We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

5,800
Open access books available

142,000
International authors and editors

180M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Compensatory Adaptive Neural Fuzzy Inference System

Rabah Mellah, Hocine Khati, Hand Talem and Said Guermah

Abstract

The traditional approach to fuzzy design is based on knowledge acquired by expert operators formulated into rules. However, operators may not be able to translate their knowledge and experience into a fuzzy logic controller. In addition, most adaptive fuzzy controllers present difficulties in determining appropriate fuzzy rules and appropriate membership functions. This chapter presents adaptive neural-fuzzy controller equipped with compensatory fuzzy control in order to adjust membership functions, and as well to optimize the adaptive reasoning by using a compensatory learning algorithm. An analysis of stability and transparency based on a passivity framework is carried out. The resulting controllers are implemented on a two degree of freedom robotic system. The simulation results obtained show a fairly high accuracy in terms of position and velocity tracking, what highlights the effectiveness of the proposed controllers.

Keywords: control, fuzzy logic, neural-fuzzy, compensatory fuzzy, Kalman filter, manipulator robot

1. Introduction

The advantage of fuzzy control is that the fuzzy system can model any continuous (sufficiently smooth) nonlinear function in a compact set and the modeling error decreases [1]. Fuzzy logic resembles human analysis in its use of inaccurate information to create decisions. Many such problems can be formulated as the minimization of functional defined over a class of admissible domains [2]. However, the difficulty in deploying fuzzy clustering strategies along with the high calculating cost and without update the parameters were their disadvantage [3].

On the other hand, a major concern of researchers is turned towards the combination of fuzzy logic and neural network. In this combination, a fuzzy reasoning is followed within multilayered hierarchical neural network. The parameters are represented by connection weights or involved in unit functions. They are learned the actual data [4]. In the near past, ANFIS (Adaptive Neural Fuzzy Inference System) models have become very popular for two reasons: the first reason is that in calibrating of non-linear relationships they offer more advantages over conventional modeling techniques, namely the ability to handle large amounts of noisy data from dynamic and non-linear systems, particularly when the underlying physical relationships are not fully understood. The second reason is that they facilitate the solving of linear systems which include the interpolation modeling such as time series [5].

The reason why authors used ANFIS is that it not only includes the characteristics of both methods but also eliminates some disadvantages in case of their lonely use [6]. Unfortunately, conventional neural fuzzy systems can only optimize the fuzzy membership functions under specially defined fuzzy operators which are unchangeable forever, which makes it use the local optimization technique rather than the global optimization technique [7–8]. Thus an adaptive neural fuzzy controller with compensatory fuzzy is most suitable in an environment where system dynamics change dramatically, become highly nonlinear, and in principle not fully known.

On the ground of these observations, several optimal and systematic methods have been developed for the design of neural fuzzy controllers with compensatory fuzzy. Among these methods, we have retained the compensatory adaptive neural fuzzy inference system approach which consists in adjusting not only fuzzy membership functions but also dynamically optimize the adaptive fuzzy reasoning. Besides that, ANFIS is a class of adaptive networks that are functionally equivalent to a first order Takagi-Sugeno fuzzy model.

Recently compensatory adaptive neural fuzzy inference system control has gained more attention from the control Community in general, as adaptive fuzzy systems are of crucial importance in several areas. The compensatory adaptive neural fuzzy inference system is preferred to deal with nonlinearities and complexity by working on data characterized by incompleteness and inaccuracy. Therefore, it offers powerful skills, such as adaptive adjustment, parallelism, tolerance error and generalization for the neural fuzzy controller. Thus the optimal methods are used to adjust and optimize the parameters of neural fuzzy controllers through an optimization algorithm in order to improve the control performance [9].

In this chapter, we will present and analyze in section 2 the structure of adaptive neural fuzzy inference system (ANFIS), based on concepts such as fuzzy logic, optimization techniques. This approach is carried out in order to remove a control constraint relating to the need to have a model as faithful as possible, knowing that the modeling errors and the imperfections of the models, contribute to significantly degrade the performance of the conventional control laws [10].

Section 3 presents the mathematical formalism appropriate to the compensatory neural fuzzy inference system controller proposed. The effectiveness of the proposed control is highlighted by some simulation results in Section 4. Finally this paper is concluded with a summary and an outlook to future research directions in Section 5.

2. Presentation of adaptive neural fuzzy inference system (ANFIS)

Jang was the first to present ANFIS as a multi-layer adaptive network-based fuzzy inference system [11]. One can compare this method to a fuzzy inference system besides that it uses back- propagation in minimizing the errors. The operation of a FIS is similar to that of both fuzzy logic (FL) and artificial neural networks (ANN). ln both (ANN) and (FL), the input passes through the input layer (via the input membership function) and the output appears in output layer (via the output membership function). This type of advanced fuzzy logic uses neural networks. Hence, a learning algorithm can be used to change the parameters until an optimal solution is found. It follows that ANFIS uses either back-propagation or a combination of least squares estimation and back-propagation to estimate the membership function parameters [12]. Neural-Fuzzy system has newly known more attraction in research communities than other types of fuzzy expert systems. The reason of it combines the advantages of learning ability of neural network and reasoning ability

of fuzzy logic to successfully solve many non-linear and complex real-world problems [13].

The regulator ANFIS is computationally very efficient, as it favors mathematical analysis, and works well with linear, adaptive, and optimization techniques. The fuzzy reasoning is performed with operators min and prod [14].

The conclusions of fuzzy rules are numeric values calculated from the inputs, so the final value is obtained by performing a weighted average of the conclusions [15, 16].

To simplify understanding and without loss of generalities, let us consider a fuzzy regulator with two inputs e_1 and e_2 and one output u. The entry x_1 is associated with two fuzzy sets A_1 and A_2. As for the two fuzzy sets associated with the second entry x_2 are B_1 and B_2. The output u is modeled by a fuzzy Sugeno-type system, composed of the following four rules [17]:

$$\text{Rule 1}: \text{if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } u_1 = f_1(x_1, x_2) = a_1 x_1 + b_1 x_2 + c_1 \quad (1)$$

$$\text{Rule 2}: \text{if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_2 \text{ then } u_2 = f_2(x_1, x_2) = a_2 x_1 + b_2 x_2 + c_2 \quad (2)$$

$$\text{Rule 3}: \text{if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_1 \text{ then } u_3 = f_3(x_1, x_2) = a_3 x_1 + b_3 x_2 + c_3 \quad (3)$$

$$\text{Rule 4}: \text{if } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } u_4 = f_4(x_1, x_2) = a_4 x_1 + b_4 x_2 + c_4 \quad (4)$$

Let us denote $O_{k,i}$ the node in the *ith* position of the *kth* layer. The node functions in the same layer are of the same function family as defined below.

The input layer is denoted Layer 1 and any node i in this layer is a square node with a node function that describes the membership function. Hence $O_{1,i}$ is the membership function of A_i, and it specifies the degree to which a given variable x satisfies its quantifier A_i. We select the membership function in such a way the maximum of which is equal to unity and the minimum equal to zero.

