Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

An Improved Clonal Selection Algorithm Based Optimization Method for Iterative Learning Control Systems

Hengjie Li*, Xiaohong Hao*, David Owens **, Lei Zhang*

** School of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, 730050, China (e-mail: lihj915 @163.com).*
*** Department of Automatic Control and Systems Engineering, the University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK*

Abstract: In this paper an improved Clonal Selection Algorithm (CSA) is proposed as a method to implement optimality based Iterative Learning Control algorithms. The strength of the proposed method is that it not only can cope with non-minimum phase plants and nonlinear plants but also can deal with constraints on input conveniently by a specially designed mutation operator. In addiction, because more priori information was used to decrease the size of the search space, the probability of the clonal selection algorithm converging rapidly to a global optimum was increased considerably. Simulations show that the convergence speed is satisfactory regardless of the nature of the plant.

1. INTRODUCTION

Iterative learning control (ILC) is a relatively new addition to the variety of control paradigms which, for a particular class of control problems, can be used to overcome some of the design difficulties associated with more conventional feedback controller synthesis. In more precise terms, iterative learning control is a technique for improving the transient response and tracking performance of processes, machines, equipment, or systems that execute the same trajectory (Arimoto, Kawamura, & Miyazaki, 1984. Moore, 1993), motion, or operation repetitively. ILC is an approach motivated by the observation that if the system's operating conditions are the same each times it executes then any errors in the tracking response will be repeated during each operation. These errors and the control input signal can be recorded during every repetition and this information can be used to compute a new input signal that will be applied to the system during the next repetition or the next trial (here trial is synonymous with repetition or iterative). The idea behind ILC is to calculate the new input signal so that the tracking accuracy would be increased as the number of repetitions increases, i.e. the ILC algorithm ultimately learns the correct input through repetition.

One very useful approach to ILC seems to be to combine ILC with optimisation based techniques. For example, Amann, Owens, and Rogers (1996) proposed a rare algorithm that results in monotonic and geometric convergence for an arbitrary linear time-invariant discrete-time plant combines ILC and optimality. The algorithm is based on posing a suitable optimisation problem during each repetition, and feeding the optimal input into the plant. In order to find optimality based algorithm that are simple to implement, do not require extensive calculation between trials but still result in good convergence properties, in Owens and Fang (2003) and Hätönen and Owens (2003) parameter-optimal ILC algorithm were introduced. These approaches are structurally simple but retain the property that the tracking error converges monotonically. However, most of the algorithms with guaranteed convergence properties work only for linear plants. This is a severe limitation because the dynamics of repetitive systems can be highly non-linear. For this reason it is necessary to derive a new class of ILC algorithms that are able to cope with nonlinearities. Furthermore, in practise process variables are subject to constraints that are set by safety considerations or physical constraints. Hence there is a real need for algorithms that can handle these hard constraints in a straightforward manner.

Recently, Hatzikos and Owens (2002a, 2002b) proposed a genetic algorithm based optimisation method for iterative learning control systems (GA-ILC).The proposed framework has been shown to give good results for linear time-invariant plants. In their following work (Vasilis E Hatzikos, David H Owens and Jari Hätönen, 2003, V.Hatzikos, J.Hätönen and D.H.Owens, 2004), the method was extended to the case where the dynamical system is nonlinear. However, no a priori information was used to decrease the size of the search space in these methods. In addition, the used simple genetic algorithm is not efficient enough and has too many parameters need selected in some instance. As a result, the probability of the algorithm converging rapidly to a global optimum also need increased imminently.

In the current paper, an improved clonal selection algorithm is proposed in an attempt to enhance optimal efficiency in iterative learning control. The rest of the paper is organized as follows: in the next section, we describe the ILC problem in mathematical terms. In section 3, the clonal selection algorithm based optimization method for iterative learning control is presented in detail. Section 4 illustrates the theoretical findings using simulations. Finally, in section 5, some general conclusions are given.

