

# 신경회로망을 이용한 불확실한 로봇 시스템의 하이브리드 위치/힘 제어

## Hybrid Position/Force Control of Uncertain Robotic Systems Using Neural Networks

김성우, 이주장  
(Sung-Woo Kim and Ju-Jang Lee)

**Abstract** : This paper presents neural networks for hybrid position/force control which is a type of position and force control for robot manipulators. The performance of conventional hybrid position/force control is excellent in the case of the exactly-known dynamic model of the robot, but degrades seriously as the uncertainty of the model increases. Hence, the neural network control scheme is presented here to overcome such shortcoming. The introduced neural term is designed to learn the uncertainty of the robot, and to control the robot through uncertainty compensation. Further more, the learning rule of the neural network is derived and is shown to be effective in the sense that it requires neither desired output of the network nor error back propagation through the plant. The proposed scheme is verified through the simulation of hybrid position/force control of a 6-dof robot manipulator.

**Keywords** : robot manipulators control, position/force control, neural networks, learning rule

### I. 서론

오늘날 산업 현장의 여러 분야에서 쓰이는 로봇 매니플레이터들은 자유 공간상에서의 작업 (free space task) -예를 들어, 물건을 옮기거나, 페인팅 작업등- 뿐만 아니라, 작업 환경에 구속되는 작업 (constrained space task) -예를 들어 표면 가공 또는 조립등-의 여러 가지 작업을 정확하게 수행하여야만 한다. 만약 매니플레이터가 자유 공간상에서만 작업한다면, 위치 제어 (position control) 또는 추적 제어 (tracking control)만으로도 제어가 가능하지만, 매니플레이터의 엔드 이펙터 (end-effector)와 작업 환경사이에 어떠한 접촉이 생기는 경우라면, 이러한 추적 제어만으로는 우리가 원하는 제어를 얻을 수 없다. 왜냐하면, 이러한 경우에는 약간의 위치 오차만으로도 커다란 힘 오차를 유발하게 되므로 매니플레이터뿐만 아니라 접촉면에도 손상이 가해질 수 있기 때문이다.

이와 같이 매니플레이터가 접촉하여 작업하는 경우에 발생하는 위와 같은 문제점을 해결하기 위하여 지금까지 다음과 같은 여러 가지 힘 제어 방식들이 제안되어 왔다. 먼저 임피던스 제어 (impedance control)은 Hogan [1]에 의하여 제안된 방식으로, 힘과 위치 사이의 관계를 임피던스의 관계로 보고, 이 관계로부터 매니플레이터의 위치를 제어함으로써 힘 제어를 하게되는 방식이다. 다음 하이브리드 위치/힘 제어 (hybrid position/force control) [2-6]로서, 힘 제어하는 방향과 위치 제어하는 방향을 어떠한 '작업 선택 행렬 (task selection matrix)'에 의하여 선택하고, 각각의 힘과 위치 제어를 독립적으로 수행하는 방식이다. 세 번째는 구속 움직임 제어 (constrained motion control) 방식 [7,8]으로 매니플레이터가 접촉면과의 구속 조건을 만족시키기에 필요한 구속력 (constrained force)을 이용한 제어 방식이다. 이외에도 여러 가지 힘 제어 방식이 있는데 이들에 대한 설명은 Whitney [9]의 논문에서 찾아 볼 수 있다.

만약, 로봇의 다이내믹 모델과 접촉면과의 운동 방정식을 정확하게 모델링할 수만 있다면, 가능한 힘 제어 방식으로, 위치 제어에서 쓰이는 CTM (computed-torque method)와 같이 제어 입력에 운동방정식의 비선형 항들을 포함함으로써 이들을 보상시키는 방법을 들 수 있다. 하지만 이 방식은 다이내믹 모델에 포함되지 않은 매니플레이터, 접촉면, 센서 등의 여러 가지 불확실성 등으로 인하여 이들 비선형 항들을 완전히 보상하지 못하므로 실제로는 정확한 제어가 가능하지 않다. 따라서 만족할 만한 제어를 이루기 위해서는 이러한 불확실성들을 극복할 수 있는 제어 방식이 필요하게 된다.

적용 제어 (adaptive control) 또는 강인 제어 (robust control)등이 이러한 불확실성을 극복하기 위한 선택적인 대안이 될 수 있으나, 적용 제어 방식은 구조화되지 않은 불확실성이 있는 경우에는 그 성능을 보장할 수 없고, 또한 강인 제어 방식들은 대부분 불확실성에 대한 사전 정보로써, 그 최대 값을 알고 있어야 한다는 가정이 필요하다 [10].

또 다른 대안으로서 신경회로망을 이용한 제어를 들 수 있다. 신경회로망은 학습 (learning), 비선형 맵핑 (nonlinear mapping), 병렬 처리 (parallel processing)등 비선형, 불확실한 시스템의 제어에 매우 유용하게 쓰일 수 있는 기능들을 갖고 있다. 학습과 비선형 맵핑능력은 지금까지 많은 비선형 시스템에 응용되어 성공적인 결과를 얻고 있음이, 또 병렬 처리를 이용하여서는 실시간 제어를 할 수 있음이 잘 알려진 사실이다 [11-14]. 더욱이 최근에, 신경회로망을 이용한 매니플레이터의 제어가 많이 수행되어져 왔다 [15-18]. 하지만 이들은 모두 위치 제어에 신경회로망을 이용한 것으로서 힘 제어에는 적합하지 않다. Fukuda [14]에 의하여 신경회로망을 이용한 힘 제어가 제안되었으나, 이 논문에서의 결과는 한 방향의 힘 제어에 대한 것 뿐으로서, 위치 제어와 힘 제어를 동시에 수행하지는 못하고 있다.

본 논문에서는 위치 제어뿐만 아니라 힘 제어를 불확실한 로봇 시스템에 대하여 동시에 수행할 수 있는 신경회로망

제어 기법을 제안한다. 이를 위하여 지금까지의 힘 제어 방식 중, 하이브리드 위치/힘 제어 방식을 도입한다. 이는 하이브리드 방식이 개념적으로 비교적 이해하기 쉽고 또한 신경회로망을 도입하기에 적합한 제어 방식이기 때문이다. 따라서 본 논문은 다음과 같이 구성된다. 2장에서 힘 제어 방식에 근간이 되는 몇 가지 이론을 도입하고, 3장에서는 로봇의 하이브리드 위치/힘 제어 방식을 살펴본다. 그리고, 불확실성이 있는 경우에 이를 극복하기 위한 신경회로망 하이브리드 위치/힘 제어기를 4장에서 설계한다. 다음 5장에서 6축 PUMA 로봇에 대하여 제안된 신경회로망 하이브리드 제어기를 시뮬레이션을 통하여 성능을 검증하고, 6장의 고찰을 거쳐, 마지막 7장에서는 결론을 맺는다.

