강화학습에 의한 선형동기 모터의 고정밀 제어

High-Accuracy Motion Control of Linear Synchronous Motor Using Reinforcement Learning

정승현 [1], 박정일 [2,✉]
Seong Hyen Jeong[1] and Jung Il Park[2,✉]

1 한국섬유기계연구소 (Korea Textile Machinery Institute)
2 영남대학교 전자공학과 (Department of Electronic Engineering, Yeungnam Univ.)
✉ Corresponding author: jipark@yu.ac.kr, Tel: 053-810-2498

Manuscript received: 2011.1.25 / Accepted: 2011.9.5

A PID-feedforward controller and Robust Internal-loop Compensator (RIC) based on reinforcement learning using random variable sequences are provided to auto-tune parameters for each controller in the high-precision position control of PMLSM (Permanent Magnet Linear Synchronous Motor). Experiments prove the well-tuned controller could be reduced up to one-fifth level of tracking errors before learning by reinforcement learning. The RIC compared to the PID-feedforward controller showed approximately twice the performance in reducing tracking error and disturbance rejection.

Key Words: Robust Internal-loop Compensator (견실내부루프보상기), Disturbance Observer (외란 관측기), Reinforcement Learning (강화학습), Linear Synchronous Motor (선형동기모터)

1. Introduction

Linear motors are becoming more important to form high speed and high precision linear motion systems. They are used instead of a rotary motor that is uses a rotary actuator and a lead screw used for the mechanical transmission unit to convert rotary motion to linear motion in semiconductor equipment, machine tools, automatic test equipment and so on. A conventional rotary motion system using a lead screw degrades the response of the dynamic characteristic by time delay backlash, deadzone, screw friction, PMLSM. Conversely, it has a simple structure, high thrust power from the direct linear motion, long life, low maintenance cost, and low thermal loss. This enables it to obtain high speed, high precision motion control. PMLSM, however, has defects sensitive to disturbance, so it is easily affected by friction and torque ripple. PMLSM should be treated with an efficient control algorithm to obtain high speed, high precision position control performance. The objective of this paper is to design a precise position controller by using reinforcement learning to adaptively tune parameters of the controller for a PMLSM motion control system.

PMLSM can be modeled by two nonlinear characteristics of friction and ripple force related to velocity and position of the moving coil, as shown in Fig. 1. Cogging and magnetic reluctance force can cause an unwanted ripple force against the thrust power related to the moving coil position. The ripple force is usually generated in iron-core type motors, where there is a lower weight of the moving core or low speed motor movement. The ripple force is difficult to predict but can be

reproducible. Thus, it can be rejected by controlling the force input with an online learning controller. Research has been published on overcoming the ripple force with a learning feedforward controller based on a neural network for PMLSM motion control.[1] Another nonlinear characteristic is friction that has an effect on actuation performance. It is inevitably generated by contact, and can be an obstacle to precise position control. It can usually be represented as static friction and Coulomb friction, as well as viscosity and Stribeck effects. Especially, viscous friction and Coulomb friction show nonlinear characteristics related to velocity. Research, such as compensation of the entire disturbance estimation of PMLSM to remove the nonlinear characteristics efficiently,[2,3] high precision position control algorithms, using a robust internal loop compensator, were developed.[4-7] In this paper, we propose an algorithm that can adapt parameters of a controller automatically using reinforcement learning with error.[8] A robust motion control structure to improve performance of the whole closed-loop system should be designed in advance to achieve this. Then, the adaptive learning to tune optimal parameters of the RIC and PID-feedforward controller with reinforcement learning based on random variable sequences is performed.

Section 2 introduces the reinforcement learning algorithm and explains the PID-feedforward controller and RIC used in this paper. Section 3 conducts the experiments for several motion controllers and evaluates the results. Section 4 draws conclusions.

