

로봇 매니플레이터의 직교공간 적응제어 방식

황석용, 유준
충남대학교 전자공학과

A Cartesian Space Adaptive Control Scheme for Robot Manipulators

Seok Yong Hwang and Joon Lyou
Dept. of Electronics Eng., Chungnam National Univ.

ABSTRACT

This paper presents a cartesian space decentralized adaptive controller design for the end effector of the robot manipulator to track the given desired trajectory in the cartesian coordinate. By the cartesian based control scheme, the task related high level motion command is directly executed without solving the complex inverse kinematic equations. The controller does not require the complex manipulator dynamic model, and hence it is computationally very efficient. Each degree of freedom of the end effector on the cartesian space is controlled by a PID feedback controller and a velocity acceleration feedforward compensation part. Simulation results for a two-link direct drive manipulator conform that the present cartesian based decentralized scheme is feasible.

1. 서론

매니플레이터의 제어는 작업공간 (Task Space) 에서 로보트 손 (End Effector)의 동작을 직접 제어하는 방식과 작업공간에서의 로보트 손의 동작을 조인트 공간(Joint Space)으로 변환시킨 후 제어하는 방식으로 대별할 수 있다. 작업과 관련된 로보트 손에 대한 일련의 동작은, 작업자의 명령이나 비전(Vision) 등의 각종 센서 데이터에 의해 수행되므로, 직교 좌표공간(Cartesian Space)에서 표현된다. 로보트 손을 조인트 공간에서 제어하는 경우 역기구학을 통한 조인트 각의 변위를 구해야하며 일반적으로 역기구학 자체가 심한 비선형 시스템이므로 Closed Form 의 해를 구할 수 없는 문제점이 있다.

한편 로보트에 적용되고 있는 제어기로 각 조인트를 독립적인 시스템으로 간주하고 미리 결정된 상수이득들에 의해 각 조인트를 제어하는 독립조인트 제어기 (Independent Joint controller)는 Pick-and-Place 와 같은 단순한 작업은 원만히 수행할 수 있으나 Payload 가 변할 때 빠른 귀격을 추적할 경우 만족할만한 결과를 얻을 수 없다. 이러한 기존의 제어기가 갖는 문제점을 해결하기 위한 방안으로 매니플레이터의 동력학적 모델을 이용하는 방법(Model-based Method)과 적응 제어기법을 이용하는 방법에 대한 연구가 활발히 수행되어왔다 [1]-[2]. Computed Torque Method[1] 등과 같은 매니플레이터의 동력학적인 모델을 이용하는 방법은 제어기설계시 정확한 모델의 구조 및 파라미터를 요구

한다. 매니플레이터의 동력학 모델 자체가 다수의 시변 파라미터를 포함하는 심한 비선형 시스템이며 더우기 정확한 측정이 힘든는 각 조인트의 마찰, Payload값 등을 포함하므로 정확한 모델링이 힘들며 이러한 모델링 오차가 제어기의 성능에 직접적인 영향을 미치며 온-라인으로 제어하기 힘들다. 이와는 달리 적응 제어기는 매니플레이터의 복잡한 동력학 모델링을 필요로 하지 않으며 제어기의 구현이 동력학모델을 이용하는 방법보다 용이하다. 이러한 적응제어기는 주로 매니플레이터의 조인트공간 제어를 목적으로 개발되었으며 작업공간 제어기에 관한 연구도 수행되어 왔다 [3]. 그러나 제어기가 집중식구조를 갖는 경우 많은 계산시간을 요구하게 되는 문제점을 내포하고 있다.

제어기의 집중적 제어구조로부터 야기되는 이러한 문제점을 해결하기 위한 방안으로 제어기를 각각의 부시스템별로 분산시키고 국부(local)제어기에 적응기구를 도입하는 비집중(decentralized) 적응제어 방식에 대한 연구가 진행되고 있다[4-6]. 즉 조인트공간 제어기의 경우 조인트를 부시스템으로 간주하고, 작업공간 제어기의 경우 로보트 팔의 각 자유도를 부시스템으로 간주하여 전체 시스템을 분산하여 제어하는 방법이다. 비집중식에 의한 간단한 제어구조를 통하여 계산량 부담이 크게 줄어들고, 매니플레이터의 넓은 동작 범위에 대해 만족스러운 성능을 얻을 수 있다. 본 연구는 [4], [7]의 설계개념을 바탕으로 로보트손을 직교좌표 공간에서 직접 제어가능한 다른 형태의 분산형 적응 제어방식에 관한 연구이다. 참고로 본 방식과 [4]의 방식을 비교해 보면, 국부 제어기가 feedforward 보상부와 feedback 제어부로 구성되는 측면에서 구조상 비슷하나, 국부제어 입력을 구하는데 필요한 제어파라미터가 다르고 따라서 적응기구에 차이가 있으며 적응법칙은 [4]의 유도과정에 비해 상당히 간단히 도출이 되며, 동일한 매니플레이터를 대상으로 시뮬레이션한 결과 대등한 성능을 보였다.

