I.J. Intelligent Systems and Applications, **2014, 11, 49-56**
Published Online October 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2014.11.07

Sequential Adaptive Fuzzy Inference System Based Intelligent Control of Robot Manipulators

Sahraoui Mustapha
RIIR Laboratory, Faculty of Applied and Exact Sciences, University of Oran, Algeria
Sahraoui.musta@gmail.com

Khelfi Mohamed Fayçal
RIIR Laboratory, Faculty of Applied and Exact sciences, University of Oran, Algeria
Mf_khelfi@yahoo.fr

Salem Mohammed
RIIR Laboratory, Faculty of Applied and Exact Sciences, University of Oran, Algeria
Salemohammed@gmail.com

***Abstract*—** The present paper is dedicated to the presentation and implementation of an optimized technique allowing an on-line estimation of a robot manipulator parameters to use them in a computed torque control. Indeed the proposed control law needs the exact robot model to give good performances. The complexity of the robot manipulator and its strong non-linearity makes it hard to know its parameters. Therefore, we propose in this paper to use neuro-fuzzy networks Sequential Adaptive Fuzzy Inference System (SAFIS) to estimate the parameters of the controlled robot manipulator.

***Index Terms*—** Artificial Intelligence; SAFIS (Sequential Adaptive Fuzzy Inference System); Neuro-Fuzzy Networks; Nonlinear System; Control; Robot Manipulator

I. INTRODUCTION

Recent years have seen rapid growth in the use of fuzzy controllers to control complex and poorly defined processes. We note that most nonlinear systems are characterized by uncertain parameters, which complicates their control to improve their performance. Especially the model of the robot manipulator is not always available and its parameters are subject to uncertainties caused by friction. These uncertainties often lead to errors and differences of control laws. In this situation is necessary to use the techniques of artificial intelligence.

The Computed Torque Control is an effective motion control strategy for robotic manipulator systems, which can ensure globally asymptotic stability. In this paper, a new approach combining Computed torque control and neuro-Fuzzy (SAFIS) is developed for trajectory tracking problems of robotic manipulator with presence of uncertainty. One solution to this problem is to introduce Neuro-Fuzzy control to ensure the robustness of the controller where the adaptability of fuzzy logic, power learning ability and generalization of neural networks are combined [1]

The neuro-fuzzy networks by their approximation property and adaptation are a major asset to be used in the control law [2],[3]. Several researchers have tried to exploit the advantages of neuro-fuzzy network to control a dynamic system and specifically in the field of robotics [4] .In this paper we use a neuro-fuzzy network Sequential Adaptive Fuzzy Inference System (**SAFIS**) to develop an adaptive control law of a nonlinear dynamic-system. It should be noted that SAFIS is a truly sequential on-line learning algorithm for the different parameters of model after we inject into a computed torque control law.

This paper is divided into five sections. Brief description of computed torque control for manipulators is presented in the second section. The Sequential Adaptive Fuzzy Inference Systems (SAFIS) is described in the third section. The proposed SAFIS based control architecture is detailed in section four .The final section is dedicated to simulations. Matlab Simulations are achieved to validate the adaptive approach SAFIS control applied to three degrees of freedom SCARA robot manipulator in a trajectory tracking control, in particular for estimating model parameters of the robot manipulators. The estimated parameters are then injected into control structures.

II. COMPUTED TORQUE CONTROL FOR ROBOT MANIPULATOR

A. Dynamic model of robot manipulator

A robot manipulator consists of a mechanical structure, usually a set of rigid bodies connected in series by joints, with an end on the ground, which is the base of the robot, and the end body or effecter's.[5][16]

The dynamic equation of a manipulator of N degrees of freedom [1] is given by:

$$\Gamma = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \Gamma_d \tag{1}$$

Where Γ is the $n\times 1$ vector of actuator joint torque, *M(q)* is the $n\times n$ symmetric positive-definite inertia matrix,

$C(q,\dot{q})$ is the $n\times 1$ vector of Coriolis and centrifugal, $G(q)$ is the $n\times 1$ vector of gravitational, $q,\dot{q},\ddot{q}$ are the joint position, velocity, and acceleration vectors, $F(\dot{q})$ is the $n\times 1$ vector of actuator joint friction forces, and n corresponds to the number of degrees of freedom of the robot.

B. Computed torque control

The computed torque control [6] is based on compensation for nonlinearities of the robot manipulator. Among the classical control laws of robotic manipulators, the computed torque control is the best in terms of results. Adaptive procedures can be used to compensate the system characteristics not considered in the dynamics modeled, such as those caused by modeling errors and friction. The controller computed torque is shown in Fig.4, is more sophisticated and precise. It uses the technique of feedback, which is to compensate for nonlinearities so that the dynamic system has a linear behavior closed loop. The computed torque control [1] given by:

$$\Gamma = M(q)\Gamma_0 + C(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) \tag{2}$$

where $q_d,\dot{q}_d$ and $\ddot{q}_d$ are respectively the vectors of desired position, desired velocity and desired acceleration. Substituting Γ in expression (1) and taking into account $M(q)$ that is a regular matrix, we have n decoupled linear systems :

