Shuffled Frog Leaping Algorithm Research Based Optimal Iterative Learning Control

Hao Xiaohong[1, a], Wang Hua[2, b], Li Zhuoyue[2, b] , Gu qun[1, a]

[1]School of Computer and Communication, Lanzhou University of Technology, Lanzhou 730050, China;

[2] College of Electrical and information Engineering, Lanzhou University of Technology, Lanzhou 730050, China

[a]e-mail: whscdx1126@163.com, [b]e-mail: xtt0112@126.com

KeyWords: iterative learning control; optimization; shuffled frog leaping algorithm; nonlinear; filter

Abstract: To solve the problems of nonlinear and input constraints in the iterative learning control system, using real-coded shuffled frog leaping algorithm to solve optimization problem1 in iterative learning control, .A shuffled frog leaping algorithm(SFLA) based optimal iterative learning control is proposed. The algorithm combines the advantages of memetic algorithm and particle swarm optimization to simplify the algorithm of parameter selection, reduce the search space and improve the convergence rate. The proposed approach benefits from the design of a low-pass FIR filter. This filer successfully removes unwanted high frequency components of the input signal, which are generated by SFLA algorithm method due to the random nature of SFLA algorithm search. Simulation are used to illustrate the performance of this new approach, and they demonstrate good results in terms of convergence speed and tracking of the reference signal.

INTRODUCTION

In automation industry we are facing with systems t-hat perform a certain task over a finite time duration. A sensible example of such systems is a robot manipulator which need to repeat a same tas k over a limited and constant time interval with a high precision. Iterative Learning Control (ILC) [1] is a successful and effective approach for the system above.

Improvement of the efficiency is always the seeking target by people for ILC. An effective approa ch combing optimization technique with iterative learning control, which can improve the learning e fficiency of algorithm. Amann, Owens et al. (1996) put forward a more advance approach which is c alled norm-optimal iterative learning control (NOILC) method. Owens, Fang et al. proposed another Parameter optimal iterative learning control (POILC). NOILC and POILC can be applied into linear system easily.It is necessary to design a new method to cope with nonlinear system. As the input var iables are always constrained, it need an algorithms to deal with constrained problem.

Hatzikos and Owens designed a new method called genetic algorithm optimal ILC (GA-ILC) [2], this new method had a good control result. In order to improve the convergence properties of iterati ve learning algorithm, improve shuffled frog leaping algorithm (SFLA) which is applied into all kin ds of optimization problem. This paper proposed an optimal ILC algorithm to cope with input constr aint of linear system and nonlinear system.

ILC problem description

Considering the linear discrete system model
$$\begin{aligned} x_k(t+1) &= Ax_k(t) + Bu_k(t) \\ y_k(t) &= Cx_k(t) \end{aligned} \tag{1}$$

where k is iteration number. This system is a multi-input and multi-output system. $x_k(t) \in R^p$, $u_k(t) \in R^m$, $y_k(t) \in R^n$ are state vector, control vector and output vector of the system respectively.

$A \in R^{p\times p}$, $B \in R^{p\times m}$, $C \in R^{n\times p}$ are all real matrices. t is the time variable on $[0,T]$.

Iterative learning control is to seek learning algorithms which can revise the input $u_k(t)$ constantly by learning the given reference output $y_d(t)$, make the actual output $y_k(t)$ inching closer to the reference output $y_d(t)$ until it can achieve an almost perfect tracking performance.

Write controlled plant is the first step of norm-optimal iterative learning control

$$y = Pu + d_0 \tag{2}$$

where P is the plant in question. $u \in U$, $y, d_0 \in Y$. U and Y is the input and output space. U and Y are defined as real Hilbert space. The vector form of the input, actual output and reference output can be written as

$$u = [u^T(0) \quad u^T(1) \quad \cdots \quad u^T(N-1)]^T, \quad y = [y^T(1) \quad y^T(2) \quad \cdots \quad y^T(N)]^T,$$

$$y_d = [{y_d}^T(1) \quad {y_d}^T(2) \quad \cdots \quad {y_d}^T(N)]^T \tag{3}$$

It can get the Markov parameter directly from (1), this parameter can be acquired by the impulse r-esponse of the system easily. The Markov parameter P is

$$p_i = CA^{i-1}B, \quad P = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ p_2 & p_1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ p_{N-1} & p_{N-2} & \cdots & p_1 \end{bmatrix}, \quad P_0 = \begin{bmatrix} CA & CA^2 & CA^3 & \cdots & CA^N \end{bmatrix}^T \tag{4}$$

