2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Design of Iterative Learning Controller based on Frequency Domain Linear Matrix Inequality

Kiyonori Inaba, Chun-Chih Wang, Masayoshi Tomizuka, Andrew Packard

Abstract— **This paper presents an iterative learning controller (ILC) design method based on a frequency domain linear matrix inequality (LMI). It modifies the LMI method for lifted system representation. In lifted system representation, the size of matrices grows as the number of samples in the system increase. In the proposed approach, the size of matrices in LMI is independent of the number of samples in the system, avoiding computational problems due to the size of matrices. The order of controllers can be low, even if the order of the system is high, unlike H_∞ method. Compatibility with frequency domain robust theory and easy extension to MIMO system is another advantage. Zero-phase weighting functions are introduced to shape the filters to increase robustness to the disturbance in high frequency range. This paper also examines the monotonic convergence of the output of the feedback controllers after implementing ILC. The performance of the designed ILC is verified by experiments performed on an industrial robot.**

I. INTRODUCTION

For the control of industrial robots, one important objective is to achieve the precise tracking. However, it is not easy to perform precise position control in either the joint space or the task space for industrial robots, mainly because there are disturbances caused by friction, backlash, and transmission error in the reducers [1]. Moreover, the disturbances change depending on the motion of the robot. Currently, the requirements of motion precision for industrial robots are becoming more and more stringent.

ILC is a promising method to overcome these problems caused by the reducers. The idea of ILC is to learn information from previous cycles of a repetitive process so as to shape the feed-forward input to improve the performance of the system. The first paper on this topic was published by Uchiyama in 1978 [2]. In 1984, Arimoto proposed a P-type learning control law for robotic applications [3]. Also, in the same year, Casalino and Bartolini [4], and Craig [5] also published papers about learning control for robotic applications.

One way to systematically design an ILC scheme is through the H_∞ method. De Roover [6], Moon and Doh [7], Xu [8] published papers about ILC design through H_∞ method. This method can be extended to MIMO systems easily, and can be analyzed for robustness in the frequency domain. However, it also has disadvantages that the order of the controller is high if the system order is high and the learning filter is limited to causal filters. Lifted system linear matrix inequality (LMI) method is another method for systematic ILC design. This method was proposed by Ahn [9]. This method can design non-causal filters as well. On the other hand, it may be difficult to calculate the learning filter when the number of samples is large, because the size of system matrix is dependent on the number of samples [10], [11]. This paper proposes a method which modifies lifted system LMI design method, to which we can apply frequency domain analysis. The rest of this paper is organized as follows. Section I I presents the proposed design method, and Section Ⅲ discusses the design of ILC for an industrial robot based on the proposed method, and introduces weighting functions to shape the learning filter to increase robustness in a high frequency range. We also examine the condition for monotonic convergence of the output of the feedback controllers after implementing ILC. Finally section Ⅳ concludes the paper.

K. Inaba is with Fanuc ltd., Oshino-mura Yamanashi, Japan. kiyo1722@gmail.com
C. -C. Wang is with FormFactor Inc., Livermore CA., USA. ccwang1120@gmail.com
M. Tomizuka is with the Faculty in the Department of Mechanical Engineering, University of California Berkeley, USA. tomizuka@me.berkeley.edu
A. Packard is with the Faculty in the Department of Mechanical Engineering, University of California Berkeley, USA. pack@me.berkeley.edu

II. ILC DESIGN METHOD

A. General ILC System

General ILC systems may be represented by (1) [12]. The system is assumed to be a discrete-time, linear time invariant system.

$$y_j(k+1) = P(z)u_j(k) + d(k) \tag{1}$$

P is an asymptotically stable closed loop system with a feedback controller, u_j is an ILC input, j is the iteration index, and k is the time index. y_j is the output which we desire to control, and d is due to other inputs to the closed loop system P. The following signals are stored in memory.

$$u_j(k), k \in 0, 1, \ldots., N-1$$
$$y_j(k), k \in 0, 1, \ldots., N-1$$
$$y_d(k), k \in 0, 1, \ldots., N-1$$

$y_d(k)$ is the desired output reference, and N is the number of samples. An error signal is defined as

$$e_j(k) = y_d(k) - y_j(k) \tag{2}$$

The following ILC update law is used in this paper.

