A new fuzzy iterative learning control algorithm for single joint manipulator

MENG WANG, GUANGRONG BIAN and HONGSHENG LI

This paper present a new fuzzy iterative learning control design to solve the trajectory tracking problem and performing repetitive tasks for rigid robot manipulators. Several times' iterations are needed to make the system tracking error converge, especially in the first iteration without experience. In order to solve that problem, fuzzy control and iterative learning control are combined, where fuzzy control is used to tracking trajectory at the first learning period, and the output of fuzzy control is recorded as the initial control inputs of ILC. The new algorithm also adopts gain self-tuning by fuzzy control, in order to improve the convergence rate. Simulations illustrate the effectiveness and convergence of the new algorithm and advantages compared to traditional method.

Key words: iterative learning control, fuzzy control, fuzzy gain adjustment, single joint manipulator.

1. Introduction

Iterative learning control (ILC) is widely used in the field of robot control. Most of industrial robots repeat an identical task. It is not difficult to see the repeated error under the same operating conditions. ILC starts from that inspiration. The main concept of ILC is to update control inputs or desired trajectories with previous error. As a result the tracking error of motion is reduced to be zero [1]. ILC is also a branch of intelligent control. Because of its simplicity and efficient, ILC gets wide attention by control bound, especially in the settlement of a class of dynamic position system with strong nonlinear coupling and high position repeatability (such as industrial robots, CNC machine tools, etc.), compared with other control methods which have the features and advantages of simpler structure and better control effect [2].

Since ILC method was proposed by Uchiyama and presented as a formal theory by Arimoto et al., this technique has been the center of interest of many researchers over the last decades [3]. After nearly thirty years' development, ILC has made great progress. At the same time, other control theories are applied to ILC. Paper [4] combines fuzzy

control with ILC with error gain self-tuning by fuzzy control. Papers [5-6] adopt adaptive control based on ILC for the trajectory tracking in order to cope with the unknown model parameters and disturbances.

Robot is a highly nonlinear and strongly coupled complex system. In most cases, the kinetic model is not entirely known. This brings some difficulties to the design of controller. Fuzzy control can overcome the difficulties caused by the uncertainty. However, there exist human factors in fuzzy control, especially in the design of fuzzy rules. ILC can provide high precision trajectory tracking for robot control, but it need several times' iterative learning to make tracking error convergence. Before system error converging, especially in the first iteration without experience, accuracy of trajectory tracking is low. In view of respective characteristics of these two control method, this paper presents a new fuzzy iterative learning control algorithm that switch between fuzzy control and ILC. Fuzzy control is used to tracking trajectory at the first learning period, and the outputs of fuzzy control are recorded as the initial control inputs of ILC. In addition, fuzzy logic system (FLS) is also used to regulate the gain matrix by adjusting learning factor according to system error information, in order to overcome disadvantage of lower convergence rate because of fixed gain matrix.

2. Iterative learning control

ILC can improve a control task by iterative correction. The improvement scheme is fully depends upon the algorithm part. By the proper selection and application of the control algorithm for the ILC technique we can get the desired task to be resolved in a faster manner [7]. A traditional P-type ILC updates the control input as a function of the previous stored control input and the stored output error, whereas a D-type ILC updates the control input as a function of the previous stored control input and the stored derivative of the output error [8]. In this paper, a close D type ILC is adopted, and convergence analysis is given.

2.1. Basic principle of ILC

Given following nonlinear time-varying system:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t), t) \\ y(t) = g(x(t), u(t), t). \end{cases} \tag{1}$$

ILC can be described as: within a finite time interval $[0, T]$, given expected response $y_d(t)$ and initial state $x_d(t)$, find a control quantity $u_k(t)$ that can contribute the system to obtaining output response $y_k(t)$ which is optimized compared to $y_{k-1}(t)$. If $k \to \infty$, and $y_k(t) \to y_d(t)$ then ILC is convergent.

