High Precision Control of Ball Screw Driven Stage Using Repetitive Control with Sharp Roll-off Learning Filter

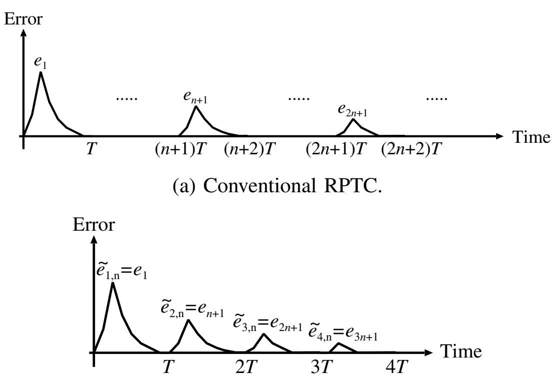
Tadashi Takemura and Hiroshi Fujimoto
The University of Tokyo
5-1-5, Kashiwanoha, Kashiwa, Chiba, 277-8561 Japan
Phone: +81-4-7136-3881
Fax: +81-4-7136-4132
Email: take@hfllab.k.u-tokyo.ac.jp, fujimoto@k.u-tokyo.ac.jp

***Abstract*—Repetitive perfect tracking control (RPTC) is one of repetitive control techniques to achieve high precision positioning. In this paper, RPTC with n-times learning filter is proposed. n-time learning filter has a sharp roll-off property than conventional learning filter. By using n-times learning filter, proposed RPTC can converge tracking error n-times faster than RPTC with conventional learning filter. Simulations show the fast convergence of proposed RPTC. Finally, experiments in ball screw driven stage also show the effectiveness of proposed system.**

I. INTRODUCTION

From the viewpoint of productivity and microfabrication, high-speed and high-precision positioning techniques are desired to industrial equipments. Repetitive control (RC) is one of techniques to achieve high precision positioning [1]. This is a widely used technique to reject periodic disturbances or to track a periodic reference signal [2],[3].

Author's research group had proposed perfect tracking control (PTC) which is 2DOF controller consisting of feedback and multi-rate feedforward controller [4]. If there are no modeling errors, this control system guarantees perfect tracking. Moreover, repetitive perfect tracking control (RPTC) which is repetitive control system using multi-rate feedforward controller had been proposed and verified its effectiveness in some industrial devices [6]–[8]. By mathematical analysis, it is verified that RPTC suppresses input and output disturbance and guarantees perfect tracking by only one time learning under ideal condition. Here, ideal condition means no modeling error and no low-pass filter. However, modeling error is unavoidable. In addition, low-pass filter is necessary because it ensures the stability of RPTC. For these reasons, we cannot use RPTC under ideal condition. Tracking error which occurred in first iteration decreases by learning, illustrated as Fig. 1(a). Here, $\boldsymbol{e}_{n+1}$ is a vector of tracking error after n-time iteration. From standpoint of high-speed and high-precision control, we hope the number of iteration gets smaller.



(a) Conventional RPTC.

(b) Proposed RPTC.

Fig. 1. Image of error convergence.

In this paper, we proposed n-time learning RPTC which is constructed with n-time learning filter instead of Q filter. Although $\boldsymbol{e}_{n+1}$ is accomplished after n-time learning by conventional RPTC, $\boldsymbol{e}_{n+1}$ is accomplished after only one time learning by proposed RPTC. n-time learning filter is multistage Q filter and has a sharp roll-off property. From mathematical analysis, it is verified that proposed RPTC can converge tracking error n-time faster than conventional RPTC. Simulations show the fast convergence of proposed RPTC and the validity of mathematical analysis. Finally, experimental results show convergence improvement by using proposed RPTC.

II. EXPERIMENTAL STAGE

The experimental ball screw driven stage is shown in Fig. 2. The ball screw is directly connected with the shaft of the servo motor through the coupling. In this paper, we only control X-axis. The rotary encoder is attached to the servo motor. High speed current feedback control is implemented to the servo motors.

Fig. 3 shows the frequency response from current reference i_q^{ref} to angular velocity ω by using FFT analyzer. From this figure, nominal plant $P_n(s)$ is determined as

$$P_n(s) = \frac{K_T}{Js^2 + Ds} \quad (1)$$

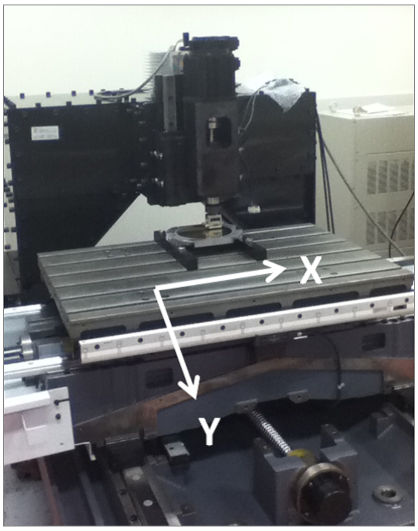


Fig. 2. Experimental stage.

