International Science Index, Computer and Information Engineering Vol:1, No:7, 2007 waset.org/Publication/14858

On Enhancing Robustness of an Evolutionary Fuzzy Tracking Controller

H. Megherbi, A. C. Megherbi, N. Megherbi, K. Benmahamed

***Abstract*—**This paper presents three-phase evolution search methodology to automatically design fuzzy logic controllers (FLCs) that can work in a wide range of operating conditions. These include varying load, parameter variations, and unknown external disturbances. The three-phase scheme consists of an exploration phase, an exploitation phase and a robustness phase. The first two phases search for FLC with high accuracy performances while the last phase aims at obtaining FLC providing the best compromise between the accuracy and robustness performances. Simulations were performed for direct-drive two-axis robot arm. The evolved FLC with the proposed design technique found to provide a very satisfactory performance under the wide range of operation conditions and to overcome problem associated with coupling and nonlinearities characteristics inherent to robot arms.

***Keywords*—**Fuzzy logic control, evolutionary algorithms, robustness, exploration/exploitation phase

I. Introduction

ONE of the key issues in the design of control system is its robustness with respect to the plant uncertainties both structured and unstructured. Robustness property of control system consists in small sensitivity of control performance (stability, accuracy, dynamic performance, etc) to inaccurate model, changes and perturbations. It is common those fuzzy controllers (FLC) are robust to plant uncertainties. As stated in [1-2], this feature arises from the fact that the fuzzy sets allow to an input data with perturbation to belong to the same fuzzy set as the same data without perturbation but with different membership function value. The support's length of membership function associated to fuzzy sets determines the perturbation level affecting the input data that will be accepted as element of the same fuzzy set. Thus, widening the membership function's support can increase the robustness to perturbation. However, the accuracy decreases due to the wider partitioning of the input and output universe of discourse. Therefore, a balance must be found during the design between robustness and accuracy. This problem in general is not computationally tractable with conventional design technique. Robust evolutionary algorithms (EAs) [3-4] are the most suitable candidate to tackle such optimization design problems. However, it was noticed that, in some cases, the EAs could locate the region in which the global optimum exists but they don't converge to this optimum. To alleviate this problem, hybrid algorithms were proposed [5-6]. In these algorithms, EAs are combined with other specific optimization techniques in the way that the EA finds the hills and the specific optimization technique climbs them. An alternative solution without EA hybridization to reach the optimal solution consists in making the exploitation effects dominant once we are in the near optimal region. There are several factors that promote exploitation, for example, the use of creep mutation; the use of elitism strategy with low replacement rate; allowing for significant mutation.

In [7] we introduced an exploration phase and an exploitation phase in the EA to overcome the above-mentioned problem. In the exploration phase, the standard genetic process is performed to explore globally the overall search space. The EA in the exploration phase, performs exploitation of favorable regions of the search space around the neighborhood of the near optimum solution found by the former phase.

It is well known that EA used for fuzzy logic system design need model of the plant to be controlled which can be quantitative or qualitative (neural, or fuzzy or neuro-fuzzy model). This model in general constitutes a nominal model. However, the controller designed once set to work has to deal with the plant affected by structured and unstructured disturbances. The latter are usually modeled as error model. To take into account these disturbances during the design, one can think to use the whole model, i.e., nominal model and error model in the evolutionary design phase. Nevertheless, Introducing error model in earlier generations yields to deterioration of convergence performance. The solution proposed is to add a third phase to the EA to search for the FLC in the vicinity of the optimal FLC issued from the previous phases that can maintain a satisfactory accuracy performance under a wide range of operation conditions.

In this paper, we investigate the use of Integer-coded EA for simultaneous optimization of the Fuzzy Rule Base (FRB) and the Fuzzy Data Base (FDB) for a robust FLC. The choice

Manuscript received July 20, 2006. This work was supported in part by the Mohamed Khieder University of Biskra under Grant J0701/02/07/03.

H. Megherbi is with the Automatic Department, Mohamed Khieder University, Biskra, 07000 Algeria (phone: +213-77-01-93-10; fax: +213-33-74-31-21; e-mail: megherbi_h@ yahoo.fr).

A. C. Megherbi, is with the Electrical Department, Mohamed Khieder University, Biskra, 07000 Algeria. (e-mail: chaouki.megherbi@ieee.org).

N. Megherbi is with Digital Imaging Research Center, Kingston University, Penrhyn road, Kingston upon Thames, Surrey, UK, KT1 2EE. (email: n.megherbi@kingston.ac.uk)

K. Benmahamed, is with Electronics Departement, Ferhat Abbes University, CP19000 Setif, Algeria. (e-mail: Khierben@ieee.org)

of integer coding is done because it has the advantage in reducing the Hamming Cliff effects [8] associated with binary coding and reduces the convergence time since the length of the chromosome is further reduced compared to the binary one [9].

Most of the proposed encoding strategies in the literature restrict adjacent membership function to fully overlap, because allowing partial overlaps during evolution requires the test of the existence of an overlapping between the adjacent membership functions in all the chromosomes. If there is no overlapping in a considered chromosome, then the latter is discarded, repaired or a penalty value is associated to it. In our encoding strategy, the overlaps are coded in the chromosome and evolved by the EA. This fact enforces the partial overlapping between the memberships functions, so that all chromosomes represent valid solutions and there is no need to discard, repair, or penalize invalid chromosomes.

