ISSN 2229-5518

Application of Imperialist Competitive Algorithm for Optimum Iterative Learning Control Model

Mahdiyeh Eslami, Reza Seyedi Marghaki, Mahdi Shamsadin Motlagh

Abstract— This study introduce the robust iterative learning control (ILC) model for uncertain single input-single output (SISO) linear time invariant (LTI) control systems. In large cases of industrial robot employments it is a fact that the robot is learned to do the same task repeatedly several times. By looking at the control fault in the independent and several iterations of the same task it becomes obvious that it is actually highly repetitive. The ILC permits to iteratively compensate for and, therefore, cancel this repetitive fault. In this article different issues of iterative learning control are investigated. Although stability margin is the main significant in practice the model aspect is also bolded. some design schemes for iterative learning control techniques are introduced, consisting 1st order as well as 2nd order iterative learning control and parameter balancing. The basic characters of the model are that: The control signal is continuous and the coefficient of controller is optimized. Therefore it is chattering-free compared with the robust ILC using conventional 1st-order sliding mode method. In the suggested method, free variables of controller have valuable figure in the performance of system. Hence we propose Imperialist Competitive Algorithm (ICA) for selecting the excellent value of these parameters. The suggested method is examined on robotic manipulator case and computer simulations show that the proposed method has high performance.

Index Terms— Optimization algorithm, ICA, Control, Chattering, Nature base algorithm

———————————— ◆ ————————————

- *Mahdiyeh Eslami, Department Electrical Engineering Kerman Branch, Islamic Azad University of Iran, m_eslami@srbiau.ir*
- *Reza Seyedi Marghaki, Department Electrical Engineering Kerman Branch, Islamic Azad University of Iran, rezaseyedi1365@yahoo.com*
- *Mahdi Shamsadin Motlagh, Department Electrical Engineering Kerman Branch, Islamic Azad University of Iran , mahdikhan92@yahoo.com*

1 INTRODUCTION

PROBLEMS in control system model can be broadly classified into two categories: stabilization and performance. In the latter category a typical problem is to force the output response of a dynamical system to follow a favor route as near as feasible, where "near" is typically defined relative to a specific norm or some other measure of optimality. Although control theory supply many approaches for attacking such problems, it is not often possible to gain a favor set of performance design requirements. This issue can be because of the presence of vague dynamics or parametric uncertainties that are exhibited through real system operation or to the shortage of sufficient model techniques for a special class of control systems.

Iterative learning control is a approach of tracking control for systems that work in a repetitive state. Examples of systems that operate in a repetitive manner include robot arm manipulators, chemical batch processes and reliability testing rigs. In each of these tasks the system is required to perform the same action over and over again with high precision. This action is represented by the objective of accurately tracking a chosen reference $r\ t(\ )$ signal on a finite time interval.

Repetition allows the system to improve tracking accuracy from repetition to repetition, in effect learning the required input needed to track the reference exactly. The learning process uses information from previous repetitions to improve the control signal ultimately enabling a suitable control action can be found iteratively. The internal model principle yields terms under which perfect tracking can be gained but the design of the control algorithm still leaves many decisions to be made to suit the application.

In last decade, there has been a emerging interest in iterative learning control draft among the automatic control community ways [1- 2]. This technique, which is specially efficient for systems performing repetitive works, includes in selecting an adequate iterative rule which permits the controller to train from the tracking errors of the previous iterations in order to enhance the tracking accuracy with every new learning iteration process. Iterative learning control was firstly introduced as a feed forward action used straightly to the open-loop system [3- 6]. Some closed-loop ILC structures were later introduced in order to benefit from the feedback features in the first iteration, e.g., see references [7- 11].

Many of the ILC techniques suggested in the other papers centralize on the determination of the convergence terms, which is a crucial section of the design and modeling, but there is few research that investigate solving these terms to gain the ILC filters chiefly under model uncertainties. In ref. [8] and ref. [10], the H1 and µsynthesis methods were applied to draft the learning filters, assuming that the feedback controller is already exist. In ref. [7] a two-steps to draft the feedback and training controllers. But, as the researchers pointed out in the previous studies, this technique cannot be actual for unstable systems and the isotropy term may just be satisfied if there is no uncertainty.