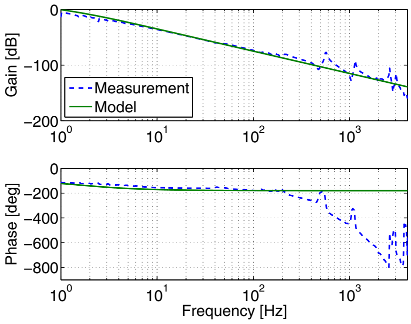


Fig. 3. Frequency response.

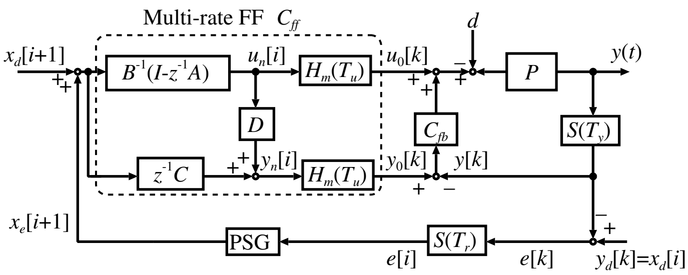


Fig. 4. Block diagram of RPTC.

where torque constant $K_T = 0.715$ Nm/A, inertia $J = 0.01$ kgm^2, and viscous friction $D = 0.1$ Nms/rad, respectively.

III. RPTC (CONVENTIONAL)

RPTC is one of repetitive control which can suppress periodic disturbance by iteration (learning) [5],[8]. In this section, structure of RPTC and convergence condition are explained.

A. Structure of RPTC

Block diagram of RPTC is described in Fig. 4. RPTC consists of FB controller C_{fb}, multi-rate FF controller C_{ff}, and learning unit which is called periodic signal generator

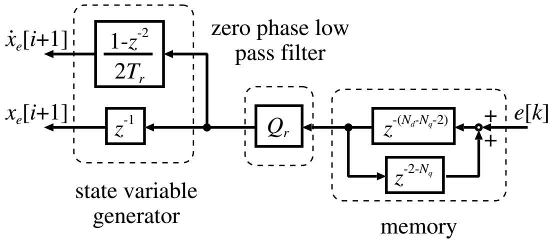


Fig. 5. Block diagram of PSG.

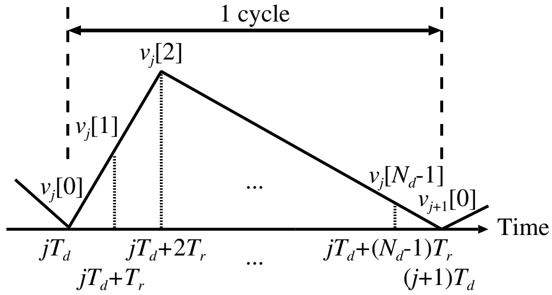


Fig. 6. Iterative signal.

(PSG). PSG which is illustrated in Fig. 5 is composed of memory, learning filter (low-pass filter) Q_r, and state variable generator. Memory plays a part of memorizing tracking error and compensation of sample delays which occur in Q_r and state variable generator. N_d is the number of memory required for RPTC. N_d is given as $N_d = T_d/T_r$, where T_d is disturbance cycle and T_r is multi-rate sampling time. Learning filter Q_r is realization of zero-phase low-pass filter Q. Q and Q_r are given as following equations, respectively.

$$Q = \left(\frac{z + 2 + z^{-1}}{4}\right)^{N_q} \quad (2)$$

$$Q_r = z^{-N_q} Q \quad (3)$$

Q_r acts as Q thanks to sample delay compensation of memory. State variable generator generates state variable, position and velocity, which is required for multi-rate FF controller. Velocity is calculated by center difference in order to avoid sample delay. Periodic disturbance is suppressed by following steps.

- Step1 (learning)
  Tracking error is stored to memory.
- Step2 (state variable calculation)
  State variable is calculated from stored error.
- Step3 (redesigning of target trajectory)
  Calculated state variable is added to target trajectory. Then error becomes smaller than previous cycle.

B. Definition of signals and transfer functions

Signals in iteration cycle $T_d(= N_d T_r)$, shown in Fig. 6, are represented as following vector.

$$\boldsymbol{v}_j = [v_j[0], v_j[1], \ldots, v_j[N_d - 1]]^T \quad (4)$$

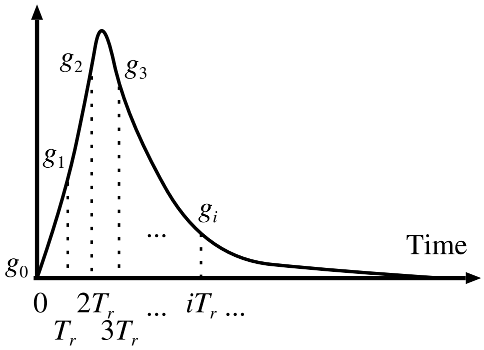


Fig. 7. Impulse response of G.