$$\ddot{q} = \Gamma_0 \tag{3}$$

where Γ_0 is an auxiliary input of the select controller. A proportional derivative control (*PD*) is a typical choice for Γ_0 given by the equation:

$$\Gamma_0 = \ddot{q}_d + K_v(\dot{q}_d - \dot{q}) + K_p(q_d - q) \tag{4}$$

III. Sequential Adaptive Fuzzy Inference Systems (SAFIS)

A. SAFIS description

In this paper, a SAFIS is developed to realize a compact fuzzy system with lesser number of rules. SAFIS uses the idea of functional equivalence between a RBF neural network and a fuzzy inference system [7]. RADIAL BASIS function (RBF) networks offer an efficient mechanism for approximating complex nonlinear mappings from the input–output data. Selection of a learning algorithm for a particular application is dependent on its accuracy and speed [7],[8]. Sequential learning algorithms are better than batch learning algorithms as they do not require retraining whenever a new data is received.

SAFIS uses the Growing And pruning RBF (GAP–RBF) [8]. The SAFIS algorithm consists of two aspects, determination of the fuzzy rules and adjustment of the premise and consequent parameters in fuzzy rules [9].

SAFIS uses the concept of influence of a fuzzy rule to add and remove rules during learning. SAFIS starts with no fuzzy rules and based on the data builds up a compact rule base. The influence of a fuzzy rule is defined as its contribution to the system output in a statistical sense. The parameter adjustment is done using a winner rule strategy where the winner rule is defined as the one Closest to the input data and the parameter update is done using an EKF algorithm.

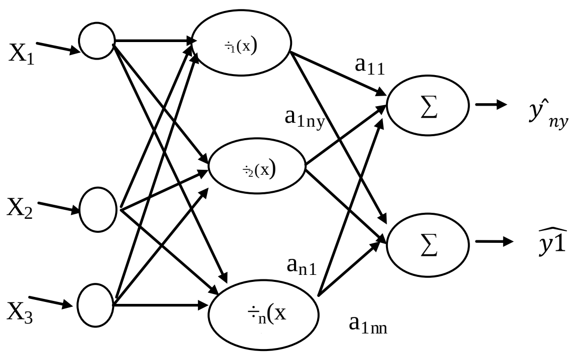

Fig. 1. Radial Basis Function Network Model

It should be noted that SAFIS is a truly sequential learning algorithm.

B. Radial Basis Function Neural Network

Radial basis function networks are a variety of artificial Neural networks (RBF), it has one hidden layer and a linear output Fig.1, $\hat{y} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_{ny}]$ which is given by (5):

$$\hat{y} = \phi_T(x)A \tag{5}$$

The vector A is the weight's vector connecting the hidden layer to the output layer. $\Phi_T(X)$ is the response of the hidden layer to the input vector $X=[1,x_1,x_2,\dots,x_{nx})$ where Φ is the Gaussian function given by [10][15]:

$$\phi(x) = \exp\left(\frac{\|x-\mu\|^2}{\sigma^2}\right) \tag{6}$$

μ is the center's vector of the hidden neurons and σ is the width of the Gaussian functions.

C.SAFIS architecture

1) Description of SAFIS architecture

Generally, a wide class of MIMO nonlinear dynamic systems can be represented by the nonlinear discrete model with an input–output description form:

$$\begin{aligned} y(n) = {} & f[y(n-1), y(n-2), ..., y(n-k+1) \\ & ...u(n), u(n-1), u(n-p+1)] \\ k = {} & 1,2,...., N_h; \end{aligned} \tag{6}$$

Where $\boldsymbol{y}$ is a vector containing N_y system outputs, $\boldsymbol{u}$ is a vector for N_u system inputs; $\boldsymbol{f}$ is a nonlinear vector function, k, p are the maximum lags of the output and input, respectively.

$$[y(n-1), y(n-2), ..., y(n-k+1); u(n), u(n-1), ... \\ u(n-p+1)], y(n) \tag{7}$$

Selecting (5) as the fuzzy system's input–output $\mathbf{x}_n$, $\mathbf{y}_n$ at time n, the above equation can be put as:

$$y_n = f(X_n) \tag{8}$$

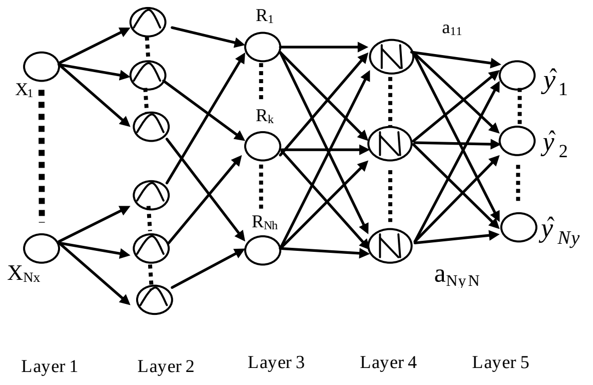

Fig. 2. SAFIS Architecture.

