2005 American Control Conference
June 8-10, 2005. Portland, OR, USA

Two-Mode Iterative Learning Control Using P-Type and Pseudo-Downsampled Learning

Bin Zhang, Danwei Wang and Yongqiang Ye

***Abstract*— In this paper, an iterative learning controller combining frequency domain and time domain design methods is proposed. The controller uses different learning mechanisms on low frequency band and high frequency band, respectively. On low frequency band, a conventional iterative learning controller is used and input update is carried out point by point. While on the high frequency band, the input update is carried out every several steps and it is step-wise constant, which is termed as pseudo-downsampled learning. This way, some high frequency error components can be learned so that the tracking accuracy can be improved while the good learning transient can be guaranteed at the same time. The design method and experimental results are presented.**

I. INTRODUCTION

The objective of iterative learning control (ILC) [1], [2], [3], [4] is to improve the tracking accuracy as the same task is repeated. It is well known that ILC may show a bad learning behavior during the learning process and this phenomenon is explained in both time domain and frequency domain in many papers [5], [6], [7].

In frequency domain, bad learning transient is explained as high frequency components of error accumulates as iteration goes on. To obtain good learning behavior so that the decay of error being monotonic, a low-pass filter is introduced into the learning law. This low-pass filter cuts off the high frequency error components that make learning diverge. But the error components reside in the high frequencies are left untouched and the tracking accuracy is sacrificed. Wang *et al.* [8] proposed a multi-channel method in frequency domain by using anticipatory ILC. There are also adaptive schemes in frequency domain to achieve good learning behavior [9], [10], [11]. However, there is always a trade-off between tracking accuracy and good learning behavior.

In time domain, some researchers proposed adaptive learning gain on iteration axis or time axis to get good learning transient [12], [13]. Longman *et al.* [5] pointed out that this scheme needs much knowledge of system model, which is often unavailable in practice. In addition, a very small learning gain can also yield bad learning transient. Chang *et al.* proposed *the bisection method* to improve the learning transient. The idea is that the learning is carried out only on a small number of evenly distributed sampling points throughout the trajectory. Once good tracking error is achieved at these steps, the density of steps involved in the learning is increased. This process is repeated until all points enter the learning. Because the learning is not carried out on all points during the learning, the learning might be slow. In addition, a difficulty is that it is hard to choose the number of steps to meet a desired error tolerance restriction [5]. Moore [14] derived the monotonic convergence condition in time domain. Hillenbrand *et al.* [15], [16] proposed a reduced sampling rate iterative learning control to deal with initial state error. It also shows that this scheme can guarantee the exponential rate of convergence. This learning law is interesting and by reducing the sampling rate, the condition derived by Moore in [14] might be satisfied. A shortcoming is that after reducing the sampling rate, the amount of sampling points is decreased. The original points between these decreased sampling points are not learned and become uncontrolled.

In this paper, we will combine the conventional ILC with low-pass filter in frequency domain with the reduced sampling rate ILC in time domain to improve the learning performance. On low frequency band, a conventional ILC can learn the error components on all sampling points. But on high frequency band, only error on evenly separated sampling points enter the learning like the beginning of *the bisection method*. This equals to reduce the sampling rate on high frequency band and it is termed as pseudo-downsampling. Hence, in our approach, all sampling points are learned during the learning process. In addition, the actual sampling rate does not need to be changed. The theoretical basis of this approach is presented and experimental results are provided to verify the approach.

II. ANALYSIS OF LEARNING TRANSIENT

Consider a discrete-time linear single input single output (SISO) system

$$\begin{cases} x_j(k+1)= Ax_j(k) + Bu_j(k) + w_j(k) \\ y_j(k)= Cx_j(k) + v_j(k) \end{cases} \tag{1}$$

where $k \in [0, p-1]$ with p being the total sampling points of the operation interval, the state x is a n dimensional vector, the input u is a scalar, the output y is a scalar, the subscript j is the index of cycle, w and v are the state disturbances and output disturbances, respectively.