$$u_{j+1}(k) = Q(z)[u_j(k) + L(z)e_j(k)] \tag{3}$$

Q is a filter and L is a learning filter. The purpose of ILC design is to choose proper Q and L.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

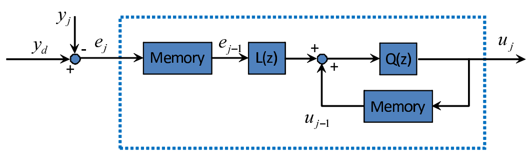

Fig. 1. ILC General Structure

B. LMI Method based on Frequency Domain

It is difficult to find the optimal solution of learning gains for a lifted system, because we need to analyze super-vector ILC system, even if it is known that the inverse of the Markov matrices is the optimized solution [9]. Also, we may face the implementation difficulty when the system is ill-conditioned [9]. Lifted system LMI method is a useful method to deal with these difficulties. We can convert the condition for stability and and monotonic convergence of the ILC system into an LMI problem. However, the analysis of the lifted system may be difficult when it has a large number of samples, because the size of the matrix is dependent on the number of samples. In that case, the computation time of the ILC gains may be too long, and the memory requirement may be excessive. In this section, we will discuss the design of ILC gains in frequency domain instead of in time domain, by modifying the lifted system LMI method.

The basic idea of the LMI method is to formulate a given problem as an optimization problem with linear objective and linear matrix inequality (LMI) constraints. An LMI constraint on a vector $x \in \mathbf{R}^m$ is of the form,

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \succeq 0 \qquad (4)$$

where the symmetric matrices $F_i = F_i^* \in \mathbf{R}^{n\times n}$, $i = 0, 1, \cdots, m$ are given [13]. $F \succeq 0$ means the matrix F is symmetric and positive semidefinite. The minimization problem is

$$minimize \quad c^T x \quad subject \quad to \quad F(x) \succeq 0, \qquad (5)$$

where $c \in \mathbf{R}^m$. Since F_i is positive semidefinite, this problem is called a semidefinite program (SDP). This framework has efficient numerical calculation by LMI programs. We will convert the conditions for stability and monotonic convergence of the system into the LMI constraints.

Letting $L_k \in \mathbf{R}^{N_y\times N_u}$, the lifted ILC filter can be represented in frequency domain as [14],

$$\begin{aligned} L(z) = L_0 z^{-N_o} + L_1 z^{-N_o+1} \cdots + L_{N_o-1} z^{-1} + L_{N_o} \\ + L_{N_o+1} z + \cdots + L_{N_o-1} z^{2N_o-1} + L_{2N_o} z^{N_o} \end{aligned} \qquad (6)$$

where $2N_o + 1$ is the order of the ILC filter.

For the ILC system defined by (1) and (6), the stability condition for the ILC system can be represented as [12],

$$||Q(z)(I - L(z)P(z))||_\infty = \gamma \qquad (\gamma < 1) \qquad (7)$$

It is equivalent to the following equation. Letting $M(z) = Q(z)(I - L(z)P(z))$, (7) will be represented as,

$$\sup_{\Omega \in \upsilon} \bar{\sigma}[\, M(e^{j\Omega})\,] = \gamma \qquad (\gamma < 1) \qquad (\upsilon = \mathbf{R}) \qquad (8)$$

To establish (8), we need to check $\sup_{\Omega\in\bar{\upsilon}} \bar{\sigma}[\, M(e^{j\Omega})\,] = \gamma$ at every Ω. Practically, however, we can consider this condition at only a finite but large number of frequencies, N_p. i.e.

$$\sup_{\Omega \in \bar{\upsilon}} \bar{\sigma}[\, M(e^{j\Omega_i})\,] = \gamma \qquad (\gamma < 1) \qquad (i = 1, 2, \cdots, N_p) \qquad (\bar{\upsilon} = \Omega_1, \Omega_2, \cdots, \Omega_{N_p}) \qquad (9)$$

which will be represented as

$$\begin{bmatrix} \gamma^2 I & M^*(e^{j\Omega_i}) \\ M(e^{j\Omega_i}) & I \end{bmatrix} \succeq 0 \qquad (\gamma < 1 \ for \ all \ \Omega_i) \qquad (10)$$