Considering equation (1), using $x_k(t)$, $y_k(t)$ and $u_k(t)$ to represent state variable, output variable and control variable, the kth iteration operation can be described as:

$$\begin{cases} \dot{x}_k(t) = f(x_k(t), u_k(t), t) \\ y_k(t) = g(x_k(t), u_k(t), t). \end{cases} \tag{2}$$

The control input is updated iteratively in a certain way by using the error measurements in the previous operation [9]:

$$u_{k+1}(t) = L(u_k(t), e_k(t), t) \tag{3}$$

where

$$e_k(t) = y_d(t) - y_k(t). \tag{4}$$

This may cause that the system output $y_k(t)$ gradually approaches the given reference trajectory $y_d(t)$. That is $\lim_{k \to \infty} y_k(t) = y_d(t)$.

The function of ILC has much to do with iterative learning law that is composed of control input and output error. The basic iterative learning law can be expressed as:

$$u_{k+1}(t) = u_k(t) + L(e_k(t), t). \tag{5}$$

The common forms of the learning law are: D type with learning law $u_{k+1}(t) = u_k(t) + \Gamma \dot{e}_k(t)$, P type with learning law $u_{k+1}(t) = u_k(t) + \Gamma e_k(t)$ and other forms, such as PD type, PI type and PID type.

Equation (5) is open loop iterative learning law, which uses the control input and output deviation of kth iteration to conduct control input of $(k+1)$ iteration. While close loop iterative learning law uses the control input of kth iteration and output deviation of $(k+1)$ iteration to conduct control input of $(k+1)$ iteration. So the close type iterative learning law of equation (5) can be wrote as:

$$u_{k+1}(t) = u_k(t) + L(e_{k+1}(t), t). \tag{6}$$

2.2. Convergence analysis of ILC

In this paper, ILC for nonlinear systems is studied. Consider the following nonlinear system:

$$\begin{cases} \dot{x}_k(t) = f(t, x_k(t) + B(t)u_k(t) \\ y_k(t) = C(t)x_k(t.) \end{cases} \tag{7}$$

According to the given system, control target is to design a nonlinear ILC that make the system continuous operating according to the iterative learning law. When $k \to \infty$, the system has $u_k(t) \to u_d(t)$ and $y_k(t) \to y_d(t)$.

In this paper, we adopt close D type iterative learning law that is described as:

$$u_{k+1}(t) = u_k(t) + \Gamma \dot{e}_{k+1}(t). \tag{8}$$

In order to realize above control target, the following assumes are given to restrict ILC system of equation (7) (8) [10]:

Assumption 1 *Function f is globally uniformly Lipschiz condition in x on $[0,T]$ in the sense of $\|f(t,x_1) - f(t,x_2)\| \leqslant M(\|x_1 - x_2\|)$, $\forall t, x_1, x_2$ where M is a positive constant.*

Assumption 2 *Expected trajectory is continuous for all $t \in [0,T]$.*

Assumption 3 *$B(t)$ and $C(t)$ are bounded for all $t \in [0,T]$. There exist $\dot{C}(t)$ and it is bounded.*

Assumption 4 *There exists one and only one $u_d(t)$ that makes system state become and system output become expected value.*

Assumption 5 *Matrix $(I + \Gamma(t)C(t)B(t))$ is invertible for all $t \in [0,T]$.*

A system that meets the above assumptions is consistent with ILC design premise. On the basis of the above assumptions, the convergence of the control is proved. When the nonlinear system satisfies the above hypothesis, the following theorems and lemma are established.

Lemma 1 *If the operator $Q : C_r[0,T] \to C_r[0,T]$ satisfied the following conditions:*

$$\|Q(x)(t)\| \leqslant M(q + \int_0^t \|x(s)\| ds), \ \forall x \in C_r[0,T], \ t \in [0,T]$$

and

$$\|Q(x)(t) - Q(y)(t)\| \leqslant M \int_0^t \|x(s) - y(s)\| ds, \ \forall x, y \in C_r[0,T], \ t \in [0,T]$$

where M and q are non negative constant, then we get that there exist one and only one $x(t)$ which make the equation $x(t) + Q(t) = y(t), t \in [0,T]$ set up for all $y \in C_r[0,T]$.