As Longman suggested, to simplify the implementation of ILC, it is better to adjust the command given to the existing feedback control system rather than adjust the

Bin Zhang is with School of EEE, Nanyang Technological University, Singapore. binzhang@pmail.ntu.edu.sg

Danwei Wang is corresponding author and with Faculty of School of EEE, Nanyang Technological University, Singapore. edwwang@ntu.edu.sg

Yongqiang Ye is with School of Information, Zhejiang University of Finance and Economics, Hangzhou, 310012, China. p149890776@ntu.edu.sg

0-7803-9098-9/05/$25.00 ©2005 AACC

torque to actuators so that the existing feedback controller can be left unchanged [7]. Longman also proved that these two approaches are mathematically equivalent [17]. The learning law has the form as follows:

$$\left\{\begin{array}{r l} u_j(k) = & y_d(k) + u_{L,j}(k) \\ u_{L,j+1}(k) = & u_{L,j}(k) + \Gamma e_j(k+1) \end{array}\right. \tag{2}$$

where Γ is learning gain, $e_j(k) = y_d(k) - y_j(k)$ is the error signal at the j-th cycle with $y_d(k)$ is the desired trajectory and $y_j(k)$ is the actual trajectory at the j-th cycle. $u_{L,j}$ is the adjustment of command in the j-th cycle and u_j is the input to the closed-loop feedback control system.

Take z-transform of (1) for the j-th cycle, we have

$$\begin{aligned} Y_j(z) &= G(z)U_j(z) + C(zI - A)^{-1}zx(0) \\ &\quad + C(zI - A)^{-1}W(z) + V(z) \end{aligned} \tag{3}$$

where $G(z) = C(zI - A)^{-1}B$ and $x(0)$ the initial state. The z-transform of (2) has the form of

$$\left\{\begin{array}{r l} U_j(z) = & Y_d(z) + U_{L,j}(z) \\ U_{L,j+1}(z) = & U_{L,j}(z) + \Gamma z E_j(z) \end{array}\right. \tag{4}$$

Combining these equations with the assumption that the disturbances are repeated and initial state are the same with the desired initial state, we have

$$E_j(z) = [1 - \Gamma z G(z)]E_{j-1}(z) \tag{5}$$

In this equation, $[1 - \Gamma z G(z)]$ represents an error evolution transfer function from cycle $(j-1)$ to j. By substituting $z = e^{i\omega}$, it yields that if

$$\|1 - \Gamma e^{i\omega} G(e^{i\omega})\| < 1 \tag{6}$$

holds for all frequencies up to Nyquist frequency, the decay of error will be monotonic. But in practice, this condition is difficult to hold for high frequencies [7]. To get good learning transient, a low pass filter F is introduced to cut off frequencies that violate the condition (6). The learning law (2) becomes

$$\left\{\begin{array}{r l} u_j(k) = & y_d(k) + u_{L,j}(k) \\ u_{L,j+1}(k) = & u_{L,j}(k) + \Gamma F e_j(k+1) \end{array}\right. \tag{7}$$

The filter F is the low-pass filter to cutoff those frequencies that violate the condition (6). Then, the condition (6) becomes

$$\|1 - \Gamma e^{i\omega} F(e^{i\omega}) G(e^{i\omega})\| < 1 \tag{8}$$

and this condition holds for all frequency components pass through the filter F so that the good learning transient can be guaranteed.

Although the introduction of low pass filter F can ensure the good learning transient, it sacrifices the tracking accuracy because the error components beyond the cutoff frequency of filter F are left unlearned. If there is much useful information in the high frequencies, the tracking performance might be very poor.

On the other hand, in time domain, the solution of system (1) is

$$\begin{aligned} y_j(k) &= CA^k x_j(0) + \sum_{i=0}^{k-1} CA^{k-i-1}Bu_j(i) + \\ &\quad \sum_{i=0}^{k-1} CA^{k-i-1}w_j(i) + v_j(i) \end{aligned} \tag{9}$$

Define an operator $\delta_j z(k) = z_j(k) - z_{j-1}(k)$ [18] to get the difference value of any variable in two successive repetitions. Then by applying this to equation (9), the result will be

$$\delta_j y = P\delta_j u \tag{10}$$

where

$$P = \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{p-1}B & CA^{p-2}B & \cdots & CB \end{bmatrix}$$

$y = [y(1), y(2), \cdots, y(p)]^T$, and $u = [u(0), u(1), \cdots, u(p-1)]^T$.