The primary grant of designing the feedback and learning controllers in two separate steps is clearly the increased number of degrees of freedom, which allows to assign the favor operation at the 1st iteration through the feedback controller, and the operation of the iterative process through the training filters. The basic practical shortage of this method, besides the increased design complexity and time consuming, is that it leads commonly to very high order ILC filters. In practice, a feedback controller drafted by robust control approaches, such as μ-synthesis, is commonly of a very high order; therefore, robust ILC draft based on the high order feedback controller will lead to higher order ILC filters if those filters available. From a actual view, it is significant to keep the order of the feedback and training filters as minimum as feasible. One potential out is to draft the feedback and training filters concurrently. In practice, in references [12, 13] it is described that if the feedback controller is drafted to satisfy the robust performance term, then the operation weighting function can be applied as a training filter guaranteeing the isotropy of the iterative process. Therefore, there is no need to draft the training filter if a feedback controller may be drafted to satisfy the robust performance term. A quite similar result has also been suggested in reference [14].

In this study, using the imperialist competitive optimization algorithm, we generate a optimized continuous robust ILC design procedure that guarantees robust operation for the feedback control system and the isotropy of the iterative process, for both stable and unstable uncertain LTI control systems. As said before, in continuous robust ILC, free variables of controller have valuable figure in the operation of control system. Hence we propose ICA optimization algorithm for discover the excellent value of these variables.

The suggested system has been actually applied and examined on robot manipulator CRS A465 case. The computer simulation results achieved support excellent convergence characters of the proposed system. In particular, we have illustrated that the anticipative property supplies some additional enhancements to the performance.

This paper is structured as next: in section two, the needed concepts are described. In section three, ICA and proposed method are defined. In section four, illustrative example is employed to show the powerfulness of the proposed chattering free robust ILC system and finally section five conclude the paper.

2 Needed concepts

Ifk□□0,1,....□. A n degree of freedom rigid fully actuated manipulator during may be defined by EulerLagrange equations of the following standard state:

Coriolis powers, and $G\, q(_k)$ is the vector of potential powers.

In this study, we address a problem of tracking a route that is repeated over a given operation time. Throughout the study, the following assumptions and theories will be considered.

Assumption 1. The basic route $q_d(.)$□$C^2[0,T]$ (*i.e.*, $q_d(t)$ is twice continuously differentiable on $[0,T]$).

In particular, this assumptions implies that, given a reference trajectory $q_d(.)$, there exist nonnegative constants $q_{d\max}$, $\dot{q}_{d\max}$, $\ddot{q}_{d\max}$ such that $\sup_{t□□0,T□} |q(t)| = q_{d\,d\max}$, $\sup_{t□□0,T□} |\dot{q}_d(t)|$ □ $\dot{q}_{d\max}$: and $\sup_{t□□0,T□} |\ddot{q}_d(t)|$ □$\ddot{q}_{d\max}$.

Assumption 2. The disturbances $d(t)$ are iteration constant and uniformly limited on □0,T□by some boundary $|k_d$□|0, *i.e.*, $\sup_{t□□0,T□} d(t) = k_d$.

Assumption 3. The joint location of the manipulator is existing for measurement subject to (not large) measurement disturbances. More detail, let $\bar{q}_k$ shows a measured fault described as next:

$$\bar{q}_k(t) □ \tilde{q}_k(t) \;□\; □_{qk}(t) \qquad (2)$$

here $\tilde{q}_k$□ □q_d $\tilde{q}_k$, *and* □$_{qk}$(.) are measurement disturbances at the k□t iteration.

note the following higher-order SISO nonlinear dynamical system defined using following equation: $\dot{x}_i(t)$ □$x_{i□1}$, i=1,....,m-1

$\dot{x}_m(t) \square \square^T(x\,t,)\square(x\,t,) \square b\,x\,t\,u\,t(,)() \square d\,t()$

(3) $y\,t() \square x_1\,t()$ t □[0,T] here the measurable system state

$$M\,q\,q(_k)\ddot{}_k \square C\,q\,q\,q(_k,\dot{}\,\dot{}_k)_k \square G\,q(_k) \square\square_k d$$

Where $q\,t_k() \square^n$, $q\,\dot{t}_k() \square^n$ are vectors of generalized coordinates and generalized speeds respectively, where $k\square\square 0,1,....\square$ is the iteration issue, and $t\square\square 0,T\square,T \square 0$ is the duration of each practice.

