

로봇 매니플레이터의 제어를 위한
강인한 적응 제어기의 설계

류 기 응^{*}, 홍 석 교
아주대학교 제어공학과

ROBUST ADAPTIVE CONTROLLER FOR ROBOT MANIPULATOR CONTROL

Kgi Wung Ryoo , Suk kyo Hong
Dept.of C.E. , Ajou University

Abstract

In this paper, the robust adaptive controller for the accurate position control of the robot manipulator is described. The proposed controller has feedforward and feedback terms. To reduce the computation time, the inertia matrix is simplified by diagonal matrix and the centrifugal and Coriolis term and gravity term are assumed to be zero. The auxiliary signal is added in the controller. This signal improves convergence time of error. The simulation results show that the proposed method is applied to the real time control.

1. 서론

로봇 매니플레이터의 제어는 공장 자동화를 위한 점에서 상당히 중요한 사항이다. 일반적으로 로봇 매니플레이터의 제어에는 위치 제어와 힘 제어로 크게 두가지가 있다. 로봇의 위치를 제어하기 위해서는 로봇 매니플레이터의 동적 방정식을 이용해야한다. 그러나 로봇의 동적 방정식은 관절들간에 서로 결합된 비선형-시변함수(Joints Coupled Nonlinear-Time Varying Function)이기 때문에 로봇 위치를 정확히 제어 하고자 할때에는 어려움이 따른다.

70년대 후반부터 로봇 매니플레이터의 제어에 관한 연구가 활발히 진행되어 왔다.[1]-[10] 로봇 매니플레이터의 제어에는 크게 비적응 제어(Nonadaptive Control)방법과 적응 제어(Adaptive Method)로 나눌수 있다. 비적응 제어에 가장 대표적인 것으로 Computed Torque Method를 들수 있다. 이방법은 로봇의 파라미터 값을 정확히 알아야 하고 부하가 변할 때마다 궤환 이득값이 부하의 변동에 대처하지 못한다는 어려움이 있다. 이와같은 어려움을 극복하기 위해서 부하의 변동에 대처하여 이득값을 조절할 수 있고 로봇의 동적 모델의 파라미터값을 정확히 알지 않아도 되는 적응 제어 방

법이 대두되었다. 적응제어 방법에는 MRAC(Model Reference Adaptive Control)방법, STAC(Self Tuning Adaptive Control)방식이 있다.[1][7] MRAC 방법은 Dubowsky와 Desforges에 의해서 처음으로 제시되었다.[1] 이들은 기존 모델의 상태 방정식과 플랜트의 상태 방정식간 오차의 이차함수를 설정하여서 이 오차 함수를 최소화 시키는 이득의 값을 찾고자 하였다. Tomizuka는 MRAC방법을 로봇 동적 방정식의 비선형을 보상하고 관절간을 디커플 하는 데 사용하였고 독립 추종과 레귤레이션 알고리즘을 이용한 이산화 시간 MRAC방법을 제시하였으며 Lee와 Chung은 로봇의 비선형 동적 방정식을 공칭궤적에 대해서 선형화시키고 선형화된 모델을 RLS에 의해서 파라미터값을 결정하는 적응 파라미터 인식 알고리즘을 이용하여 로봇을 제어하였다. LIM과 ESLAMI는 Lyapunov방정식을 이용한 MRAC방법을 제시하였다.[4][5][6]

본 논문에서는 H.Seraji의 방법에[3] 강건한 제어를 첨가한 강건한 적응 제어 방식을 연구하였다. II장에서는 로봇의 비선형 동적 방정식을 선형화 시키었고 III장에서는 강건한 적응제어기를 제시하였으며 IV장에서는 제시된 제어기를 스카라형 로봇에 적용시켜보았다.