The structure of the regulator ANFIS is given by the following figure (**Figure 1**):
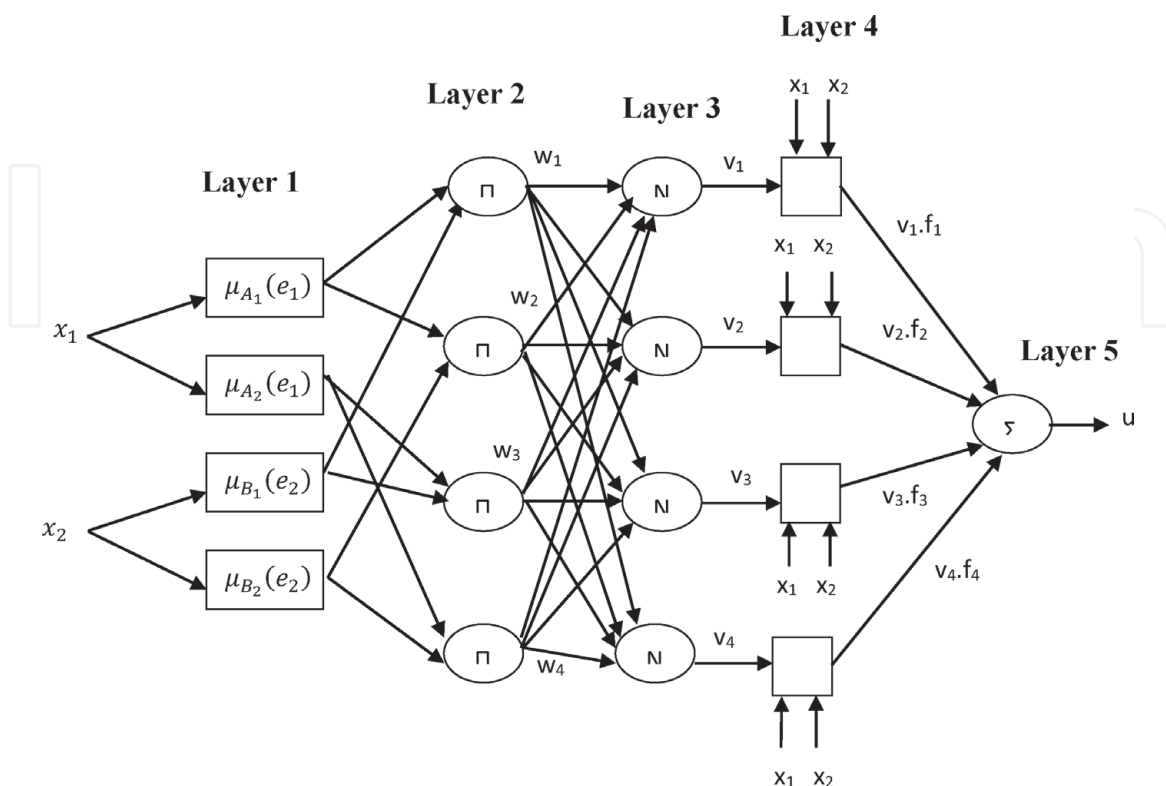

Figure 1.
Structure of the regulator ANFIS.

Through this structure we can see five layers described as follows:

Layer 1: The function of node at this layer is identical to the membership function in the fuzzification process:

Layer 2: generate the degree of activation of a rule.

$$\begin{cases} O_{2,1} = O_{1,1}.O_{1,3} = w_1 \\ O_{2,2} = O_{1,1}.O_{1,4} = w_2 \\ O_{2,3} = O_{1,2}.O_{1,3} = w_3 \\ O_{2,4} = O_{1,2}.O_{1,4} = w_4 \end{cases} \tag{5}$$

Layer 3: Each node of this layer is a circular node denoted by N. The output node represents the normalized activation degree according to the *i*th rule.

$$O_{3,i} = \frac{w_i}{\sum_{k=1}^{4} w_k} \quad \text{for i} = 1, 4 \tag{6}$$

Layer 4: Each node of this layer is a square node with a function described as follows:

$$O_{4,i} = O_{3,i} f_i = v_i(a_i e + b_i \Delta e + c_i) \tag{7}$$

where v_i is the output of the node I of layer 3 and $\{a_i, b_i, c_i\}$ is the set of update parameters.

Layer 5: In this layer, there is only one node that determines the overall output by using the following expression:

$$O_{5,i} = \sum_i O_{4,i} f_i \quad \text{for i} = 1..4 \tag{8}$$

Considering x_1 and x_2 are the position error e and its derivative Δe: $[x_1, x_2] = [e, \Delta e]$. We associate two fuzzy sets for each of the inputs x_1 and x_2 namely N (*Negative*) and P (*Positive*). μ_N and μ_P represent the degrees of membership appropriate to variables x_i with respect to the fuzzy subsets A_i and B_i, defined by the following membership functions (**Figure 2**) [17]:

For $i = 1, 2$

$$\mu_N(x_i) = \begin{cases} 1, & \text{if } x_i < -1 \\ -0.5x_i + 0.5, & \text{if } -1 < x_i < 1 \\ 0, & \text{if } x_i > 1 \end{cases} \tag{9}$$
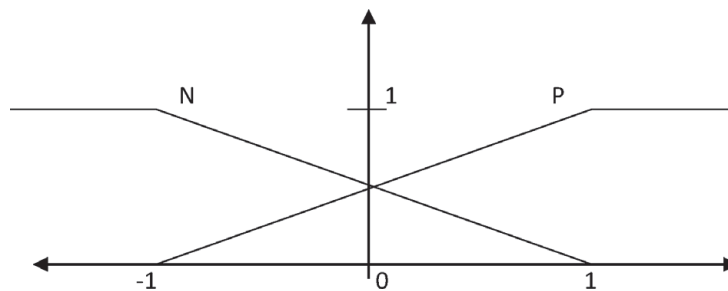

Figure 2.
Membership functions.

$$\mu_P(x_i) = \begin{cases} 0, & if \ x_i < -1 \\ 0.5x_i + \ 0.5, & if \ -1 < x_i < 1 \\ 1, & if \ x_i > 1 \end{cases} \tag{10}$$

2.1 Learning algorithm

The learning process consists of identifying the consequence parameters a_i, b_i and c_i for $i = 1,2,\ldots.4$. Thus, let us assume y_d and y are respectively the desired and actual outputs of system. In this work, we consider that the consequence parameters are adjusted by the minimization of the following objective function:

$$e(k) = \frac{1}{2}(e)^2 \tag{11}$$

where $e = y_d - y$.