□

respectively,

Furthermore, $\square_k \square^n$ is the vector of outside powers,

$M\,q(_k)\square^{n\,n\square}$,$C\,q\,q(_k,\dot{}_k)\square^{n\,n\square}$ and $G\,q(_k)\square^n$ are

(4) smooth matrix-valued (vector- valued) functions of their arguments, $M\,q(_k)$ shows the inertia matrix of the manipulator, $C\,q\,q\,q(_k,\dot{}\,\dot{}_k)_k$ is the vector of centrifugal and

$x(t)=[x\,,\ ....\,,\ x\,]_{1\,m}$,

3 PROPOSED METHOD (ICA-ILC)

A. Imperialist Competitive Algorithm (ICA)

In computer science, Imperialist Competitive Algorithm (ICA) is a computational method that is applied to solve optimization problems of different types. Like most of the methods in the area of evolutionary computation, ICA does not need the gradient of the function in its optimization process. From a specific point of view, ICA may be thought of as the social counterpart of genetic algorithms (GAs). ICA is the mathematical model and the computer simulation of human social evolution, while GAs are based on the biological evolution of species.

(1)$u\,t()$ and $y\,t()$are the control input, $b\,x\,t(,)$ is vague non-zero function, $\square(x,t)$ is a p $1\square$ vague and time varying function to be train, $\square(x\,t,)$ is identified vector-valued function with dimension of p $1\square$. The variable $d\,t()$ present the vague disturbance.

The suggested continuous 2nd-order sliding mode ILC at kth iteration is drafted that described as follow equation:

$$u_k(t) \square b\,(x\,t,)(\square_d\,y_i(t) \square^m_{i\square}\square_d\,x_{i\square,k}^{(\,)i} \square\square\square^{\frown T}(x\,t,)_k \square y_k\,t())$$

$$\square\square_1|\square_k|^{/}{}_{2\,3}\ \mathrm{sgn}(\square_{i\square}\ \square_k^{i\square})\ \square\ _3\ _k\ (\ \ ))t$$

here k assign the issue of iteration, $\square_1$ and $\square_3$ are positive constant.

ICA is a population-based stochastic search optimization algorithm. It has been proposed by Mr. Atashpaz and Mr. Lucas in last decade [15, 16]. Since then, it is applied to solve some kinds of optimization problem. The algorithm is inspired by imperialistic competition of real world. It work to present the social policy of imperialisms to govern more countries and use their sources when colonies are dominated by some international and human life rules. If one empire loses its power, the rest of them will compete to take its possession. In ICA, this process is simulated by particles that are known as countries. This algorithm starts with a randomly primary population and objective function which is computed for the mentioned countries. The number powerful countries are assigned as imperialists country and the others are colonies countries of these imperialists country .Then the racing between imperialists countries arise to get more colonies country .The best imperialist has more luck to possess more colonies. Then one imperialist with its colonies makes an empire. Fig. 1 illustrate the primary populations of each empire country. If the empire is bigger, its colonies are larger and the weaker ones are less. In this figure Imperialist 1 is the most powerful and has the greatest number of colonies (more number country or number of colony).

After dealing colonies between imperialists countries, these colonies close to their respective imperialist countries. Fig.2 illustrate this movement behavior. Based on this concept any colony closes toward the imperialist by a units and reaches its new location and its relative fitness value. Where a is a random variable with uniform (or any proper) distribution, b, a number greater than 1, causes colonies move toward their imperialists from several direction and S is the distance among colony and imperialist as described in next equation:

$$\square\square U(0,\square\square\$) \qquad (5)$$

If after this motion one of the colonies possess more power than its relevant imperialist, they will exchange their location. To begin the competition between empires and all colonies, total objective function of each empire should be computed exactly base on its fitness function that defined before. It depends on objective function of both an imperialist and its colonies. Then the racing begins, the worst empire loses its possession and strong ones try to take it (that weak colony). The empire that has lost all its relative colonies will destroy. At last the most powerful empire will take the possession of other empires and wins the racing and will be empire all over the world of human kinds.