## II. 힘 제어의 기본 이론

기존의 여러 가지 힘 제어 이론 중에, 본 논문에서는 하이브리드 위치/힘 제어 방식을 도입한다. 하이브리드 제어는 기본적으로 위치 제어를 할 방향과 힘 제어를 할 방향을 어떤 '선택 방식 (selection mechanism)'에 따라 서로 분리한 뒤, 각각의 위치 또는 힘 제어를 별개의 제어로 보고 독립적으로 수행하는 것이다. 즉,  $n$ -dof의 로봇 매니퓰레이터에 대하여,  $m$ -dof에 대하여는 위치 제어를 수행하고,  $l$ -dof에 대하여는 힘 제어를 수행하도록 하는 것이다. (단,  $m+l \leq n$ ).

다음은 이러한 하이브리드 위치/힘 제어에 필요한 몇 가지 이론들을 정리한 것이다 [2,3]. 먼저,  $n$ -link 로봇이 접촉면에 의하여 그 움직임이 구속될 때의 운동 방정식은 다음과 같이 주어진다 가정한다.

$$M(q)\ddot{q} + b(q, \dot{q}) + g(q) + f(\dot{q}) = \tau - J^T(q)F_c, \quad (1)$$

여기서,  $q$ 는 조인트 공간에서의  $n \times 1$  벡터이고,  $M(q)$ 는 조인트 공간에서의 관성 행렬,  $b(q, \dot{q})$ 는 원심력과 Coriolis 벡터,  $g(q)$ 는 중력 벡터,  $f(\dot{q})$ 는 마찰력 벡터이다. 또한  $J(q)$ 는 매니퓰레이터의 자코비안 행렬이고,  $\tau$ 는 토크 벡터, 그리고  $F_c$ 는 접촉 힘 (contact force) 벡터를 나타낸다.

이제 우리가 관심을 갖고 있는 것은 매니퓰레이터의 엔드 이펙터의 움직임과 그에 따른 접촉면과의 힘의 관계이므로, 로봇의 운동 방정식을 조인트 공간에서보다는 작업 공간 (task space) 상에서 표시하는 것이 더 자연스러울 것이다.  $x$ 를 작업 공간에서의  $n \times 1$  벡터라고 하면, 작업 공간상에서의 운동방정식은 아래의 식과 같이 표현될 수 있다.

$$\Lambda(x)\ddot{x} + \mu(x, \dot{x}) + p(x) + v(x, \dot{x}) = F - F_c, \quad (2)$$

여기서,  $\Lambda(x)$ 는 작업 공간상에서의 관성 행렬이고,  $F$ 는 엔드 이펙터에서의 힘 벡터이다. 조인트 공간상에서의 관계식들이 성립한다.

$$\begin{aligned} \Lambda(x) &= J^{-T}(q)M(q)J^{-1}(q) \\ \mu(x, \dot{x}) &= J^{-T}(q)b(q, \dot{q}) - \Lambda(x)\dot{J}(q)\dot{q} \\ p(x) + v(x, \dot{x}) &= J^{-T}(q)[g(q) + f(\dot{q})] \\ \tau &= J^T(q)F. \end{aligned} \quad (3)$$

접촉 힘은 아래와 같이 간단한 스프링으로 모델 된다고 가정한다.

$$F_c = K_s(x - x_0) \quad (4)$$

여기서,  $x_0$ 는 정지 상태에서의 평형점이고,  $K_s$ 는 탄성 계수이다. 단, 위와 같은 관계식은 매니퓰레이터가 접촉하였을

때에만 성립한다.

하이브리드 제어를 위해서는 그림 1에서 나타난 것과 같은 3개의 좌표축을 도입하는 것이 편리하다. 좌표 축  $\{B\}$ ,  $\{C\}$ , 그리고  $\{E\}$ 는 각각 로봇의 베이스 좌표 축 (base coordinate frame), 구속 좌표 축 (constraint coordinate frame), 엔드 이펙터 좌표 축 (end-effector coordinate frame)을 나타낸다. 로봇의 운동 방정식은  $\{B\}$ 에서 표현되고, 구속 면의 기하학적인 표현과 위치 및 힘 제어의 방향 선택은  $\{C\}$ 에서, 그리고 엔드 이펙터에 장착된 힘 센서에 의한 힘의 측정은  $\{E\}$ 에서 각각 나타내어지게 된다. 여기서, 좌표 축  $\{B\}$ 와  $\{C\}$ 는 고정되어 있으며, 좌표 축  $\{C\}$ 는 좌표 축  $\{B\}$ 을  $R_B^C$ 에 의하여 회전 변환 (rotation transform)시키면 얻어 질 수 있다고 가정한다.

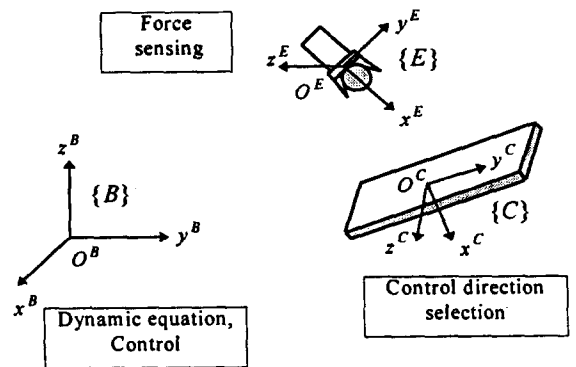


그림 1. 좌표 축:  $\{B\}$ ,  $\{C\}$ , 그리고  $\{E\}$ .

Fig. 1. Coordinate frames:  $\{B\}$ ,  $\{C\}$ , and  $\{E\}$  denote the base, constraint, end-effector coordinate frames, respectively.