In addition, let be Φ_i the vector of parameters to be adjusted. Our objective is to find the parameters a_i, b_i and c_i of the vector Φ_i using the gradient descent method combined with the approach of extended Kalman filter. This is equivalent to writing:

$$\Phi_i(k+1) = \Phi_i(k) - \alpha(k)\frac{\partial J}{\partial \Phi_i} \tag{12}$$

We have:

$$\frac{\partial J}{\partial \Phi_i} = -e\frac{\partial y}{\partial \Phi_i} = -e\frac{\partial y}{\partial u}\frac{\partial u}{\partial \Phi_i} \tag{13}$$

From Eqs. (12) and (13), it follows:

$$\Phi_i(k+1) = \Phi_i(k) + \alpha(k)\frac{\partial y}{\partial u}\frac{\partial u}{\partial \Phi_i}e \tag{14}$$

In our case, $\frac{\partial y}{\partial u}$ cannot be evaluated, but can be estimated using the extended Kalman filter equations. Consequently, Eq. (14) can be written as:

$$\Phi_i(k+1) = \Phi_i(k) + K'\Psi_i e \tag{15}$$

where

$$K' = \alpha(k)\frac{\partial y}{\partial u} \tag{16}$$

$$\Psi_i = \frac{\partial u}{\partial \Phi_i} = \begin{bmatrix} \frac{\partial u}{\partial a_i} \\ \frac{\partial u}{\partial b_i} \\ \frac{\partial u}{\partial c_i} \end{bmatrix} \tag{17}$$

The Eq. (15) can be identified to extended Kalman filter equation:

$$\Phi_i(k+1) = \Phi_i(k) + K(k)e \tag{18}$$

Where $K(k)$ is the Kalman gain defined as follows:

$$K(k) = \frac{P(k)H^T(k)}{H(k)P(k)H^T(k) + R(k)} \tag{19}$$

Where $H(k)$ is the Jacobian matrix (observation matrix of the system); $P(k)$ is the covariance estimation matrix of the error and is the covariance matrix of the process noise.

Taking $H(k) = (\Psi_i)^T, P(k) = \lambda_1$ and $R(k) = \lambda_2$, the gain $K(k)$ can be written:

$$K(k) = \frac{\lambda_1}{(\Psi_i)^T \lambda_1 (\Psi_i) + \lambda_2}(\Psi_i) = \frac{\lambda_1}{\lambda_1 (\Psi_i)^T (\Psi_i) + \lambda_2}(\Psi_i) \tag{20}$$

Hence Eq. (18) reduces:

$$\Phi_i(k+1) = \Phi_i(k) + \frac{\lambda_1}{\lambda_1 (\Psi_i)^T (\Psi_i) + \lambda_2}(\Psi_i)e \tag{21}$$

By identification between Eqs. (15) and (21), we have:

$$K' = \frac{\lambda_1}{\lambda_1 (\Psi_i)^T (\Psi_i) + \lambda_2} \tag{22}$$

Finally, the vector of consequence parameters Φ_i can be adjusted by the following relation:

$$\Phi_i(k+1) = \Phi_i(k) + K'(\Psi_i)e \tag{23}$$

2.2 Stability analysis of the control system

From Eq. (23), we can consider for a very short time T_e, this relation:

$$\dot{\Phi}_i = \frac{\Phi_i(k+1) - \Phi_i(k)}{T_e} = \frac{K'(\Psi_i)e}{T_e} = K_1(\Psi_i)e \tag{24}$$

Where $K_1 = \frac{K'}{T_e}$
Hence:

$$\dot{\Phi}_i = K_1(\Psi_i)e_u \tag{25}$$

Where $e_u = K_1 e$ is the error between the controller's desired output u_d and actual output u.

Let be $\tilde{\Phi}_i = \Phi_{id} - \Phi_i$, where Φ_i is the vector of the consequence parameters and Φ_{id} the vector of the desired consequence parameters.

$$\dot{\tilde{\Phi}}_i = \dot{\Phi}_{id} - \dot{\Phi}_i = -(\Psi_i)e_u \tag{26}$$

For linear variation, the error e_u is defined by [18]:

$$e_u = u_d - u = \sum_{i=1}^{4}\left((\Psi_i)^T \Phi_{id} - (\Psi_i)^T \Phi_i\right) = \sum_{i=1}^{4}\left((\Psi_i)^T(\Phi_{id} - \Phi_i)\right) = \sum_{i=1}^{4}\left((\Psi_i)^T \tilde{\Phi}_i\right) \tag{27}$$

Consider the following Lyapunov function [19–21]:

$$V = \frac{1}{2} \sum_{i=1}^{4} \left((\tilde{\Phi}_i)^T (\tilde{\Phi}_i) \right) \tag{28}$$

Differentiating V with respect to time yields, we obtain [17]:

$$\dot{V} = \sum_{i=1}^{4} \left(\left(\dot{\tilde{\Phi}}_i \right)^T (\tilde{\Phi}_i) \right) \tag{29}$$

From Eqs. (26), (27) and (29), we obtain:

$$\dot{V} = \sum_{i=1}^{4} \left((-(\Psi_i)e_u)^T (\tilde{\Phi}_i) \right) = -(e_u)^T \sum_{i=1}^{4} \left((\Psi_i)^T (\tilde{\Phi}_i) \right) = -(e_u)^T (e_u) \tag{30}$$

Consequently, from Eq. (30), we find that $\dot{V} \leq 0$, so we conclude that the system is asymptotically stable in the sense of Lyapunov according to the LaSalle theorem.

3. Compensatory adaptive neural fuzzy inference system (CANFIS)

The other class of inference systems that can deal this type of analytic information in conclusion of rules inference was proposed by Sugeno and his staff.

As for our contribution here, it consists in adding a compensatory fuzzy part to adjust consequence parameters and as well to dynamically optimize the adaptive fuzzy reasoning. In addition to this, ANFIS represents a class of adaptive networks that are functionally equivalent to a first order Takagi-Sugeno fuzzy model. As performed above, by taking a center-average deffuzzifier mapping, the crisp value of the output u is given as:

$$u = \frac{\sum_{i=1}^{4} (a_i e + b_i \Delta e + c_i) w_i}{\sum_{i=1}^{4} w_i} \tag{31}$$

We consider the pessimistic and optimistic operation given respectively as follows:

$$z_i = w_i \tag{32}$$

$$m_i = [w_i]^{\frac{1}{2}} \tag{33}$$

By using these two operations, our contribution is to add the compensatory form formulated as [7]:

$$C_i(z_i, q_i, \gamma_i) = (z_i)^{1-\gamma_i} (m_i)^{\gamma_i} \tag{34}$$

Where $\gamma_i \epsilon [0 \ 1]$ is compensatory degree. Finally, the crisp value of the compensatory neural-fuzzy inference is derived as [13, 14]:

$$u = \frac{\sum_{i=1}^{4} (a_i e + b_i \Delta e + c_i)[w_i]^{1-\frac{\gamma_i}{2}}}{\sum_{i=1}^{4} [w_i]^{1-\frac{\gamma_i}{2}}} \tag{35}$$

For simplicity, we define:

$$\alpha_i = 1 - \frac{\gamma_i}{2} \tag{36}$$

Then we have:

$$u = \frac{\sum_{i=1}^{4}(a_i e + b_i \Delta e + c_i)[w_i]^{\alpha_i}}{\sum_{i=1}^{4}[w_i]^{\alpha_i}} \tag{37}$$

The structure of the CANFIS controller with compensatory fuzzy for two input and one output, is shown by **Figure 3** [9].