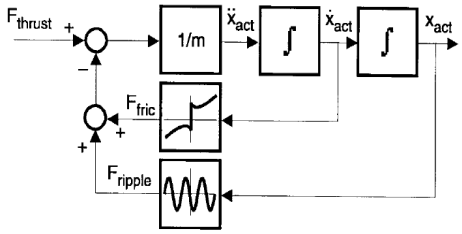


Fig. 1 PMLSM model with friction and ripple force

2. Design of controller with reinforcement learning

2.1 Reinforcement learning based on random variable sequence

This paper introduces a reinforcement learning algorithm to adaptively tune parameters of the controller as in.[8] Reinforcement learning adapts the strength of the synapse with eq.(1). Here η is the learning rate, f the activation function, θ is 0.5, as the bias, n is a discrete random process between 0 and 1.

$$m(t+1) = m(t) + \eta r(t) f(n(t) - \theta) \tag{1}$$

where, $r(t)$ is the reinforcement signal as eq.(2).

$$r(t) = u(J(t) - J(t-1))$$
$$Here, u(x) = \begin{cases} 1, & if \quad x \le 0 \\ 0, & otherwise \end{cases} \tag{2}$$

$J(t)$ is the performance index; it represents the entire sum of errors and the squares of the error derivative, as shown in eq.(3), in the discrete time domain; a lower value means better performance. The synapse learning can be done if the performance index is lower; otherwise, recover the previous one. The activation function is a bipolar step function, as in eq.(4).

$$J(t) = \sum_{t=0}^{\infty} \left[e^2(t) + e'^2(t) \right] \tag{3}$$

$$f(x) = \begin{cases} +1, if \quad x \ge 0 \\ -1, otherwise \end{cases} \tag{4}$$

2.2 Design of PID controller based on reinforcement learning

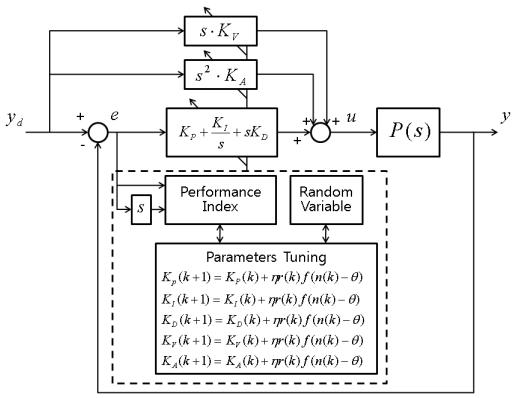


Fig. 2 PID-feedforward controller structure based on reinforcement learning

Fig. 2 shows the control structure of PMLSM with the PID-feedforward controller based on reinforcement learning using random variable sequences. It uses a sinusoidal position profile and several parameters, such as PID gains (K_P, K_I, K_D), velocity gain (K_V), acceleration feedforward gain (K_A), are auto-tuned by reinforcement learning using random variable sequences.

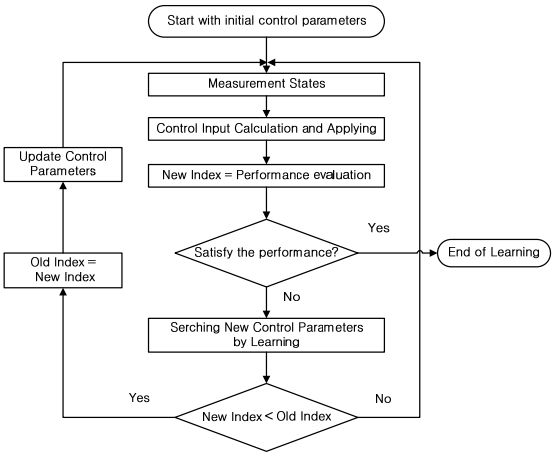


Fig. 3 Flow of control parameters tuning using reinforcement learning with random variable sequence

Fig. 3 shows the flow of reinforcement learning, parameters of each controller learn to minimize the performance index that evaluates learning results.