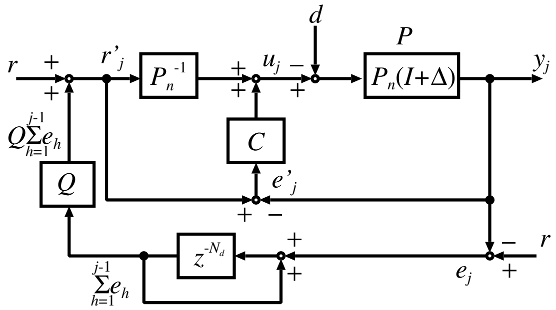


Fig. 8. Equivalent block diagram of RPTC.

Here, sampling time is T_r and subscript is number of iterations. Transfer function of 1 cycle can be written as impulse response matrix $\boldsymbol{G}$.

$$\boldsymbol{G} = \begin{bmatrix} g_0 & 0 & \dots & 0 \\ g_1 & g_0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ g_{N_d-1} & g_{N_d-1} & \cdots & g_0 \end{bmatrix} \tag{5}$$

Here, $g_i (i = 1, 2, \dots, N_d-1)$ are impulse response coefficient of transfer function G which is described in Fig. 7. Input-output relation of each sampling time T_r in iteration cycle T_d is described by input-output vectors and impulse response matrices as follows:

$$\boldsymbol{y}_j = \boldsymbol{G}\boldsymbol{H}\boldsymbol{u}_j = \boldsymbol{H}\boldsymbol{G}\boldsymbol{u}_j. \tag{6}$$

Multiplication of impulse response matrices is commutative because they are lower triangular and Toeplitz matrices.

C. Convergence condition of RPTC

Convergence condition of RPTC is derived by using an equivalent block diagram which is illustrated in Fig. 8. This equivalent block diagram is based on following assumptions.

- RPTC is multi-rate control system. However, RPTC is treated as single-rate system for simplicity.
- Multi-rate FF controller C_{ff} is treated as $C_{ff} = P_n^{-1}$ because this is a perfect inverse system, both gain and phase, of the plant.
- In order to simplify the system, only position is used for input of C_{ff} although C_{ff} requires state variable for its input. Due to this assumption, state variable generator can be ignored.

Convergence condition of RPTC is derived from relation of error vector between jth and $(j+1)$th iteration. Output vector of jth iteration $\boldsymbol{y}_j$ is given as,

$$\begin{aligned} \boldsymbol{y}_j &= \boldsymbol{P}(\boldsymbol{u}_j - \boldsymbol{d}) \\ &= \boldsymbol{P}\left\{(\boldsymbol{P}_n^{-1} + \boldsymbol{C})\boldsymbol{r}'_j - \boldsymbol{C}\boldsymbol{y}_j - \boldsymbol{d}\right\}, \end{aligned}$$

$$\begin{aligned} \boldsymbol{y}_j &= (\boldsymbol{I} + \boldsymbol{P}\boldsymbol{C})^{-1}\boldsymbol{P}\left\{(\boldsymbol{P}_n^{-1} + \boldsymbol{C})\boldsymbol{r}'_j - \boldsymbol{d}\right\} \\ &= \boldsymbol{S}\boldsymbol{P}\left\{(\boldsymbol{P}_n^{-1} + \boldsymbol{C})(\boldsymbol{r} + \boldsymbol{Q}\sum_{h=1}^{j-1}\boldsymbol{e}_h) - \boldsymbol{d}\right\}, \end{aligned} \tag{7}$$

where

$$\boldsymbol{S} = (\boldsymbol{I} + \boldsymbol{P}\boldsymbol{C})^{-1}. \tag{8}$$

From (7), error vector of jth iteration $\boldsymbol{e}_j$ is represented as

$$\begin{aligned} \boldsymbol{e}_j &= \boldsymbol{r} - \boldsymbol{y}_j \\ &= \left\{\boldsymbol{I} - \boldsymbol{S}\boldsymbol{P}(\boldsymbol{P}_n^{-1} + \boldsymbol{C})\right\}\boldsymbol{r} \\ &\quad -\boldsymbol{Q}\boldsymbol{S}\boldsymbol{P}(\boldsymbol{P}_n^{-1} + \boldsymbol{C})\sum_{h=1}^{j-1}\boldsymbol{e}_h - \boldsymbol{S}\boldsymbol{P}\boldsymbol{d}. \end{aligned} \tag{9}$$

Error vector of the next iteration $\boldsymbol{e}_{j+1}$ is also represented as

$$\begin{aligned} \boldsymbol{e}_{j+1} &= \left\{\boldsymbol{I} - \boldsymbol{S}\boldsymbol{P}(\boldsymbol{P}_n^{-1} + \boldsymbol{C})\right\}\boldsymbol{r} \\ &\quad -\boldsymbol{Q}\boldsymbol{S}\boldsymbol{P}(\boldsymbol{P}_n^{-1} + \boldsymbol{C})\sum_{h=1}^{j}\boldsymbol{e}_h - \boldsymbol{S}\boldsymbol{P}\boldsymbol{d} \\ &= \left(\boldsymbol{I} - \boldsymbol{Q}\boldsymbol{S}\boldsymbol{P}\boldsymbol{P}_n^{-1} + \boldsymbol{Q}\boldsymbol{S}\boldsymbol{P}\boldsymbol{C}\right)\boldsymbol{e}_j. \end{aligned} \tag{10}$$