Lemma 2 *Suppose*

   (a) *Serial $\{b_k\}_{k \geqslant 0}(b_k \geqslant 0)$ converges to zero.*

   (b) *Operator $Q : C_r[0,T] \to C_r[0,T]$ satisfies $\|Q(u)(t)\| \leqslant M(b_k + \int_0^t \|u(s)\| ds)$ where constants $M \geqslant 1$.*

   (c) *Operator $P : C_r[0,T] \to C_r[0,T]$ is defined as $P(u)(t) = P(t)U(t)$, where $P(t)$ is $r \times r$ matrix of continuous functions.*

(d) $\rho(P(t)) \leqslant 1, \; \forall t \in [0,T]$.

Then equation $\lim\limits_{n\to\infty}(P+Q_n)(P+Q_{n-1})\cdots(P+Q_0)(u)(t) = 0$ *holds consistently for all* $t \in [0,T]$.

Theorem 5 *Consider the nonlinear system described by equation (7), and it satisfies the assumption conditions that discussed above. Choose equation (8) as closed loop D-type Iterative learning law. If there exist Γ that makes $\rho[(I+\Gamma(t)C(t))^{-1}] < 1, \; t \in [0,T]$ hold, then the system has* $\lim\limits_{n\to\infty} y_k(t) = y_d(t)$.

Proof Let

$$\begin{cases} \delta x_k(t) = x_d(t) - x_k(t) \\ \delta y_k(t) = y_d(t) - y_k(t) \\ \delta u_k(t) = u_d(t) - u_k(t) \end{cases} \quad \text{and} \quad f_1(t,x) = f(t,x_d(t)) - f(t,x_d(t) - x(t)).$$

According to the kth iteration operation, we have:

$$\begin{cases} \delta \dot{x}_k(t) = f_1(t,\delta x_k(t)) + B(t)\delta u_k(t) \\ \delta \dot{y}_k(t) = C(t)\delta x_k(t) \\ \delta u_{k+1}(t) = \delta u_k(t) - \Gamma(t)\delta \ddot{y}_{k+1}(t). \end{cases} \tag{9}$$

According to equation (9), there is

$$\begin{aligned} \delta \dot{y}_k(t) &= C(t)\delta \dot{x}_k(t) + \dot{C}(t)\delta x_k(t) \\ &= C(t)f_1(t,\delta x_k(t)) + C(t)B(t)\delta u_k(t) + \dot{C}(t)\delta x_k(t) \end{aligned} \tag{10}$$

so we can get

$$\delta \dot{y}_{k+1} = \dot{C}(t)\delta x_{k+1}(t) + C(t)f_1(t,\delta x_{k+1}(t)) + C(t)B(t)\delta u_{k+1}(t). \tag{11}$$

From equation (9) (10) (11), we can get

$$\begin{aligned} \delta u_{k+1}(t) &= \delta u_k(t) - \Gamma(t)C(t)B(t)\delta u_{k+1}(t) - \Gamma(t)\dot{C}(t)\delta u_{k+1}(t) - \Gamma(t)C(t)f_1(t,\delta x_{k+1}(t)) \\ &= (I - \Gamma(t)C(t)B(t))^{-1}\delta u_k(t) - (I+\Gamma(t)C(t)B(t))^{-1}(\Gamma(t)\dot{C}(t)\delta u_{k+1}(t) \\ &\quad + \Gamma(t)C(t)f_1(t,\delta x_{k+1}(t))). \end{aligned} \tag{12}$$

Define operator $G_k : C_r[0,T] \to C_r[0,T]$ as

$$G_{k+1}(\delta u)(t) = (I + \Gamma(t)C(t)B(t))^{-1}(\Gamma(t)\dot{C}(t)\delta u_{k+1}(t) + \Gamma(t)C(t)f_1(t,\delta x_{k+1}(t))). \tag{13}$$

Define $P(t) = (I + \Gamma(t)C(t)B(t))^{-1}$ as the solution of $x_k(t)$ based on dynamic equation. Then equation (12) can turn into

$$\delta u_{k+1}(t) = P(t)\delta u_k(t) - G_{k+1}(\delta u_{k+1})(t). \tag{14}$$

According to Assumption (1) and equation (9), we have

$$
\|x(t)\| \leqslant \left\| x(0) + \int_0^t f_1(\tau, x(\tau)) d\tau + \int_0^t B(\tau) u_k(\tau) d\tau \right\|
$$
$$
\leqslant \|x(0)\| + K_{f1} \int_0^t \|x(\tau)\| d\tau + \int_0^t \|B(\tau) u_k(\tau)\| d\tau \tag{15}
$$

where $K_{f1} > 0$.