2.3 Design of RIC with reinforcement learning

The controller design using disturbance observer (DOB), adaptive robust control[2,3] are research to compensate disturbances. Generally, the DOB structure consists of a low-pass filter $Q(s)$ and an inverse model of the plant for external disturbance compensation. The RIC structure is equivalent to the DOB one. Fig. 4 is the relationship diagram that shows the equivalence between the RIC and DOB structure. y is plant output, u_r is the reference input, $P_m(s)$ is the reference plant model, y_r is the reference output, d_{ex} is external disturbance, ξ is the sensing noise.

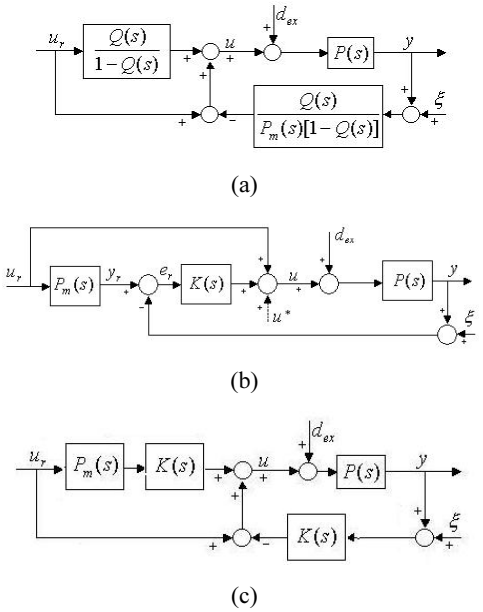


Fig. 4 Equivalent block diagram between RIC and DOB structure : (a) DOB structure, (b) Normal RIC structure, (c) Modified RIC structure

In Fig. 4, if the relation of eq.(5) is satisfied, the DOB structure of Fig. 4(a) and RIC structure of Fig. 4(c) are equivalent. Fig. 4(b), the normal structure of RIC, is a special one of the DOB structures with reference model $P_m(s)$.

$$Q(s) = \frac{P_m(s)K(s)}{1 + P_m(s)K(s)}, K(s) = \frac{Q(s)}{P_m(s)(1 - Q(s))} \quad (5)$$

Control input can be calculated as in eq.(6) from Fig. 4(b).

$$u = u_r + K(s)e_r + u^* \quad (6)$$

Here u^* is the control input term to compensate for the nonlinearity and uncertainty. $K(s)$ is the internal loop feedback compensator. The PMLSM plant model is set as in eq.(7).

$$P_m(s) = \frac{1}{J_m s^2 + B_m s} \quad (7)$$

where, J_m and B_m are the nominal value of the mass and viscous friction of the linear motor, respectively.

If $Q(s)$ is a first order low pass filter, as $1/(\tau s+1)$, $K(s)$ of eq.(5) will be same as eq.(8).

$$K(s)=\frac{1}{\tau}(J_m s+B_m) \tag{8}$$

In eq.(8), if $1/\tau=D$, the internal loop $K(s)$ is $(J_m s+B_m)D$, which is the PD controller, then $Q(s)=D/(s+D)$ by eq.(5). It becomes the same structure as the first-order low pass filter proposed by Ohnishi.[9]

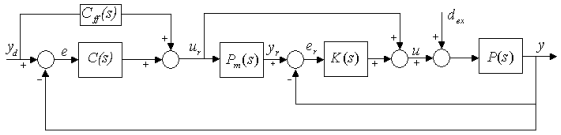


Fig. 5 RIC structure with outer feedback loop and feedforward loop

RIC, as shown in Fig. 5, can be structured by adding the outer loop controller to the Fig. 4(c) structure. $C_{ff}(s)$ is the feed-forward controller, $C(s)$ is the feedback controller and y_d is the desired output. The outer loop controller $C(s)$ and $C_{ff}(s)$ are shown as in eq.(9).

$$\begin{aligned} C_{ff}(s) &= J_m s^2 + B_m s, \\ C(s) &= (K_1 s + K_2)\Lambda, \\ K(s) &= (J_m s + B_m)D \end{aligned} \tag{9}$$