1.1. 로봇 동적 방정식

로봇 매니플레이터가 n개의 관절과 n개의 링크를 가진다면 Euler-Lagrange 방정식을 써서 로봇의 동적 방정식을 쓰면 식(1)과 같이 표현되고

$$D(\Theta)\ddot{\Theta}(t) + H(\Theta, \dot{\Theta}) + G(\Theta) = \tau(t) \quad (1)$$

식(1)은 관절간에 강하게 커플드된 비선형 미분방정

식이다. 여기서 $\Theta(t), \dot{\Theta}(t), \ddot{\Theta}(t)$ 는 로보트의 위치, 속도, 가속도를 나타내고 $\tau(t)$ 는 로보트에 가해 주는 토오르 입력이다. $D(\Theta)$ 는 $n \times n$ 인 대칭 정치 (positive definite) 관성 행렬이고 $H(\Theta, \dot{\Theta})$ 는 $n \times 1$ 인 Coriolis와 원심력 행렬이고 $G(\Theta)$ 는 $n \times 1$ 인 중력 행렬이다.

식(1)을 $t_i \leq \Delta t_i \leq (t_i + \Delta t_i)$ 인 기간에서 테일러 전개하여서 근사화시켜 선형화시키면

$$A\ddot{\Theta}(t) + B\dot{\Theta}(t) + C\Theta(t) = T(t) \quad (2)$$

인 결합된 선형 미분 방정식이 된다.

여기서

$$A = [D]_p, \quad B = \left[\frac{\partial H}{\partial \dot{\Theta}} \right], \quad C = \left[\frac{\partial (H+G)}{\partial \Theta} \right] \quad (3)$$

이다.

식(2)에 원하는 각 관절의 $\Theta_d(t), \dot{\Theta}_d(t), \ddot{\Theta}_d(t)$ 를 입력시켜 로보트가 원하는 위치에 가도록 하는 제어기를 식(4)와 같이 구성할 수 있다.

$$T(t) = A\ddot{\Theta}_d(t) + B\dot{\Theta}_d(t) + C\Theta_d(t) \quad (4)$$

또한 시스템의 안정을 기하기 위한 궤환 제어이득 값은 식(5)와 같다.

$$K(t) = K_p(t) \cdot e(t) + K_v \cdot \dot{e}(t) \quad (5)$$

여기서 K_p 와 K_v 는 $n \times n$ 위치 이득 행렬과 속도 이득 행렬이다. 식(5)에 제어 법칙을 적용하면

$$T(t) = K_p(t) \cdot e(t) + K_v(t) \cdot \dot{e}(t) \quad (6)$$

이다. 여기서 $e(t) = \Theta_d(t) - \Theta(t)$ 인 $n \times 1$ 위치 오차 벡터이고 $\dot{e}(t) = \dot{\Theta}_d(t) - \dot{\Theta}(t)$ 인 $n \times 1$ 속도 오차 벡터이다. 식(4)와 식(6)을 더해서 로보트 매니플레이터의 토오르 입력을 식(7)과 같이 만들 수 있다.

$$\begin{aligned} T(t) &= T_{ff}(t) + T_{fb}(t) \\ &= A\ddot{\Theta}_d(t) + B\dot{\Theta}_d(t) + C\Theta_d(t) \\ &\quad + K_p \cdot e(t) + K_v(t) \cdot \dot{e}(t) \end{aligned} \quad (7)$$

식(2)와 식(7)은 토오르 $T(t)$ 식이 같으므로

$$\begin{aligned} &A\ddot{\Theta}(t) + B\dot{\Theta}(t) + C\Theta(t) \\ &= K_p \cdot e(t) + K_v \cdot \dot{e}(t) + A\ddot{\Theta}_d(t) + B\dot{\Theta}_d(t) \\ &\quad + C\Theta_d(t) \end{aligned}$$

이 되고

$$\begin{aligned} &A(\ddot{\Theta}_d(t) - \ddot{\Theta}(t)) + B(\dot{\Theta}_d(t) - \dot{\Theta}(t)) + C(\Theta_d(t) - \Theta(t)) \\ &\quad + K_p \cdot e(t) + K_v \cdot \dot{e}(t) = 0 \end{aligned} \quad (8)$$

식(8)을 오차의 방정식으로 만들면 다음과 같다.

$$\begin{aligned} &A \cdot \ddot{e}(t) + (B + K_v) \cdot \dot{e}(t) + (C + K_p) \cdot e(t) = 0 \\ &\ddot{e}(t) + A^{-1}(B + K_v) \cdot \dot{e}(t) + A^{-1}(C + K_p) \cdot e(t) = 0 \end{aligned} \quad (9)$$

