Refrences

[1] C. C. Lee, "Fuzzy logic in control systems: Fuzzy logic controller," IEEE Trans. Sys., Man, Cyber., vol. 20, No. 2, pp. 404-434, Mar. 1990.

[2] N. Nariman-Zadeh and A. Darvizh, "Genetic and Evolutionary Design and Robustification of Fuzzy-logic Controllers for Manufacturing Systems," IUST, Journal of Iranian Mechanical Engineering, No. 5, pp. 1-12, Summer 2000.

[3] J. J. Craig, "Introduction to Robotics," Addison-Wesley, 1989.

[4] N. Nariman-Zadeh, "Genetic Design of Controllers for Robotic Manipulators," PhD Thesis, The University of Salford, UK, 1996.

[5] B. Porter, "Genetic design of control systems," J.SICE, Vol. 34, pp 343-402, 1995.

[6] B. Porter and N. Nariman-Zadeh, "Genetic design of computed-torque/fuzzy-logic controllers for robotic manipulators," Proc. IEEE International symposium on Intelligent control, Monterey, USA, 1995.

[7] J. H. Holland, "Adaptation in Natural and Artificial System," University of Michigan press, 1975.

[8] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley, 1989.

[9] B. Porter, B A Sangolola, N Nariman-Zadeh, "Genetic design of computed-torque controllers for robotic manipulators," IASTED Int. Conf. On Sys. And Control, Lugano, Switzerland, 1994.

[10] J. J. Craig, "Adaptive Control of Mechanical Manipulators," Addison-Wesley, 1988.

[11] B. Porter and N. N. Nariman-Zadeh, "Evolutionary Robustification of Fuzzy-logic Controllers for Manufacturing Systems," World Manufacturing Congress (WMC'97), November 18-21, 1997, Auckland, New Zwaland.

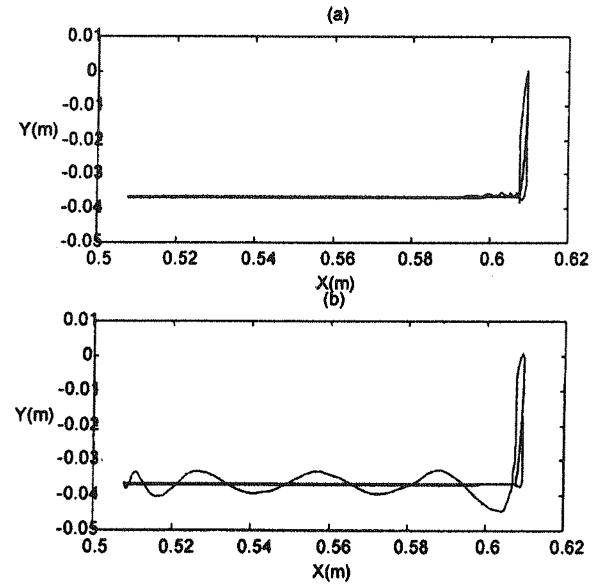

Figure (8) Workspace trajectory tracking Genetically tuned (without payload) (a) Without payload (b) With payload.

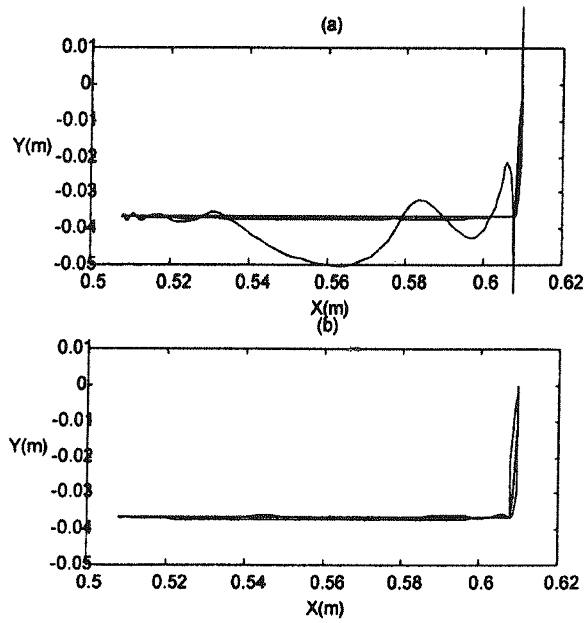

Figure (9) Workspace trajectory tracking Genetically tuned (with payload) (a) Without payload (b) With payload.