2. ILC PROBLEM DEFINITION

Consider the following possibly non-linear discrete-time dynamical system defined over finite time interval, $t \in [0, T_s, 2T_s, \cdots, T_f]$

$$\begin{cases} x(t+T_s) = f(x(t),u(t),t) \\ y(t) = g(x(t),u(t),t) \end{cases} \quad (1)$$

with a suitable initial condition $x(0) = x_0$ and $T_f = NT_f$ .In addition, a reference signal $y_d(t)$ is specified and the control objective is to design a learning algorithm that will drive the output variable $y(t)$ to track this reference signal as closely as possible by manipulating the input variable $u(t)$. The special feature of the problem is that when the system (1) has reached the final time point $t = T_f$, the state of the system is reset back to x_0, and after resetting, the system is supposed to follow the same reference signal $y_d(t)$ again. This repetitive nature of the problem opens up possibilities for modifying iteratively the input function $u(t)$ so that as the number of repetitions or trials increases, the system learns the input function that gives perfect tracking. To be more precise, the idea is to find a control law

$$u_{k+1} = f(u_k, u_{k-1}, \cdots, u_{k-r}, e_{k+1}, e_k, \cdots e_{k-s}) \quad (2)$$

so that

$$\lim_{k\to\infty}\|e_k\| = 0 \qquad \text{and} \qquad \lim_{k\to\infty}\|u_k - u^*\| = 0 \quad (3)$$

where

$$y_k = [y_k(0), y_k(T_s), y_k(2T_s), \cdots, y_k(T_f)]^T \quad (4)$$

$$u_k = [u_k(0), u_k(T_s), u_k(2T_s), \cdots, u_k(T_f)]^T \quad (5)$$

$$e_k = [y_d(0) - y_k(0), y_d(T_s) - y_k(T_s), y_d(2T_s) - y_k(2T_s), \cdots, y_d(T_f) - y_k(T_f)]^T \quad (6)$$

and u^* is the input function that gives perfect tracking (i.e. we are assuming the reference signal belongs to the range of the plant). Note that if the original plant model is a linear time-invariant model

$$\begin{cases} x(t+T_s) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases} \quad (7)$$

it can be represented equivalently with a matrix equation $y_k = G_e u_k$,where

$$G_e = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{T_l-1}B & CA^{T_l-2}B & \cdots & \cdots & 0 \end{bmatrix} \quad (8)$$

where $T_l = T_f / T_s$. This equivalent representation can typically simplify considerably the convergence analysis of ILC algorithms.

3. THE CLONAL SELECTION ALGORITHM BASED OPTIMIZATION METHOD FOR ITERATIVE LEARNING CONTROL

3.1 The CSA-ILC optimization structure

The ILC theory was introduced independently by several researchers in the beginning of the 1980s. Most important of all was Arimoto et al. who defined the principles that underlie 'learning control'. After that the number of published papers has increased significantly. However, algorithms based on optimality have been proved to be the most popular ones amongst the researchers.

There exist nowadays several optimisation based algorithms that can solve the ILC problem introduced in the previous section when applied to linear systems. One particular approach is the so-called Norm-Optimal ILC method. The basic idea behind this method is to solve the following optimisation problem on-line during iteration:

$$J_{k+1} = \|u_{k+1} - u_k\|^2 + \|e_{k+1}\|^2 \quad (9)$$

with the constraint equation $y_{k+1}(t) = [G_e u_{k+1}](t)$,where G_e is the plant in question. The advantages of this approach are immediate from the simple interlacing result (10) which is a consequence of optimality (it is assumed that (9) has at least one optimal solution) and furthermore from the fact that the choice of $u_{k+1} = u_k$ would lead to the relation $J_{k+1}(u_k) = \|e_k\|^2$ and hence

$$\|e_{k+1}\|^2 \le J_{k+1}(u_{k+1}) \le \|e_k\|^2 \quad (10)$$

in other words the algorithm results in monotonic convergence. If the plant G_e in the constraint equation $y_{k+1}(t) = [G_e u_{k+1}](t)$ is a linear time-invariant (LTI) system, it is straightforward to show that the optimizing solution is given by