3.1 Learning algorithm

Consider as for ANFIS two dimensional data vectors, $x = [e, \Delta e]$ and one dimensional output data vector u_2. In order to limit the computation time, we have optimally adjusted the consequence parameters and compensatory degree by minimizing the following objective function:
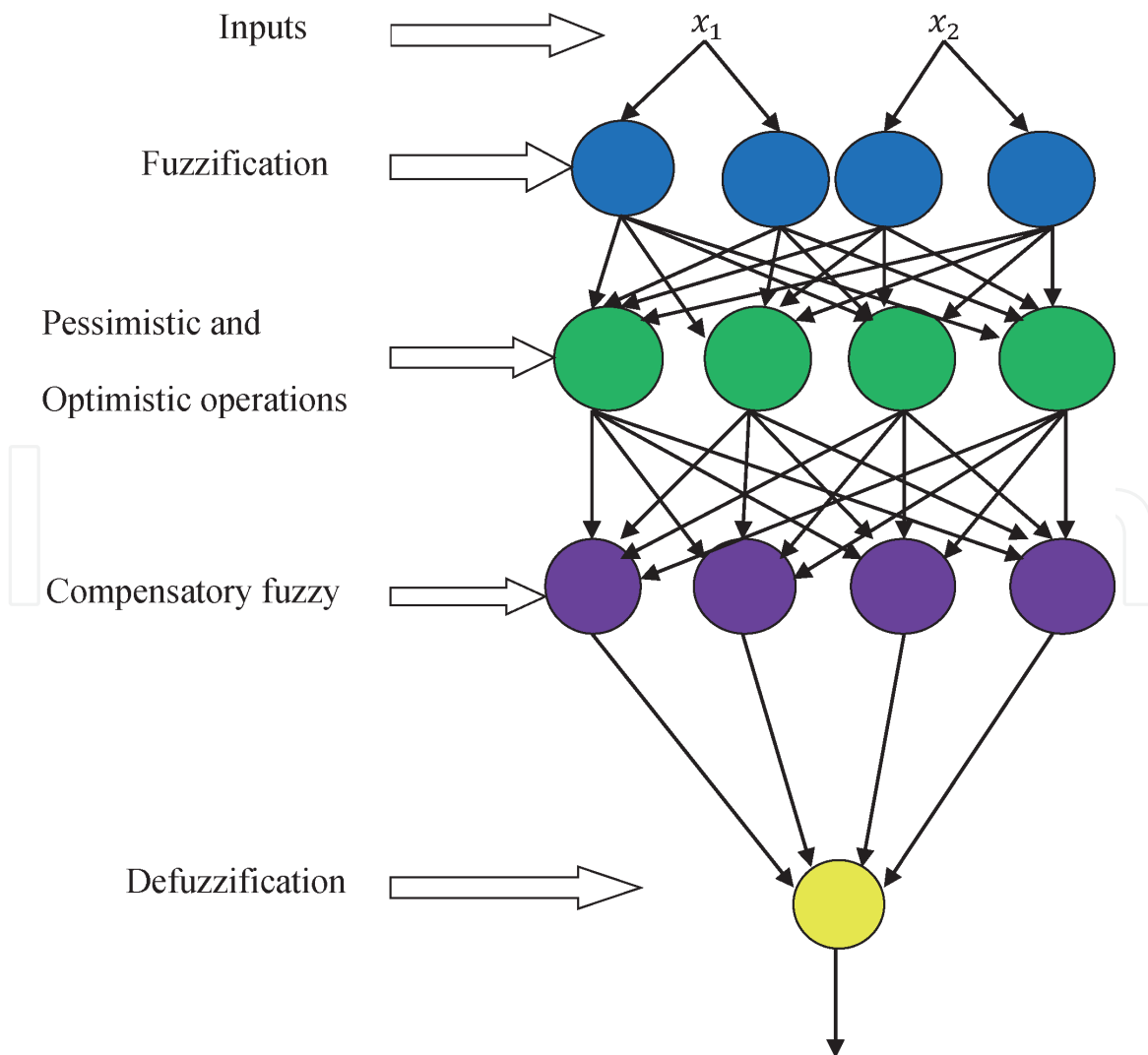
$$J = \frac{1}{2}(y_d - y)^2 \tag{38}$$

Figure 3.
Structure of CANFIS controller.

Where y_d and y are respectively desired and actual values of output system. Now let Φ_{2i} for $i = 1 \dots 4$, be the vector of update parameters. We aim to determine vector Φ_{2i} through the extended Kalman filter which consists in linearizing the output around the control input at each sampling period. This is equivalent to writing [16, 19, 20]:

$$\frac{\partial J}{\partial \Phi_{2i}} = \frac{\partial J}{\partial u_2} = -(y_d - y)\frac{\partial y}{\partial u_2}\frac{\partial u_2}{\partial \Phi_{2i}} = -K'.\Psi_{2i}e \tag{39}$$

In which

$$\Psi_{2i} = \frac{\partial u_2}{\partial \Phi_{2i}} \tag{40}$$

$$K_2 = \frac{\lambda_1}{\lambda_1 \Psi_{2i}^T \Psi_{2i} + \lambda_2} \tag{41}$$

Where λ_1 and λ_2 are adaptation gains for varying the convergence rate. Further, to eliminate the constraint $\gamma_i \in [0, 1]$, we redefine γ_i as follows [7]:

$$\frac{(p_i)^2}{(p_i)^2 + (r_i)^2} \tag{42}$$

Where p_i and r_i are update parameters such that $\gamma_i \in [0, 1]$. Consequently, the vector of update parameters is given as $(\Phi_{2i})^T = [a_i, b_i, c_i, p_i, r_i]$ for CANFIS. According to the definition, we have [17]:

$$\frac{\partial u_2}{\partial a_i} = \frac{e[w_i]^{\alpha_i}}{\sum_{i=1}^{4}[w_i]^{\alpha_i}} \tag{43}$$

$$\frac{\partial u_2}{\partial b_i} = \frac{\Delta e[w_i]^{\alpha_i}}{\sum_{i=1}^{4}[w_i]^{\alpha_i}} \tag{44}$$

$$\frac{\partial u_2}{\partial c_i} = \frac{[w_i]^{\alpha_i}}{\sum_{i=1}^{4}[w_i]^{\alpha_i}} \tag{45}$$

$$\frac{\partial u_2}{\partial \gamma_i} = -\frac{1}{2}\left[\sum_{i=1}^{4}(a_i e + b_i \Delta e + c_i)\right]\frac{z_i \ln(w_i)}{\sum_{i=1}^{4}z_i} \tag{46}$$

$$\frac{\partial u_2}{\partial p_i} = -\left\{\frac{2p_i(q_i)^2}{(p_i)^2 + (q_i)^2}\right\}\frac{\partial u_2}{\partial \gamma_i} \tag{47}$$

$$\frac{\partial u_2}{\partial r_i} = \left\{\frac{2q_i(p_i)^2}{(p_i)^2 + (r_i)^2}\right\}\frac{\partial u_2}{\partial \gamma_i} \tag{48}$$

Finally, the vector of parameters Φ_{2i} is adjusted using the following equation:

$$\Phi_{2i}(k + 1) = \Phi_{2i}(k) + K'\Psi_{2i}e \tag{49}$$

Where $(\Psi_{2i})^T = \left[\frac{\partial u_2}{\partial a_i}, \frac{\partial u_2}{\partial b_i}, \frac{\partial u_2}{\partial c_i}, \frac{\partial u_2}{\partial p_i}, \frac{\partial u_2}{\partial r_i}\right]$.