If plant $\boldsymbol{P}$ has a multiplicative modeling error $\boldsymbol{\Delta}$, $\boldsymbol{P}$ is expressed as

$$\boldsymbol{P} = \boldsymbol{P}_n(\boldsymbol{I} + \boldsymbol{\Delta}). \tag{11}$$

From (8),

$$\boldsymbol{S}\boldsymbol{P}\boldsymbol{C} = \boldsymbol{I} - \boldsymbol{S} \tag{12}$$

is given. Then $\boldsymbol{e}_{j+1}$ is written as follows by (10), (11), and (12).

$$\begin{aligned} \boldsymbol{e}_{j+1} &= \left\{\boldsymbol{I} - \boldsymbol{Q}\boldsymbol{S}(\boldsymbol{I} + \boldsymbol{\Delta}) - \boldsymbol{Q}(\boldsymbol{I} - \boldsymbol{S})\right\}\boldsymbol{e}_j \\ &= (\boldsymbol{I} - \boldsymbol{Q} - \boldsymbol{Q}\boldsymbol{S}\boldsymbol{\Delta})\,\boldsymbol{e}_j \\ &= (\boldsymbol{I} - \boldsymbol{Q} - \boldsymbol{Q}\boldsymbol{S}\boldsymbol{\Delta})^j\,\boldsymbol{e}_1 \end{aligned} \tag{13}$$

(13) indicates error vector $\boldsymbol{e}_1$ which happened in 1st iteration converges with $(\boldsymbol{I} - \boldsymbol{Q} - \boldsymbol{Q}\boldsymbol{S}\boldsymbol{\Delta})$ by iteration. Moreover, $\boldsymbol{e}_2 = \boldsymbol{0}$ (zero vector) under ideal condition which is $\boldsymbol{Q} = \boldsymbol{I}$ and $\Delta = \boldsymbol{O}$. This means perfect tracking is achieved after first learning. This is the most interesting property of RPTC.

Designing of Q is important because stability and error convergence depend on Q. The sufficient condition of

$$||\boldsymbol{e}_1||_2 > ||\boldsymbol{e}_2||_2 > \cdots > ||\boldsymbol{e}_n||_2 > \dots \tag{14}$$

is to satisfied following inequation [9].

$$||\boldsymbol{I} - \boldsymbol{Q} - \boldsymbol{Q}\boldsymbol{S}\boldsymbol{\Delta}||_\infty < 1 \tag{15}$$

IV. N-TIME LEARNING RPTC WITH N-TIME LEARNING FILTER $\tilde{Q}_n$ (PROPOSED)

By using RPTC, tracking error converges as number of iterations increase. In this part, in order to improve tracking error convergence, we propose n-time learning RPTC with n-time learning filter $\tilde{Q}_n$. Proposed RPTC is realized by replacement of learning filter Q with $\tilde{Q}_n$. As Q is replaced with $\tilde{Q}_n$, convergence condition changes. By this, proposed RPTC can learn n times faster than conventional under $\Delta = \boldsymbol{O}$. Change of convergence condition by replacement of learning filter is verified. Furthermore, characteristic of $\tilde{Q}_n$ is described.

A. Change of convergence condition by replacement of learning filter

When $\boldsymbol{e}_1$ occurred in conventional RPTC, $\boldsymbol{e}_3, \boldsymbol{e}_4, \ldots, \boldsymbol{e}_{n+1}$ are written as follows by using binomial theorem.

$$\begin{aligned} \boldsymbol{e}_3 &= (\boldsymbol{I} - \boldsymbol{Q} - \boldsymbol{QS\Delta})^2 \boldsymbol{e}_1 \\ &= \left\{(\boldsymbol{I}-\boldsymbol{Q})^2 - 2(\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{QS\Delta} + (\boldsymbol{QS\Delta})^2\right\} \boldsymbol{e}_1 \end{aligned} \tag{16}$$

$$\begin{aligned} \boldsymbol{e}_4 &= (\boldsymbol{I} - \boldsymbol{Q} - \boldsymbol{QS\Delta})^3 \boldsymbol{e}_1 \\ &= \left\{(\boldsymbol{I}-\boldsymbol{Q})^3 - 3(\boldsymbol{I}-\boldsymbol{Q})^2\boldsymbol{QS\Delta} \right. \\ &\quad \left. +3(\boldsymbol{I}-\boldsymbol{Q})(\boldsymbol{QS\Delta})^2 - (\boldsymbol{QS\Delta})^3\right\} \boldsymbol{e}_1 \end{aligned} \tag{17}$$