Here Λ, D is constant

By Fig. 5, the control input becomes eq.(10) and the model reference input is the same as eq.(11).

$$u = C_{ff}(s)y_d + C(s)e + K(s)e_r \tag{10}$$

$$u_r = y_d C_{ff}(s) + eC(s) \tag{11}$$

If the well-tuned inter-loop works as $P_m(s)$, then the entire feedback closed loop characteristic equation is represented as eq.(12), using eq.(7) and eq.(9).

$$\begin{aligned} &1 + C(s)P_m(s) = \\ &1 + (K_1 s + K_2)\Lambda\left(\frac{1}{J_m s^2 + B_m s}\right) = 0 \end{aligned} \tag{12}$$

It can be rearranged as eq.(13).

$$s^2 + \frac{B_m + \Lambda K_1}{J_m}s + \frac{\Lambda K_2}{J_m} = 0 \tag{13}$$

The feedback closed loop characteristic equation will be a second-order equation, as shown in eq.(14).

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \tag{14}$$

where, ζ is the damping ratio and ω_n is the natural frequency.

If the desired poles are given by eq.(14), then the internal loop feedback controller gains K_1, K_2 can be calculated as eq.(15) by comparing coefficients of eq.(13) and (14).

$$K_1 = \frac{2\varsigma\omega_n J_m - B_m}{\Lambda}, \quad K_2 = \frac{J_m\omega_n^2}{\Lambda} \tag{15}$$

The structure of Fig. 6 shows that each parameter is automatically adjusted by tuning each parameter, such as Λ, ζ, ω_n and D using reinforcement learning. The external loop gain K_1, K_2 is calculated by three factors Λ, ζ, ω_n. At this time, the flow of reinforcement learning is performed, as in Fig. 3, except for the different learning parameters.

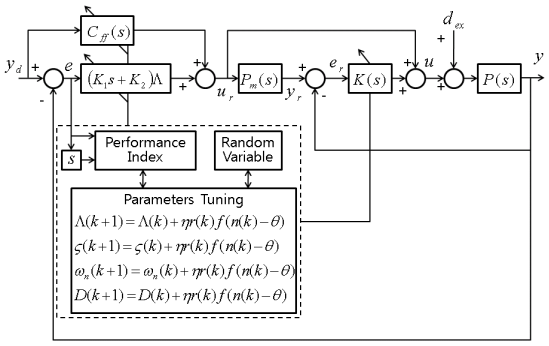


Fig. 6 Structure of RIC using reinforcement learning

3. Experiments of Motion Control

The entire control system shown in Fig. 7 consists of PMLSM, servo driver, DSP controller and XDS 510 USB JTAG. The servo driver can make three-phase torque control signals and acquire the motion states of position and velocity through the linear scale encoder. The control software compiled is downloaded to flash memory of the target control board through JTAG XDS510. Position, velocity, acceleration states are saved to external memory of the control board and transferred to the PC through the RS232C port for evaluation.

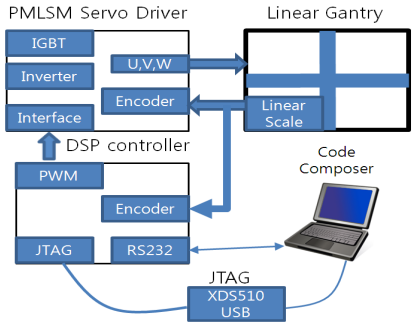


Fig. 7 Structure of PMLSM motion control system

3.1 Experiments

Experiments of two controllers, the PID-feedforward controller and the RIC based on reinforcement learning, were performed with the motion control system of PMLSM.

3.1.1 Generation of motion profile

The sine wave position profile, shown in Fig. 8 is used, to prevent the discontinuous jerk that can greatly influence motion error. It is 5cm circular motion during 0.8 sec including stopping time. Reinforcement learning is performed comparing the sum of squared tracking errors of the current sample time with one of the previous sample times. One learning cycle takes 0.8 sec.