This can be represented as,

$$\gamma^2 \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & Q(e^{j\Omega_i})^* \\ Q(e^{j\Omega_i}) & I \end{bmatrix} + \begin{bmatrix} 0 & (-QLP(e^{j\Omega_i}))^* \\ -QLP(e^{j\Omega_i}) & 0 \end{bmatrix} \succeq 0 \qquad (11)$$

We will represent the third term in the above expression as a standard LMI constraint.
Define,

$$\phi_k(z) \;=\; z^{-N_o+(k-1)} \qquad (12)$$

Then,

$$\begin{aligned} L(e^{j\Omega_i}) &= L_0 \phi_0(z)|_{z=e^{j\Omega_i}} + \cdots + L_{2N_o} \phi_{2N_o}(z)|_{z=e^{j\Omega_i}} \\ &= \sum_{k=0}^{2N_o} L_k \phi_k(z)|_{z=e^{j\Omega_i}} \end{aligned} \qquad (13)$$

We will represent L_k as a linear combination of scalars $\alpha_{k,j}$, multiplied with matrices V_j. The dimension of V_j are the same as that of L_k. V_j are matrices of all zeros except for 1 at entry indexed by j, where j is counted horizontally starting from the top-left entry and wrapped around at the end of each row. For example, supposing $N_y = 2$ and $N_u = 2$,

$$\begin{aligned} L_k &= \begin{bmatrix} L^k_{1,1} & L^k_{1,2} \\ L^k_{2,1} & L^k_{2,2} \end{bmatrix} \\ &= L^k_{1,1} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} + L^k_{1,2} \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \\ &+ L^k_{2,1} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + L^k_{2,2} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ &= \alpha_{k,1} V_1 + \alpha_{k,2} V_2 + \alpha_{k,3} V_3 + \alpha_{k,4} V_4 \end{aligned}$$

Therefore,

$$L(e^{j\Omega_i}) = \sum_{k=0}^{2N_o} \sum_{j=1}^{N_y N_u} \alpha_{k,j} V_j \phi_k(z)|_{z=e^{j\Omega_i}} \qquad (14)$$

Noting (14), $Q(e^{j\Omega_i})L(e^{j\Omega_i})P(e^{j\Omega_i})$ is expressed as,

$$\begin{aligned} &Q(e^{j\Omega_i})L(e^{j\Omega_i})P(e^{j\Omega_i}) \\ &= Q(e^{j\Omega_i}) \sum_{k=0}^{2N_o} \sum_{j=1}^{N_yN_u} \alpha_{k,j} V_j \phi_k(z)|_{z=e^{j\Omega_i}} P(e^{j\Omega_i}) \\ = &\sum_{l=1}^{(2N_o+1)N_yN_u} \beta_l F_l^i \end{aligned} \quad (15)$$

where,

$$\begin{aligned} \beta_l &= \alpha_{k,j} \\ F_l^i &= Q(e^{j\Omega_i}) V_j P(e^{j\Omega_i}) \end{aligned}$$

Finally, (10) is represented as,

$$\gamma^2 \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} - \sum_{l=1}^{(2N_o+1)N_yN_u} \beta_l \begin{bmatrix} 0 & (F_l^i)^* \\ F_l^i & 0 \end{bmatrix} + \begin{bmatrix} 0 & Q^* \\ Q & I \end{bmatrix} \succeq 0 \quad (16)$$

This representation is the LMI constraint form (4). Letting the first and the second terms be $\sum_{i=1}^{m} x_i F_i$ and the third term be F_0, c and x in (5) are respectively $c = [1, 0, \cdots, 0]$ and $x = [\gamma^2, \beta_1, \cdots, \beta_{(2N_o+1)N_yN_u}]$. This LMI problem is to find learning gains to minimize γ^2. Notice that the size of the matrices in the inequality is $2N_y \times 2N_y$ and is independent of the number of samples.
The monotonic condition for the ILC system

$$\begin{aligned} ||Q(z)(I - P(z)L(z))||_\infty = \gamma \\ (\gamma < 1\ ) \end{aligned} \quad (17)$$

can be solved in a similar way, Also notice if the system is SISO system, the stability condition (7) and the monotonic convergence condition (17) are the same equation.