The aim of the SAFIS algorithm is to approximate $\boldsymbol{f}$ such that:

$$\hat{y}_n = \hat{f}(x_n) \tag{9}$$

Where $\hat{\boldsymbol{y}}_n$ is the output of SAFIS. This means that the objective is to minimize the error between the system output and the output of SAFIS, $\boldsymbol{y}_n - \hat{\boldsymbol{y}}_n$ Before describing the details of the algorithm, the structure of SAFIS network is first described below.

The structure of SAFIS illustrated by Fig. 1 consists of five layers to realize the following fuzzy rule model:

Rule k:

$$if\ (x_1 is\ A_{1k})...(x_{Nx} is\ A_{Nxk})\ then\ (\hat{y}_1 is\ a_{1k})...(\hat{y}_{Ny} is\ a_{Nyk})$$

Where a_{jk} (j=1,2….,N_y,k=1,2,..,N_h) is a constant consequent parameter in rule k. A_{ik}(i = 1, 2, . . . , Nx) is the membership value of the i^{th} input variable x_i in rule k. N_x is the dimension of the input vector x $(x=[x_1, .. ,x_{Nx}]^T)$. N_h is the number of fuzzy rules. Ny is the dimension of the output vector $\hat{y} = [\hat{y}_1 ... \hat{y}_{Ny}]^T$

In SAFIS, the number of fuzzy rules Nh varies. Initially, there is no fuzzy rule and then during learning fuzzy rules are added and removed.[9]

1) *Layer 1*: In layer 1, each node represents an input variable and directly transmits the input signal to layer 2.
2) *Layer 2*: In this layer, each node represents the membership value of each input variable. SAFIS utilizes the function equivalence between a RBF network and a FIS and thus it's antecedent part (if part) in fuzzy rules is achieved by Gaussian functions of the RBF network [11]. The membership value $A_{ik}(x_i)$ of the i th input variable x_i in the k th Gaussian function is given by

$$A_{ik}(x_i) = \exp\left(-\frac{(x_i - \mu_{ik})^2}{\sigma_k^2}\right), k = 1, 2, ..., N_h \tag{10}$$

where N_h is the number of the Gaussian functions, μ_{ik} is the center of the k th Gaussian function for the i th input variable, σ_k is the width of the k th Gaussian function. In SAFIS, the width of all the input variables in the k th Gaussian function is the same.

3) *Layer 3:* Each node in the layer represents the if part of if–then rules obtained by the sum–product composition and the total number of such rules is N_h. The firing strength (if part) of the k th rule is given by:

$$\begin{aligned} R_k(X) &= \prod_{i=1}^{N_x} A_{ik}(x_i) \\ &= \exp\left(-\sum_{i=1}^{N_x} \frac{(x_i - \mu_{ik})^2}{\sigma_k^2}\right) \\ &= \exp\left(-\frac{\|X - \mu_k\|^2}{\sigma_k^2}\right) \end{aligned} \tag{11}$$

4) *Layer 4:* The nodes in the layer are named as normalized nodes whose number is equal to the number of the nodes in third layer. The k th normalized node is given by:

$$\bar{R}_k = \frac{R_k(X)}{\sum_{k=1}^{N_h} R_k(X)} \tag{12}$$

5) *Layer 5:* Each node in this layer corresponds to an output variable, which is given by the weighted sum of the output of each normalized rule. The system output is calculated by:

$$\hat{y} = \frac{\sum_{k=1}^{N_h} R_k(X) a_k}{\sum_{k=1}^{N_h} R_k(X)} \tag{13}$$

where $\hat{\boldsymbol{y}} = [\hat{\boldsymbol{y}}_1, \hat{\boldsymbol{y}}_2, ..., \hat{\boldsymbol{y}}_{Ny}]^T, \boldsymbol{a_k} = [\boldsymbol{a}_{1k}, \boldsymbol{a}_{2k}, ..., \boldsymbol{a}_{Nyk}]^T$

2) *Influence of a fuzzy rule*

As in (13), the contribution of the k^{th} rule to the overall output for an input observation x_l is given by:

$$E(k,l) = |a_k| \frac{R_k(x_l)}{\sum_{k=1}^{N_h} R_k(x_l)} \tag{14}$$

Then the contribution of the k th rule to the overall output based on all input data N received so far is obtained by:

$$E(k) = |a_k| \frac{\sum_{l=1}^{N} R_k(x_l)}{\sum_{k=1}^{N_h}\sum_{l=1}^{N} R_k(x_l)} \tag{15}$$

Dividing the numerator and denominator by N in (14) and using the significance concept of GAP–RBF [4], the influence of the k^{th} fuzzy rule is defined as its statistical contribution to the overall output of SAFIS.

Thus, based on [8] the influence of the k^{th} rule is given by:

$$E_{\inf}(k) = |a_k| \frac{(1.8\sigma_k)^{Nx}}{\sum_{k=1}^{N_h} (1.8\sigma_k)^{Nx}} \tag{16}$$

3) Apply the criterion for adding rules

We proceed by adding a rule if the criterion as in (17) is verified.

$$\|X_n - \mu_{nr}\| > \varepsilon_n \text{ and } \quad E_{\inf}(N_h + 1) > e_g \tag{17}$$

The parameters of the allocated rule are given by:

$$\begin{cases} a_{Nh+1} = e_n \ ; \ \mu_{Nh+1} = X_n \\ \sigma_{Nh+1} = k\|X_n - \mu_{nr}\| \end{cases} \tag{18}$$

And then the Fuzzy inference system will contain *Nh + 1* rules.