3.2 Stability analysis of the control system

From Eq. (48), we can consider for a very short time T_e, this relation:

$$\dot{\Phi}_{2i} = \frac{\Phi_{2i}(k+1) - \Phi_{2i}(k)}{T_e} = \frac{K'(\Psi_{2i})e}{T_e} = K_1(\Psi_{2i})e \tag{50}$$

Where $K_1 = \frac{K'}{T_e}$

Hence:

$$\dot{\Phi}_{2i} = K_1(\Psi_{2i})e_u \tag{51}$$

Where $e_u = K_1 e$ is the error between the controller's desired output u_d and actual output u.

Let be $\tilde{\Phi}_{2i} = \Phi_{2id} - \Phi_{2i}$, where Φ_{2i} is the vector of the consequence parameters and Φ_{2id} the vector of the desired consequence parameters.

$$\dot{\tilde{\Phi}}_{2i} = \dot{\Phi}_{2id} - \dot{\Phi}_{2i} = -(\Psi_{2i})e_u \tag{52}$$

For linear variation, the error e_u is defined by:

$$e_u = u_d - u = \sum_{i=1}^{4}\left((\Psi_{2i})^T \Phi_{2id} - (\Psi_{2i})^T \Phi_{2i}\right)$$

$$= \sum_{i=1}^{4}\left((\Psi_{2i})^T(\Phi_{2id} - \Phi_{2i})\right) = \sum_{i=1}^{4}\left((\Psi_{2i})^T \tilde{\Phi}_{2i}\right) \tag{53}$$

Consider the following Lyapunov function:

$$V = \frac{1}{2}\sum_{i=1}^{4}\left((\tilde{\Phi}_{2i})^T(\tilde{\Phi}_{2i})\right) \tag{54}$$

Differentiating V with respect to time yields, we obtain:

$$\dot{V} = \sum_{i=1}^{4}\left((\dot{\tilde{\Phi}}_{2i})^T(\tilde{\Phi}_{2i})\right) \tag{55}$$

From Eqs. (51), (52) and (54), we obtain:

$$\dot{V} = \sum_{i=1}^{4}\left((-(\Psi_{2i})e_u)^T(\tilde{\Phi}_{2i})\right) = -(e_u)^T\sum_{i=1}^{4}\left((\Psi_{2i})^T(\tilde{\Phi}_{2i})\right) = -(e_u)^T(e_u) \tag{56}$$

Consequently, from Eq. (55), we find that $\dot{V} \leq 0$, so we conclude that the system is asymptotically stable in the sense of Lyapunov according to the LaSalle theorem.

4. Simulation results and interpretation

We applied in simulation the neural-fuzzy command equipped with a compensator explained above, to the two-joint robot in a performance environment described by the following joint trajectories:

$$q_{ir} = \frac{\pi}{6}(1 - \cos(6t)) \tag{57}$$

With $i = 1 \ldots 2$.

The compact form of the dynamic model relating to the two-joint robot is given as follows:

$$\tau = \underbrace{\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}}_{M(q)} \ddot{q} + \underbrace{\begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q}) \\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q}) \end{bmatrix}}_{C(q,\dot{q})} \dot{q} + \underbrace{\begin{bmatrix} G_1(q) \\ G_2(q) \end{bmatrix}}_{G(q)} \tag{58}$$

Where.

$q = [q_1; q_2]$: Vector of joint position variables.

$\dot{q} = [\dot{q}_1; \dot{q}_2]$: Vector of joint velocity variables.

$\ddot{q} = [\ddot{q}_1; \ddot{q}_2]$: Vector of joint acceleration variables.

$\tau = [\tau_1; \tau_2]$: Vector of torques applied to joint.

$M(q)$: Inertial matrix.

$C(q,\dot{q})$: Matrix of terms centripetal and coriolis.

$G(q)$: Vector of gravitational effects

$$M_{11}(q) = m_1 l_{c1}^2 + m_2 l_1^2 + m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_1 + I_2$$

$$M_{12}(q) = M_{21}(q) = m_2 l_{c2}^2 + m_2 l_1 l_{c2} \cos(q_2) + I_2$$

$$M_{22}(q) = m_2 l_{c2}^2 + I_2$$

$$C_{11}(q,\dot{q}) = -m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2$$

$$C_{12}(q,\dot{q}) = -m_2 l_1 l_{c2} \sin(q_2) [\dot{q}_1 + \dot{q}_2]$$

$$C_{21}(q,\dot{q}) = m_2 l_1 l_{c2} \sin(q_2)$$

$$C_{22}(q,\dot{q}) = 0$$

$$G_1(q) = [m_1 l_{c1} + m_2 l_1] g \sin(q_1) + m_2 g l_{c2} \sin(q_1 + q_2)$$

$$G_2(q) = m_2 g l_{c2} \sin(q_1 + q_2)$$

The parameters relating to the dynamic model of this robot are given in the following table (**Table 1**):