III. Application to a Robot Manipulator

In this section, we will apply the proposed method to a robot manipulator. In this paper, our goal is precise position tracking in joint space. When the same task is repeated for industrial robots, ILC is a promising control method to improve performance.

A. Description of the robot manipulator

The robot manipulator, used in this paper, is an M-16*i*B/20 industrial robot shown in Fig. 2. It is a six-axis, medium size robot and can carry objects up to 20 *kg* at a maximum speed of 2000 *mm*/*sec*. It is mainly used for high-speed applications such as spot welding, material handling, sealing and water-jet cutting. The specification of this robot can be found in [15]. The hardware connection diagram of the experimental setup is shown in Fig. 2. Controllers are implemented by *xPC Target* [16] installed in the target PC. In experiments, the available sensors are motor encoders.

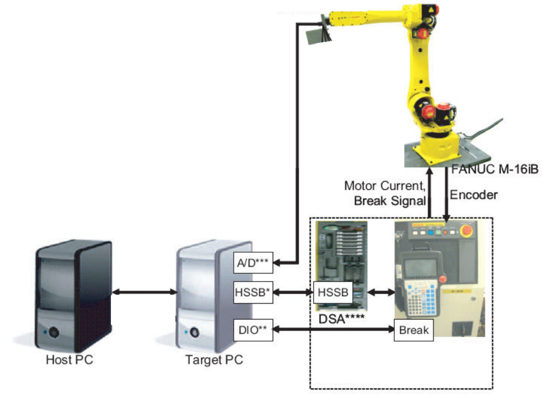

Fig. 2. M-16*i*B/20 and hardware connection

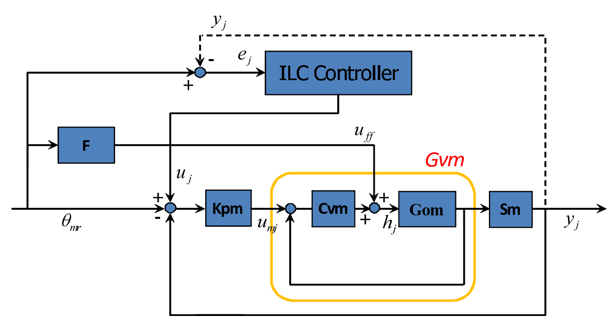

Fig. 3. Controller Structure

B. Structure of Controller

Fig. 3 illustrates the control structure for the robot. F is a feed-forward controller. K_{pm} is a proportional controller, and C_{vm} is a velocity controller. G_{om} is the plant model from the current in the motor to the motor velocity. S_m is a integrator. j is a iteration index. θ_{mr} is the motor reference trajectory. u_{ff} is the control output of the feed-forward controller. y_j is the measured motor position by encoder. u_j is the ILC input, and u_{mj} is the control output of the proportional controller, and h_j is the output of velocity controller. y_j is the measured motor position. Therefore, the overall system can be described by the following set of equations by defining the converged ILC input as $u_\infty(k) = \lim_{j\to\infty} u_j(k)$, and the converged error as $e_\infty(k) = \lim_{j\to\infty} e_j(k)$.

$$y_j(k) = T_p(z)u_j(k) \quad (18)$$

$$\begin{aligned} u_{j+1}(k) - u_\infty(k) = Q(I - L(z)T_p(z)) \\ (u_j(k) - u_\infty(k)) \end{aligned} \quad (19)$$

$$\begin{aligned} e_{j+1}(k) - e_\infty(k) = Q(I - T_p(z)L(z)) \\ (e_{j+1}(k) - e_\infty(k)) \end{aligned} \quad (20)$$

where,

$$
\begin{aligned}
e_j(k) &= \theta_j - \theta_{mr} && (21)\\
S_p(z) &= (I + S_m G_{vm} K_{pm})^{-1} && (22)\\
T_p(z) &= (I + S_m G_{vm} K_{pm})^{-1}(S_m G_{vm} K_{pm}) && (23)\\
S_v(z) &= (I + G_{om} C_{vm})^{-1} && (24)\\
T_v(z) &= (I + G_{om} C_{vm})^{-1} G_{om} C_{vm} && (25)
\end{aligned}
$$

S_p, T_p represent the sensitivity and complementary sensitivity functions for the position loop, and S_v, T_v represent the sensitivity and complementary sensitivity functions for the velocity loop. The bode diagram of T_p is shown in Fig. 4. The bandwidth of the closed-loop system is around 3 Hz.