The genetic design of the two joint fuzzy controllers is done simultaneously by considering the input-output pairs of a simulated robot manipulator. In so doing, the coupling effects are considered as an inherent aspect in the robot arm behavior similar to the separate joint motions.

To sum up, the FLC design problem and its robustness enhancement is considered as the problem of fuzzy knowledge base optimization, where we seek to minimize the tracking error and alleviate chattering or avoid drastic changes in the control signal that leads to high stress of the actuators to be controlled. A multi-objective EA is used with weighted-sum approach.

The remainder of this paper is organized as follows. First, a background of robust optimization design is outlined in section II. The application of three-phase EA to FLCs robust optimization design is described in section III. Finally, simulation results and discussions are given in section IV.

II. Robust Optimization Design

Engineering design methods can often be cast in terms of optimization design, where the objective function is maximized or minimized by altering the design parameters while meeting various constraints. However, such approaches suffer from the presence of uncertainties as almost all the other disciplines related to engineering. Sources of uncertainties include, to name but a few, physical measurement limitations, the use of stochastic simulation models, complexity of the phenomena to handle, and implementation effects (discretization, quantization). As consequences of these practically inevitable uncertainties, the optimization design technique yield to a solution design not at the precise point in the design space but somewhere in a neighborhood of the nominal point of interest. Since the practical optimization problems may be very ill conditioned, the resultant design solution can have disastrous consequences once put to work in real world.

Robust optimization design is one way to effectively and efficiently deal with these uncertainties. It is a powerful tool that has already proven its capabilities in several application areas, to minimize the effect of uncertainties in a design solution without eliminating the source of the uncertainties, which is difficult, if not impossible task [10-13].

III. Evolutionary Algorithm For Robust Optimization Design Of A FLC

Evolutionary algorithms (EAs) are frequently reported to be able to cope well with noisy environments. In fact, the latter's are considered as a prime application domain for EAs, and that noise can even be helpful in evolutionary search. Indeed, design solutions that are far apart in the design space may have similar objective function values but may have significantly different sensitivities with respect to uncertainties. Thus, allowing for perturbations and parameter variations in the plant during optimization design is potentially the best means of influencing the robust character of the design.

Fig. 1 shows the structure of the robust evolutionary fuzzy control system for robot manipulators that includes a fuzzy control system (FLS) and a three-phase EA.

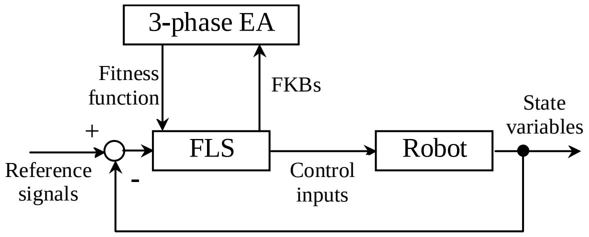

Fig. 1 Configuration of the evolutionary fuzzy control system for robot manipulator

A. Dynamic model of the robot arm

To validate the proposed control scheme for control of robot arms a direct-drive two-axis robot, with two degrees of freedom in the rotational angles q_1 and q_2, is selected as an example problem. Each of its axes is driven by a DC servo-motor which can be described by the following equations:

$$E_a = R_a.I_a + L_a\frac{dI_a}{dt} + K_b.\dot{q} \qquad (1)$$

$$T_m = K_i.I_a \qquad (2)$$

Where E_a is the input voltage, I_a the rotor current, $\dot{q}$ the angular velocity of the joint, and T_m the generated torque. The other parameters and their numerical values are given on table I.

The dynamic equations of motion of the robot are represented by the following coupled nonlinear differential equations:

$$\begin{aligned} T_{m1} = & \left[a_1 + a_2\cos(q_2)\right]\ddot{q}_1 + \left[a_3 + a_2\cos(q_2)\right]\ddot{q}_2 - \\ & \left[a_2\sin(q_2)\right]\left[\dot{q}_1\dot{q}_2 + \frac{\dot{q}_2^2}{2}\right] + a_4\cos(q_1) + \\ & a_5\cos(q_1+q_2) + T_{l1} \end{aligned} \tag{3}$$

$$\begin{aligned} T_{m2} = & \left[a_3 + \left(\frac{a_2}{2}\right)\cos(q_2)\right]\ddot{q}_1 + a_3\ddot{q}_2 + \left[a_2\sin(q_2)\right]\frac{\dot{q}_1}{2} \\ & a_5\cos(q_1+q_2) + T_{l2} \end{aligned} \tag{4}$$

where q_i is the angular position of joint *i*; $\dot{q}_i$ and $\ddot{q}_i$ are the velocity and acceleration of the joint *i* respectively; $T_{l\,i}$ is load torque on joint *i;* T_{mi} is the torque generated by the DC servo-motor; and a_1, ..., a_5 are constant parameters depending on the mechanical parameters illustrated on table II.

