

# 고정밀 CNC 머신을 위한 신경망 윤곽제어

## A Neuro-contouring Controller for High-precision CNC Machine Tools

이현철\* · 주정홍 · 전기준\*

Hyun Chul Lee\*, Jeong Hong Joo and Gi Joon Jeon\*

경북대학교 전자전기공학부,  
\*경북대학교 전자전기공학부, ERC-ACI

### 요 약

CNC 공작기계의 두 서보축을 대상으로 가공 정밀도를 향상시키기 위한 신경망 윤곽제어 알고리즘을 제안한다. 이 연구에서는 두 축 상호간에 미치는 영향을 신경망의 학습 능력을 이용하여 보상하고자 한다. 윤곽제어를 위해서는 매 샘플링 주기마다 윤곽오차를 계산하여야 하나, 윤곽오차는 직선경로를 이동하는 경우 쉽게 계산가능하나 원호, 인볼루트곡선등 비선형 경로를 가공하는 경우에는 정확하게 계산하기가 힘들다. 먼저 이 논문에서는 임의의 비선형 곡선경로에 대하여도 윤곽오차를 정확히 구해낼 수 있는 새로운 윤곽오차 모델링 방법을 제안한다. 또한 이러한 윤곽오차에 대한 항을 포함하는 성능지수를 정의하고, 신경망 윤곽제어를 위한 온라인 학습법칙을 유도한다. 이러한 신경망윤곽제어의 사용으로 시스템이 비선형 특성을 가지거나 외부 환경이 변화하는 경우에도 좋은 윤곽제어 성능을 유지 할 수 있다.

### ABSTRACT

In this paper, a neuro-contouring control scheme for the high precision machining of CNC machine tools is described. The proposed control system consists of a conventional controller for each axis and an additional neuro-controller. For contouring control, the contour error must be computed during real-time motion, but generally the contour error for nonlinear contours is difficult to be directly computed. We, therefore, propose a new contour error model to approximate real error more exactly, and here we also introduce a cost function for better contouring performance and derive a learning law to adjust the weights of the neuro-controller. The derived learning law guarantees good contouring performance. Usefulness of the proposed control scheme is demonstrated by computer simulations.

### 1. 서 론

최근 모션 제어 분야에서 여러 개의 축을 동시에 고려하여 제어하는 문제에 대한 관심이 높아지고 있으며, CNC 공작기계의 경우 이는 바로 공작물의 가공 정밀도와 직접 관련이 된다[1]. CNC 공작기계를 이용한 가공작업은 2차원, 또는 3차원 공간상의 연속된 궤적을 따라 공구를 이동시킴으로써 이루어지며, 실제 공구의 이동 경로가 원하는 궤적을 얼마나 정확히 추종되도록 제어하느냐 하는 것이 공작물의 가공 정밀도를 결정하게 된다. 기존의 다축의 CNC 공작기계의 경우 각 축은 완전히 분리된 형태로 각각 독립적으로 제어되고, 그 제어기는 각 축의 추적성능을 고려하여 설계된다[2]. 그러나 이러한 분리된 형태의 제어 방법으로는 제어기 설계 과정에서 축 상호간에 미

치는 영향에 대한 정확한 고려가 없다면 좋은 윤곽제어 성능을 기대할 수 없으며, 서로 다른 드라이브 동특성, 외부 부하 외란, 시스템 파라미터의 변화, 시스템의 비선형성 등에 의해 필연적으로 윤곽오차가 발생하게 된다. CNC 시스템의 경우 윤곽오차가 가공 정밀도를 결정하는 중요한 특성이 되고, 이를 고려하는 제어 방법이 필연적으로 요구된다.