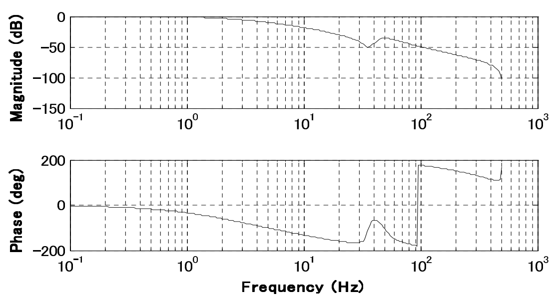

Fig. 4. Bode plot of T_p

C. Iterative learning control design

We consider SISO (single-input single-output system) ILC design. The system order is 4th order, and Q-filter is designed as a zero-phase 4th order butterworth filter with cut-off frequency equal to 20 Hz. The Bode plot of the Q-filter is shown in Fig. 5. Let $N_o = 1$, so that the resulting controller is 3rd order, which is much lower than the order of the controller by H_∞ design. As mentioned before, in SISO system, the condition for system stability and the condition for monotonic convergence become the same. In other words, $||Q(1 - L(z)T_p(z))||_\infty < 1$. We will use the condition as a constraint condition for LMI problem. The numerical calculation of LMI was performed by *Sedumi* software [17]. γ converged to a value of $\gamma = 0.2165$. The resulting learning filter is shown with $T_p(z)^{-1}$ in Fig. 6. In Fig 6, we notice that the Bode plot of $L(z)$ follows that of $T_p^{-1}(z)$. As a result, the gain of $L(z)$ is high at high frequencies. The gain of $L(z)$ is close to T_p^{-1} up to 300 Hz, as shown in Fig. 6.

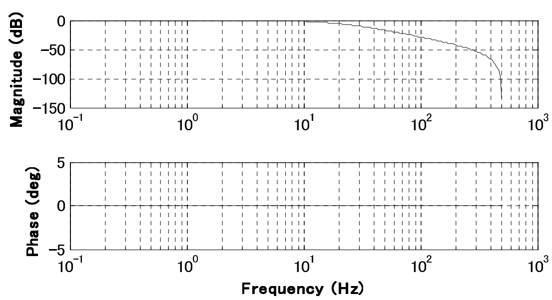

Fig. 5. Bode plot of Q-filter

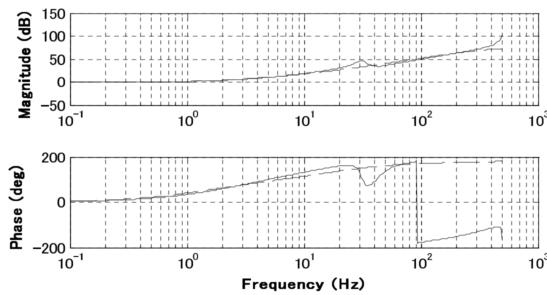

Fig. 6. Bode plot of T_p^{-1} (solid), Bode plot of L (dashed)

D. Shaping the learning filter

The high learning filter gain at high frequencies, where T_p is close to zero, may make the learning system sensitive to the initial condition error, the disturbance, and the uncertainty of the model at high frequencies. We do not need to make the learning gain track the inverse of T_p above the Q-filter bandwidth, because the learning process is needed only within Q-filter bandwidth. We will shape the learning filter by introducing a weighting function, W_p. Also, we may want to make learning filter active at some frequencies within the Q-filter bandwidth. Therefore, we will introduce another weighting function, W_t , for it. (7) and (17) can be represented after introducing weighting functions as,

$$\|W_t Q(1 - L W_p T_p)\|_\infty = \gamma \quad (\gamma < 1) \qquad (26)$$

The Bode plot of W_p and W_t are shown in Fig. 7 and Fig 8. W_p is a zero-phase weighting function to raise the gain of T_p at high frequencies in order to shape the learning filter. W_p is 1 within Q-filter bandwidth and increase at high frequencies which is at least above Q-filter bandwidth. We set W_p to increase around 50 Hz. W_t is also a zero-phase weighting function for adjusting the value of $(1 - LP)$ at some frequencies within Q-filter bandwidth. As the value of $(1 - LP)$ become smaller, the learning process become more active. W_t is 1 outside of Q-filter bandwidth. In this case, we set the W_t to be 3 up to 0.1 Hz. It means the value of $(1 - LP)$ will be less than $\frac{1}{3}$ up to 0.1 Hz.