다음, 힘 및 위치 제어를 위한 각각의 방향을 선택하여 보자. 좌표 축  $\{C\}$ 에서 '위치 결정 행렬 (position specification matrix)'은 다음과 같이 나타내어진다 [3].

$$\Sigma_F = \begin{pmatrix} \sigma_{F_x} & 0 & 0 \\ 0 & \sigma_{F_y} & 0 \\ 0 & 0 & \sigma_{F_z} \end{pmatrix} \quad (5)$$

여기서,  $\sigma_{F_x}, \sigma_{F_y}, \sigma_{F_z}$ 는 모두 2진수로서 각각  $x^C, y^C, z^C$ 축을 따라 위치 제어를 하게 되면 그 값으로 1을 갖고, 그렇지 않은 경우 (즉, 힘 제어)이면 그 값으로 0을 갖는다. 이 경우 '힘 결정 행렬 (force specification matrix)'은 다음과 같이 결정된다.

$$\overline{\Sigma}_F = I_3 - \Sigma_F. \quad (6)$$

여기서,  $I_3$ 는  $3 \times 3$  단위 행렬 (identity matrix)을 나타낸다.

(5)와 (6)과 유사한 방식으로 '회전 결정 행렬 (rotation specification matrix)'과 '모멘트 결정 행렬 (moment specification matrix)'은 아래와 같이 결정되어 진다.

$$\begin{aligned} \Sigma_M &= \text{diag}[\sigma_{M_x}, \sigma_{M_y}, \sigma_{M_z}] \\ \overline{\Sigma}_M &= I_3 - \Sigma_M. \end{aligned} \quad (7)$$

여기서,  $\sigma_{M_x}, \sigma_{M_y}, \sigma_{M_z}$ 는 모두 2진수로서 각각  $x^C, y^C, z^C$ 축을 따라 자유롭게 회전 (free rotation)할 수 있으면 그 값으로 1을 갖고, 그렇지 않은 경우이면 0을 갖는다. 지금까지의 결정행렬들은 모두 좌표 축  $\{C\}$ 에서 선택된 것들이다. 그러나 실제로 로봇의 운동 방정식은 좌표 축  $\{B\}$

에서 표현되므로, 이들 결정 행렬들 {C}에서 {B} 변환시킬 필요가 있다. 이와 같이 변환된 결정 행렬을 '일반화된 작업 결정 행렬 (generalized task specification matrix)'라고 부르며, 위치/힘을 결정하는 행렬  $\Omega$ 와 오리엔테이션/모멘트를 결정하는 행렬  $\bar{\Omega}$ 는 각각 다음과 같다.

$$\Omega = \begin{pmatrix} R_C^B \Sigma_F R_B^C & 0 \\ 0 & R_C^B \Sigma_M R_B^C \end{pmatrix} \quad (8)$$

$$\bar{\Omega} = I_6 - \Omega.$$

위의 식이 나타내는 것은, 일단 제어 입력이 처음에 좌표 축 {B}에 근거하여 주어지면, 이것은 회전 변환  $R_B^C$ 에 의하여 좌표 축 {C}로 옮겨진다. 그리고  $\Sigma_F$ 에 의하여 위치 제어 방향이 결정되고, 다시 회전 변환  $R_C^B$ 에 의하여 처음의 좌표 축 {B}로 다시 돌려 보내지게 되는 것을 나타낸다. 즉, 다시 말하면, 좌표 축 {C}에서 결정된 위치 또는 힘 제어의 방향이 결국 제어를 수행하게 되는 좌표축인 {B}에서 어떻게 결정되는 가를 나타내는 것이다.

### III. 하이브리드 위치/힘 제어

주어진 기준 위치 벡터,  $x_d$ 와 기준 힘 벡터,  $F_d$ 에 대하여 각각의 오차를  $e_x = x_d - x$  그리고  $e_F = F_d - F$ 와 같이 정의한다. 이 경우, 우리의 제어 목적은  $e_x$ 와  $e_F$ 를 모두  $e_x \rightarrow 0$ ,  $e_F \rightarrow 0$ 가 되도록 만드는 것이다.

만약, 로봇과 접촉면의 운동방정식을 모두 정확하게 알고 있다면, 이러한 제어 목적은 아래의 제어 입력을 가하여 줌으로써 달성될 수 있을 것이다.

$$F = F_b + F_f, \quad (9)$$

여기서,  $F_b = \Lambda(x)\Omega F_b^* + \mu(x, \dot{x}) + p(x) + v(x, \dot{x})$  이고,  $F_f = \Lambda(x)\bar{\Omega}F_f^* + F_c$ 이다. 그러면, 페루프 시스템은 다음의 두개의 시스템으로 나뉘어 진다.

$$\begin{aligned} \Omega \dot{x} &= \Omega F_b^* \\ \bar{\Omega} \dot{x} &= \bar{\Omega} F_f^*. \end{aligned} \quad (10)$$

이 경우에,  $F_b^*$ 와  $F_f^*$ 를 다음과 같이 선택하면,

$$\begin{aligned} F_b^* &= \ddot{x}_d + k_1 \dot{e}_x + k_2 e_x \\ F_f^* &= k_3 e_F + k_4 \int_0^t e_F dt, \end{aligned} \quad (11)$$

(10)의 페루프 시스템의 오차 방정식은 아래와 같이 되어,

$$\begin{aligned} \Omega [\ddot{e}_x + k_1 \dot{e}_x + k_2 e_x] &= 0 \\ \bar{\Omega} [k_3 e_F + k_4 \int_0^t e_F dt] &= 0, \end{aligned} \quad (12)$$

이고 상수  $k_1, \dots, k_4$ 를 적당히 선택하여 주면, 우리가 원하는 제어 목적  $e_x \rightarrow 0$ ,  $e_F \rightarrow 0$ 을 얻을 수 있게 된다.