$$u_{k+1}(t) = u_k(t) + [G_e^* e_{k+1}](t) \quad (11)$$

where G_e^* is the adjoint operator of G_e . This is non-causal implementation of algorithm but it can be shown that with LTI systems there exists an equivalent causal feedback-law (Aman 1996). Furthermore, in the case of discrete-time LTI system one can show that

$$\|e_{k+1}\| \le \frac{1}{1+\sigma}\|e_k\| \quad (12)$$

where $\sigma > 0$ is the smallest singular value of the plant G_e . Therefore (12) shows that the convergence is in fact geometric for this particular class of plants. However, with nonlinear plants it is not always possible to use the adjoint of the plant to implement the algorithm (the adjoint does not exist or it is not clear how to find an equivalent causal implementation).

Hence in this paper it is suggested that for nonlinear plants the optimization problem (9) is to be solved numerically between trials by using a CSA approach. It is important to understand that if the optimization problem (9) has at least one optimizing solution with the given nonlinear plant, and the chosen CSA method is able to find one of the optimizing solutions, then the interlacing result (10) still holds. Consequently by using the CSA method, it is still possible to achieve monotonic convergence with nonlinear plants.

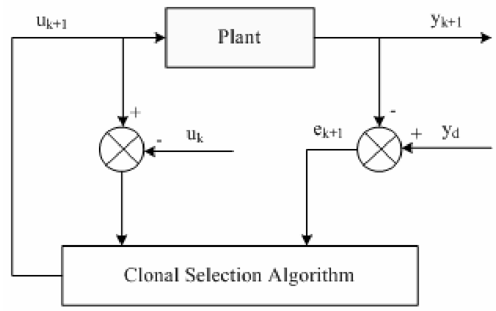

Fig. 1. Plot of the CSA-ILC structure

A block diagram representing the CSA-ILC optimization structure can be seen in Figure 1.All the procedure is coded in

MATLAB's workspace. It should be pointed out that, for each iterative the data of the last memory antibody pool in CSA should be collected, which will be used to initialize population of CSA in next iterative. In this structure, the operation of the system is directly influenced by the CSA. So a efficient CSA become very necessary.

3.2 Improved Clonal selection algorithm in iterative learning control

The Artificial Immune System (AIS) is a new kind of computational intelligence methodologies inspired by the natural immune system to solve real-world problems (D. Dasgupta, 1999). As an important partner of the AIS, the Clonal Selection Algorithm (CSA) methods have been successfully applied to handle challenging optimization problems with superior performances over classical approaches (L. N. de Castro and F. J. von Zuben, 2002. X. Wang, X. Z. Gao, and S. J. Ovaska, 2004). It is based on the Clonal Selection Principle (CSP), which explains how an immune response is mounted, when a non-self antigenic pattern is recognized by the B cells (G. L. Ada and G. J. V. Nossal, 1987). It is an evolutionary process in the natural immune systems, during which only the antibodies that can recognize intruding antigens are selected to proliferate by cloning (L. N. de Castro and J. Timmis, 2002). More precisely, the fundamental of the CSA is the theory that the cells (antibodies) capable of recognizing non-self cells (antigens) can proliferate. The main ideas of the proposed CSA borrowed from the CSP are (L. N. de Castro and F. J. von Zuben, 1999):

- maintenance of memory cells functionally disconnected from the repertoire.
- selection and cloning of most stimulated antibodies.
- suppression of non-simulated cells.
- affinity maturation and re-selection of clones with higher affinity.
- mutation rate proportional to cell affinity.