According to Bellman-Gronwall lemma and equation (15), there exist $K_{f2} > 0$, such that

$$
\|x(t)\| \leqslant K_{f2}(\|x(0)\| + \int_0^t \|B(\tau) u_k(\tau)\| d\tau). \tag{16}
$$

Due to equation (13) (16), and Assumption 3, we can get $\|G_k(u)(t)\| \leqslant K_{f3}(\|x(0)\| + \int_0^t \|u_k(\tau)\| d\tau)$. Furthermore, we can get

$$
\|G_{k+1}(\delta u)(t)\| \leqslant K_{f3}(\|\delta x_{k+1}(0)\| + \int_0^t \|\delta u(\tau)\| d\tau). \tag{17}
$$

Supposing x_w and x_v are the solution of equation (7) when $u = u_w$ and $u = u_v$ separately, from equation (13), we can obtain

$$
\|G_{k+1}(\delta u_w)(t) - G_{k+1}(\delta u_v)(t)\|
$$
$$
= \left\| P(t)\Gamma(t)\dot{C}(t)(\delta x_w(t) - \delta x_v(t)) + P(t)\Gamma(t)C(t)(f_1(t, \delta x_w(t)) - f_1(t, \delta x_v(t))) \right\|
$$
$$
\leqslant \left\| P(t)\Gamma(t)\dot{C}(t) \right\| \cdot \|\delta x_w(t) - \delta x_v(t)\| + \left\| P(t)\Gamma(t)C(t) \right\| \cdot \|f_1(t, \delta x_w(t)) - f_1(t, \delta x_v(t))\|. \tag{18}
$$

From equation (15), we can get

$$
\|x_w(t) - x_v(t)\| \leqslant \left\| \int_0^t (f_1(\tau, x_w(\tau)) - f_1(\tau, x_v(\tau))) d\tau \right\|
$$
$$
+ \int_0^t \|B(\tau)\| \cdot \|u_{w(k+1)}(\tau) - u_{v(k+1)}(\tau)\| d\tau. \tag{19}
$$

According to Bellman-Gronwall lemma, equation (15), and Assumption 3, we can get

$$
\|x_w(t) - x_v(t)\| \leqslant K_{f4} \int_0^t \|u_{w(k+1)}(\tau) - u_{v(k+1)}(\tau)\| d\tau. \tag{20}
$$

Considering (18) and (20), there exists K_f, such that

$$\left\| G_{k+1}(\delta u_w)(t) - G_{k+1}(\delta u_v)(t) \right\| \leqslant K_f \int_0^t \left\| u_{w(k+1)}(\tau) - u_{v(k+1)}(\tau) \right\| d\tau. \qquad (21)$$

Because equation (17) (21) satisfied Lemma 1 and Lemma 2, there exist $\overline{G}_{k+1}$, such that

$$\delta u_{k+1}(t) = P(t)\delta u_k(t) - \overline{G}_{k+1}(P\delta u_{k+1})(t) \qquad (22)$$

where $\overline{G}_{k+1}$ satisfied $\left\| \overline{G}_{k+1}(\delta u(t)) \right\| \leqslant K_{f5}(\|\delta x_{k+1}(0)\| + \int_0^t \|\delta u(\tau)\| d\tau)$, $K_{f5} > 0$.

Define operators Q_{k+1} : $C_r[0,T] \rightarrow C_r[0,T]$ as $Q_{k+1}(\delta u_{k+1})(t) = -\overline{G}_{k+1}(P\delta u_{k+1})(t)$, so we can rewrite equation (22) as $\delta u_{k+1}(t) = (P+Q_{k+1})(\delta u_k)(t) = (P+Q_{k+1})(P+Q_k)\cdots(P+Q_1)(\delta u_0)(t)$.