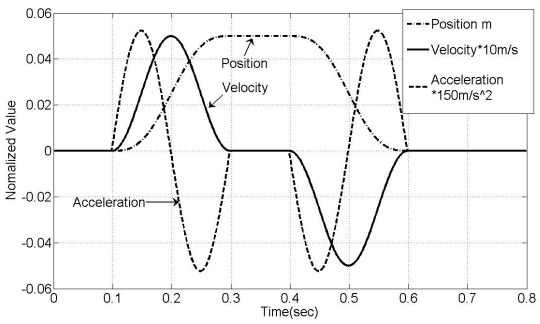
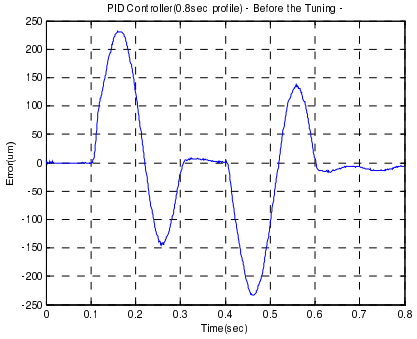


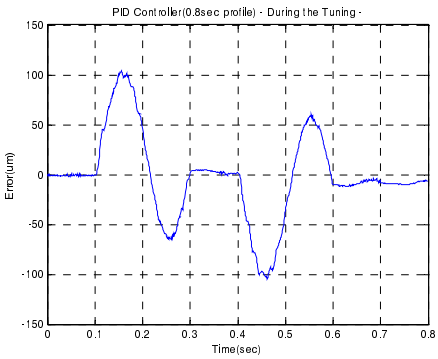
Fig. 8 Profiles based on sine wave

3.1.2 Experiments using PID feedforward controller based on reinforcement learning

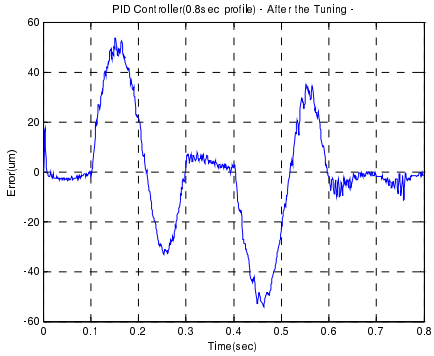
Fig. 9(a) shows the control results before learning using the PID-feedforward controller; maximum tracking error is about 240μm. The form of tracking error keeps up a similar acceleration profile shape shown in Fig. 8. Fig. 9(b) is the result of the PID-feedforward controller after a hundred times self-reinforcement learning. The maximum error is about 100μm; this is largely enhanced. Fig. 9(c) is the experimental results after eight hundred learning times. The maximum tracking error greatly decreased to 50μm. This is one-fifth the level of tracking error before learning. This shows the performance of the PID controller is improved more than fivefold after learning.



(a)



(b)



(c)

Fig. 9 Tracking error using PID-feedforward controller: (a) before learning, (b) 100 times learning, (c) 800 times learning

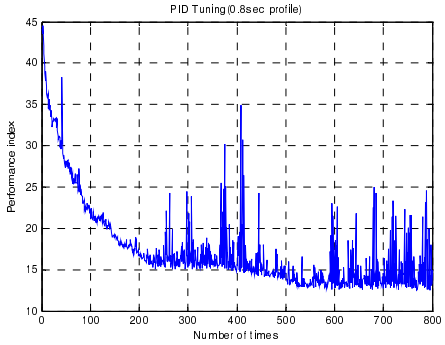
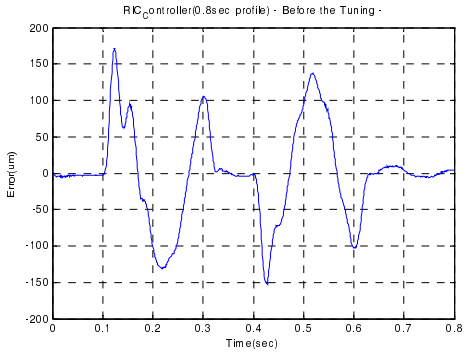