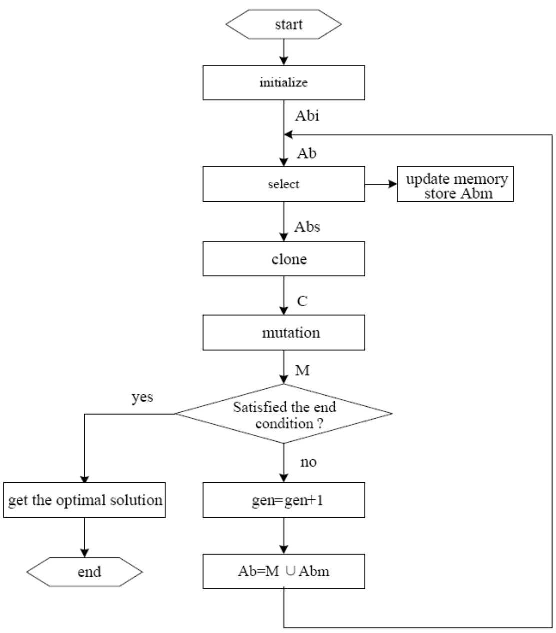

Fig. 2. Block diagram of the improved clonal selection algorithm in iterative learning control.

In order to deal with the optimal problem in ILC, an improved clonal selection algorithm is proposed in the current paper. Its block diagram is shown in Fig. 2, in which the corresponding steps are explained in detail as follows:

1) Initialize: Randomly generate *popsize* individuals composing initial antibody pool (Abi),and define an empty memory antibody pool (Abm).All the individuals in the CSA are real coded and their fitness are measured according to equation (9).

2) Select: select the *numS* best individuals from Abi according to their fitness composing the selected antibody pool (Abs). Update Abm at the same time and make sure it always keeps the best *numM* individuals got by the algorithm.

3) Clone: clone the best individuals in Abs into a temporary pool (C).Each individual in Abs is cloned into same *numC* individuals in C.

4) Gauss mutation: Generate a mutated antibody pool (M) from C. Different to other evaluation algorithms, every individual in C is chosen for mutation in the CSA in this paper. Select a bit from the real coded individual randomly and then replace it by a new bit got according to the following expression:

$$x_i^{'} = x_i + \gamma \cdot (u_i^{\max} - u_i^{\min}) \cdot N(0,1) \tag{13}$$

where $x_i^{'}$ and x_i is old and new gene bit respectively (Because all the individuals in the algorithm are real coded, they are real input value in fact.). γ is mutation factor which is used to restrict the search range. $N(0,1)$ is a Gaussian random variable of zero mean and standard deviation $\sigma = 1$. $u_i^{\max}$ and $u_i^{\min}$ is the largest value and least value of the gene bit .

5) Stop: If the stop criterion is satisfied, then output the result, or else let $k = k + 1$ and $Ab = M \cup Abm$, then return to 2).

The Gauss mutation operation is the key to the success of this algorithm, which is the only way to generate new individuals. The search space is decreased considerably by setting of the border parameters $u_i^{\max}$ and $u_i^{\min}$. All the generated new individuals are must in the range. In addition, the real coded Gauss mutation makes the inputs produced by the CSA smoother.

4. NUMERICAL EXAMPLES

To demonstrate the effectiveness of the Clonal Selection Algorithm in Constrained Linear and Non-linear Iterative Learning Control, numerical simulation results are provided in this section. These examples show that CSA can be effectively used to solve linear and non-linear problems in ILC.

4.1 Constrained non-minimum-phase linear system

In this simulation example we will investigate the proposed algorithm's performance by using for a more demanding dynamical system. The equations describing the dynamical model are:

$$\begin{aligned} x_1(i+1) &= -0.1x_3(i) + u(i) \\ x_2(i+1) &= x_1(i) \\ x_3(i+1) &= x_2(i) \end{aligned} \tag{14}$$

Output is:

$$y = x_1 \tag{15}$$

The ILC algorithm needs to find the input that tracks the following desired output.

$$y_d(i) = 0, \quad i = 1$$
$$y_d(i) = \sin(0.05\pi(i-2)), \quad 2 \le i \le 23 \tag{16}$$

Inequality constraint on input signal is:

$$u_i^{\min} \le u_i \le u_i^{\max} \qquad \mathrm{i} = 1,2,\cdots 23 \tag{17}$$

For this specific problem the CSA-ILC algorithm needs to be able to minimise the following cost function:

$$J_{k+1} = \|u_{k+1} - u_k\|^2 + \alpha\|e_{k+1}\|^2 \tag{18}$$

where the weight α is equal to 0.01. Introducing a weight factor to the input difference part of the objective function we simply provide the CSA with the ability to concentrate in them in imitation of the error rather than executing the easier task of input difference reduction. The parameters of the CSA used during this particular example are shown in Table 1.