2. 분산형 적응 제어기 설계

2.1. 매니플레이터의 동력학 모델

n 차원의 직교 좌표계 공간(Cartesian Space)에서 작업을 수행하는 n 개의 조인트로 구성된 매니플레이터의 동력학 방정식은 식 (1) 과 같이 2차 비선형 벡터 미분방정식으로 표현될 수 있다.[9]

$$H(X)\ddot{X} + C(X, \dot{X})\dot{X} + G(X, \dot{X}) = F \quad (1)$$

여기서 X 는 $n \times 1$ 차원의 직교 좌표계공간에서 정의된 로보트 손의 위치 및 오리엔테이션 벡터, $H(X)$ 는 $n \times n$ 차원의 관성 행렬(Symmetric Positive Definite), $C(X, \dot{X})X$ 는 $n \times 1$ 차원의 원심력과 코리올리력 벡터, $G(X, \dot{X})$ 는 $n \times 1$ 차원의 중력과 마찰력을 포함하는 벡터를 나타내며 F 는 로보트 손에 가할 $n \times 1$ 차원의 직교 좌표계공간에서 정의되는 힘과토크로크 벡터이다. H, C, G 의 각 요소들은 Payload 의 관성파라미터 및 X 와 \dot{X} 의 비선형 함수이며, H 가 비대각 행렬이므로 매니플레이터의 동역학 방정식은 직교좌표계공간의 자유도 사이에 심한 상호 작용이 존재함을 나타낸다. 분산형 제어기의 설계를 위하여 직교 좌표계공간에서 정의된 로보트 손의 각 자유도에 대한 시스템을 하나의 부시스템으로(subsystem)으로 간주하고, 식 (1) 로 표현되는 시스템을 관성결합, 코리올리력, 중력등에 의해 부시스템간의 상호 간섭작용이 존재하는 시스템으로 생각하고 (1)의 동적 모델을 n 개의 비선형 미분방정식으로 분해하여 (2) 식과 같이 쓸 수 있다.

$$h_{ii}(x)\ddot{x}_i + c_{ii}(x, \dot{x})\dot{x}_i + d_i(x, \dot{x}, \ddot{x}) = f_i, \quad i = 1, 2, \dots, n \quad (2)$$

$$d_i = \sum_{j=1, j \neq i}^n \{h_{ij}(x)\ddot{x}_j + c_{ij}(x, \dot{x})\dot{x}_j\} + g_i(x, \dot{x}) \quad (2.a)$$

여기서 i 번째 미분방정식은 f_i 를 입력으로, x_i 를 출력으로 하는 i 번째 부시스템의 입력력 모델이고, d_i 는 i 번째 부시스템에 총체적으로 작용하는 외란 토크로크로 간주한다.

2.2. 제어기의 구조

매니플레이터의 로보트 손을 직교좌표 공간에서 제어하는 경우 제어문제는 주어진 궤적을 충실히 추적할 수 있도록 로보트 손에 가해줄 힘 및 토크로크 벡터 F 를 구하는 것이다. 본 연구에서 제안하는 제어기는 (2) 식으로 표현되는 각 부시스템에 적응기구(Adaptive Mechanism)을 도입하여 힘 및 토크로크 벡터 F 를 구한후 이를 조인트 공간에서의 토크로크 벡터 τ 로 변환하여 각 조인트에 부착된 역추어터러를 구동하게 된다. 부시스템의 국부(local)제어기는 로보트 손의 위치(x_i) 및 속도(\dot{x}_i)만을 요구하며 직접적인 측정이 불가능할 경우 매니플레이터의 기구학 방정식을 이용하여 조인트 공간의 각 변위와 각속도로부터 구할 수 있다. 힘, 토크로크 벡터 F 를 구하는 과정에서 분산화된 제어기가 토크로크 벡터 τ 와 상태 변수들을 구하는 과정에서 집중화가 되지만 제어파라미터를 자동생성하는 적응기구(Adaptive Mechanism)의 도입으로 계산량이 집중식 제어기에 비해 많이 준다. 조인트 공간에서의 각변위와 각속도만이 측정가능한 경우 전체 제어기의 구조는 그림 1과 같다.