These results indicate that the genetic design procedure involved in (III) readily produces robust controllers. Thus, it should be noted that $\Gamma'_5 < \Gamma'_3$ event though $\Gamma'_5 > \Gamma'_1$; and $\Gamma'_6 < \Gamma'_2$ even though $\Gamma'_6 > \Gamma'_4$; in addition, $\Gamma'_5 + \Gamma'_6 < \Gamma'_1 + \Gamma'_2$ and $\Gamma'_5 + \Gamma'_6 < \Gamma'_3 + \Gamma'_4$. In this way, it is also interesting to note that the genetic design procedure provides effective controllers by observing their performances in Fig. 8 (a), Fig. 9 (b), and Figs. 10 (a & b), for which such optimal design procedure has been applied individually.

4-Conclusion

In this paper, genetic algorithms have been used in the design and robustification of various monovariable and multivariable fuzzy-logic controllers. In particular, it has been shown that genetic algorithm provides effective mean of designing the optimal set of fuzzy rules as well as the optimal domains of associated fuzzy sets simultaneously both in monovariable model-based and in multivariable non-model-based fuzzy-logic controllers. Furthermore, it has been shown that genetic algorithms are very effective in robustification in order to cope with uncertainties in the dynamical characteristics of robotic manipulators.

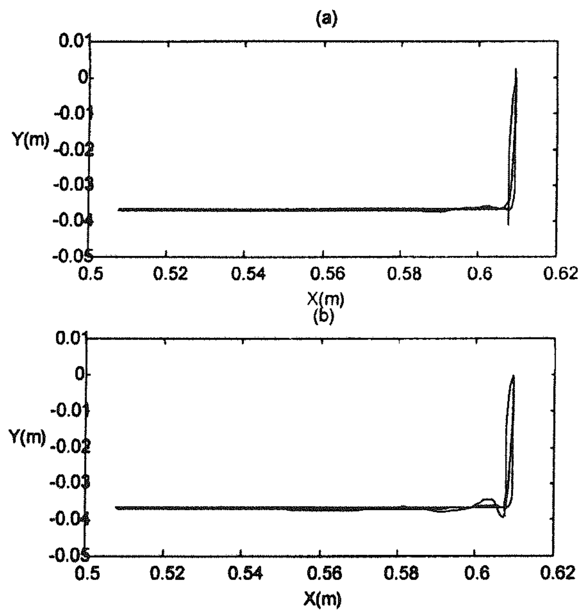

Figure (10) Workspace trajectory tracking Genetically tuned (without payload) (a) Without payload (b) With payload:

controller (C_1) which has been only genetically domain selected. In the case of multivariable fuzzy-logic controller in which no information regarding the dynamical parameters is used, the genetic design procedure has been

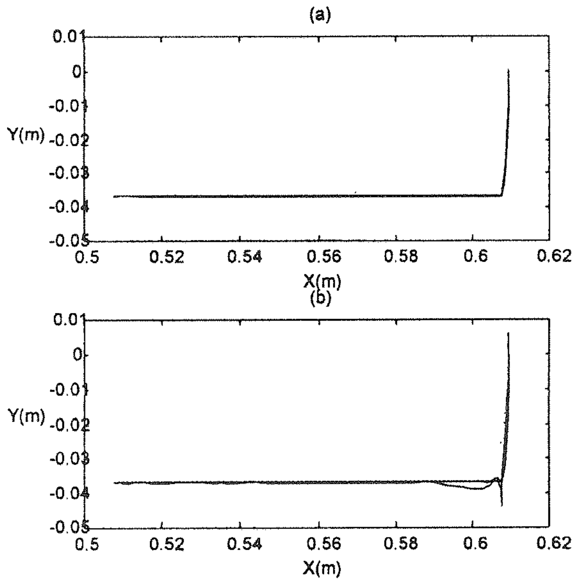

Figure (6) Workspace trajectory tracking Genetically domain selected (without payload) (a) Without payload (b) With payload.