$$\vdots$$

$$\begin{aligned} \boldsymbol{e}_{n+1} &= (\boldsymbol{I} - \boldsymbol{Q} - \boldsymbol{QS\Delta})^n \boldsymbol{e}_1 \\ &= \left\{\sum_{m=0}^{n} {}_nC_m(\boldsymbol{I}-\boldsymbol{Q})^{n-m}(-\boldsymbol{QS\Delta})^m\right\} \boldsymbol{e}_1 \\ &= \left\{(\boldsymbol{I}-\boldsymbol{Q})^n + n(\boldsymbol{I}-\boldsymbol{Q})^{n-1}(-\boldsymbol{QS\Delta}) \right. \\ &\quad \left. + \sum_{m=2}^{n} {}_nC_m(\boldsymbol{I}-\boldsymbol{Q})^{n-m}(-\boldsymbol{QS\Delta})^m\right\} \boldsymbol{e}_1 \end{aligned} \tag{18}$$

Now, we define $\tilde{Q}_n$ as

$$\tilde{Q}_n = \sum_{m=1}^{n} {}_nC_m Q^m (-1)^{m+1} \tag{19}$$

and replace Q with Q_n. Then convergence condition changes into

$$\tilde{\boldsymbol{e}}_{j+1,n} = \left(\boldsymbol{I} - \tilde{\boldsymbol{Q}}_n - \tilde{\boldsymbol{Q}}_n\boldsymbol{S\Delta}\right)^j \boldsymbol{e}_1. \tag{20}$$

At this time, tracking error vector of 2nd iteration $\tilde{\boldsymbol{e}}_{2,n}$ is written as follows.

- When $n = 2$ ($\tilde{Q}_2 = 2Q - Q^2$),

$$\begin{aligned} \tilde{\boldsymbol{e}}_{2,2} &= \left\{\boldsymbol{I} - (2\boldsymbol{Q} - \boldsymbol{Q}^2) - (2\boldsymbol{Q} - \boldsymbol{Q}^2)\boldsymbol{S\Delta}\right\} \boldsymbol{e}_1 \\ &= \left\{(\boldsymbol{I}-\boldsymbol{Q})^2 - 2(\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{QS\Delta} - \boldsymbol{Q}^2\boldsymbol{S\Delta}\right\} \boldsymbol{e}_1. \end{aligned} \tag{21}$$

- When $n = 3$ ($\tilde{Q}_3 = 3Q - 3Q^2 + Q^3$),

$$\begin{aligned} \tilde{\boldsymbol{e}}_{2,3} &= \left\{\boldsymbol{I} - (3\boldsymbol{Q} - 3\boldsymbol{Q}^2 + \boldsymbol{Q}^3) \right. \\ &\quad \left. -(3\boldsymbol{Q} - 3\boldsymbol{Q}^2 + \boldsymbol{Q}^3)\boldsymbol{S\Delta}\right\} \boldsymbol{e}_1 \\ &= \left\{(\boldsymbol{I}-\boldsymbol{Q})^3 - 3(\boldsymbol{I}-\boldsymbol{Q})^2\boldsymbol{QS\Delta} \right. \\ &\quad \left. -3(\boldsymbol{I}-\boldsymbol{Q})\boldsymbol{Q}^2\boldsymbol{S\Delta} - \boldsymbol{Q}^3\boldsymbol{S\Delta}\right\} \boldsymbol{e}_1. \end{aligned} \tag{22}$$

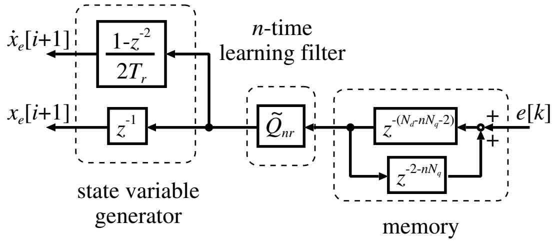


Fig. 9. PSG with n-time learning filter.

$$\vdots$$

- When $n = 3$ ($\tilde{Q}_n = \sum_{m=1}^{n} {}_nC_m Q^m (-1)^{m+1}$),

$$\begin{aligned} \boldsymbol{e}_{2,n} &= \left[\boldsymbol{I} - \sum_{m=1}^{n} {}_nC_m\boldsymbol{Q}^m(-1)^{m+1} \right. \\ &\quad \left. - \left\{\sum_{m=1}^{n} {}_nC_m\boldsymbol{Q}^m(-1)^{m+1}\right\}\boldsymbol{S\Delta}\right] \boldsymbol{e}_1 \\ &= \left[(\boldsymbol{I}-\boldsymbol{Q})^n + n(\boldsymbol{I}-\boldsymbol{Q})^{n-1}(-\boldsymbol{QS\Delta}) \right. \\ &\quad \left. - \left\{\sum_{m=2}^{n} {}_nC_m(\boldsymbol{I}-\boldsymbol{Q})^{n-m}\boldsymbol{Q}^m\right\}\boldsymbol{S\Delta}\right] \boldsymbol{e}_1. \end{aligned} \tag{23}$$

Compared with (18) and (23), right-hand side first and second terms are same. If effect of right-hand side third term is very small, $\tilde{\boldsymbol{e}}_{2,n}$ is written as

$$\tilde{\boldsymbol{e}}_{2,n} \approx \boldsymbol{e}_{n+1}. \tag{24}$$

(24) indicates that error which happened after one time learning using $\tilde{Q}_n$ is almost the same as error which happened after n times learning using Q.