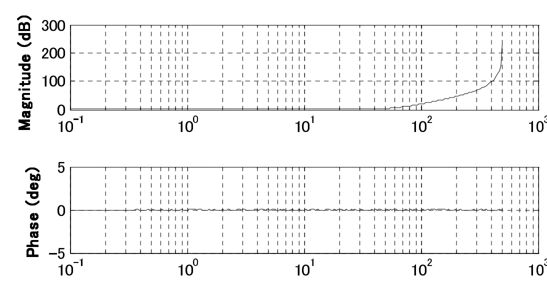

Fig. 7. Bode plot of W_p

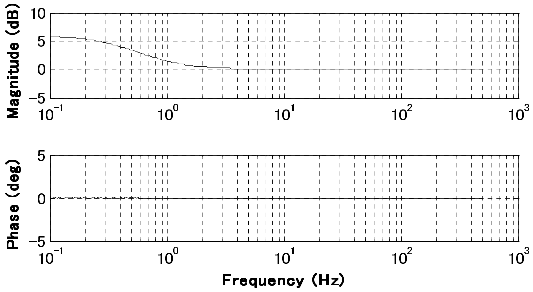

Fig. 8. Bode plot of W_t

γ converged to a value of 0.3462. The resulting learning filter is shown with $T_p(z)^{-1}$ in Fig. 9. As we can see in Fig 10, the designed ILC satisfies the condition for stability and monotonic convergence. Also, $||Q(1-LP)||_\infty$ is less than $\frac{1}{3}$ up to 0.1 Hz. In Fig 9, we can notice the difference between the previous designed $L(z)$ and the new designed $L(z)$. The Bode plot of the new designed $L(z)$ follows the bode plot of T_p up to around 20 Hz, Q-filter bandwidth. However, the learning gain does not increase at high frequencies , and goes down.

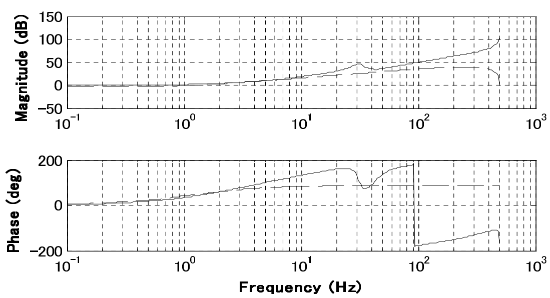

Fig. 9. Bode plot of T_p^{-1} (solid), Bode plot of L (dashed)

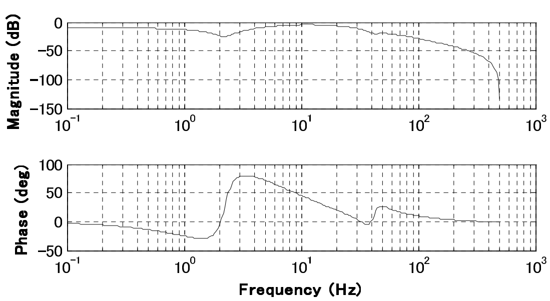

Fig. 10. Bode plot of $Q(1-LP)$

E. Observation of Output of Feedback Controller after Implementing ILC

We designed a learning filter by considering two conditions, stability and monotonic convergence. However, we did not consider the characteristics of the output of the feedback controller after implementing ILC. Even if ILC input itself is acceptable, it is possible that the output of the feedback controller is not acceptable, for example, its norm of the output of the feedback controller flutters a lot in each iteration. In this section, we will observe the relationship between the norm of feedback input and the design of Q and L. h_j is the output of the feedback controller as shown in Fig. 3.