The presentation of $y = Pu + d_0$ can greatly simplified the convergence analysis of ILC. d_0 is used to describe the impact of the non-zero initial state of the actual output. For convenience of analysis, taking $d_0 = 0$. The ultimate goal of all ILC algorithm is to solve the following optimization problem

$$\min_{u\in U} \|e\|^2 \tag{5}$$

where the constraint is $e = y_d - Pu$. For convenience of analysis, it is assumed that all the reference signals are among the range of P. Thus, there must be an optimal input u^* satisfying the target of tracking error $e^* = y_d - Pu^* = 0$.

Shuffled Frog Leaping Algorithm Research Based Optimal Iterative Learning Control

The Convergence analysis of SFLA-ILC and Strcture of SFLA-ILC

Theoretically, if P, y_d is known and y_d is in the range of P, the optimal input u^* can be calculated from $e^* = y_d - Pu^*$. However this case cannot exist in practical problem, we have to find another approach to solve the optimal problem in (5). In norm optimal iterative learning control, the basic idea behind this method is to solve the following optimization on-line during each iteration

$$\begin{cases} \min\limits_{u_{k+1}\in U} J_{k+1}(u_{k+1}) \\ J_{k+1}(u_{k+1}) = \| e_{k+1} \|^2 + \| u_{k+1} - u_k \|^2 \end{cases} \tag{6}$$

with the constraint equation $y_{k+1}(t) = [Pu_{k+1}](t)$ (7)

where P is the plant in question. $e_{k+1} = y_d - y_{k+1}$. The norm in equation (6) can be defined from the induced norm of inner product $<\cdot,\cdot>_Y$ and $<\cdot,\cdot>_U$ which are respectively from the space Y and U. The previous item in equation (6) reflects the basic reason is to minimize the tracking error $\|e_{k+1}\|^2$ and the latter of which is in order to make the input not to change quickly from u_k to u_{k+1}. The advantage of this approach is only need to use input u_{k+1} (it is assumed that (6) has at least one optimal solution) to next iterative process, which can make sure [3]:

$$\| e_{k+1} \|_2 \le J_{k+1}(u_{k+1}) \le \| e_k \|_2 \tag{7}$$

establish, in other words the algorithm results in monotonic decrease in error norm. If the plant P is a linear time-invariant (LTI) system, the optimizing solution is given by

$$u_{k+1} = u_k + P^* e_{k+1} \quad (8)$$

where P^* is the adjoint operator of P. This is a non-causal implementation of algorithm. Furthermore, in the ca-se of invertible discrete-time LTI systems which can show that

$$\| e_{k+1} \| \leq \frac{1}{1+\sigma} \| e_k \| \quad (9)$$

where $\sigma > 0$ is the smallest singular value of the plant P. Therefore (9) shows that the convergence is in fact geometric for this particular class of plants.

Implementation of the algorithm above is for linear system. However, as the adjoint operator ma y not exist or could not find any equivalent causal implementation of algorithm, with non-linear plan ts it is not clear how to use the adjoint operator. In this paper, for the nonlinear problem, it is solved numerically between trails by using SFLA algorithms. It is important to point that if the optimization problem, equation (6) has at least one optimizing solution with the given non-linear plant, and the SFLA method is able to find one of the optimizing solutions, then the interlacing result, equation (7) still holds.

Algorithm is done in the MATLAB workspace, it have to point that the solution should be reserv ed to participate in the next optimization arithmetic, which is solved optimizing problem by SFLA method each time. The solution successfully or not of optimizing problem in the optimal ILC directl y affect the overall performance of the ILC algorithm. It is essential to design a more efficient optimi zation algorithm.

Optimization ILC based on SFLA

Shuffled Frog Leaping Algorithm (SFLA)

Eusuff M and Lansey K[4] proposed a new heuristic swarm intelligence evolutionary algorithm. The algorithm can accomplish the solving problem, which can imitate the collaboration and informa tion interaction from the foraging process of the frog groups in the nature. SFLA method combines t he advantages of Memetic algorithm (MA) based on genetic behavior and particle swarm optimizati on (PSO) based on social behavior [5]. SFLA method was used to solve some complex problems [6-7] and had good result..