그러나, 실제로는 모델에 대한 정확한 값을 알지 못하여 추정 값만을 사용하고 또한 모델링에 포함되어 있지 않은 불확실성으로 인하여 (12)의 결과는 얻어지지 않는다. 이와 같이 추정 값만을 사용하는 경우에는

$$\begin{aligned} F_b &= \hat{\Lambda}(x)\Omega F_b^* + \hat{\mu}(x, \dot{x}) + \hat{p}(x) \\ F_f &= \hat{\Lambda}(x)\bar{\Omega}F_f^* + F_s, \end{aligned} \quad (13)$$

위와 같이 되어, 페루프 시스템의 오차 방정식은 아래와 같이 불확실성을 포함하게 된다.

$$\begin{aligned} \Omega [\ddot{e}_x + k_1 \dot{e}_x + k_2 e_x] &= \Omega \Delta U \\ \bar{\Omega} [k_3 e_F + k_4 \int_0^t e_F dt] &= \bar{\Omega} \Delta U, \end{aligned} \quad (14)$$

단,  $\Delta U = \hat{\Lambda}^{-1}(\Delta\Lambda \dot{x} + \Delta\mu + \Delta p + \Delta F + v)$  이고,  $\Delta\Lambda = \Lambda - \hat{\Lambda}$ ,  $\Delta\mu = \mu - \hat{\mu}$ ,  $\Delta p = p - \hat{p}$ ,  $\Delta F = F_c - F_s$ 이다. 결국, 불확실한 항들이 (14)에서와 같이 존재하면, 원하는 제어 목적은 달성할 수 없게 된다. 따라서 다음 장에서는 신경회로망을 이용하여 이러한 불확실성을 극복할 수 있는 제어 방식을 제안할 것이다.

### IV. 신경회로망을 이용한 하이브리드 위치/힘 제어

본 장에서는 신경회로망을 도입하여 (14)에서의 불확실성을 극복하고, 최종적으로 페루프 시스템의 오차 방정식을 (12)와 같이 만드는 제어 방식을 제안한다. 먼저 그러한 목적을 달성하기 위하여 제어 구조에는 신경회로망 제어기가 추가된다. 즉,

$$F = F_b + F_f + F_{nn}, \quad (15)$$

여기서,  $F_{nn}$ 은 신경회로망 제어기이다. 사용된 신경회로망 제어기의 구조는 3층 퍼셉트론 (three-layer perceptron) 구조를 가지며, 첫 번째와 두 번째 층은 은닉층으로 비선형 시그모이드 함수,  $f(x) = \tanh(x)$ 를 활성 함수 (activation function)으로 취하며, 마지막 출력 층에서는 선형 함수,  $f(x) = x$ 를 활성 함수로 취한다.

이때 (15)와 같은 제어 입력을 (2)의 로봇 시스템에 인가하면, 페루프 시스템은 다음과 같이 표현될 수 있다.

$$\Omega \dot{x} + \bar{\Omega} \dot{x} + \Delta U = \Omega F_b^* + \bar{\Omega} F_f^* + \hat{\Lambda}^{-1} F_{nn} \quad (16)$$

위로부터, 만약 신경회로망이 불확실성을 학습할 수 있다면, 즉,  $\hat{\Lambda}^{-1} F_{nn} \rightarrow \Delta U$  또는  $F_{nn} \rightarrow \hat{\Lambda} \Delta U$ 이 될 수 있다면, 시스템의 오차 방정식은 결국 (12)와 같이 되어 원하는 제어 목적을 달성할 수 있게 된다. 따라서 신경회로망은 다음의 비용 함수 (cost function)을 최소화하도록 학습시키는 것이 타당할 것이다.

$$E = \frac{1}{2} \| F_{nn} - \hat{\Lambda} \Delta U \|^2. \quad (17)$$

학습 법칙의 유도를 위하여 신경회로망의 각 가중치 (weight),  $w_j$ 에 대한 (17)의 그레디언트 (gradient)를 취하면,

$$\frac{\partial E}{\partial w_j} = (F_{nn} - \hat{\Lambda} \Delta U) \frac{\partial F_{nn}}{\partial w_j}, \quad (18)$$

따라서, 신경회로망의 학습 법칙은 아래와 같이 얻어진다.

$$\Delta w_j = -\eta \frac{\partial E}{\partial w_j} = -\eta (F_{nn} - \hat{\Lambda} \Delta U) \frac{\partial F_{nn}}{\partial w_j}, \quad (19)$$

여기서,  $\Delta w_j = w_j(t+1) - w_j(t)$ 이며,  $\eta$ 는 신경회로망의 학습률로서 0보다 큰 상수이다.

한편, (13)을 (16)에 대입하면 다음의 관계를 얻을 수 있고,

$$\begin{aligned}
 F_{nn} - \widehat{\Lambda} \Delta U &= \widehat{\Lambda} \Omega (\dot{x} - F_p^*) + \widehat{\Lambda} \overline{\Omega} (\dot{x} - F_f^*) \\
 &= -\widehat{\Lambda} \left[ \Omega (e_x + k_1 e_x + k_2 e_x) \right. \\
 &\quad \left. + \overline{\Omega} (k_3 e_F + k_4 \int_0^t e_F dt) \right], \tag{20}
 \end{aligned}$$

따라서, 위의 관계로부터, 최종적인 신경회로망의 학습법칙은 다음과 같이 구해진다.

$$\begin{aligned}
 \Delta w_j &= \eta \widehat{\Lambda} \left[ \Omega (e_x + k_1 e_x + k_2 e_x) \right. \\
 &\quad \left. + \overline{\Omega} (k_3 e_F + k_4 \int_0^t e_F dt) \right] \frac{\partial F_{nn}}{\partial w_j} \tag{21}
 \end{aligned}$$

위와 같은 신경회로망의 학습 법칙은 다음과 같은 장점들이 있다.

- ◆ 로봇 매니퓰레이터의 위치 제어뿐만 아니라 위치/힘 제어를 동시에 수행할 수 있다.
- ◆ 신경회로망의 학습을 위하여 신경회로망의 기준 출력 값 (desired output)이 필요 없고, 시스템의 오차 방정식만으로 학습이 가능하다.
- ◆ 플랜트를 통한 오차 역전달 (error back propagation through the plant) [13]이나, 플랜트의 모사기 (identifier)를 통한 역전달 [12]등의 과정을 필요로 하지 않는다.
- ◆ 제어와 학습이 동시에 이루어진다.

본 장에서 제안된 신경회로망을 이용한 하이브리드 위치/힘 제어의 구조 및 학습 방식을 그림 2에 나타내었다.

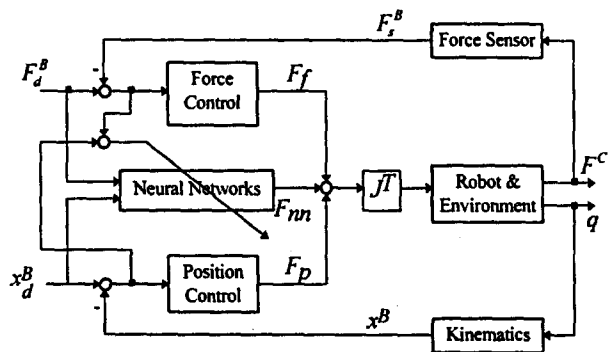


그림 2. 신경회로망을 이용한 하이브리드 위치/힘 제어의 구조.