Table 1.CSA settings

CSA parameter	Setting
Initial population size (*popsize*)	100
Total generations (*Gmax*)	100
No. of Iterations (*Kmax*)	6
Coding	Real, one number per decision variable
No. of Ab which be selected during per generation (*numS*)	20
Clone factor (*numC*)	5
Elitism	Some best Abs are kept in memory Ab pool (Abm)
No. of Ab in Abm (*numM*)	6
Mutation	Gauss mutation
Mutation factor (γ)	0.1

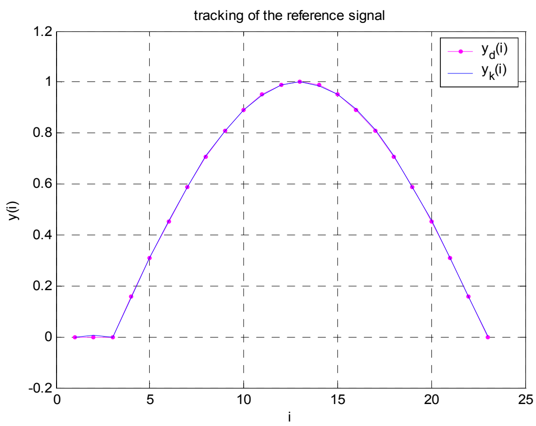

Fig. 3. Tacking of the reference function

Using the above settings, the obtained results were satisfactory. Figure 3 and figure 4 shows the tacking of the reference signal and the value of the error e in logarithmic form where the range of the search space is constrained in $[-1.5,1.5]$ ($-1.5 \le u_i \le 1.5$, $\mathrm{i} = 1,2,\cdots 23$) respectively. They indicate that the CSA-ILC structure is able to find the optimal solution of the optimization problem, and the tracking error converges monotonically to zero fast. Figure 5 and figure 6 shows the tacking of the reference signal and the best input obtained by CSA-ILC where the range of the search space is constrained in $[-0.5,0.5]$ respectively. They indicate that the priori information can be easily coded into CSA to decrease the size of the search space.