If there is no modeling error, $\boldsymbol{\Delta} = \boldsymbol{O}$, (18), and (23) are completely same.

$$\tilde{\boldsymbol{e}}_{2,n} = \boldsymbol{e}_{n+1} \tag{25}$$

$$= (\boldsymbol{I} - \boldsymbol{Q})^n \boldsymbol{e}_1 \tag{26}$$

Therefore proposed RPTC can converge error n times faster than conventional RPTC.

B. Realization of n-time learning filter $\tilde{Q}_n$

From (19), n-time learning filter $\tilde{Q}_n$ is multistage Q filter. Thereby $\tilde{Q}_n$ is expressed as FIR filter.

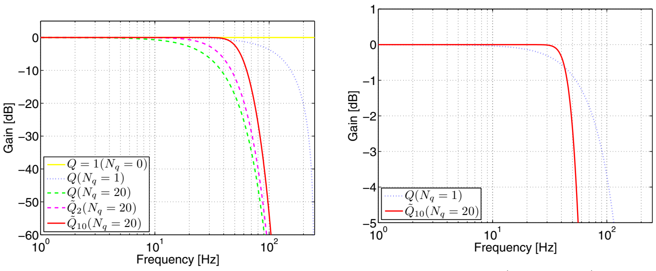
$$\tilde{Q}_n = \sum_{m=-nN_q}^{nN_q} a_m z^m$$

This non proper filter is realized as

$$\tilde{Q}_{nr} = z^{-nN_q}\tilde{Q}_n. \tag{27}$$

In comparison with (3) and (27), each number of sample delays is different. Accordingly, we have to change the number of sample delays of memory in order to realize $\tilde{Q}_n$, like Fig. 9. Here, n is restricted as follows.

$$\begin{aligned} nN_q + 2 &< N_d \\ n &< \frac{N_d - 2}{N_q} \end{aligned} \tag{28}$$



(a) Property of $\tilde{Q}_n$.

(b) Zoom of $Q(N_q = 1)$ and $\tilde{Q}_{10}(N_q = 20)$

Fig. 10. Gain characteristic of $\tilde{Q}_n$.

C. Characteristic of n-time learning filter $\tilde{Q}_n$

In this section, characteristic of $\tilde{Q}_n$ is revealed. From (13), $\boldsymbol{e}_2$ becomes $\boldsymbol{0}$ after first iteration when $\boldsymbol{\Delta} = \boldsymbol{O}$ and $\boldsymbol{Q} = \boldsymbol{I}$. In other words, the fastest convergence is achieved when $\boldsymbol{Q} = \boldsymbol{I}$. Gain characteristic of $\boldsymbol{Q} = \boldsymbol{I}$ is 0 dB in all frequency. However, we have to design Q as a low-pass filter because control system becomes unstable by high frequency signal which arises from noise or mechanical resonance. Therefore, in order to realize stable and fast convergence, Q should be a filter which has wide 0 dB region in low frequency and sharp roll-off property in high frequency.

Fig. 10 shows the gain characteristic of Q and $\tilde{Q}_n$. Here, sampling time is 2 ms. Phase characteristic is zero in all frequency because Q and $\tilde{Q}_n$ are zero-phase low-pass filter. Fig. 10(a) is plots of $Q(N_q = 0, 1, 20)$, and $\tilde{Q}_2, \tilde{Q}_{10}(N_q = 20)$. As N_q increases, high frequency roll-off characteristic improves. However, frequency which can pass signals with 0 dB becomes low. In comparison with $Q(N_q = 20), \tilde{Q}_2(N_q = 20)$, and $\tilde{Q}_{10}(N_q = 20)$, frequency which passes signals with 0 dB becomes high as n increases. From Fig. 10(b), compared with $Q(N_q = 1)$ and $\tilde{Q}_{10}(N_q = 20)$, $\tilde{Q}_{10}$ has better roll-off property and higher frequency whose gain is 0 dB. Hence, $\tilde{Q}_{10}(N_q = 20)$ is more suitable learning filter than $Q(N_q = 1)$.

In conventional learning filter Q, roll-off characteristic is improved by sacrificing low-pass characteristic. In contrast, proposed n-time learning filter $\tilde{Q}_n$ can improve both roll-off and low-pass characteristic by increasing n.

V. Simulation

The effectiveness of n-time learning RPTC is verified by two simulations.