Fig. 2. Neural networks for hybrid position/force control.

**V. 시뮬레이션**

시뮬레이션 대상 시스템으로는 6축 PUMA 로봇 매니퓰레이터가 사용된다. 이 로봇 시스템의 마찰력, 중력, Coriolis 힘 등을 포함한 자세한 운동 방정식은 [19,20]에서 구할 수 있다. 단, 로봇이 구속 면과 접촉하였을 때의 운동 방정식은 (2)와 같고 접촉 힘은 (4)와 같다고 가정한다.

본 시뮬레이션에서는 로봇의 위치, 오리엔테이션, 접촉 힘의 제어를 다루게 된다. 먼저, 로봇의 작업은 좌표 축 (C)에 대하여 결정된다. 매니퓰레이터는 \$x^C, y^C\$ 축 방향으로 자유롭게 움직일 수 있으며, \$z^C\$ 방향으로 힘을 가하며, \$x^C, y^C, z^C\$ 축 방향으로 자유롭게 회전할 수 있도록 작업이 결정된다. 즉, \$x^C, y^C\$ 방향으로 위치 제어를, \$z^C\$ 방향으로 힘 제어를, 그리고 \$x^C, y^C, z^C\$ 방향으로 오리엔테이션 제어를 함을 나타낸다. 이때의 \$y^B-z^B\$ 평면상에서 로봇의

작업을 그림 3에 나타내었다. 이 경우 위치와 회전 결정 행렬은 아래와 같이 표현된다.

$$\Sigma_F = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \Sigma_M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{22}$$

그리고 구속 좌표 축 {C}은 베이스 좌표 축 {B}로부터 다음의 회전 변환, \$R\_B^C\$를 하여 얻어질 수 있다고 가정한다.

$$R_B^C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \tag{23}$$

여기서, \$\alpha\$는 구속 면의 기울기를 나타낸다(그림 3). 추적하고자 하는 기준 위치와 오리엔테이션 (desired position and orientation)은 다음의 T행렬로 주어진다.

$$\begin{aligned}
 T_d(t) &= \begin{pmatrix} n_d(t) & o_d(t) & a_d(t) & p_d(t) \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} -1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & y_0 + r_0 \cos(\pi t) \\ 0 & 0 & -1 & z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{24}
 \end{aligned}$$

여기서, \$[n\_d(t) \ o\_d(t) \ a\_d(t)]\$는 매니퓰레이터의 오리엔테이션을, \$p\_d(t)\$는 위치를 각각 나타내며, \$x\_0=0.1, y\_0=0.5, z\_0=0\$ 그리고 \$r\_0=0.1\$의 값을 갖는다. 식(24)에서 나타내는 오리엔테이션은 로봇의 엔드 이펙터와 구속 면이 서로 수직을 이룸을 나타낸다(그림 3). 또한 기준 힘 (desired force)는 \$z^C\$를 따라 인가되도록, \$F\_d^C = [0 \ 0 \ f\_d]^T\$와 같이 설정되고, 이때 \$f\_d\$는 아래의 2차 시스템에 의하여 생성되어진다.

$$f_d + 6\dot{f}_d + 9f_d = 90. \tag{25}$$

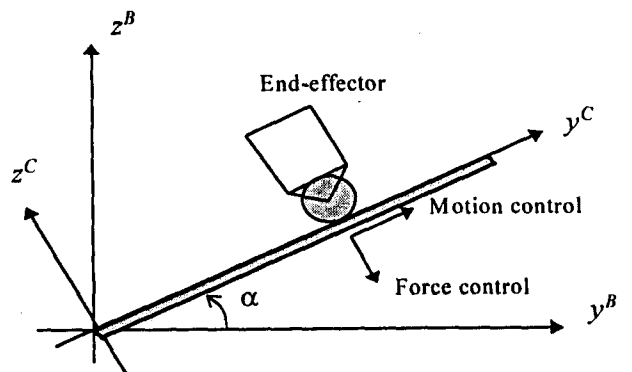


그림 3. 시뮬레이션에서 사용된 좌표축과 위치 및 힘 제어 방향.

Fig. 3. Coordinate frames and control directions.

위에서 설정된 제어 목적을 달성하기 위하여 (15)의 제어 입력이 가하여 지며, 이때

$$\begin{aligned}
 F_p &= \widehat{\Lambda}(x) \Omega [\ddot{x}_d + 20\dot{e}_x + 100e_x] \\
 F_f &= \widehat{\Lambda}(x) \overline{\Omega} \left[ 3.1e_F + 1.5 \int_0^t e_F dt \right] + F_s, \tag{26}
 \end{aligned}$$

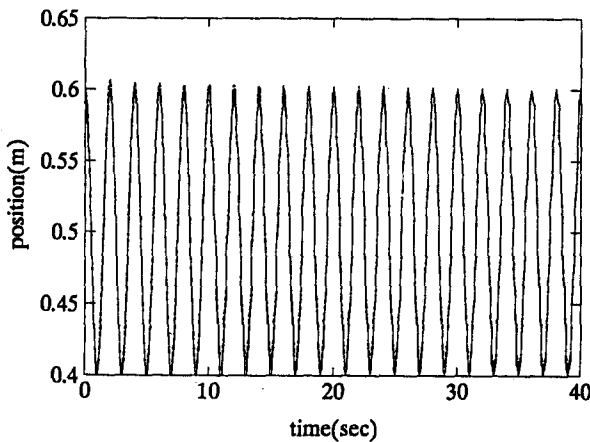
이며, \$F\_{nn}\$은 3층 신경회로망으로서 입력층, 첫 번째 은닉층, 두 번째 은닉층, 출력층의 노드의 수가 각각 15-20-12-6과 같으며, 학습률은 \$\eta=0.005\$가 사용되었다. 그리고 신경회로망의 입력으로는 \$p\_d^B, \dot{x}\_d^B, \ddot{x}\_d^B\$가 사용되고, 신경회로망의 가중치들은 학습을 시작하기 전에 \$(-0.5, 0.5)\$의

범위에서 랜덤하게 초기화되었다.