TABLE I
DC SERVO-MOTOR PARAMETERS

Parameter	Notation	Joint1	Joint2	Unit
Torque constant	Ki	0.54	0.41	N. m/A
Voltage constant	Kb	5.44	4.19	V/rad/s
Stator resistance	Ra	2.8	10.8	Ω
Stator inductance	La	1.1	2.2	mH

TABLE II
MECHANICAL PARAMETERS OF THE ROBOT

Parameter	Notation	Joint1	Joint2	Unit
Axis length	l	0.29	0.1	M
Distance from the joint to the gravity center	r	0.1	0.04	M
Axis weight	m	1	0.6	Kg
Axis inertia moment	I	0.0974	0.0155	N.m.s^2/rad
Viscous friction constant	D	1.74	0.52	N.m.s/rad

In fact, this example problem is prototypical problem in literature due to the fact that many nonlinear systems of practical interest (such as flexible space structures, twin-lift helicopter systems, power systems consisting of generators, turbine/governor dynamics, and hypersonic flight vehicles) have dynamical equations similar to that of a robot arm, and therefore it seems a suitable example to explore the potentials and limitations of the proposed approach.

It's well worthy to note that the mathematical model described above is used to simulate the dynamic behavior of the robot arm, not to generate the control actions.

B. *Fuzzy Logic Controller*

The SFLC to be evolved by the proposed EA is a MIMO control system including two MISO FLCs. As almost all FLCs set to work nowadays, we have chosen the inputs of our FLCs to be the error x_1 and the change error x_2, on the concerned joint position. At the output, the FLC provides the input voltage *y* to be applied to the DC actuator to bring the corresponding joint in the desired position.

The FLC used in our application can be viewed as a mapping from crisp inputs $\underline{x} = (x_1,x_2)^T \in U \subset IR^2$ to crisp output $y \in V \subset IR$, and this mapping can be expressed quantitatively as $y=f(\underline{x})$ where *f* is non-linear. Let the universe of discourse be $U=U_1 \times U_2$, where $U_1=U_2=$ [-0.05, 0.05], and *V*=[-24, 24].

The FLC consists of the following components:

A ***singleton fuzzifier*** that converts a crisp value $\underline{x} \in U$ into a fuzzy singleton *Ax* within *U* such that:

$$\mu_{Ax}(\underline{x}') = 1 \qquad if\ \underline{x}' = \underline{x} \tag{5}$$

$$\mu_{Ax}(\underline{x}') = 0 \qquad if\ \underline{x}' \neq \underline{x} \tag{6}$$

The ***fuzzy data base***: The space of x_1 is partitioned into three triangular and symmetric membership functions associated to the following labels: negative (N), zero (Z) and positive (P). The space of the second input x_2 and the output y are partitioned into seven membership functions associated to the following labels: negative big (NB), negative medium (NM), negative small (NS), zero (Z), positive big (PB), positive medium (PM), and positive small (PS).

The ***fuzzy rule base*** consists of a collection of fuzzy IF-THEN rules expressed as:

$$R^l:\ \text{IF } (u_1 \text{ is } A_1^l \text{ and } u_2 \text{ is } A_2^l\ ) \text{ THEN } (\ v \text{ is } C^l\ ) \tag{7}$$

where, u_i and *v* are linguistic variables; A_i^l and C^l are terms associated to the fuzzy sets F_i^l and G^l defined in *Ui* and *V*, respectively, with *l* = 1,2,....., *M*. *M* is the number of rules in the FRB. Here we have chosen *M* = 3x7 = 21 to account for every possible combination of input fuzzy sets.

Each fuzzy IF-THEN rule defines a fuzzy implication:

$$R^l = F_1^l \text{ x } F_2^l \rightarrow G^l \tag{8}$$

$$= \{\ ((\underline{u},v),\ \mu_{R^l}(\underline{u},v))\ |\ \ \underline{u} \in U, v \in V\ \} \tag{9}$$

where $\mu_{R^l}(\underline{u},v)$ is defined by the following Larsen's fuzzy implication rule:

$$\mu_{R^l}(\underline{u},v) = \mu_{F_1^l x F_2^l}(\underline{u})\ .\ \mu_{G^l}(v) \tag{10}$$

$$= (\ \mu_{F_1^l}(u_1)\ .\ \mu_{F_2^l}(u_2)\ )\ .\ \mu_{G^l}(v) \tag{11}$$

The ***fuzzy inference engine*** derives from each fuzzy rule of the FRB an output fuzzy set, in the following way:

Each fuzzy rule of (3), described by a fuzzy implication R^l, determines a fuzzy set $B^l = Ax^l \circ R^l$ in *V* such that:

$$\mu_{B^l}(v) = \mu_{Ax \circ R^l}(v) \tag{12}$$

$$= \sup_{\underline{u} \in U} \{\ \mu_{Ax}(\underline{u})\ .\ \mu_{R^l}(\underline{u},v)\ \} \tag{13}$$

The ***defuzzifier*** used in our fuzzy controller is the modified height defuzzifier.

Let v^l denote the center of gravity of the fuzzy set $\overset{l}{B}$, which is associated with the activation of the l^{th} fuzzy rule. This defuzzifier evaluates $\mu_{B^l}(v^l)$ at v^l, then computes the output of the FLC as:

$$y = \frac{\sum_{l=1}^{M} v^l \frac{\mu_{B^l}(v^l)}{\delta^l}}{\sum_{l=1}^{M} \frac{\mu_{B^l}(v^l)}{\delta^l}} \qquad (14)$$

where δ^l is the support's length of the triangular membership function of the consequent for the l^{th} fuzzy rule.

With this components, the FLC is called "*fuzzy system as expansion of FBF: Fuzzy Basis Function*" [14].

