equations[4]:

$$\frac{du}{dt} = f(u,x,z,t) \qquad y = g(u,x,z,t) \tag{11}$$

The parameters estimation consists to minimize the criterion V:

$$V = E\left\{(\vec{e}(t))^2\right\} \tag{12}$$

or

$$V = E\left\{\frac{1}{N}\sum_{t=1}^{N}(\vec{e}(t))^2\right\} \tag{13}$$

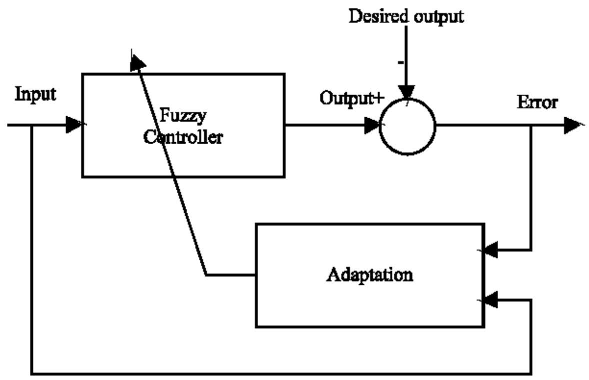

Fig. 6: Training scheme

E: means
N: number of iterations
e(t): learning error vector

In order to minimise the learning error, we will find the e minimum of the criterion function V. It can be found by solving:

$$-\nabla_z V = \left[-\frac{\partial V}{\partial z_1}, \ldots\ldots, \frac{-\partial V}{\partial z_p} \right] = 0 \quad (14)$$

$-\nabla_z V$: gradient of V
P: number of adjustable parameters

$$z_p(t+1) = z_p(t) - \Gamma_p \nabla_z V\left[z_p(t)\right] \quad (15)$$

Γ_p: is the predefined constant named learning rate.

Adaptation of parameters of Takagi and Sugeno's controller: The Takagi and Sugeno's fuzzy controller has three types of parameters to adapt:

- Centre values $a = (a_{11}, \ldots, a_{nm}, \ldots, a_{NM})^T$,
- width values $b = (b_{11}, \ldots, b_{nm}, \ldots, b_{NM})^T$,
- consequences values $c = (c_{11}, \ldots, c_{nk}, \ldots, c_{NK})^T$,

Then $\vec{Z} = (a_{11}, \ldots, a_{NM}, b_{11}, \ldots, b_{NM}, c_{11}, \ldots, c_{NK})^T$ (16)

The number of parameters to adapt is:

$$P = 2N \times M + K \times N$$

The vector which minimize the criterion function is given by:

$$\frac{-\partial V}{\partial a_{11}}, .., \frac{-\partial V}{\partial a_{NM}}, \frac{-\partial V}{\partial b_{11}}, .., \frac{-\partial V}{\partial b_{NM}}, \frac{-\partial V}{\partial c_{11}}, .., \frac{-\partial V}{\partial c_{NK}} = 0$$

And the recursive (learning) rules:

$$a_{nm}(t+1) = a_{nm}(t) - \Gamma_a \frac{\partial V(z)}{\partial a_{nm}} \quad (17)$$

$$b_{nm}(t+1) = b_{nm}(t) - \Gamma_b \frac{\partial V(z)}{\partial b_{nm}} \quad (18)$$

$$c_{nk}(t+1) = c_{nk}(t) - \Gamma_c \frac{\partial V(z)}{\partial c_{nk}} \quad (19)$$

If the membership functions of the controller are Gaussians, then the partial derivatives of the criterion V are:

$$\frac{\partial V}{\partial a_{nm}} = \sum_{k=1}^{K} (y_k - y_{dk}) \frac{u_n}{\sum_{n=1}^{N} u_n} (c_{nk} - y_k) \frac{x_m - a_{nm}}{b_{nm}^2}$$

$$\frac{\partial V}{\partial b_{nm}} = \sum_{k=1}^{K} (y_k - y_{dk}) \frac{u_n}{\sum_{n=1}^{N} u_n} (c_{nk} - y_k) \frac{(x_m - a_{nm})^2}{b_{nm}^3}$$

$$\frac{\partial V}{\partial c_{nk}} = (y_k - y_{dk}) \frac{u_n}{\sum_{n=1}^{N} u_n}$$

The adaptation of the parameters of the Gaussians and weights is done by

$$a_{nm}(t+1) = a_{nm}(t) - \Gamma_a \frac{u_n}{\sum_{n=1}^{N} u_n} \frac{x_m(t) - a_{nm}(t)}{b_{nm}(t)^2}$$
$$\sum_{k=1}^{K} (y_k(t) - y_{dk}(t))(c_{nk}(t) - y_k(t)),$$

$$b_{nm}(t+1) = b_{nm}(t) - \Gamma_b \frac{u_n}{\sum_{n=1}^{N} u_n} \frac{(x_m(t) - a_{nm}(t))^2}{b_{nm}(t)^3}$$
$$\sum_{k=1}^{K} (y_k(t) - y_{dk}(t))(c_{nk}(t) - y_k(t));$$

$$c_{nk}(t+1) = c_{nk}(t) - \Gamma_c \frac{u_n}{\sum_{n=1}^{N} u_n} (y_k(t) - y_{dk}(t))$$

Linguistic rules extraction: The problem of the linguistic rules extraction is to convert the parameters (a_{nm}, b_{nm}, c_{nk}) at the end of the adjustment to linguistic values.