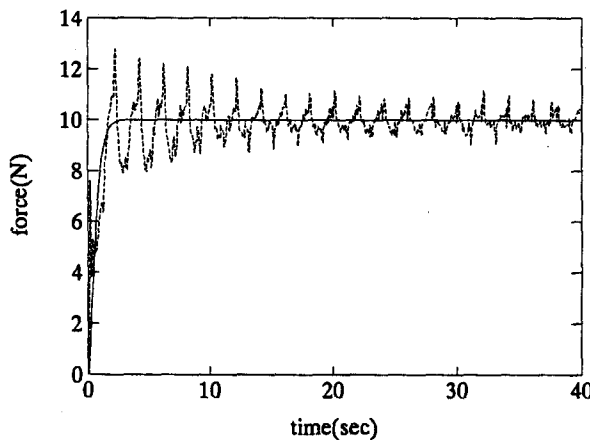
여기에서는 제어를 설계할 때에 로봇과 환경에 대하여 알고 있는 정보가 없다고 가정하여,  $\hat{M} = I$ 로 놓았으므로,  $\hat{A} = J^T J^{-1}$ 이 되고, 매니퓰레이터의 원심력, Coriolis 힘, 마찰력, 중력 등은 모두 모른다고 가정하였다. 단, 힘 센서는 접촉 힘을 정확하게 측정한다고 가정하였다, 즉  $F_s = F_c$ . 따라서, 각각의 조인트에 가하여지는 토크는 아래와 같이 표현될 수 있다.

$$\begin{aligned} \tau &= J^T F \\ &= J^{-1} (\Omega F_p + \bar{Q} F_f) + J^T F_{nn}. \end{aligned} \quad (27)$$

다음의 그림들은 시뮬레이션 결과를 나타낸다. 그림 4와 5은 신경회로망을 이용하여 하이브리드 위치/힘 제어를 수행한 결과이다. 그림 4에서는 위치와 힘의 기준 값 및 실제 응답 값을 나타내었고, 그림 5에서는 위치 및 오리엔테이션의 추적 오차를 나타내었다. 여기서 오리엔테이션 오차는  $e_o = 0.5(n \times n_d + o \times o_d + a \times a_d)$  [19]를 이용하여 계산하였다. 그림의 결과들로부터 신경회로망이 점점 더 시스템에 대하여 학습을 진행해 나감에 따라 제어 성능도 점점 더 좋아지고 있음을 볼 수 있다.



(a)  $y^c$  방향 위치: 기준 위치 (실선), 실제 위치 (점선)



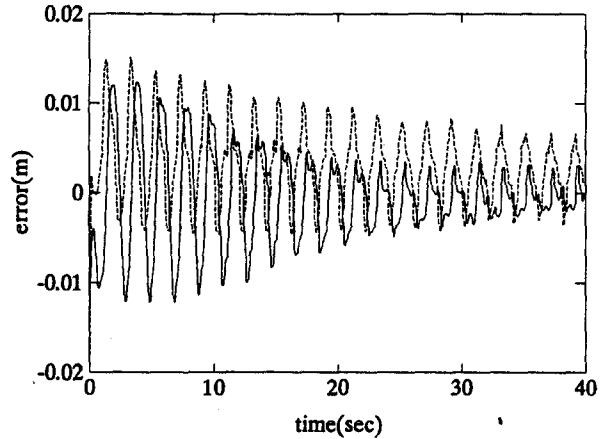
(b)  $z^c$  방향 힘: 기준 힘 (실선), 실제 힘 (점선)

그림 4. 신경회로망을 사용한 경우의 기준 및 실제 응답 특성.

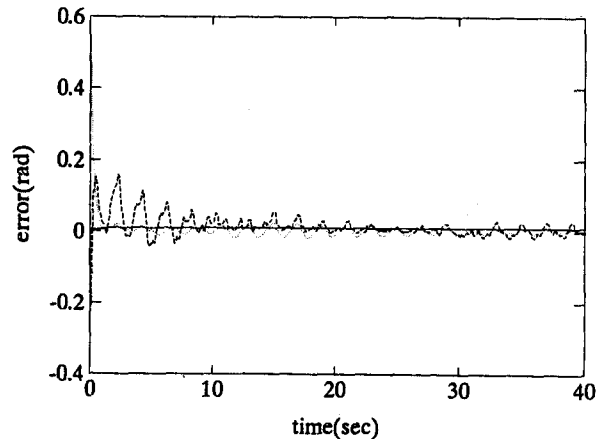
Fig. 4. Desired and actual responses using neural networks.

아울러 성능의 비교를 위하여 그림 6과 7에서는 신경회로

망을 사용하지 않고 PD와 PI 제어기만으로 하이브리드 위치/힘 제어를 수행한 결과를 나타내었다. 그림 6에서는 PD+PI 제어기의 위치 및 오리엔테이션 오차를 나타내었고, 그림 7에서는 힘 오차에 대하여 PD+PI 제어기와 신경회로망을 사용하였을 때를 비교하였다. PD+PI 제어기만으로는 시스템의 불확실성을 보상할 수 없기 때문에 위치오차는 물론, 오리엔테이션과 힘 오차도 모두 줄어들지 않음을 보이고 있다. 이에 비하여 신경회로망을 이용한 제어의 경우는 이러한 불확실성들이 극복되어 좋은 성능을 보임을 알 수 있다 (그림 4,5).



(a) 위치 오차:  $x_e$  (실선),  $y_e$  (점선)



(b) 오리엔테이션 오차  $e_o = 0.5(n \times n_d + o \times o_d + a \times a_d)$

그림 5. 신경회로망을 사용한 경우의 오차의 추이.

Fig. 5. Transition of errors using neural networks.

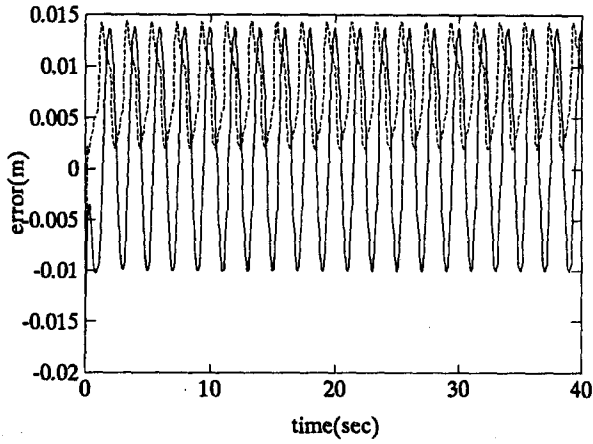
### VI. 고찰

지금까지 본 논문에서는 힘 제어의 방향과 위치 제어의 방향이 분리되어 있다는 가정 하에서의 하이브리드 위치/힘 제어 방식을 제안하여 왔다. 그러나 힘 제어 문제에서 항상 이 두 방향이 서로 분리되는 것은 아니므로, 이번 장에서는 힘 제어 방향과 위치 제어 방향이 서로 분리되지 않은 경우에 있어서 제안된 제어 방식의 효율성을 고찰하여 보겠다.