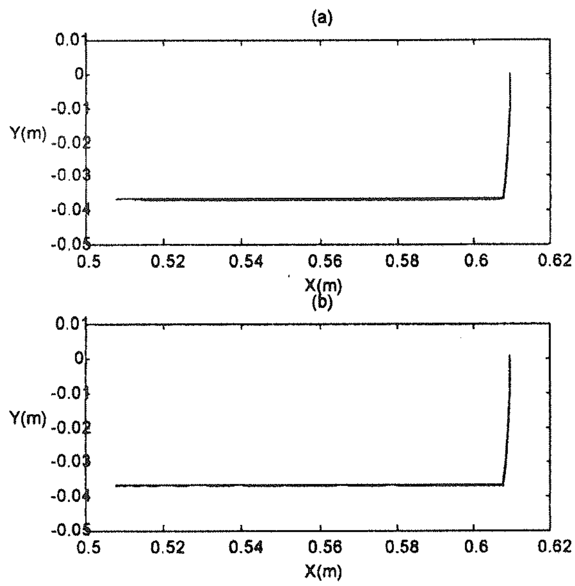

Figure (7) Workspace trajectory tracking Genetically tuned (without payload) (a) Without payload (b) With payload.

accomplished in three different ways in order to show the case of designing robust controllers:

(I) The controller is first genetically designed so that the domains of the fuzzy sets and the consequents of fuzzy rules are optimally determined simultaeously when the robotic manipulator carries no payload. The resulting workspace trajectory-tracking performances of the end-effector are present d in Fig. 8 both without payload and with payload. The associated cost function, Γ, without payload and with payload, are readily computed, respectively, $\Gamma'_1=144.\times10^{-4}$ and $\Gamma'_2=360.\times10^{-4}$.

(II) The controller is then genetically designed so that the domains of the fuzzy sets and the consequents of fuzzy rules are optimally determined simultaneously when the robotic manipulator carries a 4Kg payload. The resulting workspace trajectory-tracking performance of the end-effector are presented in Fig.9 both with payload and without payload. The associated cost function, Γ, without payload and with payload, are readily computed, respectively, $\Gamma'_3=317.\times10^{-4}$ and $\Gamma'_4=236.\times10^{-4}$.

(III) The controller is finally genetically designed so that the domains of the fuzzy sets and the consequents of fuzzy rules are optimally determined simultaneously when the robotic manipulator performs a composite task consisting of the task in the absence of any payload together with the task in the presence of a 4Kg payload.

The resulting workspace trajectory-tracking performances of the end-effector are presented in Fig.10 both without payload and with payload. The associated cost function, Γ, without payload and with payload are readily computed, respectively, $\Gamma'_5 = 185.8\times10^{-4}$ and $\Gamma'_6=259.7\times10^{-4}$.

logic controller when the robotic manipulator performs the specified task.

3-ILLUSTRATIVE EXAMPLE

These genetic methodologies can be conveniently illustrated by designing monovariable computed-torque/fuzzy-logic and multivariable fuzzy-logic controllers for a two-link direct-drive robotic manipulator presented in Fig. 4. The governing dynamic equations of this robotic manipulator is given in detail in [4].

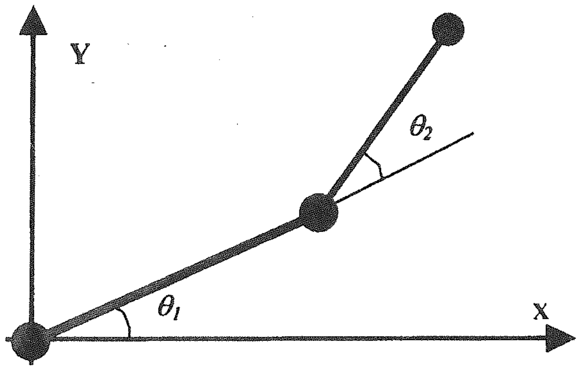

Figure (4) Schematic diagram of a two-link robotic manipulator.

In the case of monovariable computed torque/fuzzy-logic controller, these genetic design procedure have been accomplished in two different ways:

(I) The controller, C_1, is first genetically designed so that only the domains of the fuzzy sets are optimally determined when the robotic manipulator carries no payload. The progressively improving behavior of the genetically-domain-setected monovariable computed-torque fuzzy-logic controller can be observed in Fig.5.

Accordingly, the workspace trajectory tracking performances of the end effector are presented in Fig. 6 both without payload and with payload. It must be noted that the robustness of such controller designed when subjected to no payload can be tested due to structured dynamical uncertainties because of an additional unknown payload. The associated cost function, Γ, without payload and with payload are readily computed, respectively, $\Gamma_1 = 39.7 \times 10^{-4}$ and $\Gamma_2 = 161.0 \times 10^{-4}$.