When $\rho(P(t)) < 1$, $\delta u_k(t) \rightarrow 0$ holds for all $t \in [0,T]$. So $\lim_{k\to\infty} \|\delta \dot{y}_{k+1}(t)\| = 0$ and $\lim_{k\to\infty} y_k(t) = y_d(t)$.   □

3. Fuzzy iterative learning control

High precision trajectory tracking of ILC method provides an effective method for robot control. However, it needs several iterations to obtain convergence of tracking error, and tracking accuracy is not very high before convergence. Fuzzy control is another intelligent control technology that does not need to establish the precise mathematical model of the controlled object. Fuzzy control has good robustness and makes it possible to overcome the adverse effects of uncertain factors brought to the system, such as variations and nonlinear model. Considering the characteristics of fuzzy control and closed-loop ILC, we combine these two kinds of control method, and conduct fuzzy iterative learning control. Before iterative learning, the system switches to fuzzy controller and stores its control output that will be provide as the initial control quantity of ILC. Structure of fuzzy iterative learning hybrid control is shown in Fig. 1.

3.1. Structure of fuzzy logic system

The basic configuration of a fuzzy logic system consists of a fuzzifier, fuzzy IF-THEN rules, a fuzzy inference engine and a defuzzifier as shown in Fig. 2. Fuzzifier, which is the fuzzy quantizer of the inputs, allows the conversion of the inputs variables which are physical quantities, in fuzzy quantities, or linguistic variables. Defuzzifier is the inverse operation of the fuzzifier. It consists in transforming the linguistic variables into real or digital variables [11].

The fuzzy inference engine uses the fuzzy IF-THEN rules to perform a mapping from an input vector $x^T = [x_1, x_2, \cdots, x_n] \in R^n$ to an output $\hat{f} \in R$. The ith fuzzy rule is written as $R^{(i)}$: If x_1 is A_i^1 and ... and x_n is A_n^i then $\hat{f}$ is f^i where A_1^i, A_2^i, ... and A_n^i are fuzzy sets and f^i is the fuzzy singleton for output in the ith rule. By using the singleton
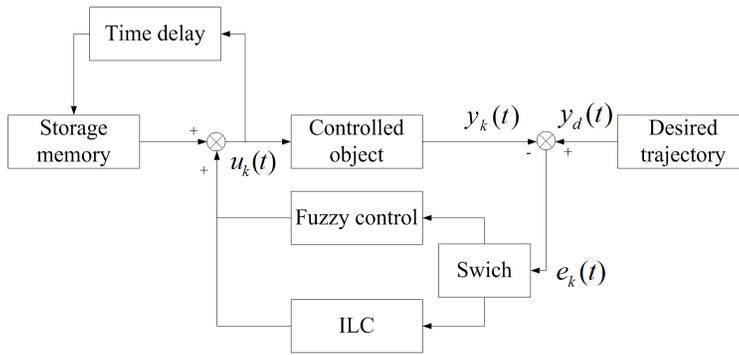
Figure 1: Fuzzy iterative learning control.
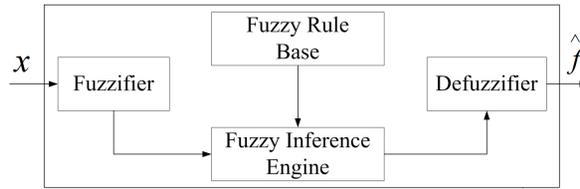
Figure 2: Fuzzy logic system.

product inference, and center-average defuzzifier, the output of the fuzzy system can be expressed as follows:

$$\hat{f}(x) = \frac{\sum_{i=1}^{m} f^i \left(\Pi_{j=1}^{n} \mu_{A_j^i}(x_j) \right)}{\sum_{i=1}^{m} \left(\Pi_{j=1}^{n} \mu_{A_j^i}(x_j) \right)} = \theta^T \psi(x) \tag{23}$$

where $\mu_{A_j^i}(x_j)$ is the degree of membership of x_j to A_j^i, m is the number of fuzzy rules, $\theta^T = [f^1, f^2, \cdots, f^m]$ is the adjustable parameter vector (composed of consequent parameters), and $\psi^T = [\psi^1, \psi^2, \cdots, \psi^m]$ with

$$\psi^i(x) = \frac{\Pi_{j=1}^{n} \mu_{A_j^i}(x_j)}{\sum_{i=1}^{m} \left(\Pi_{j=1}^{n} \mu_{A_j^i}(x_j) \right)} \tag{24}$$

being the fuzzy basis function[12].