식(9)는 궤환 이득 K_p 와 K_v 로 이루어진 궤적오차 $e(t)$ 의 미분 방정식을 보여주고 있다. 식(9)의 해를 구해보면 식(10)과 같다.

$$e(t) = \left\{ \sum_{i=1}^{2n} L_i \exp(\eta_i t) \right\} e(0) + \left\{ \sum_{i=1}^{2n} \bar{L}_i \exp(\eta_i t) \right\} \dot{e}(0) \quad (10)$$

이식에서 $e(0)$ 와 $\dot{e}(0)$ 는 초기의 위치와 속도의 오차이다. L_i 와 \bar{L}_i 는 구해진 $n \times n$ 상수 행렬이고 η_i 는

식(11)으로 표현된 오차 방정식의 특성방정식의

$$\Delta(S) = |I_n S^2 + A^{-1}(B + K_v)S + A^{-1}(C + K_p)| = 0 \quad (11)$$

해 즉, 오차 방정식의 고유치이다.

식(10)에서 보는 바와 같이 식(8)의 궤환 이득값 K_v 와 K_p 를 조정함으로써 오차방정식의 고유값을 S평면상의 좌반부로 보낼 수 있다.

지금까지 구한 토오르식을 전체 제어 법칙에 적용하면

$$\begin{aligned} \tau(t) &= \tau^*(t) + K_p \cdot e(t) + K_v \cdot \dot{e}(t) \\ &\quad + C\Theta_d(t) + B\dot{\Theta}_d(t) + A\ddot{\Theta}_d(t) \end{aligned} \quad (12)$$

지금까지 알아본 제어시스템을 살펴보면 전방입력 제어기는 로보트가 원하는 입력을 따라가도록 하는 제어기이고 궤환제어기는 시스템이 안정하도록 하기 위해 오차를 줄이는 항이다. 그림 1에 식(12)의 제어기를 구성하였다.

I I I. 강인한 적응제어 시스템

3장에서 설계된 제어기는 동작점 부근에서는 아주 효과적이다. 로보트의 전체 운동에서는 동작점 P와 선형화된 행렬 A, B, C는 시간이 변함에 따라 연속적으로 변하는 값이다. 그러므로 전방입력 제어기와 궤환 제어기의 이득을 시간에 대해서 적응 시켜야 한다. 또한 로보트 제어를 더욱 안정도를 향상시키기 위해서 보조 시그널 $F(t)$ 을 첨가한다.

$$\begin{aligned} T(t) &= F(t) + K_p(t) \cdot E(t) + K_v(t) \cdot \dot{E}(t) \\ &\quad + C \cdot \Theta_d(t) + B \cdot \dot{\Theta}_d(t) + A \cdot \ddot{\Theta}_d(t) \end{aligned} \quad (13)$$

그러나 로봇의 동적 방정식 A, B, C 값을 전부 적응시키면 로봇 실시간(real time) 제어에 상당한 어려움이 따른다. 이와 같은 문제를 해결하기 위해서 A, B, C 행렬 즉 D, H, G 행렬을 추정해야한다.

(1) 파라미터의 추정

로봇 매니플레이터의 D, H, G 행렬을 $\hat{D}, \hat{H}, \hat{G}$ 로 추정하였다면 식(1)은

$$T(t) = \hat{D}(\Theta) \ddot{\Theta} + \hat{H}(\Theta, \dot{\Theta}) + \hat{G}(\Theta) \quad (14)$$

이 되고 $\hat{D}, \hat{H}, \hat{G}$ 는 다음 조건을 만족해야한다. [9]

i) 동적 방정식이 비선형 디커플링하기 위해서는

$$\hat{D} = D, \hat{H} = H, \hat{G} = G \text{ 이 되어야 한다.}$$

ii) 조건 (i)이 선형 시불변으로 하기 위해서 \hat{D} 는

$$\text{상수 행렬이어야하고 } \hat{H} = \hat{G} = 0 \text{ 이어야한다.}$$

또한 \hat{D} 를 구하기 위해서 \hat{D} 에 대한 가정이 필요하게 된다. [8]