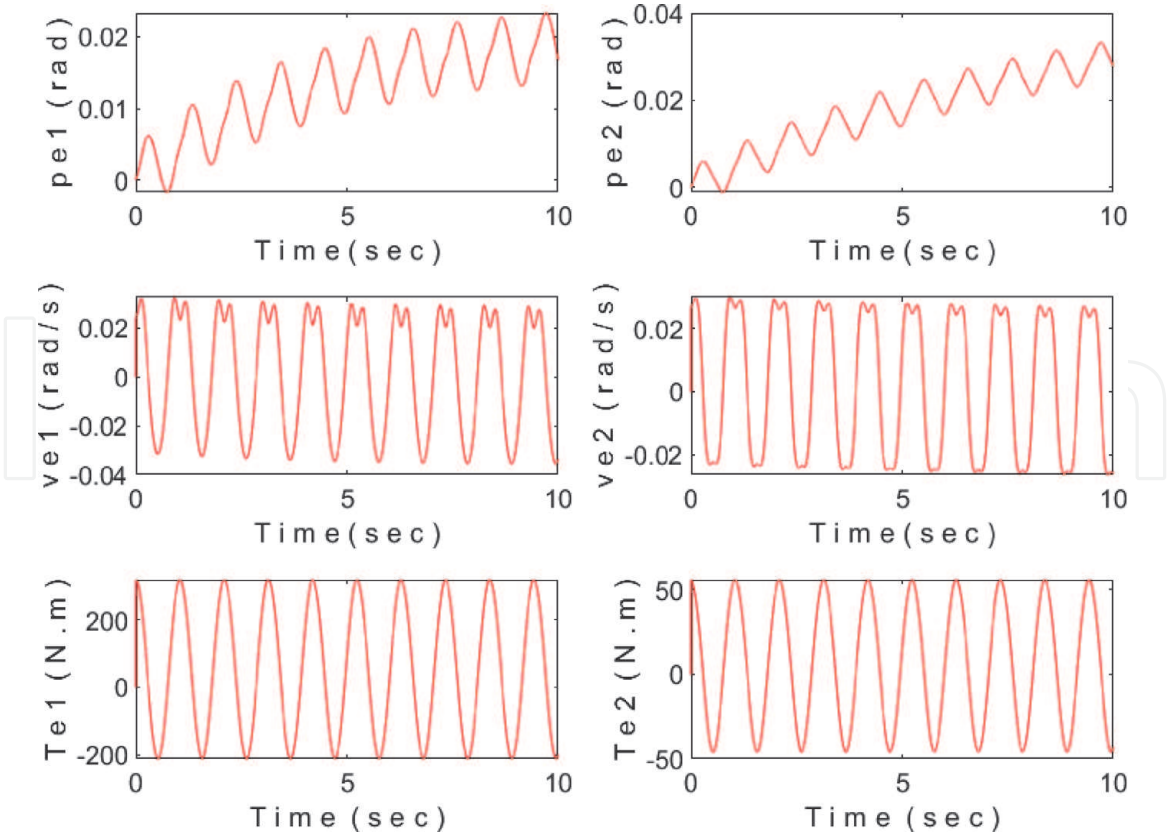| Parameter	Value
Mass of the link 1 (m_1)	6.5225 kg
Mass of the link 2 (m_2)	2.0458 kg
Link length 1 (l_1)	0.26 m
Link length 2 (l_2)	0.26 m
Gravity (g)	9.81 m/s^2
Distance to the center of mass of link 1 (l_{c1})	0.0983 m
Distance to the center of mass of link 2 (l_{c2})	0.0229 m
Moment of inertia of the center of mass m_1	0.1213 $kg.m^2$
Moment of inertia of the center of mass m_2	0.0116 $kg.m^2$

Table 1.
Robot parameters.

Figure 4.
Motion errors tracking and torques behavior with neural-fuzzy without disturbances.
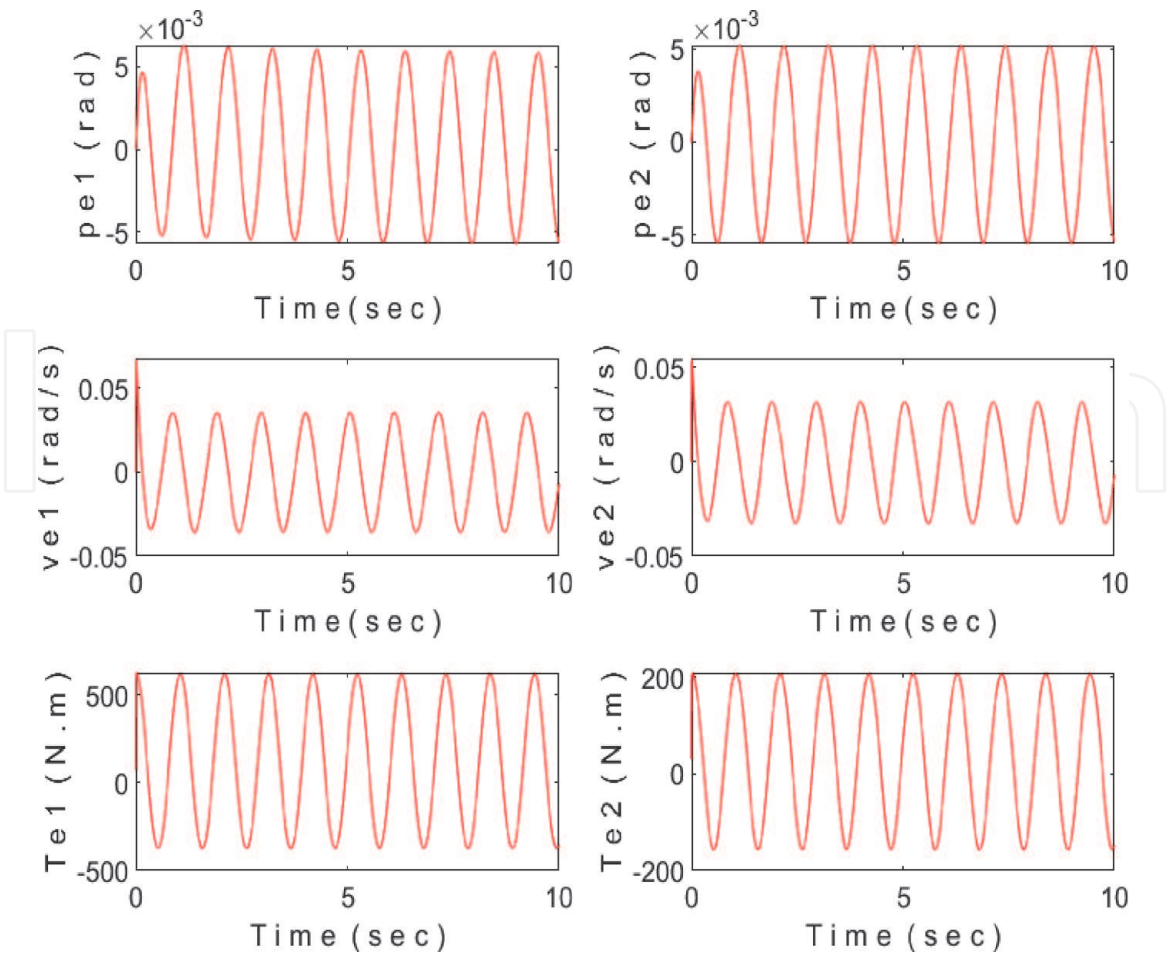

Figure 5.
Motion errors tracking and torques behavior with compensatory neural-fuzzy with without disturbances.

Figures 4 and **5** show the time evolution position and velocity errors and the torque applied to each joint of manipulator Robot with neuro-fuzzy controller and neuro-fuzzy controller respectively. Through these graphics, we can see that, the Neural –fuzzy controllers and compensatory Neural-fuzzy controllers provide a good tracking performance.

On the one hand, we can observe that the tracking errors are limited by low values, and the dynamics of the errors in position vary little compared to that of the errors in speed. This is physically explained by the fact that the position depends only on the environment while the velocity in addition to the environment depends on the Jacobean matrix. On the other hand, the command paths are smooth, which facilitates their implementation. This is achieved through the appropriate choice of parameters of the control structures.