지금까지 이러한 윤곽오차를 줄이기 위한 여러 가지 서보 제어 알고리즘이 연구되고 있다. Tomizuka [3]는 완벽한 추적 특성을 위하여 영위상 오차 추적제어기(ZPETC)를 제안하였고, Weck[4]은 순방향제어기에 저역필터를 부가함으로써 모서리 추적성능을 향상시키고자 하였다. 이러한 순방향 제어기법은 각 축의 추적오차를 줄여 나감으로써 결과적으로 윤곽오차를 줄이고자 하는 방법으로 부하 외란에 약하고

모델링 오차나 입력의 제한 정도에 따라 성능의 변화가 심하다. 또한 비선형 가공 등의 경우에 있어서는 각 축의 추적오차를 감소시키는 것이 바로 윤곽오차의 감소로 나타나지 않을 수 있다는 문제점이 있다. Koren[5]은 각 축의 추적제어기와는 별도로 윤곽오차를 고려하기 위한 상호결합제어기(cross coupled controller)를 부가하는 제어방법을 제안하였고, 그 후 여러 개의 축을 동시에 고려하여 제어하는 문제에 대한 관심이 높아지고 있다[6,7]. 이는 각 축의 추적오차와는 별도로 직접적으로 윤곽오차를 줄이거나 제거시키는 방법이다. 그러나 이 경우 제어 구조가 복잡해짐에 따라 제어기 파라미터를 결정하기 위한 작업의 어려움이 예상되고, 시스템이 가지는 비선형 특성과 제어환경의 변화에 따른 성능저하가 나타난다.

최근 비선형 특성을 가지는 시스템의 복잡한 제어 목적을 달성하기 위하여 신경회로망을 도입하는 노력이 활발히 진행되고 있다[8]. 신경회로망에 대한 연구는 80년대 초에 새로운 학습 방법들이 발표되면서 여러 공학적인 문제를 해결하는데 활용하고자 하는 연구가 진행되어 왔으며, 비선형 시스템의 모델링 및 제어 분야에도 중요한 응용 대상 분야로 많은 관심을 끌고 있다.

이 논문에서는 CNC 공작기계의 두 서보축을 대상으로 신경회로망을 이용한 윤곽제어 알고리즘을 제안한다. 두 서보축 상호간의 동특성의 차이에 의한 관계를 정확한 수학적 모델로 표현하여 보상하는 것은 매우 힘들다. 이 연구에서는 두 축 상호간에 미치는 영향을 신경망의 학습 능력을 이용하여 보상하고자 한다. 또한 윤곽제어를 위해서는 매 샘플링 주기마다 윤곽오차를 계산하여야 하나, 윤곽오차는 직선경로를 이동하는 경우 쉽게 계산가능하나 원호 보간, 인볼루트(involute) 보간등 비선형 경로를 가공하는 경우에는 정확하게 계산하기가 힘들다. 이러한 윤곽오차를 계산하기 위한 몇 가지 모델링 방법이 제안되었으나, 이들은 현재 순간의 기준위치와 실제 공구 위치에 대한 정보만을 사용함으로써 실제 윤곽오차와는 다소 거리가 있는 근사적인 윤곽오차를 찾고자 하였다[6,7]. 이 논문에서는 임의의 비선형 곡선 경로에 대하여도 윤곽오차를 정확히 구해낼 수 있는 새로운 윤곽오차 모델링 방법을 제안한다. 또한 이러한 윤곽오차에 대한 항을 포함하는 성능지수를 정의하고, 신경망 윤곽제어기를 위한 온라인 학습법칙을 유도한다. 이러한 신경망제어기의 사용으로 시스템이 비선형 특성, 시변 특성을 가지거나 외부 환경이 변화하는 경우에도 좋은 윤곽제어 성능을 유지할 수 있다.

## 2. 신경회로망을 이용한 윤곽제어

CNC 머신의 두 서보축의 위치 제어를 수행하기 위하여 고려하여야 할 오차는 각 축의 추적오차와 X-Y 공간상의 윤곽오차를 들 수 있다. 이 중 윤곽오차는 가공 정밀도를 결정하는 중요한 특성 값으로 기준궤적과 현재 공구위치와의 최단 거리로 정의된다. 윤곽제어를 위하여 이러한 윤곽오차는 매 샘플링 시간마다 실시간으로 계산되어야 하나 임의의 비선형 경로에 대하여 정확히 표현하기가 힘들다. 아래에 이러한 윤곽오차를 실시간으로 정확히 계산할 수 있는 윤곽오차 모델링 방법과 신경망 윤곽제어기에 대하여 서술하였다.