(II) The controller, C_2, is then genetically designed so that the domains of the fuzzy sets and the consequents of fuzzy rules are optimally determined simultaneously when the robtic manipulator carries no payload. of the Accordingly, the workspace trajectory-tracking performances end-effector are presented in Fig. 7 both without payload and with payload.

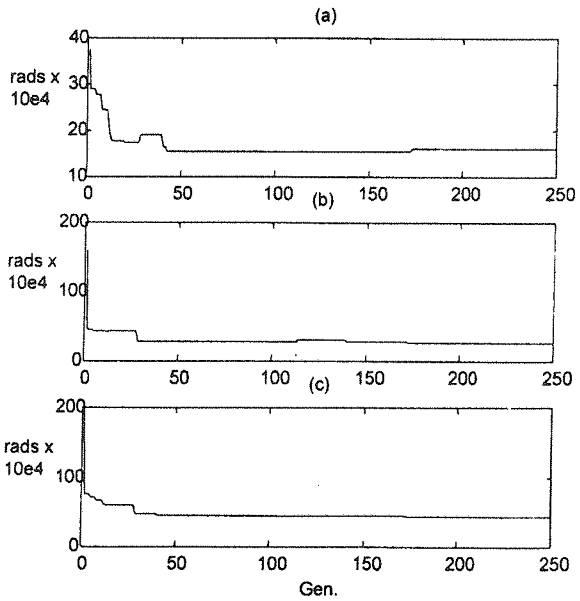

Figure (5) (a) IAE of joint 1 (b) IAE of joint (c) sum of IAE of joints.

The associated cost functions, Γ, without payload and with payload are readily computed, respectively, Γ_3=20.0$\times10^{-4}$ and Γ_4=45.0.$\times10^{-4}$ It is evident that $\Gamma_1<\Gamma_3$ and $\Gamma_2<\Gamma_4$. This means that the controller (C_2) which has been genetically rule induced and domain selected is superior both in performance and robustness than those in of progressively increaing fitness, φ, are produced until no further significant improvements is achievable. This genetic design procedure provides the optimal set of fuzzy rules as well as associated fuzzy domains simultaneously.

2-2 Multivariable fuzzy-logic controllers

In this case, in order to remove the need for precise knowledge of dynamical parameters incorporated in the dynamical equation, the control law is presented in the form of

$$\tau = E(e, \dot{e}) \quad (17)$$

expressed in terms of fuzzy-logic. In this form, the multivariable fuzzy-logic controller is governed by set of rules of the form

$$\text{IF } e_1 \text{ is } P_1^{(j_1)} \text{ and } e_2 \text{ is } P_2^{(j_2)} \text{ and } \ldots e_n \text{ is } P_1^{(j_n)}$$
$$\text{and } \dot{e}_1 \text{ is } Q_1^{(k_1)} \text{ and } \dot{e}_1 \text{ is } Q_2^{(k_2)} \text{ and } \ldots \dot{e}_n \text{ is } Q_n^{(k_n)}$$
$$\text{THEN } \tau_1 \text{ is } R_1^{(l_1)} \text{ and } \tau_1 \text{ is } R_2^{(l_2)} \text{ and } \ldots \tau_n \text{ is } R_n^{(l_n)} \quad (18)$$

The entire fuzzy sets in e_i, $\dot{e}_i$, and τ_i spaces are given in equation (14), in which $j_i \in \{1,2,...,p\}$, $k_i \in \{1,2,...,q\}$, and $l_i \in \{1,2,....,r\}$. These entire fuzzy sets are symmetric and, respectively, defined on the domains $[-\alpha_i, +\alpha_i]$, $[-\beta_i, +\beta_i]$, and $[-\gamma_i, +\gamma_i]$, $(i=1,2,...,n)$. In this case, the fuzzy partitioning and associated triangular membership function of antecedents and the consequents incorporated in the generic rule (18) are assumed to be symmetric in which, p=q=3 and r=7. It is evident from the generic rule (18) that there are $(pq)^n$ rules each with a different antecedent for each of which the appropriate consequent must be determined in terms of entire fuzzy sets $R_i(i=1,2,...,n)$. It is also evident that there are 3 different domains of fuzzy sets incorporated in the antecedents and consequents of the generic rule (18) for each joint of the robotic manipulators which hence indicates that there are 3n parameters representing the domains of fuzzy sets for the n-link robotic manipulator. The genetic design procedure accordingly represents each fuzzy-logic controller as an entire string of $[(pq)^n+3n]$ concatenated substrings of binary digits. The number of fuzzy linguistic values of each fuzzy sets in the e_i and $\dot{e}_i$ spaces are 3 (i.e.N, Z, P). The binary representation of the fuzzy consequents of rules is shown in Fig (3). However, it is assumed that optimal set of rules for each joint of the robotic manipulators includes the followinng rule