4) Parameter adjustment (EKF algorithm)

If the last criterion is not satisfied, the SAFIS will only adjust the actual parameters, SAFIS utilizes a winner rule strategy similar to the work done by [11]. The key idea of the winner rule strategy is that only the parameters related to the selected winner rule are updated by the EKF algorithm in every step [13]. The winner rule is defined as the rule that is closest (in the Euclidean distance sense) to the current input data as in ([12],[8],[14])

The parameters of the allocated rule are given by:

$$\begin{aligned} \theta_n &= [\theta_1 \ldots \theta_{nr} \ldots \theta_{Nh}]^T \\ &= [a_1, \mu_1, \sigma_1, \ldots.., a_{nr}, \mu_{nr}, \sigma_{nr}, \ldots., a_{Nh}, \mu_{Nh}, \sigma_{Nh}]^T \end{aligned} \tag{19}$$

Where $\theta_{nr} = [a_{nr}\ , \mu_{nr}\ , \sigma_{nr}\ ]$ is the parameter vector of the nearest fuzzy rule and its gradient is derived as follows:

$$\dot{a}_{nr} = \frac{\partial \hat{y}_n}{\partial a_{nr}} = \frac{R_{nr}}{\sum_{k=1}^{Nh} R_k} \tag{20}$$

$$\dot{\mu}_{nr} = \frac{\partial \hat{y}_n}{\partial \mu_{nr}} = \frac{a_{nr} - \hat{y}_n}{\sum_{k=1}^{Nh} R_k} 2R_{nr} \frac{x_n - \mu_{nr}}{\sigma_{nr}^2} \tag{21}$$

$$\dot{\sigma}_{nr} = \frac{\partial \hat{y}_n}{\partial \sigma_{nr}} = \frac{a_{nr} - \hat{y}_n}{\sum_{k=1}^{Nh} R_k} 2R_{nr} \frac{\|x_n - \mu_{nr}\|^2}{\sigma_{nr}^3} \tag{22}$$

After obtaining the gradient vector of the nearest fuzzy rule $B_{nr} = [\dot{a}_{nr}, \dot{\mu}_{nr}, \dot{\sigma}_{nr}]^T$, EKF is used to update its parameters as follows:

$$K_n = P_{n-1} B_n [R_n + B_n^T P_{n-1} B_n]^{-1} \tag{23}$$

$$\theta_n = \theta_{n-1} + K_n e_n \tag{24}$$

$$P_n = [I_{z*z} - K_n B_n^T] P_{n-1} + q I_{z*z} \tag{25}$$

where q is a scalar that determines the allowed step in the direction of the gradient vector, Z is the dimension of parameters to be adjusted. When a new rule is added, the dimension of Pn increases to:

$$\begin{pmatrix} P_{n-1} & 0 \\ 0 & p_0 I_{z1*z1} \end{pmatrix} \tag{26}$$

Where *Z1* is the dimension of the parameters introduced by the newly added rule, p_0 is an initial value of the uncertainty assigned to the newly allocated rule.

5) Pruning the rules

We proceed by adding a rule if the criterion as in (27) is verified

$$E_{\inf}(nr) = |a_{nr}| \frac{(1.8\sigma_{nr})^{N_x}}{\sum_{k=1}^{N_h+1} (1.8\sigma_k)^{N_x}} < e_p \tag{27}$$

Remove the *nr* rule, Reduce the dimensionality of EFK as in (26) [4]

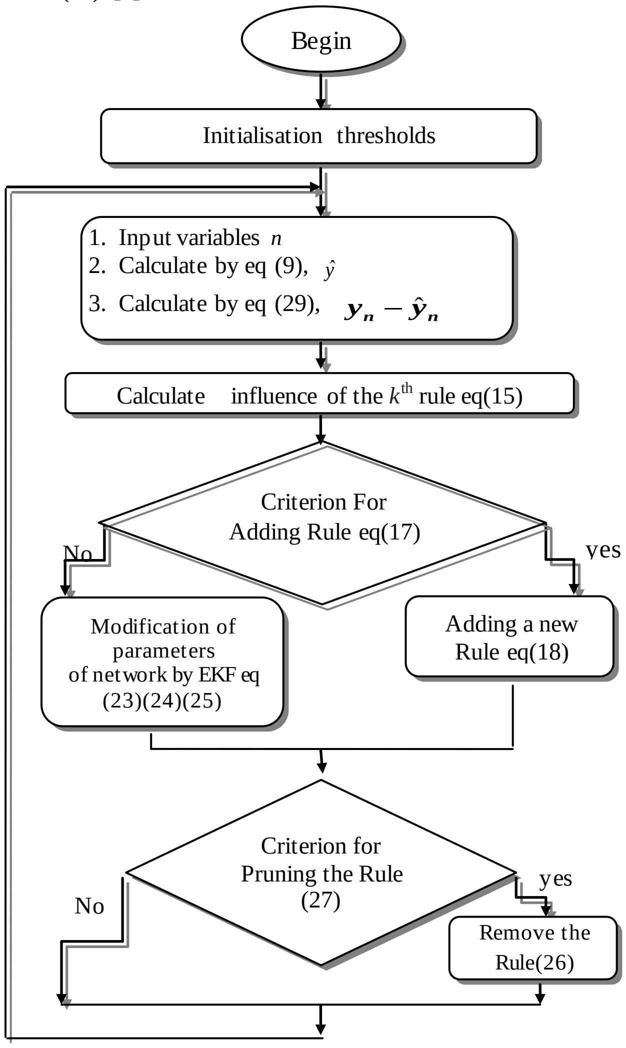
Begin
Initialisation thresholds
1. Input variables n
2. Calculate by eq (9), ŷ
3. Calculate by eq (29), $y_n - \hat{y}_n$
Calculate influence of the k^{th} rule eq(15)
Criterion For Adding Rule eq(17)
No
yes
Modification of parameters of network by EKF eq (23)(24)(25)
Adding a new Rule eq(18)
Criterion for Pruning the Rule (27)
No
yes
Remove the Rule(26)