가정 1) 식(1)의 관성 행렬 D(Θ)는 포지티브 데피니트 해야하고 식(15)을 만족하는 양의 상수 α가 존재해야한다.

$$\| [D(\Theta)]^{-1} \| \leq \alpha < \infty \quad (15)$$

가정 2) 식(16)을 만족하는 양의 상수 α (0 < α < 1)가 존재해야한다.

$$\| [D(\Theta)]^{-1} \cdot \hat{D}(\Theta) - I \| \leq \alpha \quad (16)$$

가정 1), 2)와 조건 1), ii)를 만족하는 $\hat{D}, \hat{H}, \hat{G}$ 를 구할 수 있다.

(2) 적응 제어기 설계

식(3)에서 본좌와 같이 A, B, C는 각각 D, H, G 행렬과 관계가 있으므로 추정된 D, H, G 행렬을 A, B, C에 대입하면 식(17)을 얻는다.

$$T(t) = F(t) + K_p(t) \cdot E(t) + K_v(t) \cdot \dot{E}(t) + \hat{A} \cdot \Theta_d(t) \quad (17)$$

이 되고

식(2)와 식(17)을 비교하면

$$\begin{aligned} A^* \cdot E(t) + (B^* + K_v) \cdot \dot{E}(t) + (C^* + K_p) \cdot E(t) \\ = -F(t) + (A^* - \hat{A}) \cdot \Theta_d(t) - B^* \dot{\Theta}_d(t) - C^* \Theta_d(t) \end{aligned} \quad (18)$$

이 된다.

식(18)을 2nx1의 위치-속도 오차상태 방정식을 쓰면

$$\begin{aligned} \begin{bmatrix} E(t) \\ \dot{E}(t) \end{bmatrix} &= \begin{bmatrix} 0 & I_n \\ \Delta_1 & \Delta_2 \end{bmatrix} \begin{bmatrix} E(t) \\ \dot{E}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta_3 \end{bmatrix} + \\ &\begin{bmatrix} 0 \\ \Delta_4 \end{bmatrix} \Theta_r(t) + \begin{bmatrix} 0 \\ \Delta_5 \end{bmatrix} \dot{\Theta}_r(t) + \begin{bmatrix} 0 \\ \Delta_6 \end{bmatrix} \Theta_r(t) \end{aligned} \quad (19)$$

여기서 $\Delta_1 = -A^{*-1} \cdot (C^* + K_p)$ $\Delta_2 = -A^{*-1} \cdot (B^* + K_v)$
 $\Delta_3 = -A^{*-1} \cdot F$ $\Delta_4 = -A^{*-1} \cdot C^*$ $\Delta_5 = -A^{*-1} \cdot B^*$
 $\Delta_6 = -A^{*-1} (A^* - \hat{A})$ 이다.

이제 "기준 모델"를 설정해야 하는데 이 기준 모델은 원하는 실행을 할수 있는 궤적 오차 E(t)의 형태로 이루어져야 한다.

$$E(t) + 2\xi_1 \omega_1 \dot{E}(t) + \omega_1^2 E(t) = 0 \quad (20)$$

ξ_1 와 ω_1 는 설계자에 의해서 설정되어지는 감쇠비와 비감쇠 자연 주파수 값이다. 식(20)을 상태 방정식으로 쓰면

$$\begin{aligned} \begin{bmatrix} E_m(t) \\ \dot{E}_m(t) \end{bmatrix} &= \begin{bmatrix} 0 & I_n \\ -\xi_1 & -2\omega_1 \end{bmatrix} \begin{bmatrix} E_m(t) \\ \dot{E}_m(t) \end{bmatrix} \\ &+ \begin{bmatrix} 0 & I_n \\ (D_1) & -(D_2) \end{bmatrix} \begin{bmatrix} E_m(t) \\ \dot{E}_m(t) \end{bmatrix} \end{aligned} \quad (21)$$