3.2. Fuzzy control for single joint manipulator

In this paper, fuzzy controller is designed to track motion trajectory of robot. The input values are e and ec in accordance with error signal of joint angular displacement and its variation. In order to be convenient for analysis, $e \in [-6, 6]$ and $ec \in [-6, 6]$ are regarded as the normalized input sample values of error and its variation. $u \in [-1, 1]$ is regarded as the normalized output sample values of control quantity.

The linguistic terms of each value (*e*, *ec* and *u*) is divided into seven fuzzy sets. These are characterized by the following standard designations: negative big (*NB*), negative middle (*NM*), negative small (*NS*), zero (*Z*), positive small (*PS*), positive middle (*PM*), and positive big (*PB*). For convenience, membership functions (MFs) of inputs and output are chosen as triangle-shaped MF. Figs. 3 and 4 show the MFs for the error signal, change of error signal, and output respectively.
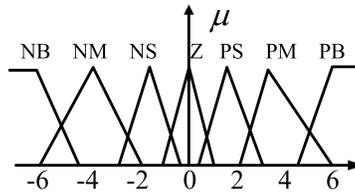
Figure 3: FMFs of inputs.
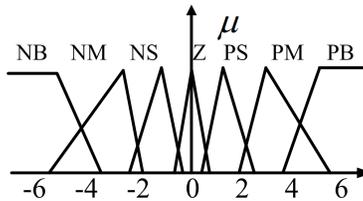
Figure 4: MFs of output.

The basis of the rules represents the strategy of control and the desired aim is determined by the linguistic control rules [13]. Considering the characteristics of robot motion, the control rules are established as Tab. 1. After the fuzzy output is calculated, it will be transformed into a numeric value which can be regard as control input of the control plant. In this work, the method of center-average defuzzification previously introduced is applied.

3.3. Fuzzy gain ILC for single joint manipulator

Considering equation (8), the fixed learning gain matrix Γ can't adapt to the changes of system operating state. Gain matrix is the same at any time, which ignores the specificity of different control states [4]. In order to improving the convergence rate, we utilize FLS to regulate the gain matrix of ILC.

Because of the influence of gain matrix on convergence rate, FLS is set up to regulate gain matrix factor, so as to regulate gain matrix on time. FLS adopts single input and single output form. $\int_0^t |e_k(\tau)| d\tau$ is designed input of FLS that reflect the error degree of iterative process, and k is adjustment factor of gain matrix. Thus gain matrix can automatically adapt to error information of each iteration. We design the fuzzy set of input as

Table 6: Control rules.

EC	DU						
	E						
	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NM	NS	PS	PM
NM	NB	NB	NM	NM	Z	PS	PM
NS	NB	NB	NS	NS	Z	PM	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NM	Z	PS	PS	PB	PB
PM	NM	NS	Z	PM	PM	PB	PB
PB	NM	NS	PS	PM	PB	PB	PB

{Z PS PM PB PVB}, the actual domain of input as $[0, E]$, where $E = \int_0^t |e_k(\tau)| d\tau$, $k = 0$. Fuzzy set of output is designed as {Z S M B}, and actual domain of output is $[0, 1]$. By this means, the membership functions of input and output variables are designed as Figs. (5) and (6).
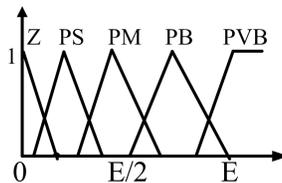
Figure 5: MFs of output.
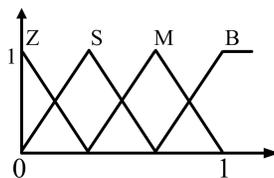
Figure 6: MFs of output.

The principle of adjustment is: in the initial stage of the control process, the error of the system is often large, so the main purpose of adjustment is to speed up the convergence rate and eliminate errors; in the late stage of the control process, the error of

Table 7: Fuzzy rules of gain adjustment.