C. FLC Parameters to be evolved

Usually EAs are initialized randomly, but if we want to incorporate some knowledge about the problem, we need to introduce that in the initial population of the EA. In our application, we have the following implicit knowledge about the joint FLC design:

The fuzzy partitions along the universe of discourse for the input and output variables are symmetric;

If the inputs are zero, the output should be zero too;

If the inputs of the two fuzzy rules are symmetric, the outputs of these rules should also be symmetric.

Instead of incorporating all of these in the initial population, we propose to make use of them in reducing the chromosome size and hence the convergence time.

Using the first piece of knowledge about the symmetrical aspect of the fuzzy partitions, just the membership functions located in either the positive or negative part of the universe of discourse and the membership function centered at zero need to be coded in the chromosome. Furthermore, it is obvious that the membership function associated to the zero term for each variable must have the center fixed at zero.

The second knowledge give already one fuzzy rule –***if*** *x_1 is Z and x_2 is Z* ***then*** *y is Z*– which must be discarded from evolution. So there's no need to encode it in the chromosome. From the last fact, we imply that we have to search only the half of the FRB and then deduce the other half by symmetry.

To sum up, by taking into account the knowledge about the joint FLC specifications, the chromosome size is reduced to less than a half. In such way the genetic FLC design for robot manipulator with high degree of freedom is possible in a reasonable convergence time.

D. Genotype

With a multi-parameter, concatenated and integer encoding the FKB parameters including those of FRB and FDB of the two joint FLC are coded on the same chromosome "*Ch*" of 60 alleles, Fig 2. Each 30 alleles of the chromosome encode one FKB. The first ten alleles of each such fragment of chromosome encode a FRB. The remaining fragment alleles are used to compute the membership function parameters. The alleles reserved for the fuzzy rules of the FRB are elements of {1,2,….,7} and correspond to the fuzzy sets associated to the output variable. This encoding correspondence is illustrated on table III with gray background and the decoding mapping is given in table IV.

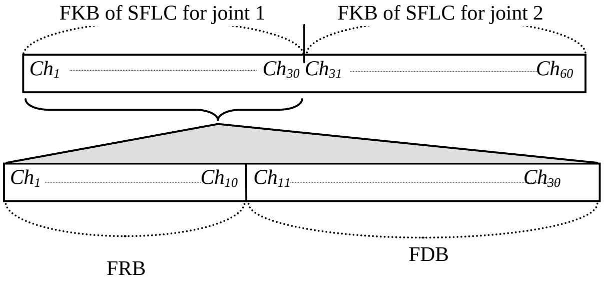

Fig. 2 Schematic representation of the FKB parameters on a chromosome

TABLE III
THE ENCODING STRATEGY OF THE FRB

$x_1\backslash x_2$	NB	NM	NS	Z	PS	PM	PB
N	Ch_1	Ch_2	Ch_3	Ch_4	$8\text{-}Ch_{10}$	$8\text{-}Ch_9$	$8\text{-}Ch_8$
Z	Ch_5	Ch_6	Ch_7	Z	$8\text{-}Ch_7$	$8\text{-}Ch_6$	$8\text{-}Ch_5$
P	Ch_8	Ch_9	Ch_{10}	$8\text{-}Ch_4$	$8\text{-}Ch_3$	$8\text{-}Ch_2$	$8\text{-}Ch_1$

TABLE IV
FRB DECODING

Allele (Ch_i, i=1,….10)	1	2	3	4	5	6	7
Fuzzy Term	NB	NM	NS	Z	PB	PM	PB

Since the membership functions used are triangular and symmetric, we need only two parameters for their description. These parameters are elements of {center, ecart, or overlap $_{\text{left}}$}, Fig. 3. The elements used for each membership function are given on table V and coded on two digits that take the values from 1 to 9.

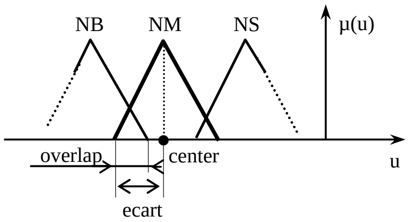

Fig. 3 Description of membership function parameters

E. Genetic Operators

Our algorithm uses tournament selection with replacement to select parents for reproduction and two-point crossover. As mentioned in the above section, the chromosome is integer based instead of binary based and each allele of this chromosome has an integer range according to which FLC parameter it is representing. For example, alleles representing FRB have an integer range from 1 to 7, and those encoding the membership function parameters have an integer range from 1 to 9. The mutation operator thus changes the allele randomly inside its range.

TABLE V
MEMBERSHIP FUNCTION PARAMETERS TO BE EVOLVED

Variables	Linguistic terms	Parameters
Error (x_1)	N	Center, ecart
	Z	overlap
Change-error (x_2) & Input voltage (y)	NB	Center, ecart
	NM	Center, overlap
	NS	Center, overlap
	Z	overlap

F. Fitness Function

The control design problem is formulated as least-squares problem where we seek the controller to minimize the quadratic error while avoiding the drastic changes of the input voltage variable. That's why the fitness function is chosen to have two components: sum of quadratic error and sum of variance of the input voltage variable. These measures are weighted and summed up so that they form a final quality value.