Fig. 3. SAFIS flowchart

D.SAFIS algorithm

The learning algorithm of SAFIS consists of two aspects; determination of fuzzy rules and adjustment of the premise and consequent parameters in fuzzy rules as in Fig.3, SAFIS can automatically add and remove fuzzy rules using ideas similar to GAP–RBF [4] for hidden neurons.

Given the growing and pruning thresholds *eg*, *ep*, for each observation $(\mathbf{x}_n, \mathbf{y}_n)$, where $\mathbf{x}_n \in R^{Nx}$, $\mathbf{y}n \in R^{Ny}$ and $n = 1, 2,..$ do:

Compute the overall system output given by (13) with

$$R_k(X_n) = \exp(-\frac{1}{\sigma_k^2}\|X_n - \mu_k\|^2) \tag{28}$$

Where N_h is the number of fuzzy rules Calculate the parameters required in the growth criterion:

$$\begin{aligned} \varepsilon_n &= \max\left\{\varepsilon_{\max}\gamma^n, \varepsilon_{\min}\right\}, \quad (0<\gamma<1) \\ e_n &= y_n - \hat{y}_n \end{aligned} \tag{29}$$

If the criterion for adding is verified as in (17) a new rule is added with new parameters as in (18).

Check the criterion for pruning the rule as in (27), if is verified then remove the nr rule and reduce the dimensionality of EKF. [8][16].

IV. SAFIS Based Computed Torque Control Of Robot Manipulateur

In this work, SAFIS based control architecture is proposed to overcome the computed torque control lack which is the difficulty to estimate online the model parameters of the robot manipulator, so to compensate uncertainties and noise.

The idea is to estimate the model matrices by a SAFIS networks.

The control law in (2) becomes:

$$\Gamma = \hat{M}(q)\Gamma_0 + \hat{C}(q,\dot{q})\dot{q} + \hat{G}(q) + \hat{F}(\dot{q}) \tag{30}$$

Where $(\hat{M}, \hat{C}, \hat{F})$ is SAFIS estimation of *M, C, G* and *F* respectively. The:

$$\begin{aligned} \hat{M} &= \phi(q) \\ \hat{C} &= \varphi([q,\dot{q}]) \\ \hat{F} &= \delta(q) \\ \hat{G} &= \theta(q) \end{aligned}$$

Where $\emptyset, \varphi, \delta, \theta$ are SAFIS network with the outputs $q, \dot{q}$.

The global SAFIS intelligent control scheme is presented in Fig.4

V. Simulation Results

To validate the proposed intelligent control approach, we have carried out simulations for adaptive control of 3 degrees of freedom SCARA robot manipulator whose dynamic model is given by (32, 33, 34,35, 36,)

Due to the complexity of the robot manipulator and its strong non-linearity make using a single network very difficult to achieve. The model matrices (Inertia matrix, Coriolis matrix and friction vector) are estimated using three SAFIS networks.

Before proceeding with the proposed approach, the algorithm parameters needed for the accomplishment of learning are initialized in Table 1.

Table 1. SAFIS initialization parameter

*E*g (pruning thresholds)	0.001
e$_p$ pruning thresholds	0.001
factor of overlap	[1.0,2.0]

$$u(t) = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + f(\dot{q}) + t_1(t) \tag{31}$$