The specific steps of SFLA are as follows

1) Initialization: Generate F solutions (Frogs) randomly, that is to generate F input quantities u_j rand omly as the initial population.

2) Division of sub-populations: calculate the fitness of each solution (11), keep a record of the best s olution u_g of the cost function (11)which is also the best solution of the whole population.

Divide the population into m sub-populations, each sub-population has n solutions, that is $F = m \times n$.

Then, put the first solution into the first sub- population, put the second solution into the second sub- population until put the m solution into the m sub- population. Next, put the $m+1$ solution int o the $m+1$ sub- population, by that analogy, until all solutions have been allocated. Take record of t he best solution u_b and the worst solution u_w of the cost function in each sub-population.

3) Local Search: in the process of each sub-population evolutionary, it is just need to update the loca tion of the worst solution each time (15), the following is the update formula

$$D_i = r_1 \times (u_b - u_w) + r_2 \times (p_w - u_w) \quad (10)$$

$$u_w = u_w + D_i, -D_{\max} \leq D_i \leq D_{\max}, i = 1, 2, ..., N \quad (11)$$

formula (10) is to calculate the moving step size D_i, $r1$ and $r2$ is the random number between 0 an d 1. p_w is the best location of u_w; formula (11) update u_w, modify the position solution. $D_{\max}$ is the maximum step size. If it can get a better solution, replace the worst solution with the better one. If n ot, calculate new solution again replacing u_b in formula (10) with u_g . Generate a new solution to rep lace the worst solution if it still cannot get a better solution. This process is repeated until it come to the predetermined evolution algebra N. Then the local search is completed.

4) Mixed operation: reformat a new complete population having F solution by mixing all the sub-po pulations. It have to resort the solution, divide population into sub-populations and go to the next ro und of local search. Re-peat this process until it can achieve the satisfied conditions.

SFLA transfers the information according to the population classification. It combines with the l ocal search effectively. Owing to these, it has a high effective calculative ability and global informat ion ability. Because it can set the maximum and minimum values $u_j^{\max}$ and $u_j^{\min}$ according to the prior i information of the controlled plant, it can reduce the search space of SFLA algorithm and calculati ng cost.

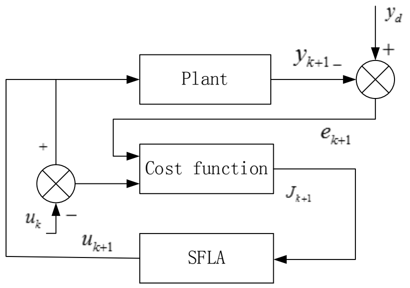

Figure 1 The structure chart of SFLA-ILC

Figure 1 shows the structure of SFLA-ILC. In the design of the algorithm, $J_{k+1}(u_{k+1})$ is the cost fu nction of the population. $e_{k+1}(t)$, $u_k(t)$, $u_{k+1}(t)$ which are generated by ILC. Each frog is real encoded of SFLA. The optimal input produced by SFLA search, which should be preserved to $J_{k+1}(u_{k+1})$ functi on participating in next SFLA search.

In the actual implementation of SFLA-ILC appears to be very robust and has a good tracking per formance for various classes of dynamical systems. The input produced by SFLA is a noisy input, as it will be demonstrated in the next section. To avoid this problem, it can use a low pass filter to smo oth the inputs from the SFLA search without impacting on the overall performance of the method.

Simulation analysis

To verify the effectiveness in solving the input with constraints of linear and nonlinear system co ntrol problem, this section select three actual systems to simulate.

Simulation of linear system

Use the following linear system

$$\begin{aligned} x_1(i+1) &= -0.1x_3(i) + u(i) \\ x_2(i+1) &= x_1(i) \\ x_3(i+1) &= x_2(i) \end{aligned} \tag{12}$$

Output:

$$y = x_1 \tag{13}$$

The cost function of SFLA-ILC:

$$J_{k+1} = \| u_{k+1} - u_k \|^2 + \alpha \| e_{k+1} \|^2 \tag{14}$$

$\alpha = 0.01$. The free parameter of the SFLA used for this example are shown in table1.