### 2.1 윤곽오차 모델

다축의 CNC 공작기계에서 임의의 곡선 가공시 주어진 곡선을 보간하여 매 샘플링 시간마다 노트 포인트(knot point)를 생성하고, 이 점들 사이는 직선으로 간주하여 기준궤적을 생성한다. 따라서 각 보간 주기 사이는 선형으로 근사화할 수 있다. 그림 1의 경우를 예로 들어 제안하는 방법을 설명하면, CNC 머신의 각 서보축은 시간 지연 양이 존재하므로, 항상 현재 순간의 기준 위치인  $P^*(t)$ 에서 일정량만큼 지연된 위치인  $P(t)$ 가 실제 위치가 된다. 따라서 현재 순간의 위치인  $P(t)$ 에서 기준 궤적의 과거의 점들과의 거리가 가장 짧은 두 점이 존재하게 된다. 이 두 점을 각각  $P_1^*$ 와  $P_2^*$ 라 놓으면, 윤곽오차  $\epsilon_c$ 는  $P(t)$ 와 두 점을 이은 직선 사이의 임의의 점인  $P_c^*(t)$ 와의 가장 가까운 거리로써 정의할 수 있다.

현재 위치와 가장 가까운 두 점을 찾기 위해 기준 궤적 위의 많은 점들과의 거리를 계산하게 되면 많은 계산 시간이 요구되므로, 가변 윈도우를 이용한 방법을 사용한다. 즉, 적당한 크기의 윈도우를 정의해서 기준 궤적의 과거 값을 메모리에 저장하고, 현재 순간의 실제 궤적의 위치와의 거리를 구하면, 기준 궤적의 과거 데이터들 중에서 거리가 가장 짧은 두 점을 찾을 수 있다. 다음 순간의 실제 위치를  $P(t+1)$ 이라 하면, 이 순간의 윈도우의 시작은 그림 1에 나타난 것처럼  $P_1^*$ 이고, 이 점에서부터 현재 위치까지의 거리를 구해보면, 거리는 계속 줄어들다가 다시 증가하게 된다. 따라서 윈도우의 끝은 거리가 다시 증가하는 점이 된다. 이와 같이 윈도우를 반복적으로 새로이 정의함으로써 계산 양을 최소화하고 실행 시간을 줄일 수 있다. 실제 위치와 가장 가까운 두 점  $P_1^*$ 와  $P_2^*$ 를 찾기 위한 알고리즘은 아래와 같다.

단계 0: (초기화 단계) 이 알고리즘에서는 현재 순

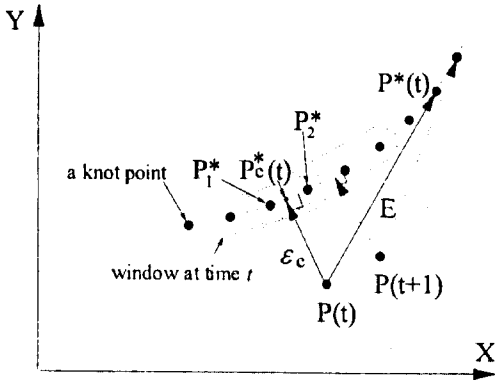


그림 1. 제안한 윤곽오차 모델  
Fig. 1. The proposed contour error model

간의 실제 위치  $P(t)$ 와 기준 궤적 위의 점들과의 거리,  $dist$ 를 계산하고, 이 거리의 변화량,  $\Delta d$ 를 이용하여 가장 가까운 두 점  $P_1^*$ 와  $P_2^*$ 를 찾아내는데 가공 시작 시점인  $t=0, T$  일 때는 비교 대상이 없기 때문에  $\Delta d$ 를 구할 수 없으므로 초기에  $P_1^*$ 와  $P_2^*$ 는

$$P_1^* = P_2^* = P^*(0), \quad \text{if } t=0$$

$$P_1^* = P^*(0), \quad P_2^* = P^*(T), \quad \text{if } t=T$$

로 둔다. 여기서  $T$ 는 샘플링 시간이다. 그리고  $k=1$ 을 메모리에 저장한다.  $k$ 는  $P_1^*$ 가 메모리에 저장되어있는 위치를 가리킨다.