G. Three-Phase EA

The three-phase scheme in an EA consists of an exploration phase, an exploitation phase and a robustness phase. In the exploration phase, the initial population is generated randomly and the standard genetic process is performed to explore globally the search space. The exploitation phase is proposed to improve the smooth and tracking performances of the motion control and ensure fast convergence of the EA. In this phase, the initial population is generated by creep mutating the best chromosome obtained from the exploration phase. A creep genetic search is proceeded after that (i.e., apply the genetic operators: selection, crossover and creep mutation) until a new best chromosome is found. In this case we initialize the population with the same manner as described above but using the new best chromosome. We repeat this process until the satisfaction of the termination criterion. In the sake of obtaining FLC providing the best tradeoff between the two conflicting performances; accuracy and robustness, we introduced a third evolutionary phase named robustness phase. This later work as an exploitation phase but using the robot arm subjected to wide range of uncertainties and free of uncertainties. Doing so, the proposed EA try to evolve FLC to be at the same time accurate and robust toward varying load, unknown external disturbances and parameter variations.

IV. Simulation Results

In this section, we investigate the proposed three-phase EA in the SFLC design for tracking control of direct-drive two-axis robot arm described in section III.A.

The population size, the mutation rate, and the crossover probability were set at 50, 0.03, and 0.8, respectively. The EA was run ten times using different random number generator seeds producing in such a way different initial populations.

The control task is to track the following trajectories:

$$q_{d\ 1} = q_{d\ 2} = 0.75(1-\cos(0.5\pi\ t))\ [\text{rad}] \qquad (15)$$

with the initial states: q_1=0 rad, q_2=0 rad, $\dot{q}_1$=0 rad.s^{-1} and $\dot{q}_2$=0 rad.s^{-1}

The uncertainties are considered to be originated from varying load, parameter variations, and unknown load torque.

During robustness phase, the load increases by 4kg at t=1.5 s while the robot is in motion. In a robot arm we distinguish between mechanical parameters and electrical parameters. The mechanical parameters can not always be obtained accurately and they may also vary during the manipulation. However, the uncertainties on the mechanical parameters can be handled as if they were due to unknown load torque. Thus, in the simulations, only the robustness against electrical variations is considered by using stator resistances as an example. At the beginning, the stator resistances have its nominal values and they change to Ra_1=4.5Ω for the first axis and Ra_2=*20.1*Ω for the second one when t=2.5s. They are assumed to have 100% uncertainty during robust optimization design. It is common that in many cases DC motors may be operated with unknown load torque. Thus, the performance of the robustness of the evolved FLC against the external load torque is enhanced. In the robustness phase, the external load torqueses are set to:

$$T_{l1} = 1.47\sin(\dot{q}_1) + 1.3\sin(\dot{q}_2) + 0.2\sin(t) \qquad (16)$$

$$T_{l2} = 0.6\sin(\dot{q}1) + 0.26\sin(\dot{q}_2) + 0.09\sin(t) \qquad (17)$$

It is of interest to note that these uncertainties constitute an extreme critical operating conditions.

In order to provide a basis for comparison the tracking error and control activities under enhanced FLCs (FLCs evolved by three-phase EA) are plotted against those of non-enhanced FLCs (FLCs evolved by bi-phase EA) which experiences similar operating conditions.

The FKBs that produce the best final objective values are used to get the input voltage with the corresponding tracking error illustrated in Fig. 4, respectively. Obviously, the tracking performance obtained under both controllers was at an excellent level.

The enhanced robustness of the control system resultant from the robust EA was validated by introducing uncertainties different than those used during optimization design. Strictly speaking, The load mass changes as 4.(1-cos(0.2π t)), the stator resistances change to Ra_1=3.5Ω and Ra_2=15.2Ω at t=3s and the external load torques were set to :

$$T_{l1} = 0.37\sin(\dot{q}_1) + 1.4\sin(\dot{q}_2) + 0.156\sin(t) \qquad (18)$$

$$T_{l2} = 0.4\sin(\dot{q}1) + 0.36\sin(\dot{q}_2) + 0.056\sin(t) \qquad (19)$$

Typical results, shown in Fig. 5, illustrate the good performance of the evolved controllers compared to those non-enhanced, when induced with uncertainties.

Additional simulations were performed to determine the influence of the coupling among the joints on the tracking control performance. We set to the robot manipulator other trajectories, where the coupling effects are prominent. These trajectories are described by:

$$q_{d\ 1} = \begin{cases} 0.75\,(1 - \cos(0.5\pi.t)) & t \leq 4 \\ 0 & t > 4 \end{cases} \qquad (20)$$

$$q_{d\ 2} = -0.75\ (1\text{-}\ \cos\,(0.5\pi\, t)) \qquad t \geq 0 \qquad (21)$$

The simulation of the FLCs evolved by the robust EA yield to the results illustrated in Fig. 6, where we can realize that the tracking performances are still at an excellent level.

The simulation results, under the same severe conditions as for the previous trajectories, are presented in Fig. 7.

While the added uncertainties and the coupling effects are clearly evident in the angular position errors, the joint controllers successfully maintain the position errors in acceptable tolerances and without significant chattering in the control inputs.

Apparently, both enhanced and non-enhanced FLCs succeed to follow the desired trajectory in free and noisy environment, but what is of interest to note is the waterbed effect. Indeed when the robustness performance increases the accuracy decreases and vice-versa.