$\int_0^t |e_k(\tau)|d\tau$	Z	PS	PM	PB	PVB
k	Z	S	M	M	B

the system is usually small, and the main purpose of the adjustment is to improve the stability. According to this principle, fuzzy rules are designed as Tab. 2.

4. Simulations

Based on the traditional fuzzy control and ILC method, fuzzy iterative learning control method is proposed and applied to trajectory tracking control for single joint robot. The dynamic model of a single joint robot system is given as follow:

$$I\ddot{\theta}_k + d\dot{\theta}_k + mgl\cos\theta_k = \tau_k \tag{25}$$

where $\ddot{\theta}_k$, $\dot{\theta}_k$ and θ_k represent acceleration, velocity and displacement of angular at kth iteration, $I = \frac{4}{3}ml^2$ represents rotary inertia, d is viscous friction coefficient, and τ_k is control input.

Parameters of single joint manipulator system are chosen as: $m = 2$kg, $l = 0.25$m, $d = 2$, and $g = 9.8$m/s^2. Desired trajectory is $y_d(t) = \sin(t)$.

Firstly, applying close D-type ILC and fuzzy control, the results are shown in Fig. 7 and 8. From Fig. 7, we can see that the real trajectory almost follows the desired trajectory after 20 times' learning iterations. It is clearly seen that the tracking errors of first few iterations are obviously too big compared to fuzzy control as shown in Fig.8.
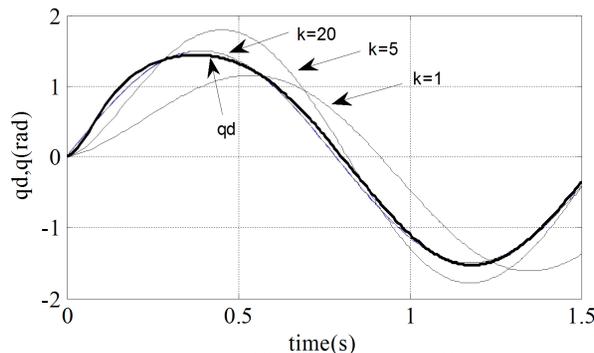
Figure 7: Close D-type ILC.

Then, applying fuzzy ILC method proposed in this paper, as shown in Fig. 9, the tacking errors of the first few iterations are much smaller than simple ILC, and the tracking errors monotonically decrease with the increase of the iteration number.
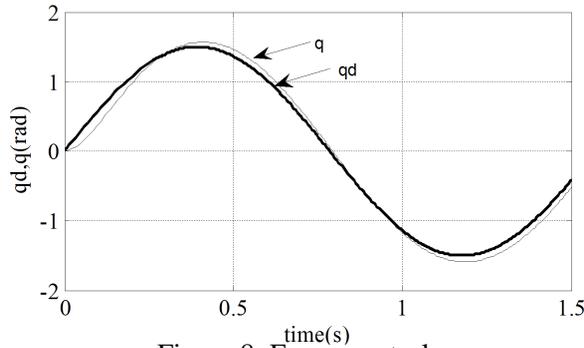
Figure 8: Fuzzy control.
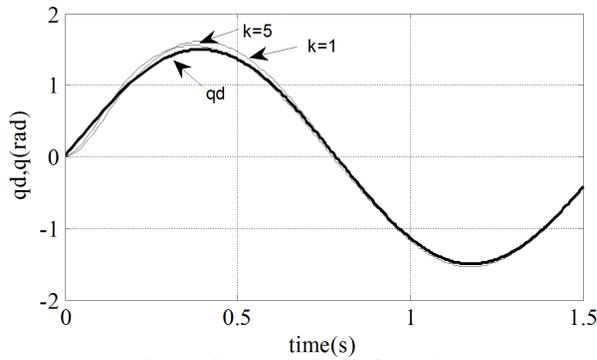
Figure 9: Fuzzy ILC for robot.
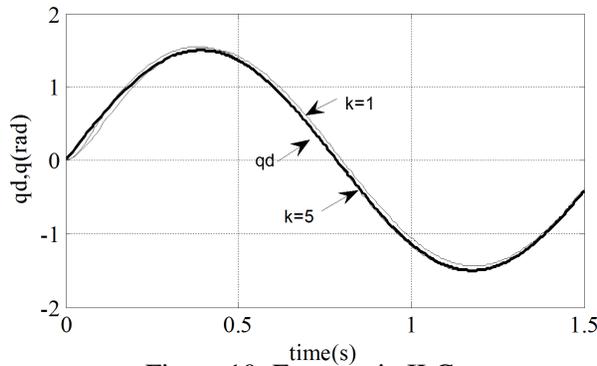
Figure 10: Fuzzy gain ILC.