$$\text{IF } e_1 \text{ is Z and } e_2 \text{ is Z}$$
$$\text{AND } \dot{e}_1 \text{ is Z and } \dot{e}_2 \text{ is Z}$$
$$\text{THEN } \tau_1 \text{ is Z and } \tau_2 \text{ is Z.} \quad (19)$$

The fitness, φ, of each entire string of binary digits is readily evaluated using equation (15). The evolutionary process involved in the genetic design of these multivariable fuzzy-logic controllers starts by randomly generating an initial population of binary strings each as a candidate solution. Then, using the standard genetic operations of roulette wheel selection, multi-point crossover, and mutation [8], entire population of binary strings are caused to evolve. In this way, multivariable fuzzy-logic controllers of progressively increasing fitness are produced until no significant further improvement is achievable. This genetic design procedure thus provides the optimal multivarible fuzzy-

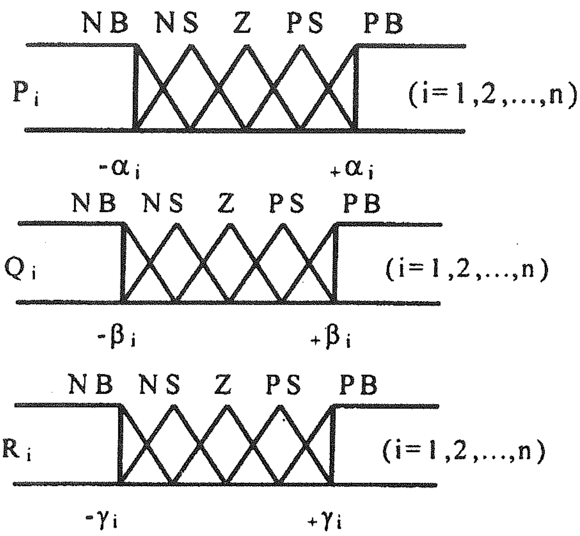

Figure (2) Symmetric triangular fuzzy membership functions:

The evolutionary process starts by randomly generating an initial population of binary strings each as a candidate solution. Then, using the standard genetic operations of roulette wheel selection, multi-point crossover, and mutation [8], entire populations of binary strings are caused to evolve. In this way, computed-torque/fuzzy-logic controllers of progressively increasing fithess, φ, are produced until no further significant improveme nt is achievable. It is evident that a controller with the largest fitness, φ_{max}, is an optimal or near-optimal solution in the sense of the smallest cost function, Γ_{min}. This genetic design procedure provides the optimal domains of fuzzy-sets associated with some prespecified set of fuzzy rules obtained heuristically using control engineering knowledge.

In case (II), the genetic design procedure represents each fuzzy-logic controller represented by equation (12) and (13) as a string of (pqn + 3n) concatenated sub-strings of binary digits. In each such entire string the first (pqn) sub-strings represent (in encoded) form the entire set of consequents of fuzzy rules, whilst the remaining 3n sub-strings represent (in encoded form) the symmetric domains of these rules. Thus, the concatenated sub-strings of binary digits involve both the consequents of rules and the domains of fuzzy membership functions. The number of fuzzy linguistic values of each fuzzy sets in the e_i and $\dot{e}_i$ spaces is 5 (i.e., NB, NS, Z, PS, PB) as it was in case (I). The binary representation of the fuzzy consequents of rules is shown in Fig 3.

However, it is assumed that optimal set of rules for each joint of the robotic manipulators includes the following rule:

$$\text{IF } e_i \text{ is Z and } \dot{e}_i \text{ is Z, THEN } u_i \text{ is Z.} \qquad (16)$$

The fitness, φ, of each entire string of binary digits which represents a controller is readily evaluated from equation (15) as before.

Sign bits	Value bits	Numeric value	Fuzzy value
0	1 1	-3	NB
0	1 0	-2	NM
0	0 1	-1	NS
0,1	0 0	0	Z
1	0 1	+1	PS
1	1 0	+2	PM
1	1 1	+3	PB

Figure (3) The binary representation of the fuzzy consequents of rules.