이고, 로봇의 식을 디커플시키기 위해서 D1과 D2는 $D_1 = \text{diag}(\omega_1^2)$, $D_2 = \text{diag}(2\xi_1 \omega_1)$ 로 만들었다. 적응제어 기법을 도입하기 위해 Lyapunov 함수를 유도해 보면

$$\begin{aligned} V &= e^T P e + [\Delta_0 - \Delta^*] Q_0 [\Delta_0 - \Delta^*] + \\ &+ \text{tr} \left\{ \sum_{i=1}^n [\Delta_i - \Delta^*] Q_i [\Delta_i - \Delta^*] \right\} \end{aligned}$$

이다. 이 Lyapunov를 시간에 대하여 미분한 값이 "0"보다 작다는 ($\dot{V} \leq 0$) Lyapunov의 두번째 방법에 의해서 적응 되어질 위치와 속도 이득 값을 식(22)에 의해서 구한다.

$$K_p(t) = K_p(0) + \alpha_1 R(t) + \alpha_2 \cdot \int_0^t R(t) \cdot E^T(t) dt \quad (22)$$

$$K_v(t) = K_v(0) + \beta_1 R(t) + \beta_2 \cdot \int_0^t R(t) \cdot E^T(t) dt$$

여기서 행렬 R(t)는 weighting 행렬이다.

또한 강인한 적응 제어기에 대해서는 보조시그널 F(t)는 시스템의 안정도 향상과 위치 오차의 수렴 속도를 빠르게 하는 항이다.

$$F(t) = F(0) + \delta_1 R(t) + \delta_2 \cdot \int_0^t R(t) dt$$

안정성의 관점에서 보면 식(17)으로 표현된 강인한 적응제어기를 사용하면 위치 오차 $e(t)$ 는 $\lim_{t \rightarrow \infty} e(t) = 0$ 으로 가는 점근적인 안정성을 띤다.

지금까지 보인 강인한 적응 제어기를 그림 2에서 나타내고 있다.

IV. Simulation

본 논문에서는 그림 3과 같은 자유도가 2인 스카라형 로봇트를 모델로 사용하였다.

여기서 링크의 길이 l_1 과 l_2 는 각각 0.23m이고 질량 m_1 는 2kg이고 m_2 의 무게는 부하에 대하여 변하는 항이다. 그리고 원하는 궤적의 경로는 초기 위치, 속도 그리고 가속도가 0이고 최종 속도와 가속도 또한 0이다. 로봇트의 궤적은 식(23)을 4초동안 따라간다.
 $\Theta_d(t) = \Theta(0) + (2\pi)^{-1} \cdot \{\Theta(tr) - \Theta(0)\}[\omega t - \sin \omega t]$
 $\dot{\Theta}_d(t) = [tr - t_0]^{-1} \cdot \{\Theta(t) - \Theta(0)\} \cdot (1 - \cos \omega t)$
 $\ddot{\Theta}_d(t) = [tr - t_0]^{-2} \cdot 2\pi \cdot \{\Theta(tr) - \Theta(0)\} \cdot \sin \omega t$ (23)

추정되어지는 \hat{A} 행렬은 III장의 조건과 가정들에 의해서 구해진다.

$$\hat{A} = \begin{bmatrix} 0.12 & 0.0 \\ 0.0 & 0.01737 \end{bmatrix}$$

그림 4은 로봇트의 기준 궤적과 속도, 가속도를 보여주고있다. 그림 5에서 보듯이 로봇트의 동적 방정식을 추정하여 강인한 적응 제어기를 설계한 것이 로봇트의 파라미터를 적용하는 기존의 제어기보다 오차가 0으로 가는 수렴하는 시간이 훨씬 짧았다.

그림 5은 무부하시의 위치 오차의 상태이고 그림 6은 1kg의 부하를 가지고 이동시에 위치 오차의 상태를 나타내고 있다.

V. 결론

본 논문은 로봇트의 비선형 동적 방정식 D,H,G행렬을 추정함으로써 보다 더 강건한 적응제어기를 구성하였다.

기존의 궤적에 의한 적응 제어 방법은 로봇트 매니퓰레이터의 매개변수의 값을 전혀 모른다고 할때 유효한 제어 방법이나, 적용되어지는 항이 많으므로 토오르크의 계산 시간이 길어져 실시간 제어는 불가능하다. 이 문제를 해결하기 위해서 로봇트 동적

방정식의 D,H,G행렬을 추정함으로써 토오르크 입력 계산 시간을 줄이고 로봇트가 부하의 변동에 상관 없이 기준 궤적을 빨리 따라가도록 하였다.