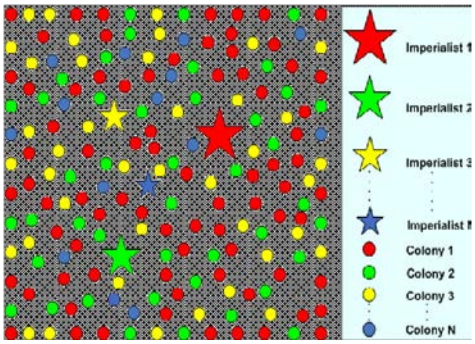

Fig. 1. Generating the initial empire

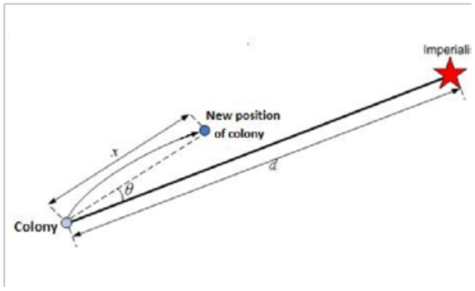

Fig. 2. Moving colonies toward their relevant Imperialist

B. ICA-ILC

In the proposed method the most important parameters of ILC are discovered by optimization algorithm, ICA. In this approach mean square error criteria and minimum iteration numbers are considered as fitness function simultaneously. In fact we are faced with multi-objective optimization problem. Equation. 7 illustrates the fitness function by details.

$$\textit{Fitness}\ \text{function}=w_1\square MSE\ \square\ \square w_2\ \ iteration \qquad (6)$$

Also, Fig. 3 shows the sample country. In fact each country consist of 6 free parameters that must be found.

$$Country\ \square[\square\square\square\square\square\square_{1\ 3\ 1\ \ 2}\ q]_{1\ 6}\square$$

Fig. 3. Sample country

4 SIMULATION RESULTS

In this section, we describe some results of experimental evaluation of the proposed iterative learning control algorithm based on ICA. The algorithm suggested in this article has been implemented and evaluated on a robot manipulator CRS A465OA (see figure 4). The CRS465 is an open chain articulated robot arm [17]. More details about case study can be found in reference [17].

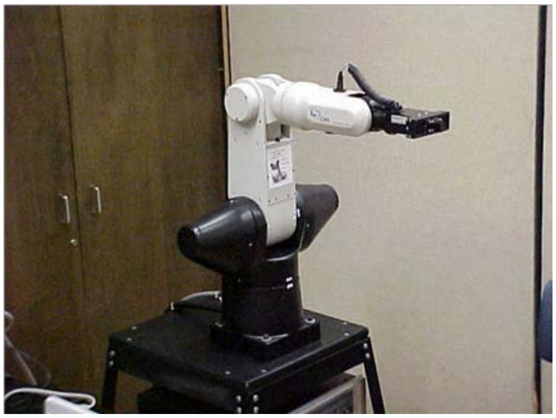

Fig. 4. CRS A465OA Robotic Manipulator

Consider the following equations that define the case study behavior:

$x\,\dot{t}_1(\,)\,□x\,t_2(\,)$

$x\,\dot{t}_2(\,)\,□\frac{1}{\;\;\;\;_2}\quad(u\,□gl.\cos(x_1)\,□d\,t(\,))$

(7) $ml\,□I$ y(t)=$x\,t_1(\,)$

here $\overset{x}{}_1$ is the joint angle, $\overset{x}{}_2$ is the angular speed, m is the mass base on kilogram, I is the torque of inertia, u is the joint point and $d\,t(\,)$ is a disturbance being $0.2\sin(\overset{x\,x}{}_{1\,2})$.Also the system output is $\overset{x}{}_1$ and the favor output route is $\sin(\,)^2 t$. The parameters take the values: $m\,□\,3kg\ l,$ $□1m$ and $I\,□0.5kg\ m.^2$.

Inserting the parameter values, (7) have an new form:

$x\,\dot{t}_1(\,)\,□x\,t_2(\,)$

$x\,\dot{t}_2(\,)\,□□2.8\cos(x_1)□0.2857\,(\,)u\,t\,□0.2\sin(x\,x_{1\,2})$
(8)
y(t)= $(\,)x\,t_1$

Comparing the above system with the considered system (1), we obtain: $b\,□\,0.2857,▱(x\,t,\,)\,□□2.8$ and $▱(x\,t,\,)\,□\,xos\,x(\,_1)$ which is assumed unknown and to be learnt. Following (2), a switching surface is designed as:

$$▱(\,)t\,□c\,e\,t_1(\,)\,□e\,\dot{t}(\,) \qquad (9)$$

Where $c_1\,□1$ and $e\,t(\,)\,□\,\sin(\,)^2 t\,□x\,t_1(\,)$. Taking derivative on the above switching surface yields:

$$\dot{▱}(\,)t\,□c\,e\,t_1\dot{}(\,)\,□\dot{}e\,t(\,) \qquad (10)$$

The proposed robust ILC can be designed as follow:

$$u(\,)_k t\,□b^{□}(x\,t,\,)\{2c_1\sin(\,)\cos(\,)t\ \ t\,□c\,x_{1\,2}(\,)t\,□$$
$$2[\cos^2(\,)t\,□\sin(\,)]^2 t\,□2.8\,▱(\,)t\,□v\,t(\,)\,□$$
$$▱▱_1\ {}_k\Big|^{2\,3}\ \operatorname{sgn}(▱\,▱▱_k)\,□_{3\,k}(\,)\}t \qquad (11)$$