Where

$$M(q) = \begin{cases} M_{11} = 2.1240 + 1.44cos(q_2) \\ M_{12} = -0.4907 - 0.72cos(q_2) \\ M_{13} = 0 \\ M_{21} = -0.4907 - 0.72cos(q_2) \\ M_{22} = -0.4907, M_{23} = 0 \\ M_{31} = M_{32} = M_{33} = 0 \end{cases} \tag{32}$$

$$G(q) = \begin{cases} G_1 = 0 \\ G_2 = 0 \\ G_3 = -4.9 \end{cases} \tag{34}$$

$$F(\dot{q}) = \begin{cases} F_1 = 12\dot{q}_1 + 0.02sign(\dot{q}_1) + \\ \quad 3sin(3t) \\ F_2 = 12\dot{q}_2 + 0.02sign(\dot{q}_2) + \\ \quad 3sin(3t) \\ F_3 = 12\dot{q}_3 + 0.02sign(\dot{q}_3) + \\ \quad 3sin(3t) \end{cases} \tag{35}$$

$$T_1(t) = \begin{bmatrix} 5sin(2t) \\ 5sin(2t) \\ 5sin(2t) \end{bmatrix} \tag{36}$$

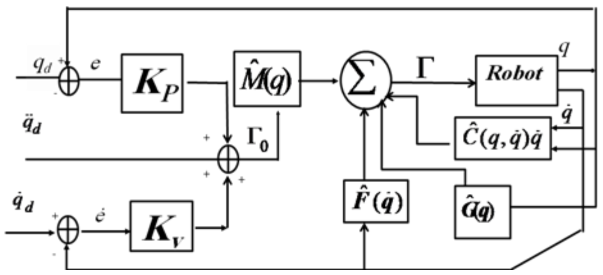

Fig. 4. Bloc diagram of computed torque control with estimation parameters by SAFIS system

After having initialized the thresholds of the learning algorithm using table.1 for the three networks SAFIS, we have injected into a computed torque control law of a SCARA robot manipulator with path tracking purpose (q_{d1}=1.2 Rad, q_{d2}=1.5 Rad, q_{d3}=0.1m).

To evaluate the performances of the proposed approach, we will start by using it in an ideal environment (without noise and uncertainties. In the next subsection, we injected some noise in the control loop and have altered the joint masses in the Inertia matrix (Δm_1=0.002Kg, Δm_2=0.001Kg, Δm_3=0.001Kg,),

Adaptive control based SAFIS was used to control the robot manipulator with a target of prosecution.

A. SAFIS Control without uncertainties

In the first step, the SAFIS neuro-fuzzy networks is used to estimate the parameters of the robot manipulator (Matrices *M*, *C and F)* without uncertainties parametric.

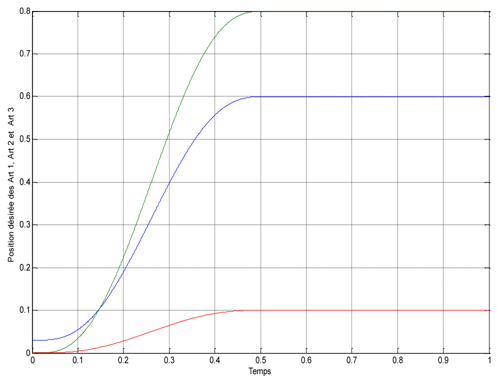

Fig. 5. desired positions

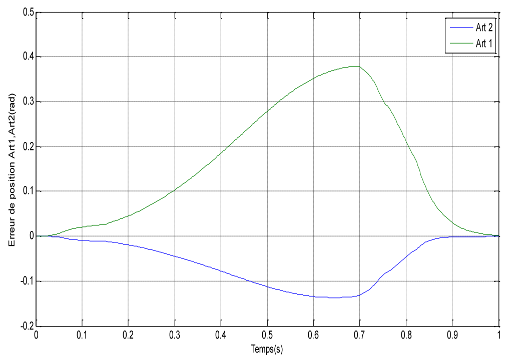

Fig. 6. Tracking error position for articulation 1,2(rad)

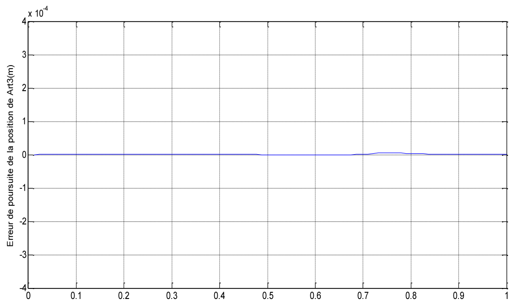

Fig. 7. Tracking error position for articulation 3(m)

Fig.6 present the position tracking errors for articulations 1 and 2, the third in fig.7

The velocity error for articulation 1and 2 is presented in fig.8, so in fig.9 presented error velocity for the third articulation.

We see that the robot joints converge quickly to the target (0.9 seconds for joints 1 and 2 and 0.01 second for joint 3). These results are obtained thanks to the good estimation of the SAFIS networks and its online learning.