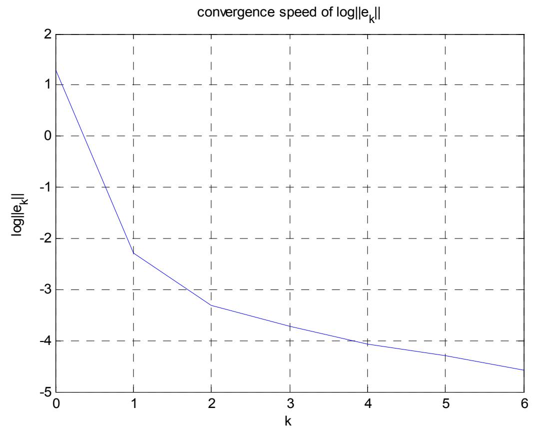

Fig. 4. Convergence speed of the $\|e_k(t)\|$ in logarithmic form

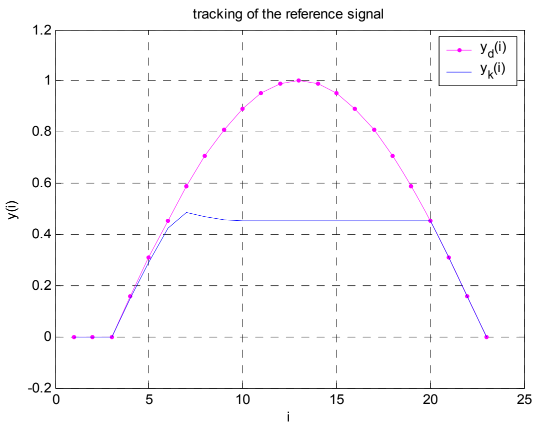

Fig. 5. Tacking of the reference function

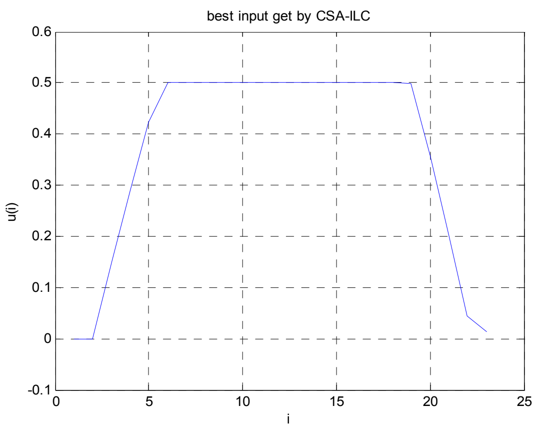

Fig. 6. The best input obtained by CSA-ILC

4.2 Saturated nonlinear industrial control system example

In this simulation example we will investigate the proposed algorithm's performance by using for a more demanding saturated nonlinear industrial control system. Its structure is shown in figure 7. The transaction function of the linear part of the plant is $G(s) = 1/(2s^2 + 2s + 1)$, and the expression of the saturated nonlinear part is:

$$z(t) = \begin{cases} k\beta & u(t) \geq \beta \\ ku(t) & |u(t)| < \beta \\ -k\beta & u(t) \leq -\beta \end{cases} \quad (19)$$

The ILC algorithm needs to find the input that tracks the following desired output.

$$y_d(t) = 1.2(1 - 1/(1+t)^3) \qquad t \in [0,25] \quad (20)$$

Inequality constraint on input signal is:

$$u_i^{\min} \leq u_i \leq u_i^{\max} \qquad i = 1,2,\cdots 25 \quad (21)$$

where $u_i^{\min} = -1.5, u_i^{\max} = 1.5$ here.

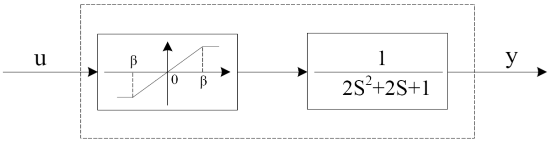

Fig. 7. Structure of the saturated nonlinear control system

The CSA settings shown in table 1 were also used in this example. Figure 8 and figure 9 shows the tacking of the reference signal and the value of the error e in logarithmic form.

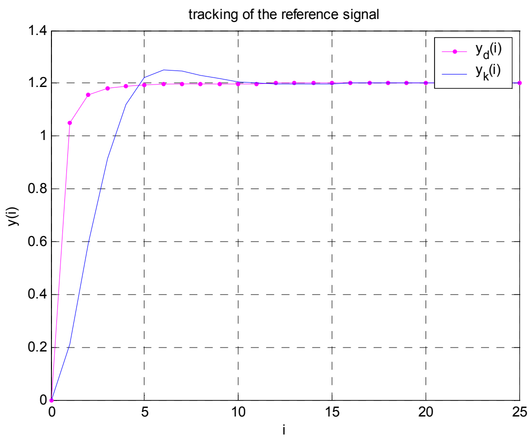

Fig. 8. Tacking of the reference function

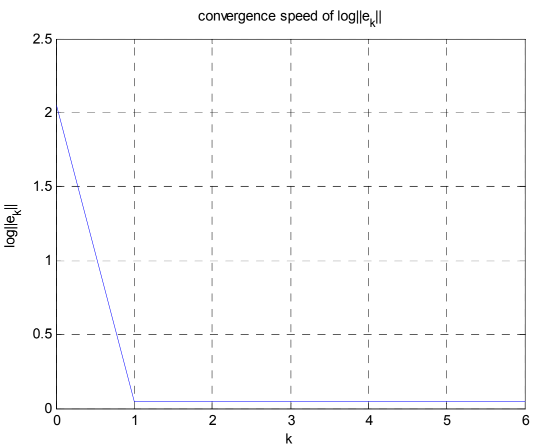

Fig. 9. Convergence speed of the $\|e_k(t)\|$ in logarithmic form

5. CONCLUSIONS

In this paper the possibility of using CSA in solving optimisation problems in ILC was investigated. It is a progress of the GA-ILC method. These projects were motivated by the fact in real world applications the plant to be controlled can be highly nonlinear and it typically constraints on its input signal. The more traditional optimality based ILC algorithms cannot cope with these kinds of plants whereas Evolutionary Algorithms such as GA and CSA on the other hand, have been proven to be able to produce at least a local optimal solution for a highly nonlinear optimisation problem.