The evolutionary process involved in this genetic design starts by randomly generating an intial population of binary strings each as a candidate solution. Then, using the standard genetic operation of selection, multi-point crossover, and mutation [8] entire population of binary digits are caused to evolve. In this way, computed-torque/fuzzy-logic controllers of fuzzy rules set and fuzzy domains set. This representation can be achieved in different ways; but the simplest form of these controllers is governed by the decoupled version

$$u_i = K_i\,(e_i, \dot{e}_i) \quad (i = 1,2,3,\ldots,n) \tag{12}$$

of equation (10b) in which each of n joins is controlled separately. Thus, each of the n decoupled PD like fuzzy logic controllers is presented by a rule of the following generic form

$$\text{IF } e_i \text{ is } P_i^{(j)} \text{ AND } \dot{e}_i \text{ is } Q_i^{(k)} \text{ THEN } u_i \text{ is } R_i^{(l)}$$

$$(i = 1,2,\ldots,n) \tag{13}$$

The entire sets of fuzzy sets in e_i , $\dot{e}_i$ and u_i spaces are, respectively

$$P_i = \{P_i^{(1)}, P_i^{(2)}, \ldots, P_i^{(p)}\}$$
$$(i = 1,2,\ldots,n), \tag{14a}$$

$$Q_i = \{Q_i^{(1)}, Q_i^{(2)}, \ldots, Q_i^{(q)}\}$$
$$(i = 1,2,\ldots,n), \tag{14b}$$

$$R_i = \{R_i^{(1)}, R_i^{(2)}, \ldots, R_i^{(r)}\}$$
$$(i = 1,2,\ldots,n). \tag{14c}$$

In this notation $j \in \{1,2,\ldots,p\}$, $k \in \{1,2,\ldots,q\}$, and $l \in \{1,2,\ldots,r\}$. This entire fuzzy sets are assumed symmetric and are, respectively, defined on the domains $[-\alpha_i, +\alpha_i]$, $[-\beta_i, +\beta_i]$, and $[-\gamma_i, +\gamma_i]$, $(i=1,2,\ldots,n)$.

The genetic design approach for such controllers presented in this paper is achieved in two different ways:

(I) C_1, a computed-torgue/PD-like fuzzy-logic controller for which only the domains of the fuzzy sets are determined genetically.

(II) C_2, a computed-torgue/PD-like fuzzy-logic controller for which the fuzzy ruler as well as the domains of the fuzzy sets are determined genetically.

In case (I), the genetic design procedure represents each fuzzy-logic controller represented by equations (12) and (13) as a string of 3n concatenated sub strings of binary digits, Each such sub-string represents the individual domains of the entire symmetric fuzzy sets P_i, Q_i, and R_i $(i=1,2,\ldots,n)$ in binary coded form. The fitness, φ, of each entire string of binary digits which represents a controller is readilly evaluted in the form

$$\varphi = \mathbf{Big} - \Gamma \tag{15}$$

where $\mathbf{Big} \in \Re^+$ is an appropriately large number and Γ is the cost function given by equation (11). It must be noted that the set of fuzzy-rules (of which the associated domains of fuzy-sets are determined genetically) is selected for each joint in accordance with the set-point control knowledge, represented in Fig. (1), [4][6]. Furthermore, the entire symmetric fuzzy sets P_i, Q_i, and R_i which are, respectively, defined on the domains $[-\alpha_i, +\alpha_i]$, $[-\beta_i, +\beta_i]$, and $[-\gamma_i, +\gamma_i]$, $(i=1,2,\ldots,n)$ have triangular membership functions of antecedents and consequence incorporated in the generic rule (13) and shown in Fig.2 when p=q=r=5.

e \ $\dot{e}$	NB	NS	Z	PS	PB
NB	NB	NB	NB	NS	Z
NS	NB	NS	NS	Z	PS
Z	NB	NS	Z	PS	PB
PS	NS	Z	PS	PS	PB
PB	Z	PS	PB	PB	PB

Figure (1) Set of fuzzy rules.