At last, we utilize FLS to regulate the gain matrix of ILC, in order to improving the convergence rate. The simulation result is show in Fig.10. Comparing with Fig. 9, by using fuzzy gain, the convergence rate is further improved.

5.   Conclusion

A new Fuzzy ILC scheme is proposed for the position tracking problem of single joint manipulator. The proposed controller is based upon fuzzy control and fuzzy gain ILC to cope with the problem of slow convergence at the beginning of iteration and the problem of fixed learning gain. The proposed method is applied to a single joint manipulator model, which indicates that the error can be obviously reduced in a short time, and a better convergence rate is obtained than simple fuzzy control and ILC.

References

[1] J.S. YEON, J.H. PARK, S.W. SON and S.H. LEE: Model-Based iterative learning control for industrial robot manipulator. *Proc.of the IEEE Int. Conf. on Automation and Logistics*, Shenyang, China, (2009), 24-28.

[2] S.P. TIAN, J. YAN and D.X. CHEN: ILC algorithm with fuzzy factor based on vector plots analysis. *Int. Conf. on System Science*, Engineering Design and Manufacturing Informatization, (2011), 148-151.

[3] F. BOUKARIF: D-type iterative learning control without resetting condition for robot manipulators. *Robotica*, **29** (2011), 975-980.

[4] Y. WANG and Y.L. FU: Fuzzy adaptive iterative learning control algorithm. *Proc. of the 6th World Congress on Intelligent Control and Automation*, Dalian, China (2006), 3719-3723.

[5] R. TANG, L. HOU and Q. ZHANG: Adaptive iterative learning control for SCARA robot manipulators. *Int. J. of Advancements in Computing Technology*, **4**(21), (2012), 50-58.

[6] X.G. JIA and Z Y. YUAN: Adaptive iterative learning control for robot manipulators. *IEEE Int. Conf. on Intelligent Computing and Intelligent Systems*, Xiamen, China, 3 (2010), 139-142.

[7] S. GOPINATH and I.N. KAR: Iterative learning control scheme for manipulators including actuator dynamics. *Mechanism and Machine Theory*, **39** (2004), 1367-1384.

[8] S.S. SAAB: Stochastic P-type/D-type iterative learning control algorithms. *Int. J. of Control*, **76**(2), (2003), 139-148.

[9] Y. FANG, Y.C. SOH and G.G. FENG: Convergence analysis of iterative learning control with uncertain initial conditions. *Proc. of the 4th World Congress on Intelligent Control and Automation*, Shanghai, China, (2002), 960-963.

[10] X.G. ZHANG and H. LIN: Robustness analysis of open-closed-loop D-type iter-
ative learning control algorithm for nonlinear systems with deviations on initial
state. *IMACS Multiconference on Computational Engineering in Systems Applica-
tions*, Beijing, China, (2006), 1707-1711.

[11] S. CHEKKAL, N.A. LAHACANI, D. AOUZELLAK AND K. GHEDAMSI: Fuzzy
logic control strategy of wind generator based on the dual-stator induction genera-
tor. *Electrical Power and Energy Systems*, **59** (2014), 166-175.

[12] A. BOULKROUNE, N. BOUNAR, M.M. SAAD and M. FARZA: Indirect adaptive
fuzzy control scheme based on observer for nonlinear systems: A novel SPR-filter
approach. *Neurocomputing*, **135** (2014), 378-387.

[13] M.T. CAO, J. NETO and H.L. HUY: Fuzzy logic based controller for induction
motor drives. *Proc. of the Canadian Conf. on Electrical and Computer Engineer-
ing*, **2** (1996), 631-634.