To solve this problem we compare the membership functions defined by a_{nm} et b_{nm} with preset ones whose linguistic terms belong to a set of trajectories provides by an expert person.

The controller parameters are identified by an off-line procedure based on the memorized data and the initial parameters.

- Number of inputs M = 2
- Number of outputs K = 1
- Number of rules N = 21
- Learning rate $= \Gamma_a = 0.05$, $\Gamma_b = 0.05$, $\Gamma_c = 0.1$

SIMULATION

The source, the target and the obstacle positions are specified.

Test:

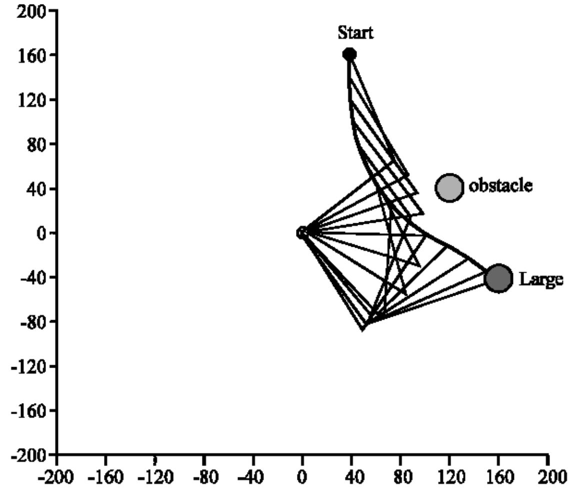

Fig. 7.1: Before training

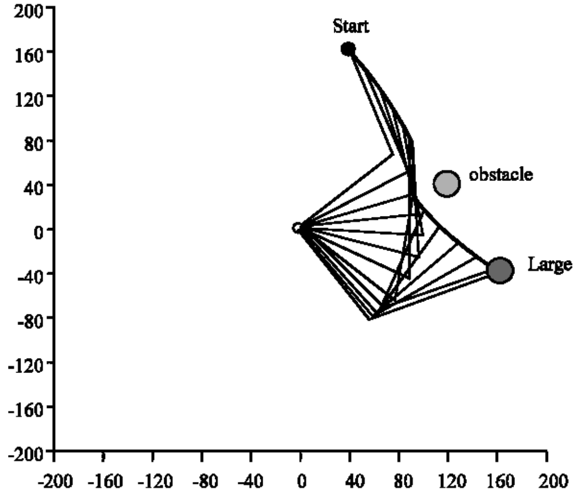

Fig. 7.2: Training

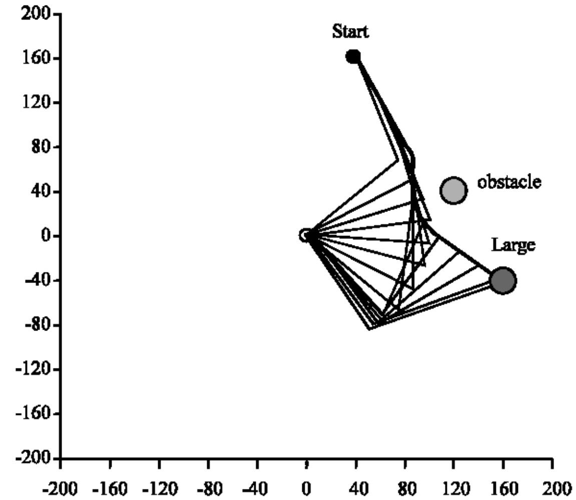

Fig. 7.3: Before training

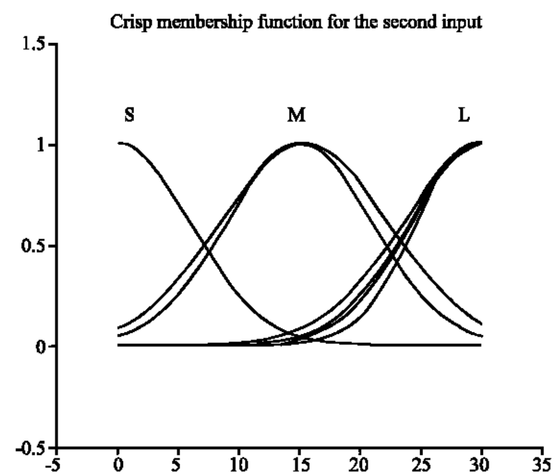

Fig. 7.4: Crisp membership functions for the second input

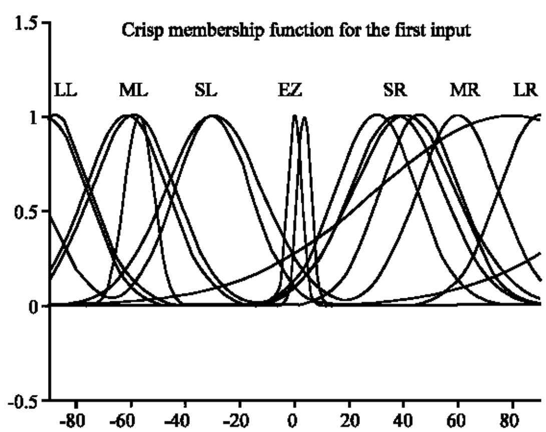

Fig. 7.5: Crisp membership functions for the first input

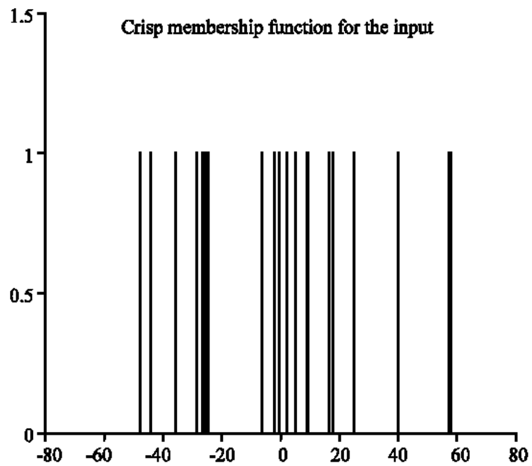

Fig. 7.6: Crisp membership functions for the output

Before training the arm moves from a start configuration to a goal configuration without collision (Fig 7.1).

Figure 7.2 shows the trajectory specified by the operator from the initial position to the final position (training).

Figure 7.3 describes the arm trajectory with collision avoidance after training.

We note that the trajectory after training is optimal compared to the one before training (Fig 7.1).

We have observed that after training, the concentration of singletons are located on left side. This is due to the avoidance of obstacle is done on the right side (Fig. 7.4, 7.5 and 7.6).

CONCLUSION

In this study, we have presented a solution to the problem of trajectory tracking without collision. The arm motion depends on the potential field approach.

The manipulator moves in a field of forces where the goal position is an attractive pole and where the obstacle is a repulsive pole. The attractive force is calculated from the goal position and the repulsive force is determined by a fuzzy logic.