of computed tprque/fuzzy logic controller is described to improve such model based controllers dynamical behaviour. In this approach, a non linear PD like fuzzy logic controller in introduced by a control law of the form

$$u = K(e, \dot{e}) \qquad (8)$$

in association with equation (6). The effectiveness of genetic algorithm is demonstrated in the optimally design of domains of the fuzzy logic sets embodied in such computed torque/PD like fuzzy logic controllers. Moreover, it is shown that the optimal fuzzy rules as well as the optimal domains of the associated fuzzy sets can be simultaneously determined using genetic algorithms. The results clearly show that such heuristic search methods significantly automate the design procedure of such controllers in an optimal sense. Secondly, in order to remove the need for dynamical parameter estimates in such model based controllers, genetic algorithm are also deployed to design optimal non model based multivariable fuzzy logic controllers governed by non linear control law of the form

$$\tau = E(e, \dot{e}) \qquad (9)$$

expressed in terms of fuzzy logic. It is demonstrated that genetic algorithms are a very effevtive means of determining the optimal domains of fuzzy sets as well as optimal set of fuzzy rules embodied in such non linear multivariable non model based fuzzy logic controllers. Furthermore, it is also demonstrated that such fuzzy logic controllers [11] can be readily robustified with respect to structured uncertainties in the dynamical characteristics of robotic manipulators.

2-Genetic Design Procedure

Fristly, a new class of monovariable model based fuzzy logic controller is introduced. The genetic design approach is then presented for two types of controllers, namely C_1 and C_2. Secondly, a multivariable non model based fuzzy logic controller is genetically designed for both set of fuzzy rules and set of fuzzy domains. Then, the robustification procedure of such controllers is also described using genetic algorithms.

2.1-Monovariable computed Troque/PD like fuzzy logic controllers

In this case, the control law is presented in the form of

$$\tau = D(\theta)(\ddot{\theta}_d + u) + h(\theta, \dot{\theta}) \qquad (10a)$$

where

$$u = K(e, \dot{e}) \qquad (10b)$$

is the PD like fuzzy logic part of such computedtorque/fuzzy logic controllers. In equation (10), it is assumed that both $D(\theta)$ and $h(\theta, \dot{\theta})$ are known precisely as dynamical characteristics of particular robotic manipulator. The performance of such controllers depends on the design of the fuzzy logic part as indicated by equation (10b). Genetic algorithms are, however, used in order to design optimally such that a cost function of the form

$$\Gamma = \int_0^T \|e\| \, dt \qquad (11)$$

is minimized when the robotic manipulator performs a specified trajectorytracking task of duration, T. The problem is now to express such fuzzy logic part function $k(e, \dot{e})$ in terms demonstrated. The dynamical behaviour of non red undant robotic manipulators neglecting all elastic effects is governed by non linear Nector matrix vector matrix differential equations of the form

$$\mathbf{D}(\theta)\ddot{\theta} + \mathbf{v}(\theta,\dot{\theta}) + \mathbf{g}(\theta) + \mathbf{f} = \tau \quad (1)$$

In equation (1), $\mathbf{D}(\theta) \in \Re^{n \times n}$ is the inertial matrix, $\mathbf{v}(\theta,\dot{\theta}) \in \Re^n$ is the vector of centrifugal and coriolis torques, $\mathbf{g}(\theta) \in \Re^n$ is the vector of gravitational torques, $\mathbf{f} \in \Re^n$ is the vector of friction torques, $\tau \in \Re^n$ is the vector of actuator torques, and $\theta \in \Re^n$ is the vector of joint angles. If the dynamical characteristics of such manipulators are known precisely, it is then possible to introduce computed torque/PD controller governed by the control law equation of the form

$$\tau = \mathbf{D}(\theta)(\ddot{\theta}_d + \mathbf{u}) + \mathbf{h}(\theta,\dot{\theta}) \quad (2)$$

where

$$\mathbf{u} = \mathbf{K}_1\mathbf{e} + \mathbf{K}_2\dot{\mathbf{e}} \quad (3)$$

In equation (2), $\theta_d \in \Re^n$ is the vector of desired joint angles, $\mathbf{e} = \theta_d - \theta \in \Re^n$ is the vector of joint angle errors, $\mathbf{K}_1 \in \Re^{n \times n}$ is the proportional gains matrix, $\mathbf{K}_2 \in \Re^{n \times n}$ is the derivative gain matrix, and

$$\mathbf{h}(\theta,\dot{\theta}) = \mathbf{v}(\theta,\dot{\theta}) + \mathbf{g}(\theta) + \mathbf{f} \in \Re^n. \quad (4)$$

It is then evident from equation (1), (2) and (3) that, the vector of joint angle errors is governed by the linear vector matrix differential equation of the form

$$\ddot{\mathbf{e}} + \mathbf{k}_2\dot{\mathbf{e}} + \mathbf{k}_1\mathbf{e} = \mathbf{0} \quad (5)$$