V. Conclusion

As mentioned earlier, the FLCs are known to be robust enough to tolerate plant uncertainties. In the sake of widening its operating conditions, we have proposed a robust optimization design methodology of FLC for robotic manipulator based on a novel three-phase evolutionary algorithm. Robust design search accommodating presence of uncertainty is possible in this algorithm through the third phase termed as robustness phase. Specifically, the robustness to be enhanced is toward varying load, parameter variations and unknown external load torque. Other uncertainty sources have to be investigated in future work.

The enhanced FLC with the proposed EA was found to provide a very satisfactory, if not excellent, performances under a very sever operating conditions and to overcome successfully problems associated with the complex and nonlinearities characteristics inherent to robot arm applications.

By considering the variance of the input voltage of the DC actuators as components of the fitness function, we get a satisfactory smooth behavior at the evolved FLC output.

Future work will focus on two points. The first one is on extending the enhancement of robustness against further uncertainty sources, eg., standstill and stall, measurement noise and startup. The second one consist in using pareto multiobjective approach instead of weighted-sum approach to produce a good set of FLCs that populate the pareto solution set, allowing the controller designer the freedom to choose one controller against others to achieve the desired performance.

References

[1] P. J. C. Branco and J. A. Dente, “An experiment in automatic modeling an electrical drive system using fuzzy logic,” *IEEE Trans. Syst. Man Cybern. C*, vol. 28, pp. 254–262, May 1998.

[2] G. C. Mouzouris and J. M. Mendel, “Dynamic nonsingleton fuzzy logic systems for nonlinear modeling,” *IEEE Trans. Fuzzy Syst.*, vol. 5, no. 2, pp. 199–208, 1997.

[3] B. Forouraghi, “A genetic algorithm for multiobjective robust design,”.*Applied Intelligence*, no. 12, pp. 151-161, 2000.

[4] S. Tsutsui, and A. Ghosh. “Genetic algorithms with a robust solution searching scheme, ” *IEEE Transactions on Evolutionary Computation*, Vol.1, No. 3, pp. 201–208, 1997.

[5] K. Hacker., J. Eddy, and K. Lewis, “Efficient Global Optimization Using Hybrid Genetic Algorithms,” *Proceedings of the 9th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, Georgia,, *2002*.

[6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1991.

[7] H. Megherbi, A. C. Megherbi, K. Benmahammed and A. Hamzaoui, “Design of smooth fuzzy controller for robot manipulators using bi-phase integer-coded genetic algorithm, ” *The seventh International Symposium on Programming and Systems*, May 9-11 2005, Algers, Algeria, pp. 133-144.

[8] Goldberg, D. E. (), Real-coded genetic algorithms, virtual alphabets, and blocking, *Complex Systems*, **5**, pp. 139-67, 1991.

[9] K.C. Ng and Y. Li, "Design of Sophisticated Fuzzy Logic Controllers using Genetic Algorithms", *IEEE World Congress on Computational Intelligence, In Proc. 3rd IEEE Int. Conf. On Fuzzy Systems*, Orlando, FL, June 1994, Vol. 3, pp. 1708-1712.

[10] Parkinson, A., “Robust Mechanical Design Using Engineering Models”, *Journal of Mechanical Design*, Vol. 117, pp. 48-54, 1995.

[11] Hacker, K., “Efficient Robust Systems Design through the Use of Hybrid Optimization and Distributed Computing”, Ph.D. Dissertation, University at Buffalo., 2002.

[12] Kouvelis, P. and G. Yu, “*Robust Discrete Optimization and its Applications”*. Kluwer Academic Publishers, Netherlands, 1997.

[13] Bonnans and A. Shapiro, “Optimization problems with perturbations, a guide tour”. *SIAM Review*, Vol 40, pp. 202-227, 1998.

[14] Wang L.X. and J. M. Mendel, “Fuzzy Basis Function, Universal Approximation, and Orthogonal Least-Squares Learning. *IEEE Trans. neural networks*, Vol. 3, No. 5, 1992.