A. Simulation of ideal condition with sinusoidal disturbance

n-time fast convergence is confirmed by simulation against $P_n(s)$ with sinusoidal disturbance. Position error caused by this disturbance is suppressed with RPTC.

Multi-rate FF controller is designed against P_n, so that modeling error is zero. PID controller whose position band frequency is 40Hz is desiged by pole placement method. Moreover, Q and 3-time learning filter $\tilde{Q}_3$ are designed with $N_q = 20$ and $T_r (= 2T_u, T_u = 1$ ms). Input disturbance shown in Fig. 11(a) is suppressed by conventional and proposed 3-time learning RPTC. In order to make clear comparison, iteration cycle T_d is determined as twice of disturbance cycle ($T_d = 0.5$). Simulation results are shown in Fig. 11. Fig. 11(b) shows tracking error. Moreover, Fig. 11(c) and Fig. 11(d) are zoom of Fig. 11(b). Tracking error is suppressed after T_d thanks to compensation of RPTC. Impulse-shaped tracking errors which appeared in Fig. 11(c) are generated by discontinuous state variables of multi-rate FF controller. Discontinuous state variables occurs when compensation of RPTC begins. From Fig. 11(d), except impulse-shaped error, conventional fifth cycle error $\boldsymbol{e}_5$ is the same as proposed third cycle error $\tilde{\boldsymbol{e}}_{3,3}$. Fifth cycle error $\boldsymbol{e}_5$ is achieved by 3 times learning with conventional RPTC. In contrast, third cycle error $\tilde{\boldsymbol{e}}_{3,3}$ is achieved by only 1 time learning with proposed RPTC. Therefore, proposed RPTC can converge tracking error 3 times faster than conventional. Although expected 3-time fast convergence cannot be seen, impulse-shaped error is also converged faster with proposed RPTC.

B. Simulation of ball screw driven stage

Effectiveness of the proposed method is also verified by simulation of ball screw driven stage. Here, P_n with nonlinear friction is determined as simulation plant. Due to nonlinear friction, periodic tracking error causes when periodic position reference is given. In this case, RPTC suppresses periodic tracking error by learning.

Simulation results are shown in Fig. 12. At this time, $\tilde{Q}_5$ is designed as a proposed learning filter. Sinusoidal position reference shown in Fig. 12(a) is given to the stage. Then tracking error occurs as Fig. 12(b). Fig. 12(c) is closeup of third iteration error. From Fig. 12(b) and Fig. 12(c), maximum tracking error of each cycle becomes smaller with 5-time learning RPTC. Therefore, $\tilde{Q}_5$ can learn tracking error more effectively than conventional Q. However, 5-time fast convergence is not achieved because disturbance caused by nonlinear friction is not perfectly periodic.

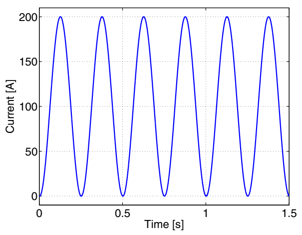
VI. Experiment

The proposed method was implemented to the experimental ball screw driven system. Controllers and learning filters were the same as simulation of ball screw driven stage. Multi-rate FF controller and PID controller are designed as same as the simulation. Q and $\tilde{Q}_5$ are designed with sampling time 2 ms and $N_q = 20$.

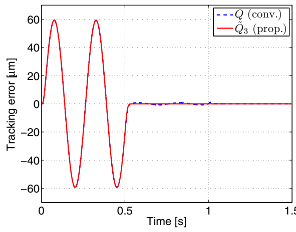
Experimental results are shown in Fig. 13. Fig. 13(a) is target trajectory the same as simulation. Fig. 13(b) and Fig. 13(c) are plots of tracking error and zoom of third cycle tracking error, respectively. From Fig. 13(b) and Fig. 13(c), maximum tracking error of proposed RPTC is smaller than conventional one, likewise simulation. Hence, fast convergence was verified from simulation and experimental results.

VII. Conclusion

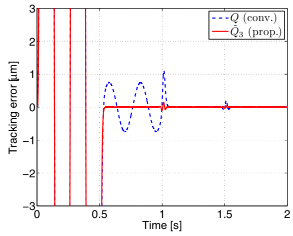
In this paper, we proposed n-time learning RPTC which can converge tracking error n times faster than conventional RPTC. n-time learning filter used in proposed RPTC has better



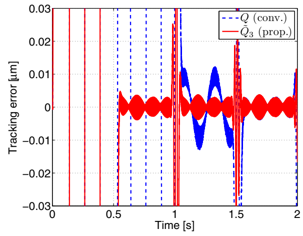
(a) Input disturbance.



(b) Error.

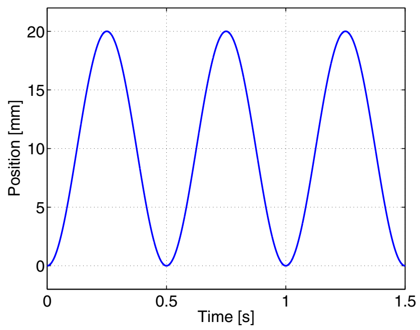


(c) Error (zoom).