This indicates that, the controller gains matrices $\mathbf{K}_1 \in \Re^{n \times n}$ and $\mathbf{K}_2 \in \Re^{n \times n}$ can be readily chosen so as to produce linear second order error dynamics with any required characteristics. In this case, they are frequently selected to be diagonal matrices, for example, in order to have decoupled dynamical behaviour. However, the dynamical characteristics of robotic manipulators are never known precisely. In such cases, the computed torque/PD controllers will not have the form of equation (2) but rather the form

$$\tau = \hat{\mathbf{D}}(\theta)(\ddot{\theta}_d + \mathbf{u}) + \hat{\mathbf{h}}(\theta,\dot{\theta}) \quad (6)$$

where $\hat{\mathbf{D}}(\theta) \neq \mathbf{D}(\theta) \in \Re^{n \times n}$ is the estimated value of inertial matrix and $\hat{\mathbf{h}}(\theta,\dot{\theta}) \neq \mathbf{h}(\theta,\dot{\theta}) \in \Re^n$ is the estimated value of the centrifugal, coriolis, gravitational and frictional vector. It is therefor evident from equation (1), (2), (3), and (6) that the vector of joint angle errors is not governed by equation (5) but rather by equation

$$\ddot{\mathbf{e}} + \mathbf{k}_2\dot{\mathbf{e}} + \mathbf{k}_1\mathbf{e} = \hat{\mathbf{D}}^{-1}(\theta)\,[\{\mathbf{h}(\theta,\dot{\theta}) - \hat{\mathbf{h}}(\theta,\dot{\theta})\} + \{\mathbf{D}(\theta) - \hat{\mathbf{D}}(\theta)\}\,\ddot{\theta}]. \quad (7)$$

Equation (7) clearly shows that diagonally chosen gain matrices embodied in $\mathbf{u} \in \Re^n$ will not produce satisfactory error dynamics in practice. This problem has already been addressed in [9] in which genetic algorithms have been deployed to identify the dynamical characteristics. Besides, adaptive computed torqve controllers have also been proposed in [10] in order to circumvent the problem. However, such adaptive controllers in which real time explicit identification scheme is incorporated are too difficult for practical industrial use. In this paper firstly, a new class

Optimal Design and Robustification of Fuzzy-Logic Controllers For Robotic Manipulators Using Genetic Algorithms

A. Moeini
Assistant Professor
Department of Engineering Sciences
Faculty of Engineering,
University of Tehran

N. Nariman-Zadeh
Assistant Professor

A.Darvizeh
Associate Professor
Department of Mechanical Engineering, Guilan University

Abstract

In this paper, genetic algorithms are used in the design and robostification of various moel-based/non-model-based fuzzy-logic controllers for robotic manipulators. It is demonstrated that genetic algorithms provide effective means of designing the optimal set of fuzzy rules as well as the optimal domains of associated fuzzy sets in a new class of model-based-fuzzy-logic controllers. Furthermore, it is shown that genetic algorithms are very effective in the optimal design and robustification of non-model-based multivariable fuzzy-logic controllers for robotic manipulators.

Keywords

Robotic Manipulators, Control, Genetic Algorithm, Fuzzy Logic, Optimization.

1-Introduction

In recent years, much research efforts have been increasingly expended on the development of high-performance trajectory-tracking controllers for robotic manipulators. In this way fuzzy-logic controllers which have been successfully used in different field of control engineering showed higher performance and robustness, [1-2]. However, there are some inherent difficulties in the design of such fuzzy-logic controllers particularly in the field of robotic manipulators in which the governing dynamical equations are highly non-linear and coupled so that complicated controllers have therefore failed to replace the relatively simple non-model-based and/or model-based PD controllers frequently used in routine industrial applications, [3]. Besides, the need for optimal design of such controllers in fuzzy-logic terms of various predefined objective functions suggests that an effective approach was employed [4]. It was shown in [5] and [6] that genetic algorithms [7-8] provide a very effective means of optimally solving various complex engineering problems particularly in the field of control engineering. Indeed, such hybridization of fuzzy logic algorithms is currently gaining much interest and attention as a powerful means in the intelligent research activities. The optimal design of such fuzzy logic controllers in terms of domains of the associated fuzzy sets and/or fuzzy rules embodied in such controllers can be readily