직교좌표계에서 로보트 손이 추종할 원하는 궤적(Desired Trajectory)이 $(\dot{x}_{d1}, \dot{x}_{d2}, \dot{x}_{d3})$ 로 주어지고, 위치오차, 속도 오차, 가속도오차를 각각 $\tilde{x}_i = x_i - x_{d1}$, $\dot{\tilde{x}}_i = \dot{x}_i - \dot{x}_{d1}$, $\ddot{\tilde{x}}_i = \ddot{x}_i - \ddot{x}_{d1}$ 로 정의하고 또한 개별 국부 제어기의 설계에 필요한 신호들을 다음과 같이 정의한다.

$$s_i = \tilde{x}_i + \lambda_i \dot{\tilde{x}}_i, \quad \lambda_i > 0 \quad (3.a)$$

$$\dot{x}_{r1} = \dot{x}_d - \lambda_i \dot{\tilde{x}}_i \quad (3.b)$$

$$\ddot{x}_{r1} = \ddot{x}_d - \lambda_i \ddot{\tilde{x}}_i \quad (3.c)$$

여기서 λ_i 는 양의 상수로 선정되는 설계변수이다. 이제 추종오차를 (가능하면 빨리) 0 으로 보내기 위하여, Lyapunov 설계법 [3,6] 에 입각한 비집중식 분산 적응제어기를 다음과 같이 제시한다.

$$f_i = \hat{d}_i(t) + \hat{c}_{ii}(t)\dot{x}_{r1} + \hat{h}_{ii}(t)\ddot{x}_{r1} - k_i s_i \quad (4)$$

$$\dot{\hat{d}}_i = -\gamma_{i1} s_i \quad (4.a)$$

$$\dot{\hat{c}}_{ii} = -\gamma_{i2} s_i \dot{x}_{r1} \quad (4.b)$$

$$\dot{\hat{h}}_{ii} = -\gamma_{i3} s_i \ddot{x}_{r1}, \quad i = 1, 2, \dots, n \quad (4.c)$$

$\hat{d}_i(t), \hat{c}_{ii}(t), \hat{h}_{ii}(t)$ 는 (2)의 d_i, c_{ii}, h_{ii} 의 추정치로서 적응기구를 통하여 매순간 계산된다. $\gamma_{i1}, \gamma_{i2}, \gamma_{i3}$ 는 가중치로서 양의상수로 선택되며, $k_i > 0$ 은 고정된 비례, 미분 이득(PD gain)으로서 추종오차 과도특성을 개선하기 위하여 첨가되었다. (3)의 정리식을 이용하여 (4)의 각 항을 풀어 쓰면, 관련항끼리 모아 정리하여 식 (5)로 표현한다.

$$f_i = \beta_1(t)\ddot{x}_{d1} + \beta_2(t)\dot{x}_{d1} + \beta_3(t)\dot{x}_i + \beta_4 f_i \dot{x}_{id} + \beta_5(t)\ddot{x}_i \quad i = 1, 2, \dots, n \quad (5)$$

여기서 $\beta_1 = \hat{h}_{ii}, \beta_2 = \hat{c}_{ii}, \beta_3 = -\lambda_i k_i - \gamma_{i1} - \lambda_i \hat{c}_{ii}, \beta_4 = -\gamma_{i1} \lambda_i, \beta_5 = -k_i - \lambda_i h_{ii}$ 로 주어진다. (5)의 처음 두항은 기준궤적 정보를 이용한 보상부이고, 나머지 세항은 추종오차에 대해 비례, 적분, 미분(PID) 작용을 취하는 부분이다. 즉 국부제어기는 적응법칙 (4.a)-(4.c)에 의해 매순간 생성되는 가변이득을 사용하여 feedforward 보상과 feedback 제어를 수행한다 (그림 2 참조).