From (2), (10) and by realizing $\delta_j y = -\delta_j e$ with $e = [e(1), e(2), \cdots, e(p)]^T$, the error propagation equation with respect to cycle can be obtained as:

$$e_{j+1} = (I - \Gamma P)e_j = Ee_j \tag{11}$$

where

$$E = \begin{bmatrix} 1-\Gamma CB & 0 & \cdots & 0 \\ -\Gamma CAB & 1-\Gamma CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\Gamma CA^{p-1}B & -\Gamma CA^{p-2}B & \cdots & 1-\Gamma CB \end{bmatrix}$$

Thus, the learning control will converge to zero tracking error as the cycle goes on, provided all eigenvalues of matrix E are less than one. That is

$$|\lambda_i(1 - \Gamma CB)| < 1 \tag{12}$$

Take ∞-norm on both sides of (11), we have

$$\|e_{j+1}\|_\infty \le \|E\|_\infty \|e_j\|_\infty \tag{13}$$

Hence, to make the error decay monotonically in the sense of ∞-norm, it requires

$$\|E\|_\infty \le 1 \tag{14}$$

That is

$$\max_i \sum_{l=1}^{p} |e_{i,l}| \le 1 \tag{15}$$

where $e_{i,l}$ is the entry of matrix E on i-th row and l-th column. Then, we have

$$|1 - \Gamma CB| + \sum_{i=1}^{p-1} |\Gamma CA^i B| \le 1 \tag{16}$$

If $(1-\Gamma CB) > 0$ and $\Gamma CB > 0$, then $|1-\Gamma CB| = 1-|\Gamma CB|$ and we have

$$1-(|\Gamma CB|-\sum_{i=1}^{p-1}|\Gamma CA^iB|) \leq 1 \tag{17}$$

This equation requires

$$|\Gamma CB|-\sum_{i=1}^{p-1}|\Gamma CA^iB| \geq 0 \tag{18}$$

Then, the monotonic decay of error in the sense of ∞-norm requires [14]:

$$|CB| \geq \sum_{i=1}^{p-1}|CA^iB| \tag{19}$$

That is, the absolute value of first Markov parameter should be larger than or equal to the sum of absolute value of all remaining Markov parameters on the entire trajectory.

For a discrete-time system with designed feedback controller and sampling rate, the Markov parameters are invariable. But for continuous system model, the condition (19) can be satisfied by reducing the sampling rate. For a continuous-time system A_c, its zero order hold equivalent with sampling period of T is [16]:

$$A = e^{A_cT} \tag{20}$$

For a stable continuous system A_c, all eigenvalues of A_c are located in the left half plane. Then, all eigenvalues of e^{A_cT} should be inside the unite circle [16]. If sampling period $T \to \infty$, then

$$\lim_{T\to\infty} A \to 0$$

In practice, suppose a system with sampling period T cannot make condition (19) hold, the error on the cycle index will not decay monotonic in the sense of ∞-norm. Suppose condition (19) holds for a sampling period of $m{\cdot}T$, then by properly changing the sampling period to $m\cdot T$, the good learning transient can be guaranteed. But the problem is that the $(m-1)$ sampling points based on sampling period T between every two sampling points based on sampling period $m{\cdot}T$ are not learned. This is undesirable. In addition, change of sampling period often needs to adjust hardware and this will increase cost in practice. In the next, we combine the convenient P-type ILC and this downsampling learning scheme together to improve the tracking accuracy and guarantee the good learning transient at the same time. The novelty is that the method is two-mode, which has different learning scheme on low frequency band and high frequency band. For low frequency band, the learning is a conventional P-type ILC and the update of input is carried out point by point. While for high frequency band, we update the input every m sampling points, which we call it pseudo-downsampled learning.