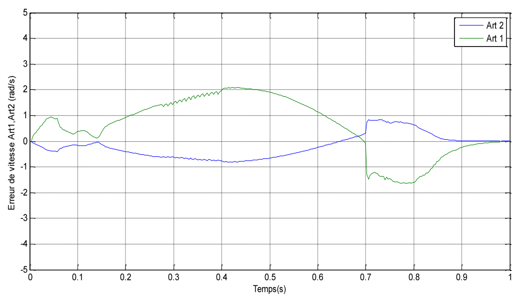

Fig. 8. Velocity error for articulation 1,2(rad/s)

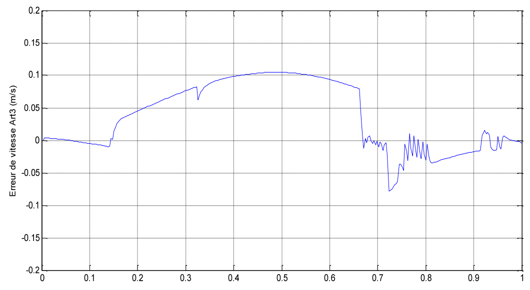

Fig. 9. Velocity error for articulation 3(m/s)

B. Simulation with uncertainties and noise

We note that most nonlinear systems are characterized by uncertain parameters, which complicates their control to improve their performance. Especially the model of the robot manipulator is not always available and its parameters are subject to uncertainties caused by friction for example. These uncertainties often lead to errors and differences of control laws.

In this subsection and to test the robustness of the proposed control, we have introduced uncertainties in the parameters of the robot manipulator, where we have altered the components of the inertia matrix (q_{d1}=1.2 Rad, q_{d2}=1.5 Rad, q_{d3}=0.1m).

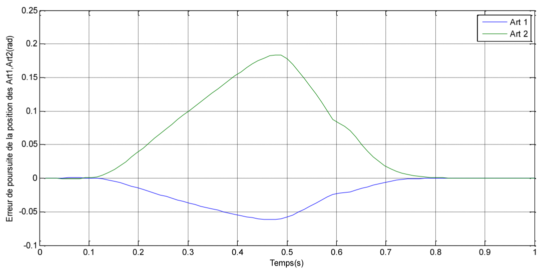

Fig. 10. Tracking error position for articulation 1,2(rad)

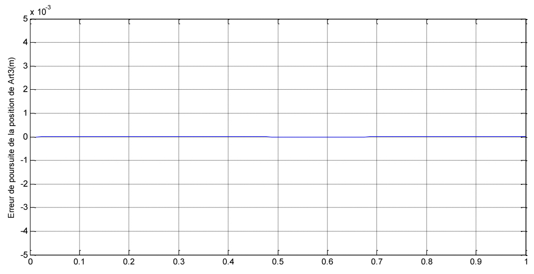

Fig. 11. Tracking error position for articulation 3(m)

In addition to the uncertainties, A white noise is introduced in the control loop.

The position errors are shown in Fig 10, and Fig 11. The velocity errors is illustrate in Fig 12 and Fig 13.

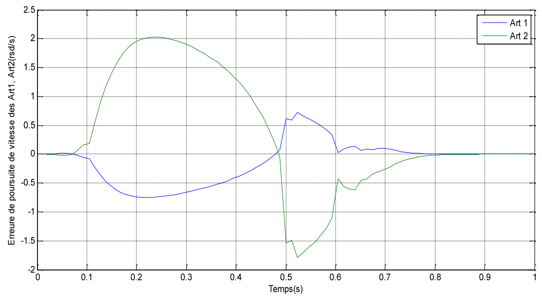

Fig. 12. Velocity error for articulation 1,2(rad/s)

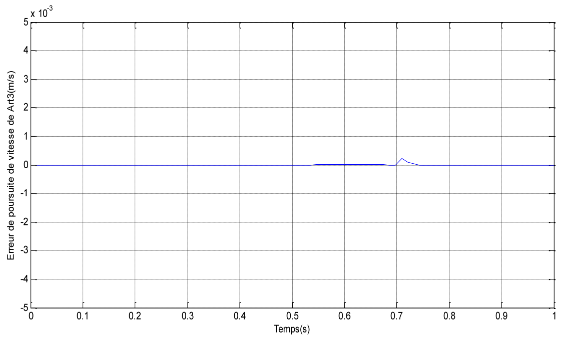

Fig. 13. Velocity error for articulation 3(m/s)

We see that errors are acceptable even a white noise is present and some uncertainties in inertia matrix.

The presence of uncertainties did not affect the control structure performances where they are compensated by the learning power of the SAFIS networks

VI. Conclusion

The robot manipulators are often subject to terms of friction and model variation, so it is very hard to get a prior knowledge of their parameters needed by the computed torque control. We presented an online SAFIS based estimation technique of robot model in order to inject it in the correspondent control law. The proposed approach consists of a self adaptive fuzzy inference systemof five layers.

In this paper, a sequential fuzzy inference system called SAFIS is developed based on the functional equivalence between a RBF (in this case, GAP–RBF network) and a FIS. In SAFIS, the fuzzy rules are added or removed based on the concept of "influence" of the rule. When there are no additions, only the parameters of the "closest" rule are updated using an EKF scheme. SAFIS is truly a sequential learning algorithm and produces a compact fuzzy inference system.

The performance of SAFIS has been proved with existing of noise and uncertainties on nonlinear systems. The simulations over three degrees of freedom SCARA robot manipulators had given good results in terms of tracking errors even in presence of noise due to the good estimation power of the SAFIS system.