그러나 (4) 식으로 표현되는 힘을 로보트 손에 물리적으로 가할 수 없기 때문에 식 (6)으로 표현되는 로보트 손에 가해지는 힘과 조인트에 부착된 역추어터러에 가해지는 토크로크의 상관관계식을 통하여 제어하게 된다 [9]. 식 (6)에서 J^T 는 조인트 공간에서의 조인트의 미소변위를 직교좌표 공간에서의 로보트 손의 변위로 변환시키는 자코비안(Jacobian) 행렬의 전치행렬이다. 각 부시스템의 상태변수 x_i 는 카메라 시스템과 거리감지 센서등을 이용하여 획득할 수 있으나 일반적으로 조인트 공간에서의 θ_i , 즉 매니플레이터의 기구부를 구성하는 각 조인트의 변위를 엔코더 등의 센서를 이용하여 구한 후 기구부의 정기구학 방정식에 의해 구해진다.

$$\tau = J^T F \quad (6)$$

2.3. 안정 해석

제안한 제어기를 포함한 전체 시스템의 안정성을 보이기 위하여, 먼저 시스템 파라미터, $h_{ii}(x), c_{ii}(x, \dot{x}), d_i(x, \dot{x}, \ddot{x})$ 들이 제어갱신(Control Update) 속도에 비해 느리게 변한다고 가정한다[2-5]. 즉 $h_{ii} \neq 0, c_{ii} \neq 0, d_i \neq 0$ 이라 하자. 참고로 본 제어방식은 분산형이므로 계산량 부담이 거의 없어, 이러한 가정을 정당화하기에 유리하다. 시스템 안정성을 해석하기 위하여 Lyapunov 함수를 다음과 같이 선정한다.

$$V(S_i, \tilde{x}_{i1}, \tilde{x}_{i2}, \tilde{x}_{i3}) = \sum_{i=1}^n (h_{ii}(x)s_i^2 + \gamma_{i1}^{-1} \tilde{x}_{i1}^2 + \gamma_{i2}^{-1} \tilde{x}_{i2}^2 + \gamma_{i3}^{-1} \tilde{x}_{i3}^2) \quad (7)$$

여기서 $\tilde{x}_{i1}, \tilde{x}_{i2}, \tilde{x}_{i3}$ 는 파라미터 오차로서 $\tilde{x}_{i1} = \hat{d}_i - d_i, \tilde{x}_{i2} = \hat{c}_{ii} - c_{ii}, \tilde{x}_{i3} = \hat{h}_{ii} - h_{ii}$ 이다. V 를 시간에 따라 미분하고, 매니플레이터 다이내믹스(Dynamics) (2) 를 따라 평가하면 다음과 같다.

$$\dot{V} = \sum_{i=1}^n (s_i(f_i - h_{ii}\ddot{x}_{r1} - c_{ii}\dot{x}_{r1} - d_i) + 1/2s_i^2(\dot{h}_{ii} - 2c_{ii}) + \gamma_{i1}^{-1} \dot{\tilde{x}}_{i1} \tilde{x}_{i1} + \gamma_{i2}^{-1} \dot{\tilde{x}}_{i2} \tilde{x}_{i2} + \gamma_{i3}^{-1} \dot{\tilde{x}}_{i3} \tilde{x}_{i3}) \quad (8)$$

여기서 $(h_{ii} - 2c_{ii})$ 항을 c_{ii} 의 적절한 표기에 의해 0 으로 소거시킬 수 있다. 또한 식 (4)로 주어진 제어 및 적응법칙을 대입하여, (8) 식은 다음 식 (9)로 간략화 된다.

$$\ddot{V} = \sum_{i=1}^n \{-k_i s_i^2 + \delta(\dot{d}_i, \dot{c}_{ii}, \dot{h}_{ii})\} \quad (9)$$

여기서 $\delta = -\gamma_{11}\dot{\psi}_{11}\dot{d}_1 - \gamma_{12}\dot{\psi}_{12}\dot{c}_{11} - \gamma_{13}\dot{\psi}_{13}\dot{h}_{11}$. (9) 식에서 δ_i 는 가정에 의해 작은 값이며, δ 가 엄밀히 "0" 가 아니어도 k_i 를 큰 값으로 취하여 $V \leq 0$ 이 되게 할 수 있다. 이상에서 $V \geq 0, \dot{V} \leq 0$ 이므로, Lyapunov 안정이론의 표준논리 [7]에 의하면, 시스템내의 신호들 $s_i(t), \dot{\psi}_{11}(t), \dot{\psi}_{12}(t), \dot{\psi}_{13}(t)$ 는 임의의 유한한 초기치 $s_i(t_0), \dot{\psi}_{11}(t_0), \dot{\psi}_{12}(t_0), \dot{\psi}_{13}(t_0)$ 및 모든 t 에 대하여 유한하다 (globally uniformly bounded). 또한 $\delta_i=0$ 이라는 가정아래서 (9) 의 V 은 모든 i 에 대하여 $s_i=0$ 일 때만 0 이므로, s_i 는 시간이 경과함에 따라 0 으로 수렴한다. 추종오차 x 는 지수함수적으로 (감쇄율 λ_i) 0 으로 수렴한다.