A. Achieved results by simulations

At first step, we select the free parameters of ILC controller with experience. This policy for choosing the parameters left the lot of time for human worker. But, Table 1 shows the found parameters of ILC controller by mentioned strategy (trial and error).

TABLE 1
PARAMETERS OF ILC WITHOUT OPTIMIZATION

▱	q	$▱_2$	▱	$▱_1$	$▱_3$	$▱_1$
0.12	0.11	0.14	0.1	0.1	1	0.12

We first of all test the ILC with the number of trials being 50. The output-tracking errors at 5th , 10th, 40th and 50th trials are shown in Figs. 5 and 6. In Fig.5 the blue lines are the desired trajectory while the red lines are the actual system output. In Fig.6 the black dashed line is the desired trajectory while the other lines are the actual system output in different iterations. Although in Fig.7 the error value of controller is shown. From Figs 5 through 7 it can be seen that with increasing the iterations, controller has better performance.

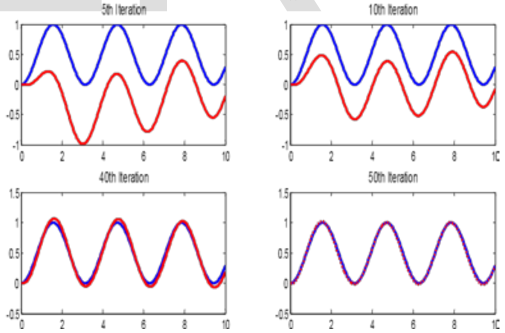

Fig. 5. System output and desired trajectory at different iterations without optimization

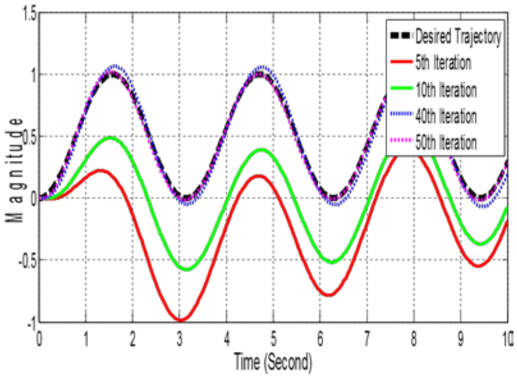

Fig. 6. System output and desired trajectory at different iterations without optimization

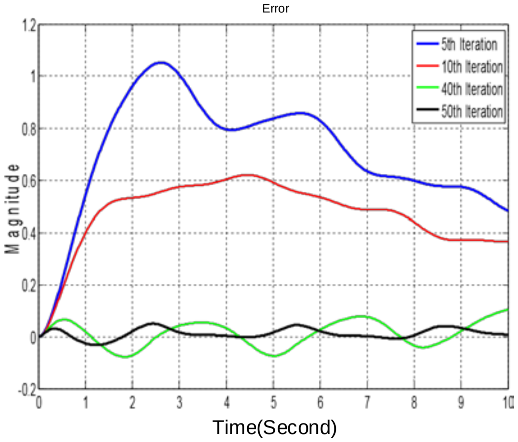

Fig.7. The error of controller in different iterations

Next, we apply ICA optimization algorithm to discover the optimum parameters of the ILC controller. Table 2 shows the ICA parameters. These values gained after some experiments. Table 3 shows the best results for ILC parameters obtained in simulation runs. We first of all evaluate the ILC controller with the number of efforts being 50. The output-route errors at 5th , 10th, 40th and 50th trials are shown in Figs. 8 and 9 respectively. In Fig.5 the blue lines are the favor route while the red lines are the actual or real system output trajectory that obtained by. In Fig.6 the black dashed line is the desired trajectory while the other lines are the actual system output in different iterations. Although in Fig.10 the error value of controller is illustrated. From Figs 8 through 10 it may be seen that with increasing the iterations, controller has better performance. Also achieved results show that optimization significantly enhance the accuracy of the system.