References

[1] Y. Gao, M. J. Er, W. E. Leithead, and D. J. Leith, "On-line adaptive control of robot manipulators using dynamic fuzzy neural networks", *Proc. American Control Conf.*, pp.4828 -4833 2001

[2] Q. H. M. Meng, "Application of an efficient neural network to identification and adaptive control of unknown robot dynamics", *Int. J. Intell. Control Syst.*, vol. 1, no. 4, pp.459 -468 1996

[3] Y. Gao and M. J. Er, "Robust adaptive fuzzy neural control of robot manipulators", *Proc. IEEE-INNS Int. Joint Conf. Neural Networks*, pp.2188 -2193 2001

[4] H. Maaraf, "Notion de base de la théorie de flou", Université d'Evry Val d'Essonne, 2002.

[5] Q. H. M. Meng, "Application of an efficient neural network to identification and adaptive control of unknown robot dynamics", *Int. J. Intell. Control Syst.*, vol. 1, no. 4, pp.459 -468 1996

[6] F.L Lewis, , C. T. Abdallah, and D. M. Dawson , "Control of Robot Manipulators", Macmillan Publishing Company, Inc, USA, 1993

[7] K.B. Cho, B.H.Wang, "Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction", Fuzzy Sets and Systems 83 (1996) 325–339

[8] G.B. Huang, P. Saratchandran and N. Sundararajan, "A Generalized Growing And Pruning RBF(GGAP-RBF) Neural Network for Function Approximation", IEEE Trans. On Neural Networks, 2004

[9] H. Jun Rong, N. Sundararajan∗, Guang-Bin Huang, P. Saratchandran "Sequential Adaptive Fuzzy Inference System (SAFIS) for nonlinear system identification and prediction", IEEE Trans, Nanyang Technological University, Singapore.2006

[10] S. Haykin, Neural Networks: A Comprehensive Foundation. Upper Saddle River, N.J.: Prentice Hall, 1999. 2nd Edition

[11] L.Yingwei, N. Sundararajan, and P. Saratchandran, “Performance evaluation of a sequental minimal radial basis function (RBF) neural network learning algorithm,” *IEEE Trans. Neural Networks*, vol. 9, pp. 308–318,Mar. 1998

[12] T. Takagi, M. Sugeno, “Derivation of fuzzy control rules from human operator’s control actions”, in: Proceedings of the IFAC Symposium On Fuzzy Information, Knowledge Representation and Decision Analysis, pp. 55–60, July 1983

[13] G. V. Puskorius and L. A. Feldkamp, “Neuro-control of nonlinear dynamics systems with Kalman filter trained recurrent networks,” *IEEE Trans. Neural Networks*, vol. 5, pp. 279–297, Mar. 1994.

[14] D.Simon, “Training radial basic neural networks with the extended Kalmar filter”, Neuro-computing 48 ,2002 ,pp.455–475

[15] F.Piltan, M.Eram, M.Taghavi,O. Sadrni and M.Jafari “Nonlinear Fuzzy Model-base Technique to Compensate Highly Nonlinear Continuum Robot Manipulator” *International Journal of Intelligent Systems and Applications (IJISA)*,PP.135-148, ijisa.2013.

[16] S. Bangia, PR Sharma, M. Garg ’’Simulation de Fuzzy Logic Basé Shunt hybride Filtre actif pour l'amélioration de la qualité de puissance’’, *International Journal of Intelligent Systems and Applications (IJISA)*, vol 5, PP.96-104, Jan 2013

Authors’ Profiles

Mustapha Sahraoui was born in Mascara, Algeria in 1975; he received an Engineer degree in computer science from Sidi Bel Abbes University Algeria on 1999 and Master degree in Industrial computing from MASCARA University on 2011.

His areas of current research are the application of artificial intelligence in the control and identification of nonlinear systems and the evolutionary computation.

He joined the Science & technology Faculty (Computer science department) University of Mascara, Algeria; he is also a Research Member of the Laboratory of Research in Industrial Computing and Networks University of Oran

Mohamed Fayçal Khelfi was born in Algiers, Algeria in 1965. He received Ph.D. degree in Automatic Control from Nancy University, France, in 1995. He is currently Professor at the Computer Science Department - Faculty of Exact and Applied Sciences - University of Oran - Algeria. He is also a research member at the Laboratory of Research in Industrial Computing and Networks. His main research interests include Automatic Control, Industrial Computing, Robotics and Networks.

Mohammed Salem was born in Mascara, Algeria in 1974; he received an Engineer degree in computer science from Sidi Bel Abbes University Algeria on 1999 and Master degree in Industrial computing from MASCARA University on 2007.

His research fields are the evolutionary computation and the application of artificial intelligence in identification and control of nonlinear systems.

He joined the Science & technology Faculty (Computer science department) University of Mascara, Algeria; he is also a Research Member of the Laboratory of Research in Industrial Computing and Networks University of Oran.

How to cite this paper: Sahraoui Mustapha, Khelfi Mohamed Fayçal, Salem Mohammed,"Sequential Adaptive Fuzzy Inference System Based Intelligent Control of Robot Manipulators", International Journal of Intelligent Systems and Applications(IJISA), vol.6, no.11, pp.49-56, 2014. DOI: 10.5815/ijisa.2014.11.07