The manipulator has to follow a trajectory specified by the operator from a start configuration to a goal configuration, which goes through a fixed obstacle. When a potential collision with the obstacle is detected, the collision avoidance redirects the arm motion thanks to the repulsive force in order to generate a new collision free path. The method has been tested on two-link robot arm and the results are very satisfactory.

REFERENCES

1. Kermiche, S. and H.A. Abbassi, 2002. Commande d'un bras manipulateur par logique floue. SNAS Annaba, Algeria.
2. Yan Shi and Masaharu Mizumoto, 2000. Some considerations on conventional neuro-fuzzy learning algorithms by gradient descent method. ELSEVIER Fuzzy Sets Systems, pp: 112.
3. Krzysztof kozlowski, 1998. Modelling and Identification in Robotics Springer-Verlag london limited.
4. Jelena Godjevac, 1997. Neuro-fuzzy Controllers, Design and Application, French University Polytechnic Press.
5. Foulloy, L. and S. Galichet, 1995. Typology of fuzzy controllers, A comparison of fuzzy and linear controllers.
6. Lallemand, J.P. and S. Zeghloul, 1994. Robotique Aspects fondamentaux, Masson.
7. Hansruadi Buhler, 1994. Roglage par logique floue, Presses polytechniques et universitaires Romandes.
8. Mark, W. Spong and M. Vidyasagar, 1989. Robot Dynamics and Control, John Wiley and Sons.
9. Li, W., K. Tanaka and H.O. Wang, 2004. Acrobatic control of a pendubot, IEEE transactions on fuzzy systems, 12: 549-559.
10. Khatib, O., 1986. Real time obstacle avoidance for manipulators and mobile robots, The Intl. J. Robotics Res., 5: 90-98.
11. Tao, C.W. and J.S. 2005. Taur, Robust fuzzy control for a plant with fuzzy linear model, IEEE Transactions on Fuzzy systems, 13: 30-41.
12. Lozano-Perez, T., 1983. Spatial planning: A configuration space approach, IEEE Transactions on computers C-32, pp: 108-1.20.
13. Dunlaing, C.O. and C.K. Yap, 1982. A retraction method for planning the motion of a disk, J. Algorithms.