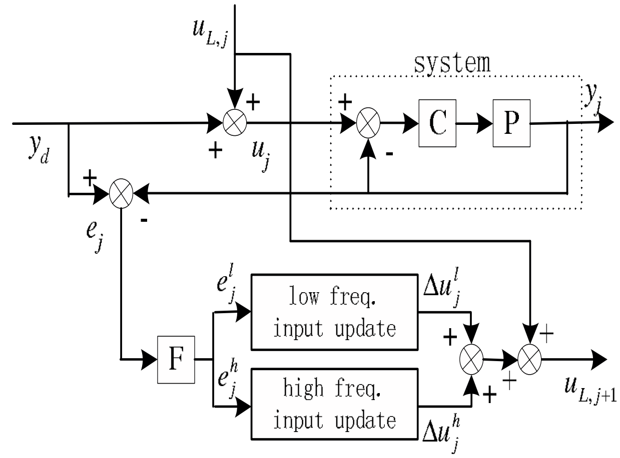

Fig. 1. The controller scheme

III. The two-mode ILC

The structure of the learning is shown in Figure 1. The ILC update law has two modes, one for low frequency error components and one for high frequency components.

In this figure, the signal Δu_j^l and Δu_j^h are input updates on low frequency band and high frequency band, respectively. The error signal is decomposed on low frequency band and high frequency band by a low-pass filter F, which has a cutoff frequency that makes condition (8) holds. The error signal of j-th iteration can be written as:

$$e_j = e_j^l + e_j^h \tag{21}$$

where e_j is the measured error signal of the j-th cycle, e_j^l the error components on the low frequency band as the result of the low-pass filter F. e_j^h is the error components on the high frequency band obtained by $e_j^h = e_j - e_j^l$. On the low frequency band, the learning is carried out point by point. The data pair is $[\Delta u_j^l(i_l), e_j^l(i_l+1)]$, $i_l \in [0,1,\cdots,p-1]$. Because the cutoff frequency of F is lower than or equal to the frequency that makes the condition (6) hold, the good learning transient on low frequency band can be guaranteed.

On the high frequency band, the learning is not point by point but carried out every m sampling points, where $m\cdot T$ is a sampling period that makes condition (19) hold. That is, for high frequency components, the update of input is based on the error signal m steps ahead. The data pair is $[\Delta u_j^h(i_h), e_j^h(i_h+m)]$, $i_h \in [0,m,2m,\cdots,d\cdot m]$ with d being the integer of $\frac{p-1}{m}$. For those $(m-1)$ points between $\Delta u_j^h(i_h)$ and $\Delta u_j^h(i_h+m)$, the input update is given as $\Delta u_j^h(i_h)$. So, the input signal on high frequency band is a step-wise constant signal with step m.

The update law is summarized as follow

$$\begin{cases} u_j(k) = y_d(k) + u_{L,j}(k) \\ u_{L,j+1}(k) = u_{L,j}(k) + \Delta u_j^l(k) + \Delta u_j^h(k); \\ \Delta u_j^l(i_l) = \Gamma_l e_j^l(i_l+1); \\ \Delta u_j^h(i_h) = \Gamma_h e_j^h(i_h+m); \\ \Delta u_j^h(i_h+i_b) = \Delta u_j^h(i_h); \end{cases} \quad (22)$$

where $i_l \in [0, 1, \cdots, p-1]$; $i_h \in [0, m, 2m, \cdots, d \cdot m]$ and $e_j^h(\xi) = e_j^h(p)$ for $\xi > p$; $i_b \in [1, m-1]$; $k \in [0, p-1]$; Γ_l and Γ_h are learning gain on low frequency band and high frequency band, respectively.

In this update law, the first equation uses the $u_{L,j}$ adjust the command. The second equation uses the input update on low frequency band and high frequency band to update the $u_{L,j}$. The third equation is the input update on low frequency band. The fourth equation is the input update on high frequency band. The last equation fills out the input update signal on high frequency band for those sampling points not included in the learning.