TABLE 2
ICA PARAMETERS

Parameters	Values
N_{pop}	50
N_{imp}	12
□	0.4
□	0.1
□	20
Max iteration	100

TABLE 3

□	q	$□_2$	□	$□_1$	$□_3$	$□_1$
0.181	0.789	0.149	0.141	0.181	0.086	0.145

PARAMETERS OF ILC WITH OPTIMIZATION

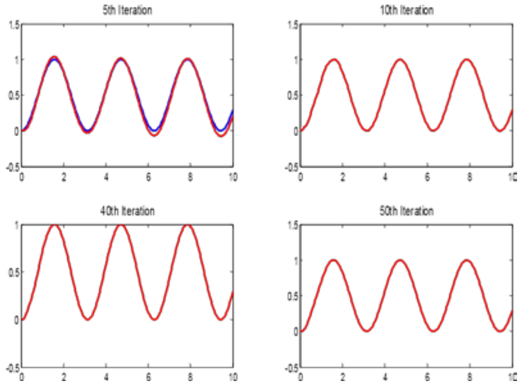

Fig. 8. System output and desired trajectory at different iterations with optimization

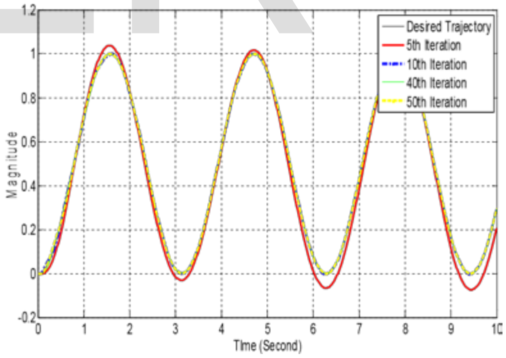

Fig. 9. System output and desired trajectory at different iterations with optimization

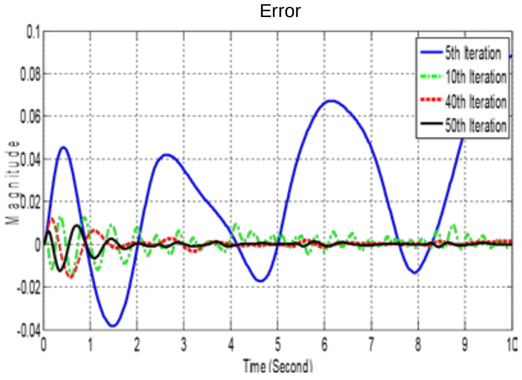

Fig. 10. The error of controller in different iterations

From figs 7 and 10 it can be seen that the optimization had high effect on operation of controller. For simplicity the error of both systems at 50th iteration, i,e, system with optimization and system with no optimization are shown in fig. 11. It can be seen that the optimized system has very excellent operation.

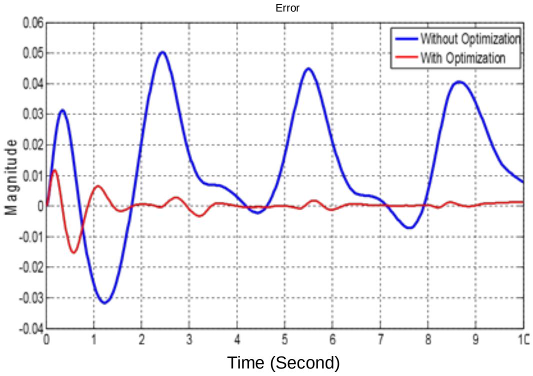

Fig. 11. Error of two systems

B. Evaluate the effects of the parameters of the ICA on the performance of the suggested system

In this part we have investigates the sensitivity of the proposed system with respect to the N_{imp} , N_{pop} ,□,□ and □, which control the manner of ICA, and hence, the effectiveness of the ICA search behavior. The obtained computer results for 10 set of parameters are mentioned in the Table 4. The mean square error (MSE) criteria of best gained value is depicted in this table. It shows that this intelligent system have a little dependency on oscillation of the its parameters.