$$\begin{aligned} h_j &= C_{vm}S_v u_{mj} + C_{vm}S_v C_{vm}^{-1} u_{ff} \\ &= C_{vm}S_v K_{pm}S_p(u_j+\theta_{mr}) \\ &+ C_{vm}S_v C_v^{-1} u_{ff} \end{aligned} \tag{27}$$

$$\begin{aligned} e_j &= \theta_{mr} - y_j \\ &= \theta_{mr} - S_m G_{mo} h_j \end{aligned} \tag{28}$$

From (3), (27) and (28),

$$\begin{aligned} h_{j+1} &= C_{vm}S_v K_{pm}S_p(u_{j+1}+\theta_{mr}) \\ &+ C_{vm}S_v C_v^{-1} u_{ff} \\ &= C_{vm}S_v K_{pm}S_p(Q(u_{j+1}+Le_j)+\theta_{mr}) \\ &+ C_{vm}S_v C_v^{-1} u_{ff} \\ &= Q(I - C_{vm}S_v K_{pm}S_p L S_m G_{mo})h_j + \\ &(I-Q)C_{vm}S_v K_{pm}S_p\theta_{mr} + (I-Q)C_{vm}S_v C_v^{-1}u_{ff} \end{aligned} \tag{29}$$

Therefore,

$$\begin{aligned} &||h_{j+1} - h_\infty||_\infty \\ &= Q(I - C_{vm}S_v K_{pm}S_p L S_m G_{mo})||h_j - h_\infty||_\infty \end{aligned} \tag{30}$$

In SISO, considering (23), it follows that,

$$||h_{j+1} - h_\infty||_\infty = Q(1 - T_p L)||h_j - h_\infty||_\infty \tag{31}$$

As a result, we find that the condition for feedback signal convergence is the same as the condition for stability and monotonic convergence.

F. Experimental Results

We applied the learning filter to the industrial robot. The sampling time of ILC is 1 *msec*, and the number of error and ILC input samples is 3500. The results are shown in Fig. 11 – Fig. 15. Fig. 11 shows reference signal and measured motor position before ILC.

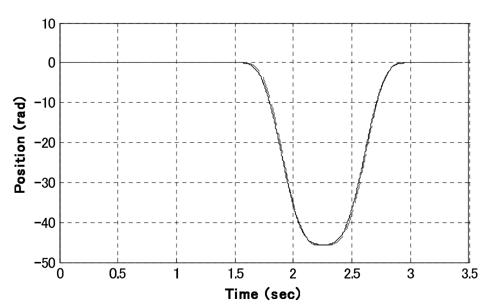

Fig. 11. Reference (Solid) and Motor Position before ILC (dashed)

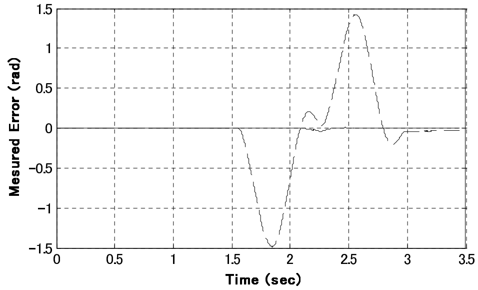

Fig. 12. Motor Position Error before ILC (dashed) and after ILC (Solid)

We can observe in Fig. 12 that the motor position error becomes close to zero after implementing ILC, which implies that Q-filter bandwidth is high enough to learn the external disturbance. Also, we can confirm stability and monotonic convergence of the system in Fig. 13−Fig. 15. The ILC input converges after 5−6 cycles, and its convergence is monotonic. Also, we can observe that the norm of error and the norm of the output of feedback controller converge monotonically after 3−4 cycles.