VI. 참고 문헌

- 1) S.Dubowsky and D.T.Desforjes, "The Application of Model Reference Adaptive Control to Robotic Manipulators," Journal of Dym Sys Meas and control, Vol.101, September 1979, pp193-200
- 2) Lee, C.S.G., and Chung, M.J., "An Adaptive Control Strategy for Mechanical Manipulator," IEEE Trans Auto Control(AC), Vol .29, No.9, 1984, pp.837-840
- 3) H.Seraji, "A New Approach To Adaptive Control of Manipulators," Journal of Dynamic System, Measurement and Control, Vol.109, Sept.1987, pp193-202
- 4) K.Y.Lim, and Eslami, M., "Adaptive Controller Design for Robot Manipulators System Using Lyapunov Direct Method," IEEE Trans. Automatic Control(AC), Vol.30, No.12, 1985, pp.1229-1233.
- 5) K.Y.Lim, "Model Reference Adaptive Control for Linear System with Improved Convergence Rate-Signal Synthesis Method," Trans. KIEE, Vol.37, No.10, pp.733-739, Oct.1988,
- 6) K.Y.Lim., "Model Reference Adaptive Control of Linear System with Improved Convergence Rate Parameter Adaptation Method," Trans. KIEE, Vol.3, No.12, pp.884-893, Dec.1988,
- 7) Mei-Hua Liu, Wei Lin, "Pole Assignment Self-Tuning Controller for Robt manipulator", Int J. of Control, Vol.46, No4, 1987, pp1037-1317
- 8) Claude SAMSON., "Robust Control Of A Class Of Non-Linear System And Application To Roboticsics," Inter Jour Adaptive Control And Signal Processing, Vol.1, 1987, pp49-68
- 9) Mark W.Spong, M.Vidyasagar, "Robust Linear Compensator Design For Nonlinear Robotics Automation, 1985, pp954-959
- 10) W.M.Grimm, N.Becker., P.M.Frank, "Robust Stability, Design Framwork For Robot Manipulator Control", IEEE Conf.on Robotics Robotics and Automation, 1988, pp1042-1047