TABLE 4
PERFORMANCE OF THE ICA-ALC FOR DIFFERENT VALUES OF PARAMETERS

Case	N_{pop}	N_{imp}	□	□	□	MSE
1	50	12	25	0.5	0.6	0.006
2	50	10	25	0.1	0.4	0.0056
3	50	8	25	0.05	0.2	0.0054
4	**50**	**12**	**20**	**0.1**	**0.4**	**0.005**
5	50	10	20	0.5	0.8	0.0058
6	50	8	20	0.05	0.6	0.0053
7	50	10	15	0.5	0.8	0.0055
8	50	8	15	0.05	0.4	0.006
9	50	10	10	0.1	0.8	0.0056
10	50	8	10	0.05	0.6	0.0059

6 CONCLUSION

This article suggested a optimized continuous robust iterative training control policy for class of nonlinear system for the purpose of output tracking. The insight of chattering –free draft is the employment of the second –order sliding mode method that can completely cancel the discontinuity of the control signal. In this system the free parameters of ILC have significant figure in the performance of the system. Hence we propose ICA to find the best values of these parameters. Simulation results show that proposed method has high performance.

REFERENCES

[1] K.L. Moore, M. Dahleh and S. P. Bhattacharyya, “Iterative learning control: A survey and new results,” Journal of Robotic Systems, 9(5), pp. 563-594, 1992.

[2] K.L. Moore, “Iterative Learning Control: An Expository Overview,” Applied and Computational Controls, Signal Processing, and Circuits, Vol. 1, pp. 151-214, 1999.

[3] S. Arimoto, S. Kawamura and F. Miyazaki, “Bettering operation of robots by learning,” Journal Robot. Syst., 1, pp. 123-140, 1984.

[4] Y. Chen, C. Wen, Z. Gong and M. Sun, “An iterative learning controller with initial state learning,” IEEE Trans. on Automatic Control, 44(2), pp. 371-376, 1999.

[5] T.W.S. Chow and Y. Fang, “An iterative learning control method for continuous-time systems based on 2-D system theory,” IEEE Trans. on Circuit and Systems-1: Fundamental theory and applications, 45(4), pp.683-689, 1998.

[6] J.E. Kurek and M. Zaremba, “Iterative learning control synthesis based on 2-D system theory,” IEEE Trans. on Automatic Control, 38(1), pp. 121-125, 1993.

[7] N. Amman, D.H. Owens, E. Rogers and A. Wahl, “An H1 approach to linear iterative learning control design,” Int. Journal of Adaptive Control and Signal Processing, 10, pp. 767-781, 1996.

[8] D. De Roover, “Synthesis of a robust iterative learning controller using an H1 approach,” Proc. of the 35th Conference on Decision and Control, Kobe, Japan, pp. 3044-3049, 1996.

[9] T.Y. Kuc, J.S. Lee and K. Nam, “An iterative learning control theory for a class of nonlinear dynamic systems,” Automatica, 28(6), pp. 1215-1221, 1992.

[10] J. H. Moon , T.Y. Doh and M. J. Chung, “A robust approach to iterative learning control design for uncertain systems,” Automatica, 34(8), pp. 1001-1004, 1998.

[11] J.X. Xu and B. Viswanathan, “Adaptive robust iterative learning control with dead zone scheme,” Automatica, Vol. 36 No. 1, pp. 9199, 2000.

[12] A. Tayebi and M.B. Zaremba, “Robust Iterative Learning Control Design is Straightforward for Uncertain LTI Systems Satisfying the Robust Performance Condition,” IEEE Transactions on Automatic Control, Vol. 48, No. 1, pp. 101-106, 2003.

[13] A. Tayebi and M.B. Zaremba, “Internal model-based robust iterative learning control for uncertain LTI systems,” In Proc. of IEEE Conference on Decision and Control, Sydney, Australia, pp. 34393444, 2000.

[14] Q. Hu, J.-X Xu and T. H. Lee, “Iterative learning control design for Smith predictor,” Systems and Control Letters, Vol. 44, pp. 201-210, 2001.

[15] E. Atashpaz-Gargari, C. Lucas. Designing an optimal PID controller using Colonial Competitive Algorithm. In: Proceedings of the First Iranian Joint Congress on Fuzzy and Intelligent Systems, Mashhad, Iran. 2007

[16] E. Atashpaz-Gargari, C. Lucas. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In: Proceedings of the IEEE Congress on Evolutionary Computation, Singapore 2007; 4661–4667.

[17] J. X. Xu. On iterative learning from different tracking tasks in the presence of time-varying uncertainties. IEEE Trans 34(2004) 589-597