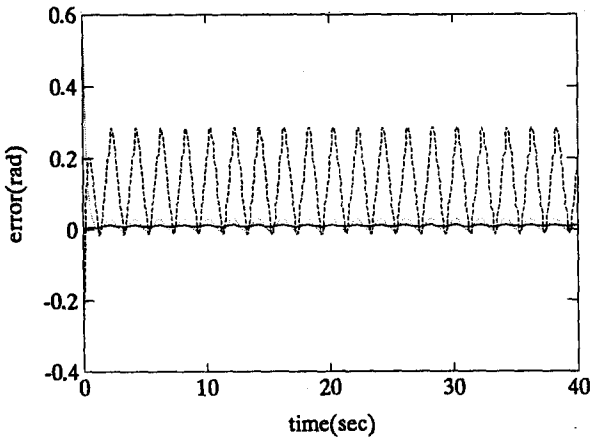
우선 (4)의 접촉 힘에 대하여 다시 살펴보자. 접촉 힘은 매니퓰레이터와 구속 면이 서로 접촉하였을 때만 성립한다는 것은 이미 앞장에서도 기술한 바와 같다. 추가로 매끄러운 접촉면을 가정한다면 (즉, 접촉면과 접선 방향으로는 마찰력이 작용하지 않는다고 가정하면), 접촉 힘은 구속 면과 수직방향으로만 작용하게 된다. 따라서 구속 면과 수직인 단위 벡터를  $n$ 이라고 한다면, 접촉 힘은 다음과 같이 쓰여질 수 있게 된다[21].

$$F_c = K_s(x-x_0) = k nn^T(x-x_0) \quad (27)$$

여기서,  $k$ 는 단위 벡터  $n$  방향으로의 탄성 계수이다.



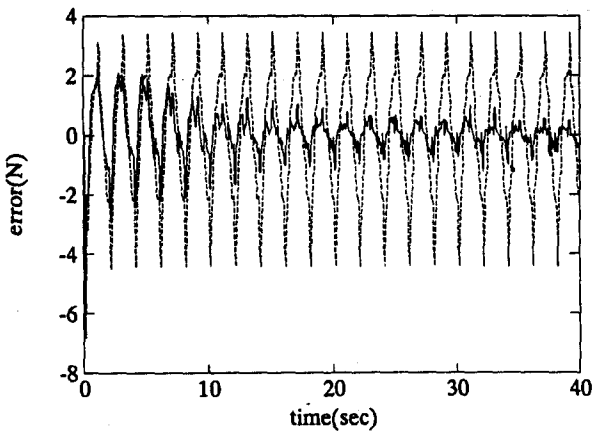
(a) 위치 오차:  $x_e$  (실선),  $y_e$  (점선)



(b) 오리엔테이션 오차,  $e_o = 0.5(n \times n_d + o \times o_d + a \times a_d)$

그림 6. 신경회로망을 사용하지 않은 경우의 오차의 추이 (PD+PI만 사용).

Fig. 6. Transition of errors without neural networks (PD+PI only).



실선:  $F = F_p + F_f + F_{nn}$ , 점선:  $F = F_p + F_f$

그림 7. 신경회로망을 사용한 경우와 사용하지 않은 경우의 힘 오차의 추이.

Fig. 7. Transition of force errors with and without neural networks.

이제 힘 제어 방향과 위치 제어 방향이 서로 완전히 분리 되지 않는 경우를 생각하여 보자. 지금까지는 힘 제어 방향

은 구속면에 수직 방향 (결국 벡터  $n$  방향)으로, 위치 제어 방향은 구속면에 접선 방향으로 분리된다고 가정하였으나, 이제는 위치 제어 방향이 임의로 주어진다고 가정한다. 또한 힘과 위치의 기준 값은 각각  $F_d$ 와  $x_d$ 로 주어진다고 가정한다. 이 경우,  $x_d$ 는 구속 면의 접선 방향 성분으로만 구성되어 있지 않으므로 (27)에 의하여 구속 힘이 발생한다. 앞으로의 전개를 쉽게 하기 위하여 이때 발생한 구속 힘을  $F_d$ 라고 가정한다(그림 8).

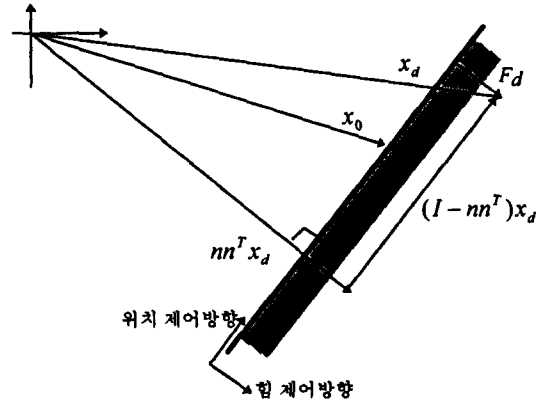


그림 8. 기준 힘과 기준 위치가 서로 분리되지 않는 경우의 하이브리드 위치/힘 제어.

Fig. 8. Hybrid position/force control when the desired position and the desired force are not decoupled.

이와 같이 힘과 위치 제어의 방향이 서로 분리되지 않았음에도 분리하고, 제안된 하이브리드 위치/힘 제어 방식을 사용한다고 하자. 제안된 방식은 우선 '위치 및 힘 결정 행렬'을 통하여 힘 및 위치 제어를 위한 방향을 선택하게 되어 있으므로, 각각의 제어 방향을 그림 8의 아래에서 표시한 것과 같이 선택한다. 그러면, 힘 제어 방향은  $F_d$ 의 방향과 일치하게 되나, 위치 제어 방향은  $x_d$ 의 방향과 어긋나게 된다. 따라서 위치 제어 방향과 기준 위치 (desired position)의 방향이 어긋나게 되므로, 기준 위치에 대하여는 완벽한 추적 제어가 이루어 질 수 없음을 알 수 있다. 실제로 추적 제어가 되는 부분은 구속면과 접선 방향인 위치 제어 방향으로만 이루어지게 되므로 추적 제어는 결국  $x_d$ 에 대하여 이루어지는 것이 아니라 위치 제어 방향으로 투영한 벡터 (projection vector),  $(I - nn^T)x_d$ 를 따라가도록 제어가 이루어지는 것이다.