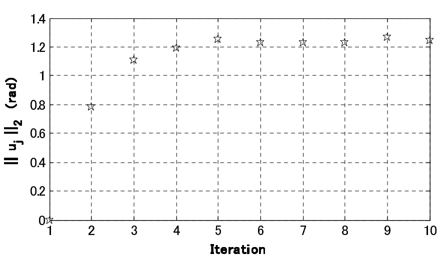

Fig. 13. Norm of ILC Input

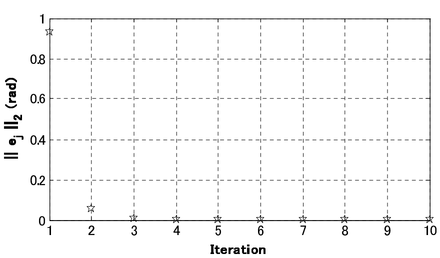

Fig. 14. Norm of Motor Position Error

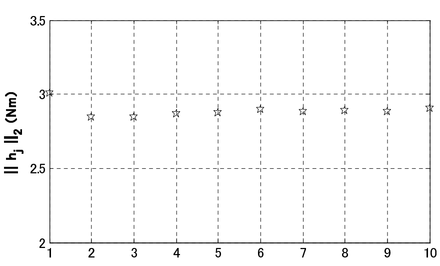

Fig. 15. Norm of Output of Feedback Controller

IV. CONCLUSIONS

We proposed a design method for ILC in the frequency domain, by modifying the lifted system LMI method. The designed ILC was applied to an industrial robot after shaping the learning filter by weighting functions. Also, we examined the condition for monotonic convergence of the output of the feedback controllers after implementing ILC. Through experiments with the robot, we demonstrated stability of the system and monotonic convergence of the norm of ILC input, the norm of error, and the norm of the feedback controller output, as well as showing the improvement of the tracking performance.

V. ACKNOWLEDGMENTS

This work was in part supported by Fanuc Ltd. The authors would also like to thank H. Stearns.

REFERENCES

[1] C. -C. Wang, "Motion Control of Indirect-Drive Robots", *Ph.D thesis, Department of Mechanical Engineering, U. C. Berkeley*
[2] M. Uchiyama, "Formulation of high-speed motion pattern of mechanical arm by trial", *Trans. SICE (Soc. Instrum. Contr. Eng.)*, vol.14, no. 6, pp.706-712(in Japanese), 1978
[3] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning", *J. of Robotic Systems*, vol.1, no. 2, pp.123-140, 1984
[4] G. Casalino, G. Bartolini, "A learning procedure for the control of movements of robotic manipulators", *IASTED symposium on robotics and automation, Amsterdam*, 1984, pp. 108-111
[5] J. J. Craig, "Adaptive control of manipulators through repeated trials", in *Proceedings of the American control conference, San Diego*, 1984, pp. 1566-1574
[6] D. D. Roover, "Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system", *international Journal of Control*, vol. 73, no. 10, 2000, pp. 968-979
[7] J. H. Moon, T. Y. Doh, and M. J. Chung, "A robust approach to iterative learning control design for uncertain systems", *Automatica*, vol. 34, no. 8, 1998, pp. 1001-1004
[8] J. Xu, M. Sun, and L. Yu, "LMI-based synthesis of robust iterative learning controller with current feedback for linear uncertain systems", *International Journal of Control Automation and Systems*, vol. 6, no. 2, 2008, pp. 171-179
[9] H. Ahn, K. L. Moore, and Y. Chen, "LMI Approach to Iterative Learning Control Design", *Adaptive and Learning Systems, 2006 IEEE Mountain Workshop on*, 2006, pp.72-77
[10] S. Mishra and M. Tomizuka, "Iterative Learning Control Design Application Waferstage Control", *MSC Seminar* , Mar 2007
[11] S. Mishra and M. Tomizuka, "Iterative Learning Control Design for Waferstage Positioning based on Orthogonal Projection", *American Control Conference* , Seattle, WA, Jun 2008
[12] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control", *Ieee Control Systems Magazine*, vol. 26, no. 3, 2006, pp. 96-114
[13] L. El. Ghaoui and S. Niculescu, *Advances in linear matrix inequality methods in control* , Society for Industrial and Applied Mathematics, PA; 1999, pp. 3-37
[14] S. -C. Wu and M. Tomizuka, "An Iterative Learning Control Design for Self-Servowriting in Hard Disk Drive", in *Proceeding of the 17th World Congress The International Federation of Automatic Control*, Seoul, Korea, 2008
[15] Fanuc Ltd., "http://www.fanucrobotics.com/24,987.html"
[16] Mathworks., "http://www.mathworks.com/products/xpctarget/"
[17] Sedumi, "http://sedumi.mcmaster.ca/"