단계 1: 현재 순간,  $t$ 에서 기준 궤적의 위치  $P^*(t)$ 를 메모리에 저장한 다음  $t-1$ 시간에서의  $k$ 를 읽어들이고,  $k=k+1$ 한 다음, 실제 위치와의 거리를 구한다.

$$dist(k) = \| P^*(k) - P(t) \|$$

단계 2:  $k$ 를 감소시켜서 거리를 구하고 거리의 변화량을 계산한다.

$$k = k - 1$$

$$dist(k) = \| P^*(k) - P(t) \|$$

$$\Delta d = dist(k+1) - dist(k)$$

단계 3:  $\Delta d \geq 0$  이면, 단계 2로 가고, 그렇지 않으면 단계 4로 간다.

단계 4:  $\Delta d$ 가 음이 되었으므로 실제 위치와 가장 가까운 점은  $P^*(k+1)$ 이 되고, 그 다음으로 가까운 점을 찾으면 된다.  $dist(k) < dist(k+2)$ 이면 단계 5로 가고, 아니면 단계 6으로 간다.

단계 5:  $P_1^*$ 와  $P_2^*$ 에 가장 가까운 두 점을 대입하고,  $P_1^*$ 가 메모리에 저장되어있는 위치  $k$ 를 저장하고 끝낸다.

$$P_1^* = P^*(k+1)$$

$$P_2^* = P^*(k)$$

$$k = k + 1$$

단계 6:  $P_1^*$ 와  $P_2^*$ 에 가장 가까운 두 점을 대입하고,  $P_1^*$ 가 메모리에 저장되어있는 위치  $k$ 를 저장하고 끝낸다.

$$P_1^* = P^*(k+2)$$

$$P_2^* = P^*(k+1)$$

$$k = k + 2$$

이 알고리즘은 기준 궤적의 과거 값을 저장하고 이 값들과 현재 위치간의 거리를 구한 다음 거리의 변화를 보고 거리의 변화가 존재하면, 그 때의  $k$ 값을 저장하고 현재 위치와 가장 가까운 두 점인  $P_1^*$ 와  $P_2^*$ 를 찾아내는 것이다. 따라서 천이 상태에서는 매 순간마다 윈도우의 크기는 변화하지만, 시스템이 정상 상태로 들어섬에 따라 윈도우의 크기는 고정되게 된다.

이렇게 해서 찾은 두 점 사이는 다시 그림 2로 나타낼 수 있으며, 이로부터 윤곽오차를 계산하는 방법은 다음과 같다. 그림 2에서  $\theta$ 는 매 보간 주기마다 계산해야 할 값으로써

$$\theta = \tan^{-1} \left( \frac{y_2 - y_1}{x_2 - x_1} \right) \quad (1)$$

이며,  $e_x$ '은  $P(t)$ 와  $P_2^*$ 의 X축의 오차 양이고,  $e_y$ '은 Y축의 오차 양이다. 그림 2에서 기하학적인 관계에 의해 현재 순간의 윤곽오차  $\epsilon_c$ 는

$$\epsilon_c = \text{sign}(x_2 - x_1) \cdot (-e_x' \sin\theta + e_y' \cos\theta) \quad (2)$$

가 된다.

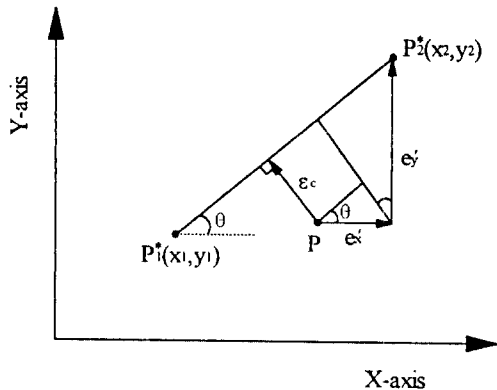


그림 2. 두 점과 윤곽오차와의 관계  
Fig. 2. The relation between two points on the reference contour and a contour error