3. 시뮬레이션

수직 평면상을 움직이고, 동력이 기어를 통하지 않고 직접 전달되는 수직 2관절 직결구동형 강체(rigid) 매니플레이터를 컴퓨터 시뮬레이션 대상으로 삼았다. 대상 모델의 동적 파라미터들은 다음과 같이 주어지며[3], 이의 실제 외관은 PUMA 560 의 링크 2, 3 이 연결된 모습이다. 기준궤적이 초기위치 ($x=0.0, y=-0.864$)에서 최종위치 ($x=0.432, y=0.432$)로 직선을 따라 3 초 동안에 완만하게 움직이도록 계획되었다. x_{d1} 는 아래와 같으며, $\dot{x}_{d1}, \ddot{x}_{d1}$ 은 x_{d1} 로부터 미분하여 구할 수 있다.

$$\begin{aligned} x_{d1} &= 0.432/2\pi(2\pi t/3 - \sin(2\pi t/3)) \text{ m}, & 0 \leq t \leq 3 \\ &= 0.432, & 3 < t \\ x_{d2} &= -0.864 + 1.296/2\pi(2\pi t/3 - \sin(2\pi t/3))\text{m}, & 0 \leq t \leq 3 \\ &= 0.432, & 3 < t \end{aligned}$$

대상 매니플레이터에 대해 제시된 국부적용제어기를 적용한 컴퓨터 모사물. Payload 조건으로 다음의 두가지 경우를 고려하여 수행하였다.

- (i) Payload 변화가 없는 경우 ($m = 10 \text{ Kg}$)
- (ii) 1.5 초에 Payload 를 떨어뜨린 경우

(4)식에 주어진 설계변수, 초기추종 오차, 적응변수의 초기 추종치를 다음과 같이 정하여 시뮬레이션한 결과가 그림 3-7에 주어져 있다.

$$\begin{aligned} \lambda_1 &= 100, \gamma_{11} = 500, \gamma_{12} = 100, \gamma_{13} = 100, k_1 = 250, \\ \hat{x}_1(0) &= 0, \hat{\dot{x}}_1(0) = 0, \hat{h}_{11}(0) = 0, \hat{c}_{11}(0) = 0, i = 1, 2 \\ \hat{d}_1(0) &= 10, \hat{d}_2(0) = 5 \end{aligned}$$

그림 3-4는 Payload 조건 (i)에 대해 직교좌표계의 각 축에 대한 로봇트 손의 위치가 기준궤적을 잘 추종하고 있음을 보여준다. X축에 대한 로봇트 손의 위치 오차는 6mm 이내에서 Y 축에 대해서는 25mm 이내에서 유지되며 최종위치 오차가 거의 존재하지 않음을 알 수 있다. 그림 5는 각 부시스템에 가해지는 힘 입력 $f_i(t)$ 를 도시한 것이다. 그림 6 은 Payload 조건 (ii) 에 대해 위치오차를 나타낸 것으로 모델파라미터가 천천히 변한다는 가정이 위배됨에도 불구하고 본 제어기가 만족할만한 성능을 유지함을 보여준다. 본 방식과의 성능 비교를 위하여 상수 PD 이득 제어기를 적용한 컴퓨터 모사를 수행하였다. (4)식에 k_1 만을 가동시켜 ($k_1 = 250, \lambda_1 = 10$) Payload 조건 (i)에 대해 시뮬레이션한 결과가 그림 7 에 나타나 있으며 위치오차가 크게 증가하고 최종위치 오차가 더 이상 줄지않음을 알 수 있다.

4. 결론.