In order to test the capacity of adaptation and robustness of the proposed approach, we have added in our simulation at time $t = 5s$ the combined friction and external torque disturbance for each joint, given as follows.

$$\tau_{fi} = 38.3\dot{q}_i + 18.9\cos\left(q_i\right) \tag{59}$$

The results obtained are illustrated by **Figures 6** and **7**, where we note that the tracking errors show peaks especially at the moment of the introduction of the disturbances, which are rejected quickly, by the Neural – fuzzy controllers and Compensatory Neural-fuzzy structure of the regulator, which allows to conclude that the tracking performance is very little affected by these disturbances. This is due to the low sensitivity to disturbance of the input data of the proposed control strategy.
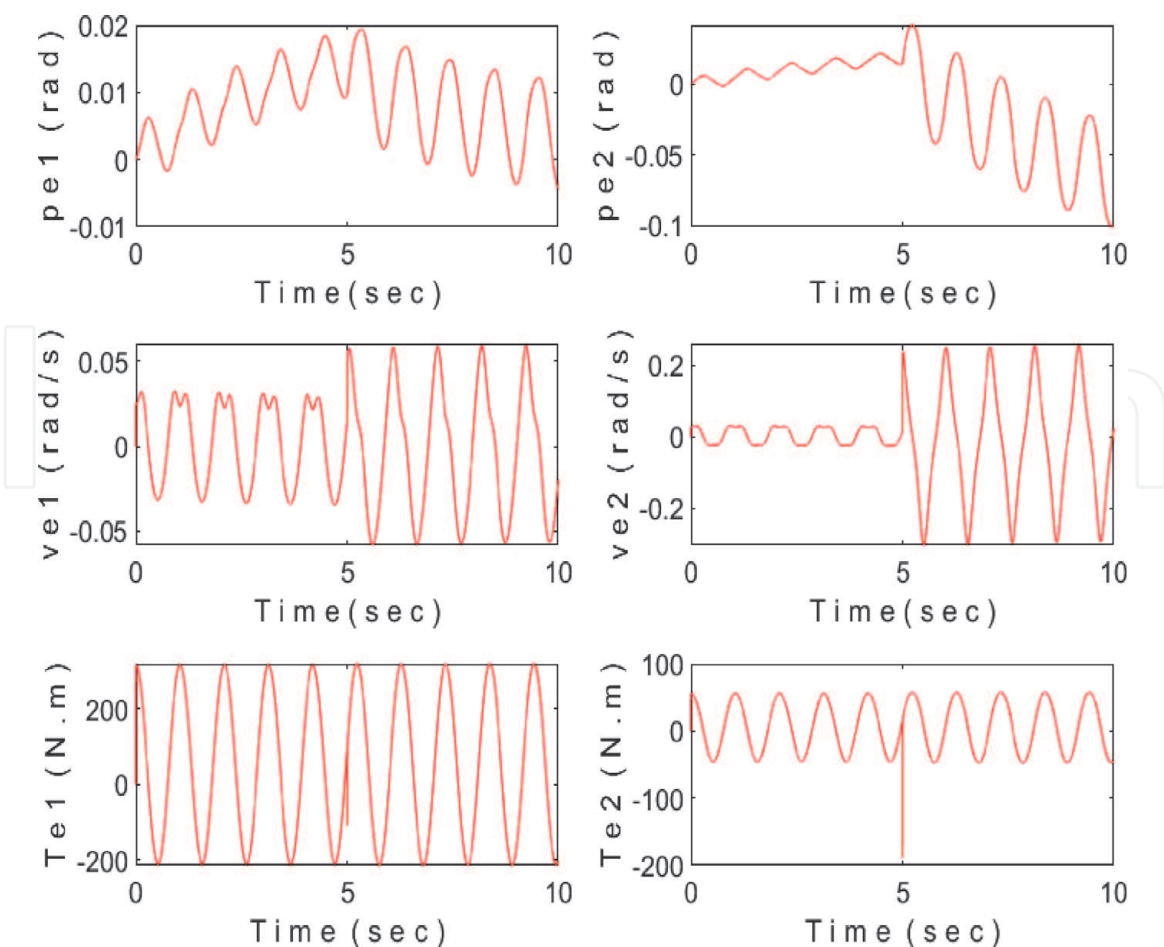

Figure 6.
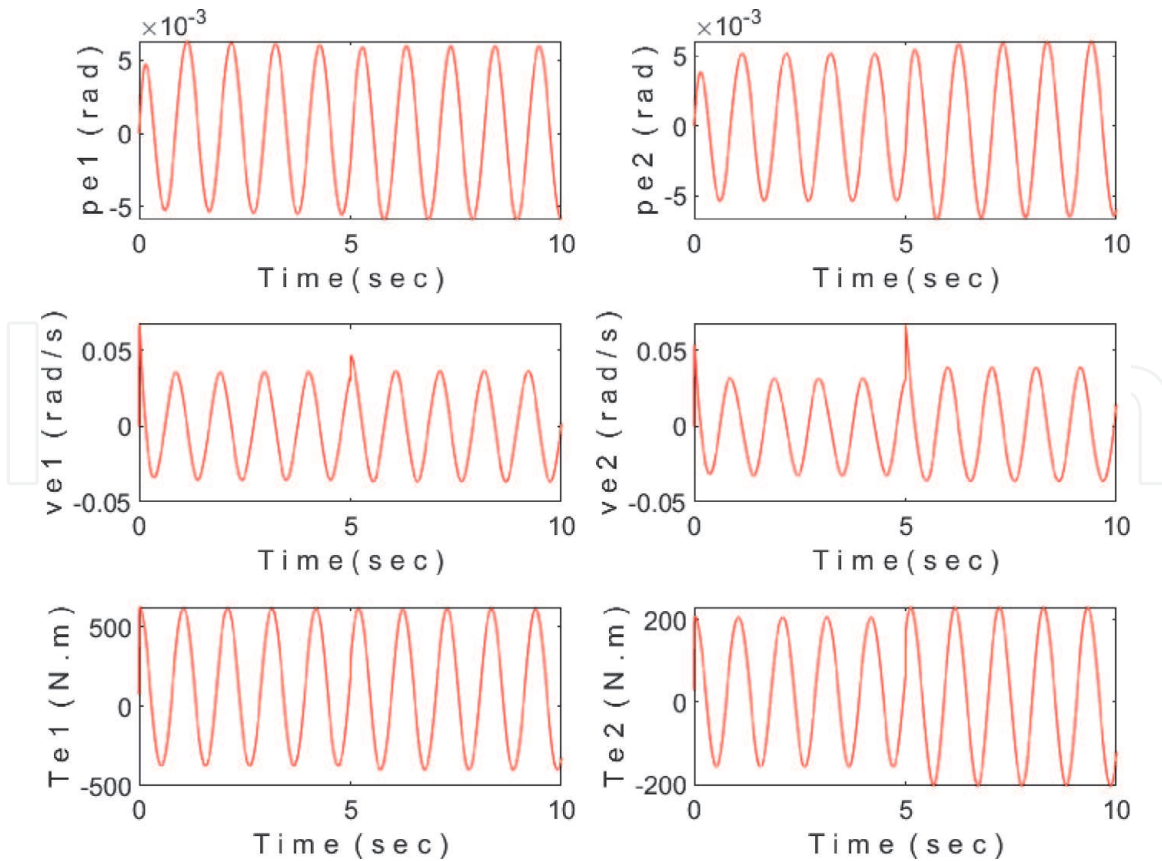*Motion errors tracking and torques behavior with neural –fuzzy with disturbances.*

Figure 7.
Motion errors tracking and torques behavior with compensatory neural –fuzzy with disturbances.