지금까지의 결과를 정리하면, 위치와 힘 제어 방향이 서로 분리되지 않는 경우에는 주어진 위치와 힘에 대한 추적 제어는 '위치 및 힘 제어 결정 행렬'에 의하여 선택된 방향으로 투영한 성분에 대하여만 제어가 수행되고 나머지 성분들에 대하여는 제어를 수행하지 않게 된다.

### VII. 결론

본 논문에서는 2장에서 제시된 힘 제어의 기본이론에 따라 신경회로망을 이용한 하이브리드 위치/힘 제어 방식을 제안하였다. 제안된 제어 방식은 하이브리드 제어 방식에 어떻게 신경회로망을 이용할 수 있을 것인가에 초점을 맞추었고, 이에 따라 하이브리드 위치/힘 제어에 적합한 신경회로망의 제어 구조 및 학습 법칙을 유도하여 내었다. 결국, 본 논문에서의 신경회로망은 매니플레이터의 불확실성을 극복하기 위하여 도입된 것이며, 제안된 학습 법칙은 시스템

의 오차 방정식만으로도 학습이 가능하며, 제어와 학습이 동시에 수행되는 장점이 있다. 그리고 제안된 제어 방식과 학습 법칙에 따라 6축 로봇 매니플레이터에 시뮬레이션하고 PD+PI 제어기와 비교하여 그 성능의 우수성을 검증하였다.

**참고문헌**

[1] N. Hogan, "Impedance control: An approach to manipulation: Part I - Theory, Part II - Implementation, Part III - Applications," *Trans. ASME: J. of Dynamic Systems, Measurement, and Control*, vol. 107, pp. 1-24, 1985.

[2] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *Trans. ASME: J. of Dynamic Systems, Measurement, and Control*, vol. 102, pp. 101-133, 1981.

[3] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. Robotics and Automation*, vol. 3, no. 1, pp. 43-53, 1987.

[4] T. Yoshikawa, "Dynamic hybrid position/force control of robot manipulators - Description of hand constraints and calculation of joint driving forces," *IEEE J. Robotics and Automation*, vol. 3, no. 5, pp. 386-392, 1987.

[5] Y. H. Chen and S. Pandey, "Robust hybrid control of robot manipulators," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 236-241, 1989.

[6] J. J. Craig, *Introduction to robotics: Mechanics and control*, 2nd Ed., Addison-Wesley Publishing Company, Inc, 1989.

[7] N. H. McClamroch and D. Wang, "Feedback stabilization and tracking of constrained robots," *IEEE Trans. Automatic Control*, vol. 33, no. 5, pp. 419-426, 1988.

[8] C.-Y. Su and Y. Stepanenko, "Adaptive variable structure tracking control for constrained robots," *IEEE Trans. Aerospace and Electronic Systems*, vol. 30, no. 2, pp. 493-503, 1994.

[9] D. E. Whitney, "Historical perspective and state of the art in robot control," *Int. J. of Robotics Research*, vol. 6, no. 1, pp. 3-14, 1987.

[10] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, Englewood Cliffs, NJ, Prentice Hall, 1991.

[11] K. J. Hunt, D. Sbarbaro, R. Zbikowski, and P. J. Gawthrop, "Neural networks for control systems - A survey," *Automatica*, vol. 28, no. 6, 1083-1112, 1992.

[12] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 4-27, 1990.

[13] D. Psaltis, A. Sideris, and A. A. Yamamura, "A multilayered neural network controller," *IEEE Control Systems Magazine*, pp. 17-21, Apr., 1988.

[14] T. Fukuda, T. Shibata, M. Tokita, and T. Mitsuoka, "Neuromorphic control: Adaptation and learning," *IEEE Trans. Industrial Electronics*, vol. 39, no. 6, pp. 497-503, 1992.

[15] M. Kawato, Y. Uno, M. Isobe, and R. Suzuki, "Hierarchical neural network model for voluntary movement with application to robotics," *IEEE Control Systems Magazine*, pp. 8-16, Apr., 1988.

[16] A. Ishiguro, T. Furuhashi, S. Okuma, and Y. Uchikawa, "A neural network compensator for uncertainties of robotics manipulators," *IEEE Trans. Industrial Electronics*, vol. 39, no. 6, pp. 565-570, 1992.

[17] M. Saad, P. Bigras, L.-A. Dessaint, and K. Al-Haddad, "Adaptive robot control using neural networks," *IEEE Trans. Industrial Electronics*, vol. 41, no. 2, pp. 173-181, 1994.

[18] 김성우·이주장, "불확실성이 있는 로봇 시스템의 역모델 학습에 의한 신경회로망 제어," 제어·자동화·시스템 공학회 논문지, 1권 2호, pp. 88-93, 1995.

[19] F. L. Lewis, C. T. Abdallah, and D. M. Dawson, *Control of robot manipulators*, Macmillan Publishing Co., New York, 1993.

[20] 김성우, "불확실한 로봇 시스템의 안정성을 보장하는 신경회로망의 학습 및 제어," 박사학위논문, 한국과학기술원 전기 및 전자공학과, 1996.

[21] S. Chiaverini and L. Sciavicco, "The parallel approach to force/position control of robotic manipulators," *IEEE Trans. Robotics and Automation*, vol. 9, no. 4, pp. 361-373, Aug., 1993.



**김 성 우**

1990년 한국과학기술원 전기 및 전자공학과 졸업 (공학사). 1996년 한국과학기술원 전기 및 전자공학과 졸업 (공학박사). 1996년 ~ 현재 현대전자산업(주) 생산기술연구소 선임연구원. 관심분야는 로봇 시스템, 신경회로망

제어, FA, 생산기술.



**이 주 장**

1948년 11월 14일 서울출생. 1973년 서울대학교 전기공학과 졸업. 동대학원 석사(1978). 미국 위스콘신대학 전기과 박사(1984). 현재 한국과학기술원 전기 및 전자공학과 부교수(1996년). 관심분야는 가변구조제어, 지능제어, 이동로봇.