Table1. Parameter setting of SFLA-ILC

SFLA parameter	Setting
Population size	100
number of memes group(m)	10
evolutional generations of inner memes group (M)	20
Maximum step $D_{\max}$	5
Iterative generations	100
encoding scheme	real-code

Design of a low pass FIR filter

Figure 2 (imaginary line) the optimal input clearly looks ‘noisy’, and this input function cannot apply into a ‘real’ system. A possible explanation for this is that the SFLA search process is designed to be random. Further-more, the decision variables in the SFLA process are the values of the input function $u^*_{k+1}(i), i \in [1, 2, \cdots 23]$. Benefit to these two facts, it is expected that the two arbitrary points $u^*_{k+1}(i)$ and $u^*_{k+1}(i+s)$ are uncorrelated to a certain degree, resulting in a ‘noisy’ input function.

As an important observation, note that the noise is significant to $u_{k+1}(i)$, but does not typically have a important effect on the output or tracking error if the original plant model is a low pass filter. It can be argued that quality if the input solution can be improved by the use of a low pass filter with a bandwidth equal to the bandwidth of a ‘real plant’. This will result in a smooth input $u_{k+1}(i)$ with the error unchanged. To verify this assumption, the noisy input in figure 2 has been filtered with the following low pass filter

$$u^F_{k+1}(i) = \alpha[u_{k+1}(i-2) \quad u_{k+1}(i-1) \quad u_{k+1}(i) \quad u_{k+1}(i+1) \quad u_{k+1}(i+2)]^T \tag{15}$$

where $u^F_{k+1}(i)$ is the filtered input. $\alpha = [0.2 \quad 0.2 \; 0.25 \quad 0.2 \quad 0.15]$. It could be seen this filter looks non-casual. However, as the input $u_{k+1}(i)$ is available before filtering, it is possible to make $u^F_{k+1}(i)$ to be a function of $u_{k+1}(s)$ for $s \geq i$. Figure 2 also shows that the filtered input function (solid line), the input is applied to the ‘real’ plant during iteration $k = 6$ resulting in the nearly perfect tracking in figure 3. Next section will give the tracking performance and tracking error convergence behavior.

Non-minimum phase linear system simulation

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

Input constraint
$$-1.5 \leq u_k(i) \leq 1.5, k = 1, 2, \cdots 23 \tag{17}$$

Figure 3 and figure 4 give out the actual tracking performance and tracking error convergence behavior when system input constraint is the duration of [-1.5, 1.5].

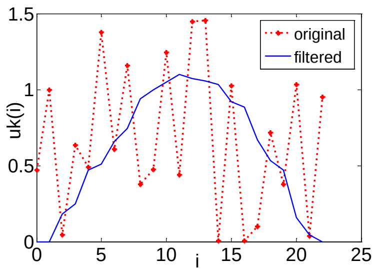

Figure 2 Original and filtered input at k=6

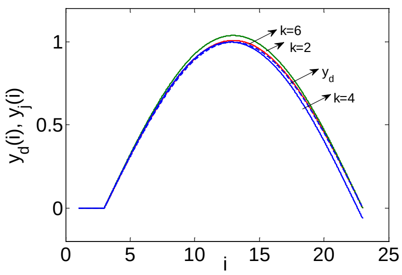

Figure 3 Tracking performance of sine function

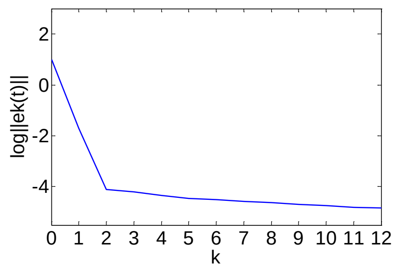

Figure 4 Convergence curve of $\log \| e_k(t) \|$

Figure 2 shows that the input produced by SFLA search commendably limited to the constraint interval. Figure 3 and figure 4 show it can nearly trace the reference output at 6th iteration by SFLA-ILC algorithm. SFLA algorithm can solve the optimal problem in the ILC effectively. It can implement the tracking error converge monotonically by SFLA-ILC. Benefit from the prior information encoded in the algorithm, it can greatly reduce the search space and save the cost of operation.

Simulation of saturated nonlinear industrial control system

A class of nonlinear saturated of industrial control system consists saturated linear part and nonlin-ear part.