Remark 1: Selection of estimated learnable bandwidth

From the above discussion, it is known that in frequency domain to guarantee the good learning transient, the condition (6) should be satisfied. With a given learning gain Γ_l and system model $G(z)$, the condition might be violated at a certain frequency f_b. All frequencies below this frequency $[0, f_b]$ form a learnable band and the value of f_b is termed as learnable bandwidth. To get good learning transient on low frequency band, the cutoff frequency of low-pass filter F should be lower than or equal to f_b.

Remark 2: Selection of sampling rate for high frequency learning

Suppose the continuous time system model is available and the sampling period of system is T, we discretize this model by using this sampling time T. If the condition (19) holds, the sampling rate does not need to be reduced. If condition (19) is violated, we increase the sampling period to $2T$, $3T$, and so on until for a sampling period $m \cdot T$, the condition (19) is satisfied.

The implementation of the method can be described as follows:

1) Use condition (6) to determine the cutoff frequency f_b of the low-pass filter F.
2) Find the minimal m such that the discretized system model with sampling period $m \cdot T$ makes condition (19) hold.
3) Use filter F to filter the error signal and obtain e_j^l and e_j^h, respectively.
4) Use update law (22) to get the input of next cycle u_{j+1}.
5) Execute next operation cycle with input u_{j+1}, record error signal e_{j+1} and return to step 3.

IV. Experimental Results

Some experiments are carried out on an industrial SCARA robot, SEIKO TT3000, that is described in [19]. One joint moving in the horizontal plane is used to test our proposed method. A nominal model of closed-loop joint is given as follows:

$$G_p(s) = \frac{948}{s^2 + 42s + 948} \quad (23)$$

The desired trajectory has a total length of 2 seconds. Since the sampling period of the robot is 0.01 second, this trajectory has total 200 sampling points. The trajectory is given as follows and is shown in Figure 2.

$$y_d(k) = \begin{cases} 7(1 - \cos\frac{6\pi k}{200}) & k \in [0, 33] \\ 4 + 5(1 - \cos\frac{6\pi k}{200}) & k \in [34, 166] \\ 7(1 - \cos\frac{6\pi(200-k)}{200}) & k \in [167, 200] \end{cases}$$

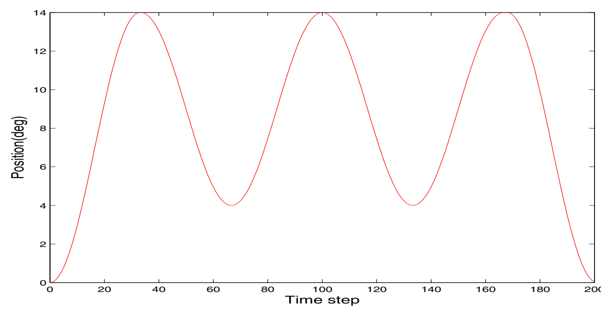

Fig. 2. The desired trajectory

A. Parameter selection

In our method, the parameters need to be determined include learning gain for low frequency error components Γ_l, learning gain for high frequency error components Γ_h, cutoff frequency of low-pass filter f_b, the pseudo-downsampling period $m \cdot T$.

For the learning in low frequency band, Longman *et al.* pointed out that learning gain is not a critical factor of learning performance and they suggested a low value learning gain for a good tracking error level [12], [20]. In addition, higher learning gain also contaminates performance in steady state response and makes system vulnerable to noises in the sense that random noise going through the learning law will be amplified. Hence, the learning gain for high frequency band should be set as a low value too. In our experiments, the learning gain Γ_l and Γ_h are both selected as 0.5.

To estimate the learnable bandwidth f_b of the system (23) with the chosen learning gain Γ_l=0.5, the condition (6) is rewritten as:

$$\|1 - \Gamma e^{i\omega} G(e^{i\omega})\| < 1 - \epsilon \quad (24)$$

where ϵ is a small positive value to enhance robustness. In this experiment, $\epsilon = 0.1$. With this consideration, the condition is illustrated in Figure 3. From this figure, the learnable bandwidth for this gain can be approximately read as 4Hz. That is $f_b = 4$Hz. Because the system model is not accurate, the estimated learnable bandwidth may lead a divergent learning behavior. In this case, the estimated learnable bandwidth can be adjusted to a lower value.