매니플레이터의 로봇트 손을 직교 좌표공간에서 직접 제어함으로써 복잡한 역기구학의 과정이 필요하지 않는 제어기 (Cartesian Space Controller)를 설계하였으며 매니플레이터

모델의 비선형성, 직교 좌표계에서 로봇트 손의 각 자유도 (Degree of Freedom)에 해당하는 부시스템간의 상호결합성, 링크 파라미터나 부하하중의 불확실성을 효과적으로 다루기 위한 방안으로, 분산형 적응제어 방식을 Lyapunov 설계법을 확장하여 제시하였다. 본 방식은 각 부시스템의 정보만을 사용하여 제어입력을 구하는, 부시스템간 정보교환이 필요치 않는 분산형구조이며, 각 국부제어기는 적응기구를 도입함으로써 직교좌표계에서 표현되는 로봇트 손의 복잡한 동적 모델에 의존하지 않고 제어 파라미터를 생성해낸다. 따라서 제어기의 신뢰도가 향상되고, 적은 제어계산량으로 넓은 동작 범위에서도 성능을 계속 유지시킬 수 있어 실제 적용에 적합하다. 부시스템들에 의해 생성된 제어입력은 매니플레이터의 기구학 방정식으로부터 유도되는 자코비안 행렬을 이용하여 조인트 공간의 토오르로 변환되고, 각 부시스템의 제어가 필요로하는 정보들은 필요에 따라 정기구학 방정식을 이용함으로써 조인트 공간의 정보로부터 구할 수 있다.

본 방식을 사용한 시뮬레이션의 결과로, 급격한 Payload 의 변화, 로봇트 손의 각 부시스템간의 심한 상호결합에도 불구하고 각 로봇트 손의 실제 위치가 기준궤적을 잘 추종함을 확인할 수 있었다.

5. 참고 문헌

- [1] R.P.Paul, Robot Manipulators: Mathematics, Programming and Control (MIT Press, Cambridge MA, 1981).
- [2] T.C.Hsia, 1986(San Francisco, Calif.). "Adaptive control of robot manipulators - A review". IEEE Int.Confer. Robotics and Automation.
- [3] H.Seraji, "Direct Adaptive Control of Manipulators in Cartesian Space". J. of Robotic Systems, 4(1), 157-178(1987).
- [4] H.Seraji, "Decentralized Adaptive Control of Manipulators : theory Simulation and Experimentation", IEEE Int.Conf. Robotics and Automation", Vol 5, pp 183-201 Apl, 1989.
- [5] D.T.Gavel and T.Hsai, "Decentralized adaptive control of robot manipulators". Proc. IEEE Int.Conf. on Robotics and Automation (Raleigh, NC, 1987).
- [6] B. J. Oh, M. Jamshidi and Seraji, "Decentralized Adaptive Control of Robot Manipulators", in Proc. IEEE Int.Conf. Robotics and Automation (Philadelphia PN, 1988), pp. 1216-1201.
- [7] Slotine, J. J. E., and Li, W., "On the adaptive control of robot manipulators". IEEE Int.Jour. of Robotics Research, Vol 6, No 3, 1987.
- [8] K.S.Narendra and A.M. Annaswamy, Stable Adaptive Systems (Prentice-Hall Inc., 1989)
- [9] O. Khatib, "A unified approach for motion and force control of robot manipulators : The operational space formulation, " IEEE J. Robotics Automat., Vol. RA-3, no.1, 1987.