The improved clonal selection algorithm instead of SGA is used in this paper because of its better search ability and the more important reason that more priori information was coded in it to decrease the size of the search space. As a result, not only the probability of the optimization algorithm converging rapidly to a global optimum was increased considerably but also the constraints on input signals are easily solved.

Finally, this paper assumes that there is no uncertainty in the plant model, which is an unrealistic assumption in typical applications of ILC. Therefore future work could consider adding feedback from the real plant so that the clonal selection algorithm based optimization method modifies its behaviour based on the information received from the real plant. One way to achieve this could be to use the experimental data from each trial to update the plant model in inside the CSA process. The work has started, and the progress will be reported separately.

REFERENCES

Amann. N, Owens, D. and Rogers, E. (1996). Iterative Learning Control for Discrete-time Systems with Exponential Rate of Convergence. *IEE Proceeding of Control Theory and Applications,*143:217–244.

Arimoto, S., Kawamura, S. & Miyazaki, F. (1984), Bettering Operations of Robots. *Learning Journal of Robotic Systems,*1:123–140.

D.Dasgupta (1999). Information processing mechanisms of the immune system. in *New Ideas in Optimization*, D. W. Corne, M Dorigo, and F. Glover (Ed.), Berkshire, UK: McGraw-Hill.

G.L.Ada and G. J. V. Nossal (1987). The clonal selection theory. *Scientific American*, vol. 257, no. 2, pages 50-57.

J. Hätönen, T. J. Harte, D. H. Owens and J. D. Ratcliffe. A New Robust Iterative Learning Control Algorithm for Application on a Gantry Robot. In: *Proceedings of the IEEE Conference Emerging Technologies and Factory Automation*, pages 305–312, Lisbon.

L. N. de Castro and J. Timmis (2002), *Artificial Immune Systems: A New Computational Intelligence Approach*. London, UK: Springer Verlag.

L.N.de Castro and F.J.von Zuben (1999). Artificial immune systems Part I-Basic theory and applications," Technical Report RT DCA 01/99, FEEC/UNICAMP, Brazil.

L.N.de Castro and F.J.von Zuben (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pages 239-251.

Moore, K. (1993) Iterative Learning Control for Deterministic Systems. Springer-Verlag. Berlin

Owens, D. & Fang, K. (2003), Parameter optimization in Iterative Learning Control. *International Journal of Control,* 76(11):1059–1069

V.Hatzikos, J.Hätönen and D.H.Owens, 2004), Genetic algorithms in norm-optimal linear and non-linear iterative learning control. *International Journal of Control*, vol. 77, no. 2, pages 188-197.

V.Hatzilos, D.Owens (2002a). Introducing Evolutionary based Frameworks into Iterative learning Control Applications. In: *4th Internation Conference on Technology and Automation,* Thessaloniki, Greece.

V.Hatzilos, D.Owens (2002b). A Genetic Algorithm Based Optimization Method for Iterative learning Control Systems. In: *ROMOCO'02,4th Workshop on robot mutation and control,* Poland.

Vasilis E Hatzikos, David H Owens and Jari Hätönen (2003). An Evolutionary Based Optimization Method for Nonlinear Iterative Learning Control Systems. In: *Proceedings of the American Control Conference,* Denver, Colorado, pages 3415-3420.

X.Wang, X. Z. Gao, and S. J. Ovaska (2004). Artificial immune optimization methods and applications–a survey. in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, The Hague, The Netherlands, pages 3415-3420.