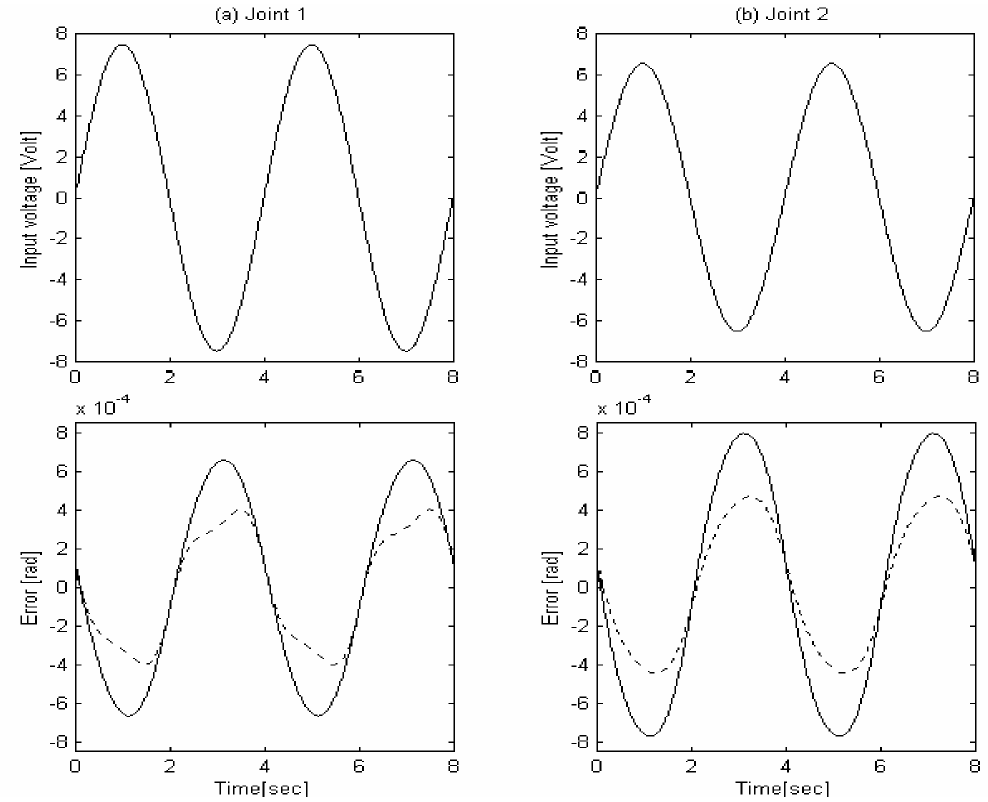

Fig. 4 A Comparison of tracking and smoothness performances of the evolved FLCs using three-phase EA (solid line) and bi-phase EA (dashed line)

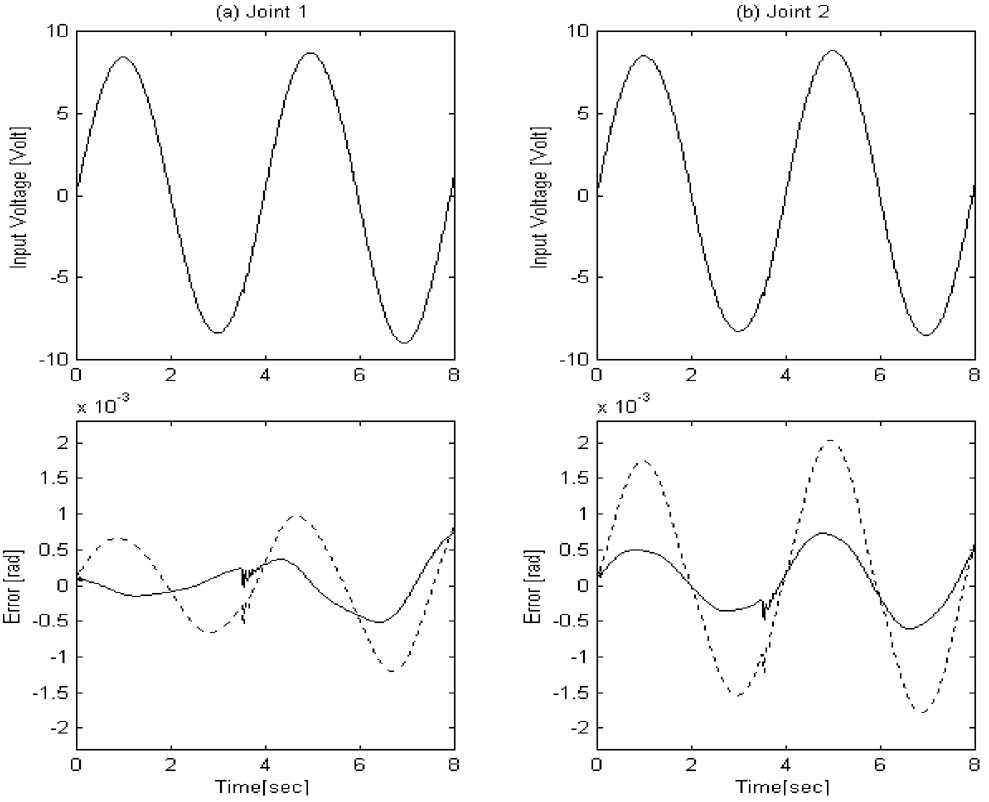

Fig. 5 A Comparison of tracking and smoothness performances of the evolved FLCs under uncertainties using three-phase EA (solid line) and bi-phase EA (dashed line)