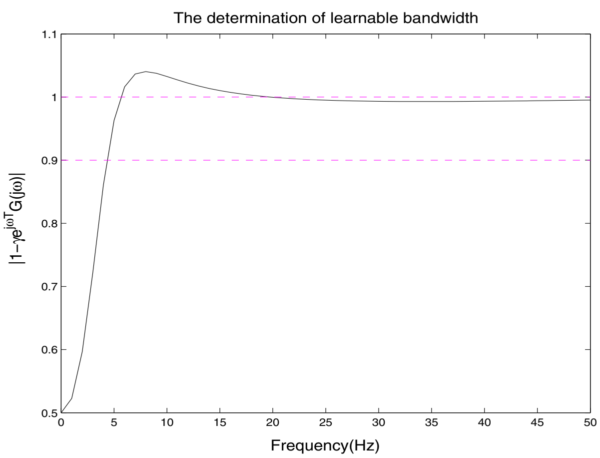

Fig. 3. The learnable bandwidth

With sampling period $T = 0.01$ second, the discretized system has Markov parameters as shown in the upper subfigure of Figure 4. It is clear that the condition (19) is not satisfied. Increase the sampling period and when it becomes 0.05 second, the discretized system has Markov parameters shown in the lower subfigure of Figure 4. Under this sampling period of 0.05 second, the first Markov parameter is 0.5548, while the sum of all other Markov parameters' absolute value is 0.5523. The condition (19) is satisfied. It is noticed that when sampling period becomes 0.05 second, the number of sampling points on the trajectory is 40. In this case, the value of m is chosen as 5.

B. Experimental results

The experimental results are shown in Figure 5. From this figure, it is clear that the estimated learnable bandwidth $f_b = 4$Hz leads a divergent learning behavior. Hence, f_b is adjusted to 3.5Hz. With this updated estimated learnable bandwidth, the learning process shows a monotonic decay of RMS error. After about 100 cycles of learning, the RMS error reaches about $0.11°$. In the later cycles, the RMS error has no obvious improvement. At 200th cycle, the RMS error is $0.10°$. The improvement is very limited.

On the other hand, the proposed method has a great improvement of RMS error. The learning converges monotonically in the entire learning process. After 150 cycles learning, the RMS has no further improvement and has little oscillations. The average RMS error in the cycles from 151 to 500 is $0.014°$. This value is less than one-seventh of RMS error of learning with only low-frequency error components.

At the 500th cycle, the error signal of the proposed method and conventional ILC with low-pass filter are shown in Figure 6. It is clear to see that the error signal of the