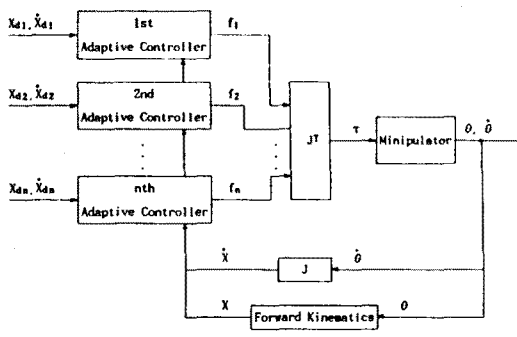


그림 1. 전체 제어기의 구조

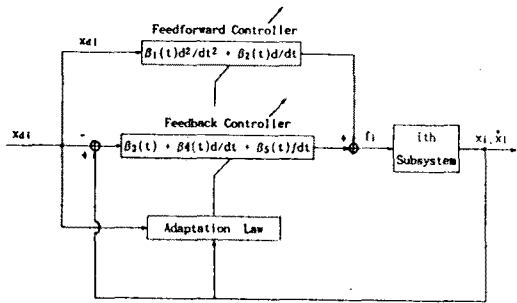
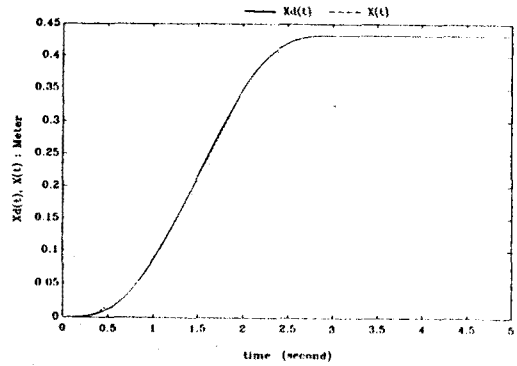
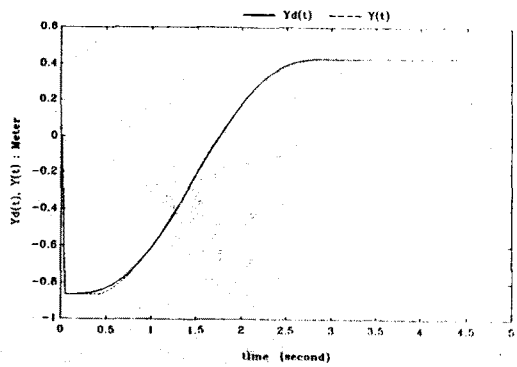


그림 2. 분산형 적응 제어기의 구조



(a) x 축



(b) y 축

그림 3. 기준궤적(—) 및 각 축의 실제위치(-----)

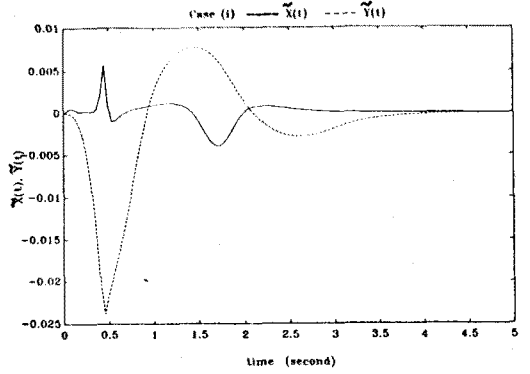


그림 4. 위치오차의 확대

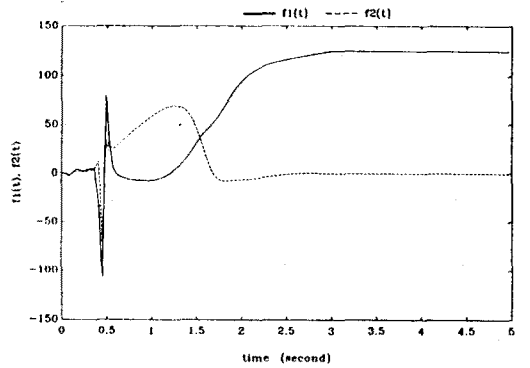


그림 5. 힘 입력 $f_1(t)$

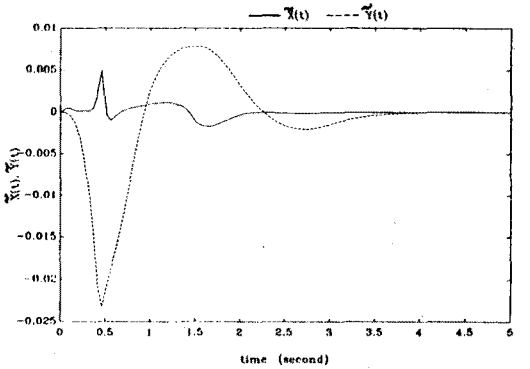


그림 6. 경우 (ii)에 대한 위치 오차

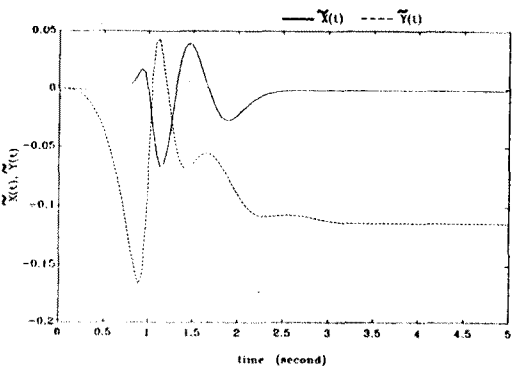


그림 7. 상수 PD 이득 제어에 의한 위치오차