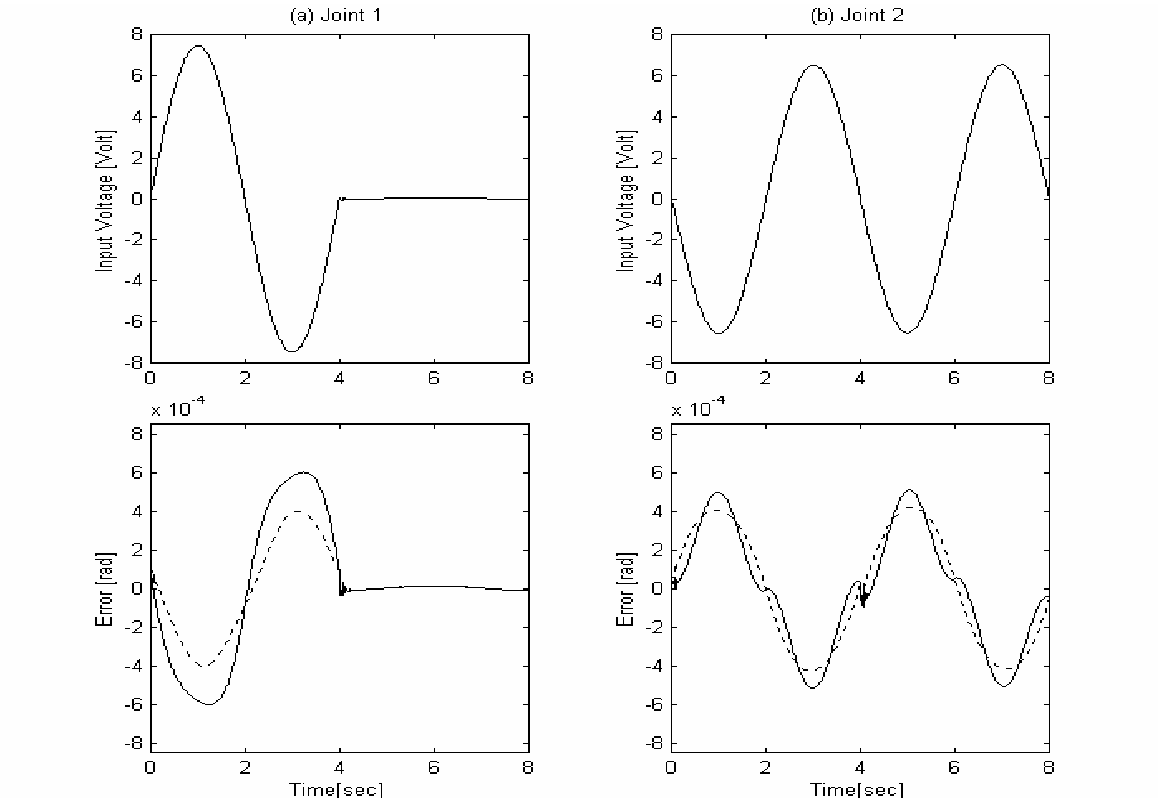

Fig. 6 A Comparison of tracking and smoothness performances of the evolved FLCs with prominent coupling effects using three-phase EA (solid line) and bi-phase EA (dashed line)

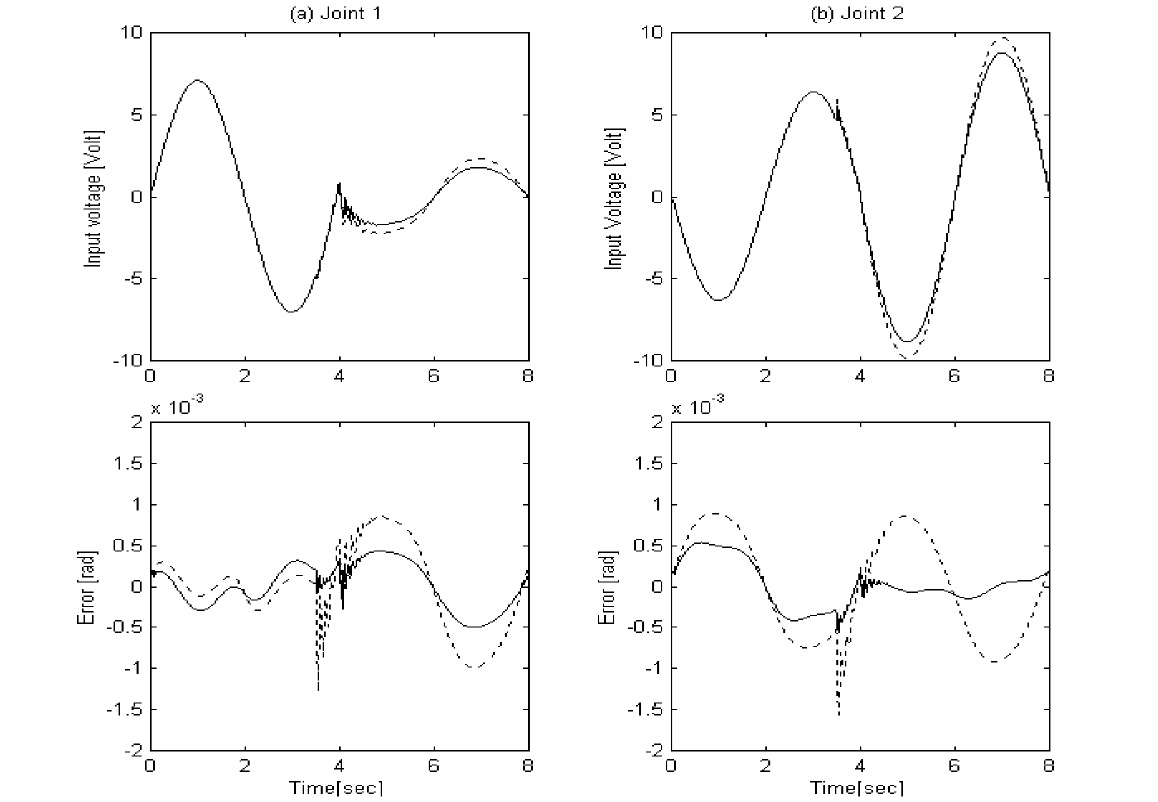

Fig. 7 A Comparison of tracking and smoothness performances of the evolved FLCs under uncertainties and prominent coupling effects