Fig. 10 Transition of the performance index during PID-feedforward controller's learning

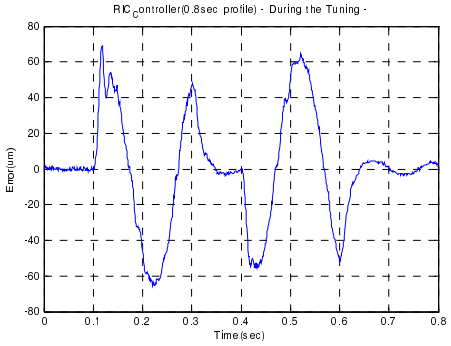
Fig. 10 shows the transition of the performance index during the PID-feedforward controller's learning. It shows a rapid decrease until about 200 learning times, but no further decrease after 600 learning times. The performance index decrease from 45 to 13, shows about a 420% performance improvement.

3.1.3 Experiments using RIC with reinforcement learning.

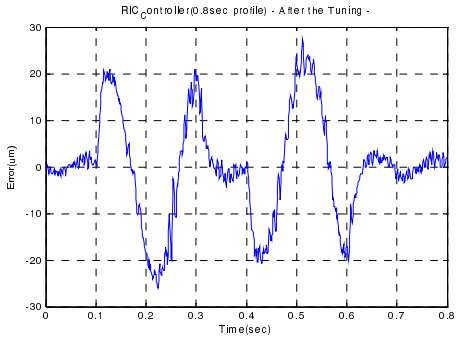
Fig. 11(a) shows the results of using RIC before reinforcement learning. The maximum tracking error is 170μm. It performs better compared to the maximum error, 250μm, of the PID-feedforward controller, shown in Fig. 9(a).



(a)



(b)



(c)

Fig. 11 Tracking error of RIC based pole placement control: (a) before learning, (b) 100 learning times, (c) 800 learning times

Fig. 11(b) shows the RIC's results after 100 learning times. Performance is largely enhanced due to decreasing tracking error to the 65μm level. Fig. 11(c), the experimental results after 800 learning times, shows that maximum tracking error is 25μm. This shows that RIC's

performance is greatly enhanced and less than the maximum tracking error 50μm using the PID-feedforward control. The shape of the steady state error during the regulation time district of 0.3~0.4 second differs from the shape of the PID-feedforward controller's error.

Fig. 12 shows the transition of the performance index. It shows a rapid and soft decrease until 300 learning times, but there is no further decrease after this. That is, there is no more learning progress. The performance index decrease from 38 to 7 shows about 542% improvement from learning.

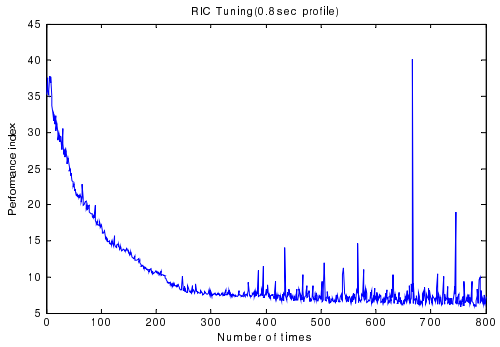


Fig. 12 Transition of performance index for RIC

3.1.4 Constant low speed motion experiment

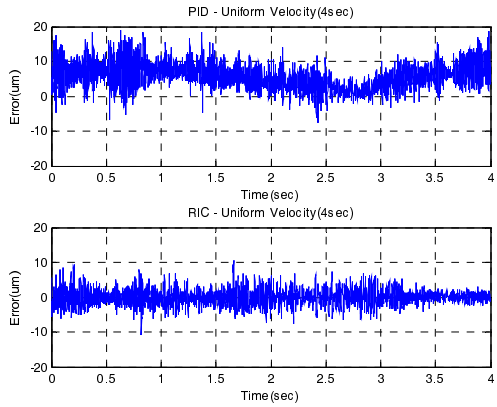


Fig. 13 Tracking error of PID and RIC at constant velocity

Fig. 13 depicts the results of constant low speed motion during four seconds between 30mm district with the PID-feedforward controller and the RIC after learning. The offset error appears in the case of the PID-feedforward controller, but there is no offset error of the RIC. RMS tracking error greatly decreased to 2.2 μ*m* from 6.8 μ*m*. The The RIC showed approximately threefold performance in reducing tracking error at constant low speed compared to the PID-feedforward controller.