Saturated linear part
$$z(t) = \begin{cases} k\beta & u(t) \geq \beta \\ ku(t) & |u(t)| < \beta \\ -k\beta & u(t) \leq -\beta \end{cases} \tag{18}$$

The transfer function model of linear part
$$G(s) = \frac{1}{2s^2 + 2s + 1} \tag{19}$$

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

The free parameter of the SFLA used for this example are shown in table 1.The filter (15) is appl ied to $u_{k+1}(t)$ produced by SFLA. where $\alpha=[0.15 \quad 0.2 \quad 0.3 \quad 0.2 \quad 0.15]$.

Figure 5 and figure 8 give out the actual tracking performance and tracking error convergence be h-avior of nonlinear system.

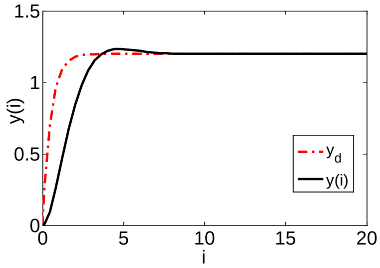

Figure 5 Tracking performance of nonlinear system

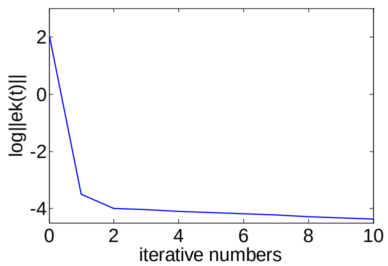

Figure 6 Convergence curve of nonlinear system

Figure 5 and figure 6 show it has a good dynamical performance by using SFLA-ILC. However, t -he constrained input and the characteristics of the controlled plant, which lead to actual output does not have a perfect tracking.

Conclusion

In this paper, the shuffled frog leaping algorithm is improved and applied to the iterative learnin g control, then the optimization iterative learning control algorithm based on shuffled frog leaping al gorithm is put forward. Comparing with traditional optimization method, SFLA-ILC has an advanta ge that it can deal with nonlinear an-d input constraint problems directly. SFLA-ILC combines MA and PSO, the advantages of it is less algorithm parameters, faster calculating speed, a good global s earch and capable to cope with the problems on the input constraints. The low pass filter is used to r emove the undesired frequencies from input produced by SFLA and the real code of algorithm, these two facts make the tracking curves more smooth.

Acknowledgments

This work is supported by National Nature Science Foundation (NO.61263008).

Author: Hao Xiaohong (1960). Male. Professor. Research Area: Automatic control on electrical power system, Advanced Process Control.

Wang Hua (Corresponding author). Master. Research Area: learning control.

References

[1]Suguru Arimoto, Sadao Kawamura, and Fumio Miyazaki, Bettering operation of robots by learni ng[J], Journal of Robotic Systems, 1984, 1(2):123-140.

[2]V Hatzikos, J Hätönen, D H Owens. Genetic algorit-hms in norm-optimal linear and non-linear it erative learning control. International[J] Journal of Control, 2004, 77(2): 188-197.

[3]AMANN N,OEWNS D H,ROGERS E. Iterative Learning Control for Discrete-time Systems wit h Expo-nential Rate of Convergence[J]. Proceedings of the IEE Control Theory Applications, 19 96, 143(2): 217-224.

[4]Eusuff M, Lansey K.E. Optimization of water distribution network design using shuffled frog lea ping algorithm[J]. Journal of Water Resources Planning and Management. 2003, 129(3): 210-225.

[5]Elbeltagi E, Hezagy T, Grierson D. Comparison among five evolutionary based optimization alg-orithms[J]. Advanced Engineering Informatics. 2005, 19(1): 43-53.

[6]Li Ying-hai, Zhou Jian-zhong, Yang Jun-jie, et al. A modified shuffled frog leaping algorithm bas

ed on the threshold selection strategy[J]. Computer Engineering and Applications, 2007, 43(35): 19-21.
[7]Amiri B, Fathian M, Maroosi A. Application of shuffled frog leaping algorithm on clustering[J]. Int J of Advanced Manufacturing Technology, 2009, 45(1/2): 199-209.
[8]Zhang X C, Hu X M, Cui G Z. et al. An improved shuffled frog leaping algorithm with cognitive behavior[C]. 7th WCICA Conf. New York: IEEE Press, 2008: 6179-6202.