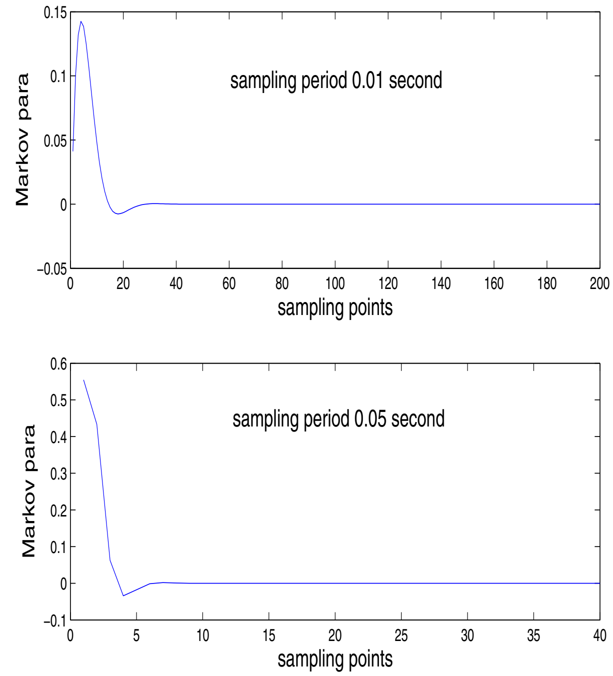

Fig. 4. Markov parameters of different sampling rate

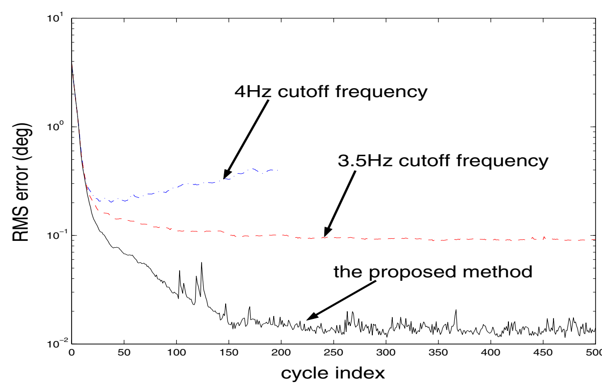

Fig. 5. RMS error

proposed method is much smaller and uniform on the entire trajectory.

Let's investigate the input signal of the proposed method. The input update on low frequency band and high frequency band are calculated separately. The input signal at the 500th cycle is shown in Figure 7. The input update on the low frequency band is step by step on every sampling point, which is shown in Figure 7(a). This signal is quite smooth. While, the input update on the high frequency band is carried out every 5 steps in the experiment. This signal is a step-wise constant one and is shown in Figure 7(b). The final input signal sends to the closed-loop feedback control system is shown in Figure 7(c).

The experimental results show that the proposed method can produce a good learning transient. In addition, it can

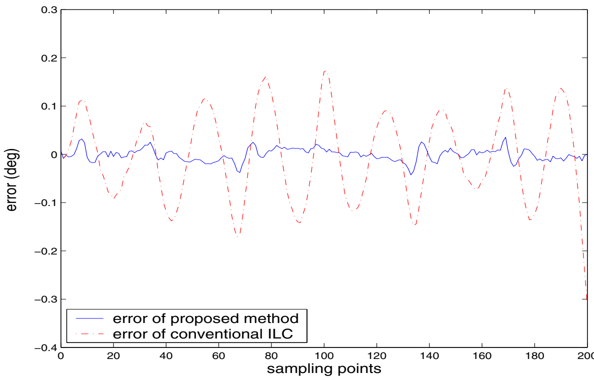

Fig. 6. Error signals comparison

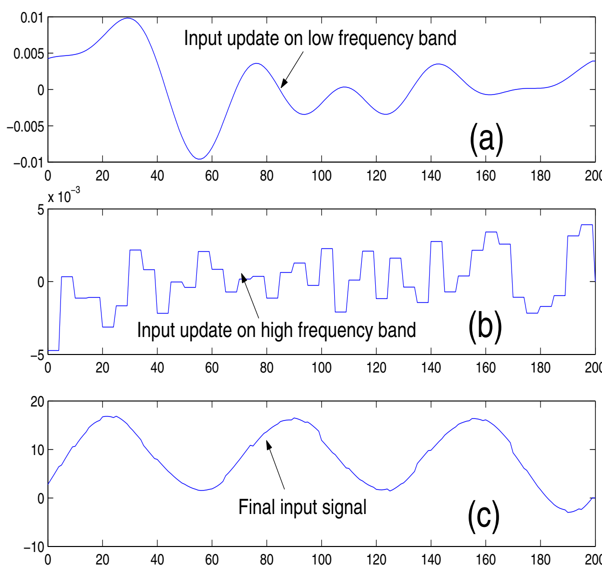

Fig. 7. Input signal of the proposed method

greatly improve the tracking accuracy.

V. Conclusion

In this paper, a two-mode iterative learning controller is proposed. In the proposed method, the error signal is decomposed on low frequency band and high frequency band according to estimated learnable bandwidth. On low frequency band, the learning is carried out step by step, which is a conventional P-type ILC. While on high frequency band, the learning is pseudo-downsampled by carrying out input update every m steps, where m is multiple of system sampling period to realize reduction of sampling rate. This method can yield a good learning transient on both low frequency band and high frequency band. Experimental results show that the proposed method can not only get a good learning transient but also improve the tracking accuracy substantially.