3.1.5 Experiment of load insertion using a spring

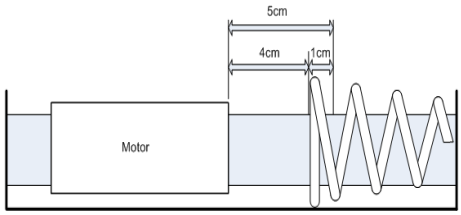


Fig. 14 Mechanism to apply several loads using a spring

Fig. 14 shows the apparatus to apply positional load with a spring system. The spring w is equipped at the end of PMLSM's linear motion system to insert the accumulated power.

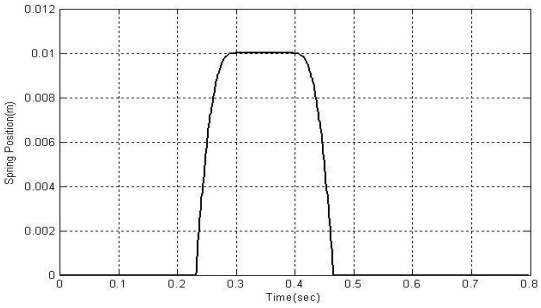
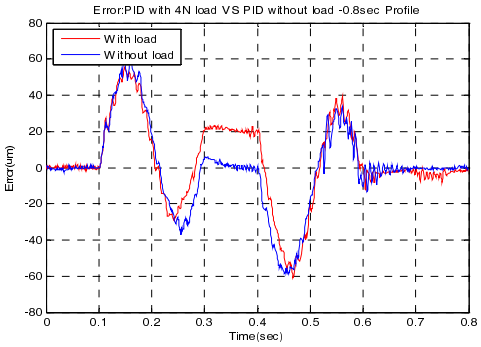


Fig. 15 Insertion of load using spring

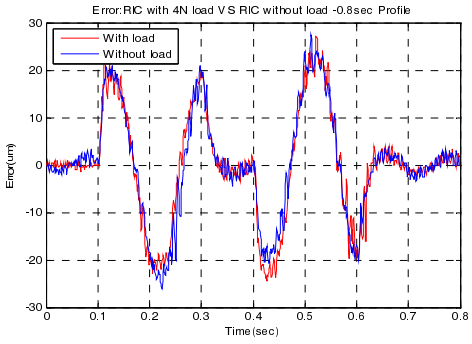
Fig. 15 is the ideal graph that shows change in the spring position related to the position profile of PMLSM. The spring constant is 0.1N/mm. 1N load for one spring can be inserted during 0.1sec from 0.3sec to 0.4sec, when it moves 10mm of the maximum position change.

Fig. 16 compares each controller's tracking error with or without 4N load using four springs. The PID-feedforward controller shows that the external disturbance caused by the load cannot be eliminated

effectively due to the tracking error increase to 20μm when a 4N load is applied during the position regulation block from 0.3sec to 0.4sec. However, in case of RIC under the same load, the external disturbance caused by the load is efficiently eliminated, as there is not much difference shown in the tracking error during the regulation block from 0.3sec to 0.4sec. Therefore, it can be seen that the well-tuned RIC with reinforcement learning performs better than the well-tuned PID-feedforward controller does, when an external disturbance is applied. Thus, the RIC structure itself forms a robust internal loop and removes load disturbance efficiently.