5. Conclusion

In this chapter, we have proposed and presented a control strategy based on a neuro-fuzzy inference system regulator with a fuzzy compensator to control the manipulator robot with two joints.

It is important to note that the compensatory neural fuzzy inference system is more powerful than fuzzy systems or neural networks, since it can incorporate these advantages:

- Adaptive fuzzy reasoning method using fuzzy compensator can make the fuzzy system adaptive and efficient more.

- The neuro-fuzzy compensation system tolerates errors, because it is effective regardless of the choice of initial fuzzy rules (good or bad) for learning.

- The speed of convergence of the learning algorithm is faster than that of the back propagation algorithm.

- The learning algorithm not only adjusts membership functions, but also optimizes the dynamics of fuzzy reasoning by adjusting the degree of compensation. As a result, fuzzy neuro systems with a fuzzy compensator are more efficient than conventional neuro-fuzzy systems.

This control strategy has the advantage of requiring only measurements of output variables. The simulations indicate that a complete stabilization of the system is indeed observed.

In conclusion, we wanted, at the end of this chapter, to try to identify a broad-spectrum methodology that opens the field to a possible standardization of the design of control laws based on a neuro-fuzzy structure equipped with a compensator, in order to controlling a poorly understood and imprecise dynamic system. It would be interesting in the context of experimental tests to judge the performance of the methods proposed on real systems.

Author details

Rabah Mellah*, Hocine Khati, Hand Talem and Said Guermah
Design and Control Laboratory Automated Systems, Faculty of Electrical
Engineering and Computer Science, UMMTO, Tizi-Ouzou, Algeria

*Address all correspondence to: mellah.rab@gmail.com

IntechOpen

References

[1] JYH-SHING R., CHUEN-TSAI S., "*Neuro-Fuzzy Modeling and Control*" Procedings of the IEEE, Vol. 83, N⁰ 3, March 1995 pp. 378–404.

[2] S. Ghosh, M. Majumder. "Prediction of Wave Energy Potential in India: a Fuzzy–ANN Approach." Intech-2019, DOI: http://dx.doi.org/10.52772/intech open.84676.

[3] S.D. Nguyen. "ANFIS: Establishing and Applying to Managing Online Damage." Intech-2019, DOI: http://dx.doi.org/10.5772/intechopen.83453.

[4] B. Ergün, C. Sahin and U. Kaplan. "ANFIS Definition of Focal Length for Zoom Lens Via Fuzzy Logic Functions." Intechopen-2017, DOI: http://dx.doi.org/10.5772/67823

[5] L. Moradi and R. Ehtehamrasi. "ANFIS to Quantify Maintenance Cost of IT Services in Telecommunication Company." Intechopen-2020, DOI: http://dx.doi.org/10.5772/intechope n.82827

[6] HAO Y. "*Constructing Nonlinear Variable Gain Controllers via the Takagi - Sugeno Fuzzy Control*." IEEE Transactions on Fuzzy Systems, Vol. 6, N°. 2, May 1998, pp. 226–234.

[7] YAN-QING., ABRAHIM K., "*Compensatory Neurofuzzy Systems with Fast Learning Algorithms*." IEEE Transactions On Neural Networks, Vol. 9, N⁰ 1, January 1998 pp. 83–105.

[8] BYUNG K., WOON C. H., "*Adapive Control of Robot Manipulator Using Fuzzy Compensator*." IEEE Transactions on Fuzzy Systems, Vol. 8, N°. 2, April 2000.

[9] R. Mellah, S. Guermah, and R. Toumi. "Adaptive control of bilateral teleoperation system with compensatory neural fuzzy controllers."

International Journal of Control, Automation and Systems, vol. 15, no. 4, pp. 1949–1959, 2017.

[10] R. Mellah and R. Toumi, "Compensatory neuro-fuzzy control of bilateral teleoperation system," *Proc. of 20th Internationale Conference on Methods and Models in Automation an Robotic*, 24–27 August 2015, Miedzyzdroje, Poland, pp. 382–387.

[11] J. R. Jang. "ANFIS: adaptive-network based fuzzy inference system." IEEE Transactions on Systems, Man and Cybernetics N° 23, vol. 13 pp 665–685, 1993.

[12] S. R. Jang and C. T. Sun. "Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence." Prentice Hall, Englewood Cliffs, NJ, 1997.

[13] M. N. M. Salleh, N. Talpur and K. H. Talpur. "A Modified Neuro-Fuzzy System Using Metaheuristic Approach for Data classification." Intechopen-2018, DOI: http://dx.doi.org/10.5772/intechopen.75575

[14] KARAKASOGLU A., SUNDARESHAN M., "A Recurrent Network-based Adaptive Variable structure Model- following Control of Robotic Manipulators." Automatica, Vol. 31, N0 10, pp. 1495–1507, 1995.

[15] YIH-GUANG L., WEI-YEN W., TSU-TIAN L., *"Robust Adaptive Fuzzy-Neural Controllers for Uncertain Nonlinear Systems*" IEEE Transactions on Robotics and Automation, Vol. 15, N°. 5 Oct 1999.

[16] JEAN B. M., "Robust Neuro-Fuzzy Sensor-Based Motion Control Among Dynamic Obstacles for Robot Manipulators." IEEE Transactions on Fuzzy Systems, Vol. 11, NO. 2, April 2003, pp. 249–261.

[17] MENG J., "*Robust Adaptive Control of Robot Manipulators Using Generalized Fuzzy Neural Networks*." IEEE on Industrial Electronics, Vol. 50, N°. 3, June 2003.

[18] R. Mellah and R. Toumi, "Control bilateral teleoperation by compensatory ANFIS," Advanced Mechatronics Solutions, vol. 393, pp. 167–172, 2016.

[19] LIMIN P., PENG-YUNG W., "*Neural-Fuzzy Control System for Robotic Manipulators*." IEEE Control Systems Magazine, February 2002. pp. 53–63

[20] H Khati, R Mellah, H Talem. "Ne uro-fuzzy Control of a Position-Position Teleoperation System Using FPGA." 24th International Conference on Methods and Models in Automation and Robotics, 26–29 August 2019, Miedzyzdroje, Poland, pp. 64–69.

[21] H Khati, H Talem, R Mellah, A Bilek. "Neuro-fuzzy control of bilateral teleoperation system using FPGA." Iranian Journal of Fuzzy Systems vol. 16, N°. 6, pp. 17–32, 2019.