References

[1] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *Journal of Robotic System*, vol. 1, pp. 123–140, 1984.

[2] G. Casalino and G. Bartolini, "A learning procedure for the control of movements of robotic manipulators," in *Proceedings of the 4th IASTED Symposium on Robotics and Automation*, (Amsterdam, Netherlands), pp. 108–111, 1984.

[3] J. J. Craig, "Adaptive control of manipulator through repeated trials," in *Proceedings of the 1984 American Control Conference*, (San Diego, CA USA), pp. 1566–1573, 1984.

[4] R. H. Middleton, G. C. Goodwin, and R. W. Longman, "A method for improving the dynamic accuracy of a robot performing a repetitive task," *International Journal of Robotic Research*, vol. 8, no. 5, pp. 67–74, 1989.

[5] C.-K. Chang, R. W. Longman, and M. Q. Phan, "Techniques for improving transients in learning control systems," *Advances in Astronautical Sciences*, vol. 76, pp. 2035–2052, 1992.

[6] Y.-C. Huang and R. W. Longman, "The source of the often observed property of initial convergence followed by divergence in learning and repetitive control," *Advances in the Astronautical Sciences*, vol. 90, pp. 555–572, 1996.

[7] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *International Journal of Control*, vol. 73, no. 10, pp. 930–954, 2000.

[8] D. Wang and Y. Ye, "Multi-channel learning using anticipatory ILCs," *International Journal of Control*, (to appear).

[9] Y.-Q. Chen and K. L. Moore, "Frequency domain adaptive learning feedforward control," in *Proceedings of 2001 IEEE Symposium on Computational Intelligence in Robotics and Automation*, (Banff, Alberta Canada), pp. 396–401, Jun-Aug 2001.

[10] D.-N. Zheng and A. Alleyne, "An experimental study for an iterative learning control scheme with adaptive filtering," in *Proceedings of 2002 ASME International Mechanical Engineering Congress and Exposition*, (New Orleans, Louisiana USA), pp. 1–9, Nov 2002.

[11] B. Zhang, D. Wang, and Y. Ye, "Experimental study of time-frequency based ILC," in *Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision*, (Kunming, China), Dec 2004(to be presented).

[12] R. W. Longman and S.-L. Wirkander, "Automated tuning concepts for iterative learning and repetitive control laws," in *Proceedings of the 37th Conference on Decision and Control*, (Tampa, Florida USA), pp. 192–198, 1998.

[13] H.-S. Lee and Z. Bien, "A note on convergence property of iterative learning controller with respect to sup norm," *Automatica*, vol. 33, no. 8, pp. 1591–1593, 1997.

[14] K. L. Moore, "An observation about monotonic convergence of discrete-time, P-type iterative learning control," in *Proceedings of the IEEE International Symposium on Intelligent Control*, (Mexico), pp. 45–49, 2001.

[15] S. Hillenbrand and M. Pandit, "A discrete-time iterative learning control law with exponential rate of convergence," in *Proceedings of the 38th Conference on Decision and Control*, pp. 1575–1580, 1999.

[16] S. Hillenbrand and M. Pandit, "An iterative learning controller with reduced sampling rate for plant with variations of initial states," *International Journal of Control*, vol. 73, pp. 882–889, 2000.

[17] E. J. Solcz and R. W. Longman, "Disturbance rejection in repetitive controller," *Advances in the Astronautical Sciences*, vol. 76, pp. 2111–2130, 1992.

[18] H. Elci, R. W. Longman, M. Phan, J.-N. Juang, and R. Ugoletti, "Discrete frequency based learning control for precision motion control," in *Proceeding of the 1994 IEEE International Conference on Systems, Man, and Cybernetics*, (San Antonio, Texas USA), pp. 2767–2773, 1994.

[19] D. Wang, "On D-type and P-type ILC designs and anticipatory approach," *International Journal of Control*, vol. 73, pp. 890–901, 2000.

[20] S.-L. Wirkander and R. W. Longman, "Limit cycles for improved performance in self-tuning learning control," *Advances in the Astronautical Sciences*, vol. 102, pp. 763–781, 1999.