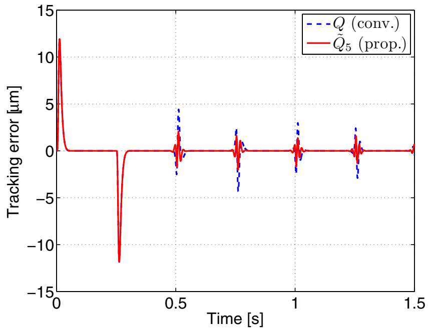


(d) Error (further zoom).

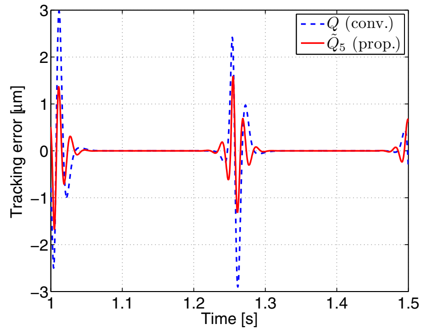
Fig. 11. Simulation results (ideal condition).



(a) Target trajectory.

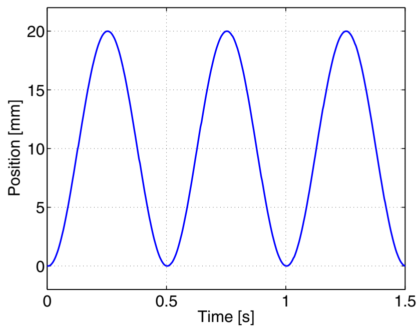


(b) Error.

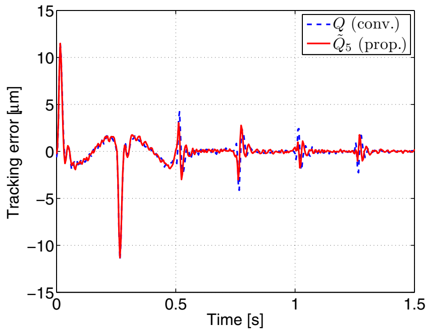


(c) Error (zoom of third iteration).

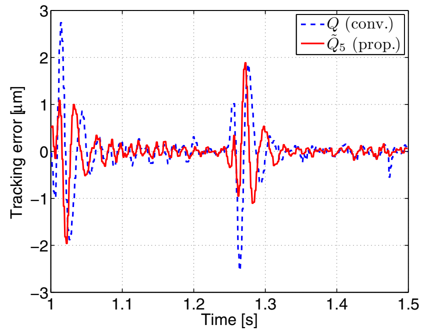
Fig. 12. Simulation results (with nonlinear friction).



(a) Target trajectory.



(b) Error.



(c) Error (zoom of third iteration).

Fig. 13. Experimental results.

low-pass and sharp roll-off characteristic than conventional learning filter. n-time fast convergence is verified with ideal simulation. Finally, the effectiveness of proposed method is shown by simulation and experiment of ball screw driven stage.

References

[1] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A survey of iterative learning control", Control Systems Magazine, IEEE, 26(3):96-114, 2006.

[2] K. K. Chew and M. Tomizuka, "Digital control of repetitive errors in disk drive systems", IEEE Control Syst. Mag., vol. 10, no. 1, pp. 16-20, 1990.

[3] S. Hara, Y. Yamamoto, T. Omata, and M. Nakano, "Repetitive control system–A new-type servo system", IEEE Trans. Autom. Control, vol. 33, no. 7, pp. 659-668, 1988.

[4] H. Fujimoto, Y. Hori, A. Kawamura, "Perfect Tracking Control based on Multirate Feedforward Control with Generalized Sampling Periods", IEEE Trans. Industrial Electronics , vol. 48, No. 3, pp. 636-644, 2001.

[5] H. Fujimoto, "RRO Compensation of Hard Disk Drives with Multirate Repetitive Perfect Tracking Control", IEEE Trans. Industrial Electronics, vol. 56, no. 10, pp. 3825-3831, 2009.

[6] T. Nakai, H. Fujimoto, "Harmonic Current Suppression for PMSM by Repetitive Perfect Tracking Control", T. IEE Japan, vol.128-D, no. 9, pp. 1083-1089, 2008 (in Japanese)

[7] H. Asaumi, H. Fujimoto, "Proposal on Presice Positioning Control of Ball Screw Stage Based on ILC with PTC", IIC-07-128, pp.71-76, 2007 (in Japanese).

[8] H. Endo, H. Fujimoto, "Short-Span Seeking Control of Hard Disk Drive Based on Integral Type Learning PTC", in Proc. The 11th IEEE International Workshop on Advanced Motion Control Proceedings, Niigata, pp.554-559, 2010.

[9] K. L. Moore, "Iterative Learingn control for Deterministic Systems", Sringer-Verlag, pp. 23-29