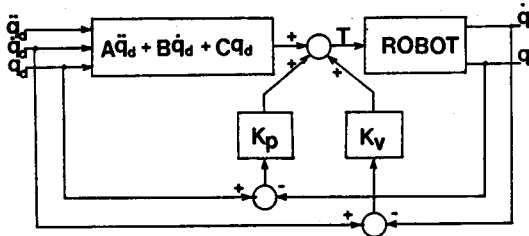


그림 1. 선형화 시킨 제어기 모델

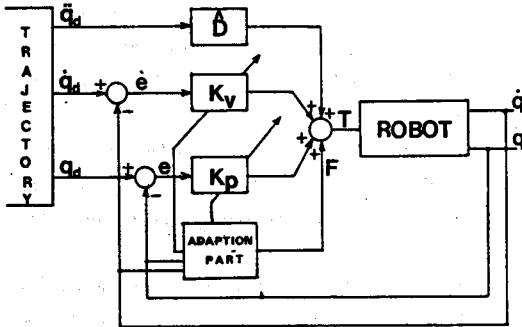


그림 2. 제안된 강인한 적응 제어기

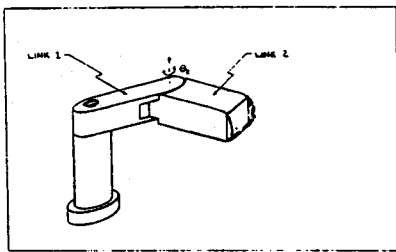


그림 3. 스카라형 로봇

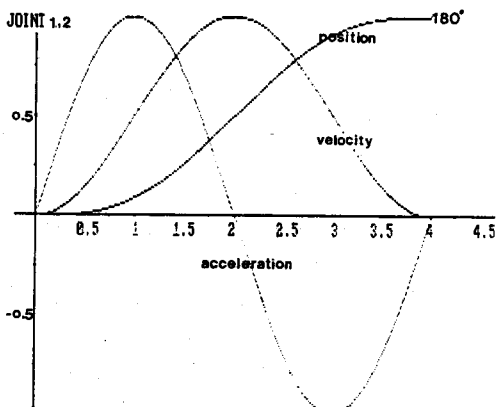


그림 4. 기존 궤적, 속도, 가속도

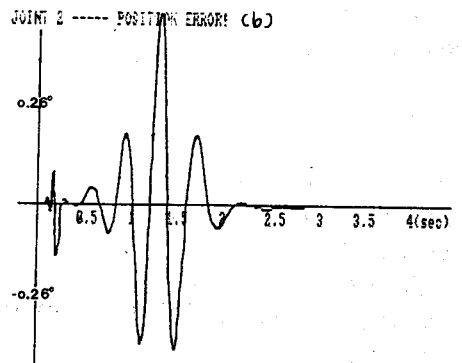
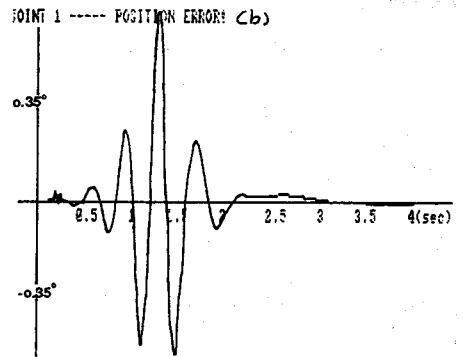
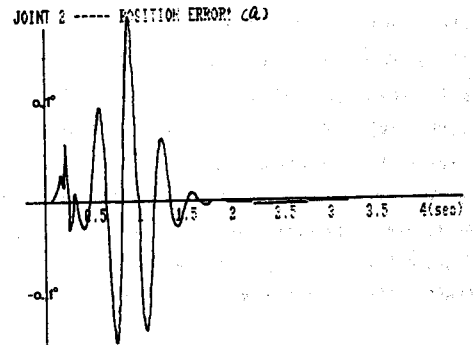
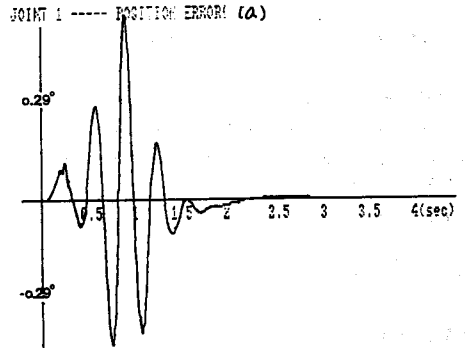


그림 5. 제안된 제어기(a)와 기존의 제어기 비교(b) 위치 오차의 비교 (무부하시)

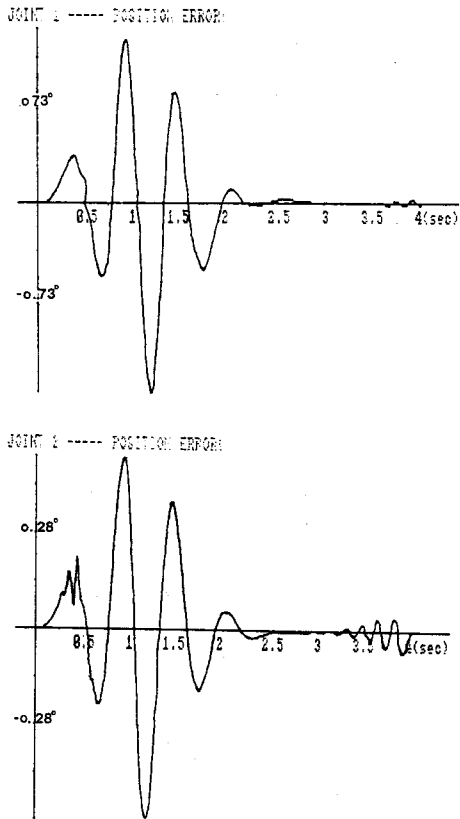


그림 6. 제안된 제어기의 위치 오차(1Kg 부하시)