(a)



(b)

Fig. 16 Tracking error comparison with/without 4N load: (a) PID-feedforward controller, (b) RIC

3.2 Evaluation of experiments.

Table 1 shows the RMS value of tracking error before and after learning under the conditions of no load and 4N external load. As the RMS value of the PID-feedforward controller and RIC are remarkably lowered to 11.76μm and 6.43μm respectively, after reinforcement learning, it can be shown that reinforcement learning based on random variable sequences can tune parameters of the controller to improve performance adaptively. When 4N load disturbance is applied in the specified block, reinforcement learning experimental results of the PID-feedforward controller shows 13.24μm. This has a 1.48μm difference to the 11.76μm of no load learning. Thus, the performance of the PID-feedforward controller can be influenced by disturbances. In contrast, the results of the RIC after reinforcement learning were 6.45μm, very similar to the no load results of 6.43μm before learning. This implies that the RIC is rarely influenced by disturbances.

Table 1 Comparison of each controller's RMS tracking error(unit:um)

Learning \ Controller		PID-feedforward controller	RIC
No load	before learning	50.52	35.21
	after learning	11.76	6.43
Insertion of 4N load	before learning	55.12	36.53
	after learning	13.24	6.45

4. Conclusion

In this paper, an efficient parameter tuning method for the PID-feedforward controller and RIC using reinforcement learning based on random variable sequences was proposed for high precision position motion control of PMLSM. The performance of each controller was evaluated by experiments with a linear motion system equipped with a high resolution linear scaler. A control board, which included 32bits 150MIPS DSP and the high density FPGA, was developed to conduct experiments. Tracking error was reduced by tuning parameters of each controller with reinforcement learning. Comparisons for each controller were also performed. RIC performs better than does the PID-feedforward controller from the viewpoint of reducing tracking error and disturbance rejection.

Acknowledgement

This research was supported by the Yeungnam University research grants in 2010.

References

1. Otten, G., De Vries, T. J. A., Van Amerongen, J., Rankers, A. M. and Gaal, E. W., “Linear motor motion control using a learning feedforward controller,” IEEE/ASME Trans. on Mechatronics, Vol. 2, No. 3, pp. 179-187, 1997.
2. Xu, L. and Yao, B., “Adaptive robust precision motion control of linear motors with ripple force compensations,” Proceeding of IEEE International Conference on Control Applications, pp. 373-378, 2000.
3. Xu, L. and Yao, B., “Adaptive Robust Precision Motion Control of Linear Motors with Negligible Electrical Dynamics: Theory and Experiments,” IEEE/ASME Trans. on Mechatronics, Vol. 6, No. 4, pp. 444-452, 2001.
4. Kim, B. K. and Chung, W. K., “Unified Analysis and Design of Robust Disturbance Attenuation Algorithms using Inherent Structural Equivalence,” Proceeding of American Control Conference, Vol. 5, pp. 4046-4051, 2001.
5. Kim, B. K. and Chung, W. K., “Performance Tuning of Robust Motion Controllers for High-Accuracy Positioning Systems,” IEEE/ASME Trans. on Mechatronics, Vol. 7, No. 4, pp. 500-514, 2002.
6. Kim, B. K. and Chung, W. K., “Advanced Disturbance Observer Design for Mechanical Positioning Systems,” IEEE Trans. on Industrial Electronics, Vol. 50, No. 6, pp. 1207-1216, 2003.
7. Park, J. I., Jeong, S. H. and Sung, J. Y., “High-accuracy Motion Control of Linear Synchronous Motor,” Journal of the Korean Society for Precision Engineering, Vol. 22, No. 6, pp. 76-82, 2005.
8. Jeong, S. H., Han, C. W, Park, J. I. and Kwon, S. H., “A Study on Learning Scheme of Self-Learning Rule-based Fuzzy Controller using Random Variable Sequence,” Proceeding of American Control Conference, pp. 1862-1863, 1998.
9. Godler, I., Honda, H. and Ohnishi, K., “Design guidelines for disturbance observer's filter in discrete time,” Proceeding of 7th International Workshop on Advanced Motion Control, pp. 390-395, 2002.