### 2.2 신경망 동기 제어

CNC 머신의 두 서보축을 대상으로 제한하는 신경망 윤곽제어 구조는 그림 3과 같다. 여기서 각 축의 위치제어를 위한 PID 형태의 제어기와 X-Y 공간상의 공구의 움직임에 고려하여 두 축 상호간의 영향을 보상하는 위한 신경망제어기가 결합되어 사용된다. 각 서보축의 PID 제어기는 실험적인 방법 또는 각 축의 선형화된 모델을 이용하는 해석적 방법을 통해 설계될 수 있으며, 이 경우 각각은 각 축의 추적 성능을 고려하여 설계된다. 또한 신경망제어기는 윤곽제어 특성을 함께 고려하여 온라인으로 학습되며, 플랜트의 비선형성이나 모델링 되지 않은 동특성에 대한 영향을 보상하는 기능을 수행한다. 또한 신경망제어기의 사용으로 제어 환경의 변화나 플랜트 파라미터의 변화에 대한 제어 성능의 저하를 보상할 수 있다. 각 축을 구동시키는 서보모터에 인가되는 제어 입력은 PID 제어기와 신경망제어기 출력의 합으로서 아래와 같이 표현된다.

$$u_1(t) = u_1^H(t) + u_1^N(t) \quad (3)$$

$$u_2(t) = u_2^H(t) + u_2^N(t) \quad (4)$$

여기서  $u_1(t)$ 와  $u_2(t)$ 는 각각 x축과 y축의 제어 입력,  $u_1^H(t)$ 와  $u_2^H(t)$ 는 각 축의 PID 제어기의 출력이고,  $u_1^N(t)$ 와  $u_2^N(t)$ 는 신경망제어기의 보상 입력을 의미한다.

신경망제어기는 가장 널리 이용되는 다층 퍼셉트론의 구조를 가지며, 그림 4와 같다. Cybenko[9]는 하나의 은닉층을 가지는 다층신경망으로 임의의 정밀도의 함수를 근사화 하는데 충분함을 보였다. 또한 두 개 이상의 은닉층을 사용하는 경우 보다 많은 CPU 수행 시간을 요구하게 된다. 이러한 이유로 이 연구에서는 하나의 은닉층을 가지는 3층 신경망 모델을 사용하였다. 신경망제어기는  $L_1$ 개의 입력층 뉴런,  $L_2$ 개의 은닉층 뉴런, 2개의 출력층 뉴런을 가지며, 2개의

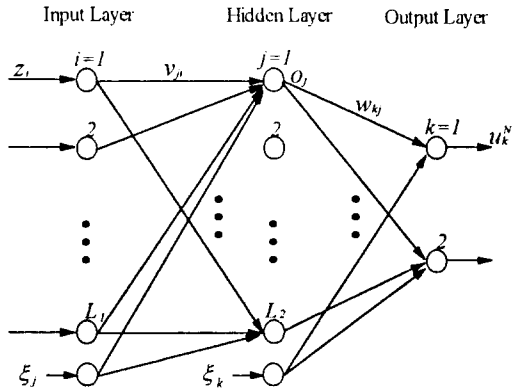


그림 4. 다층 신경회로의 구조  
Fig. 4. Structure of a multilayer neural network

출력 뉴런은 각 축의 보상 제어입력으로 사용된다. 은닉 뉴런의 출력은 다음과 같이 표현된다.

$$o_j = f(q_j) = \frac{1 - e^{-q_j}}{1 + e^{-q_j}} \quad (5)$$

여기서  $q_j = \sum_{i=1}^{L_1} v_{ji} \cdot z_i + \xi_j$ ,  $j=1, 2, \dots, L_2$ 이고,  $o_j$ 는  $j$ 번째 은닉 뉴런의 출력,  $v_{ji}$ 는  $i$ 번째 입력 뉴런과  $j$ 번째 은닉 뉴런 사이의 연결강도를 의미한다. 또한  $f(\cdot)$ 는 hyperbolic tangent 함수이며,  $\xi_j$ 는 은닉층에서의 문턱값을 의미한다. 출력 뉴런은 선형 특성을 가지며 아래 식과 같이 표현된다.

$$u_k^N = \sum_{j=1}^{L_2} w_{kj} \cdot o_j + \xi_k, \quad k=1, 2. \quad (6)$$

여기서  $w_{kj}$ 는  $k$ 번째 출력 뉴런과  $j$ 번째 은닉 뉴런 사이의 연결강도이며,  $\xi_k$ 는 출력층에서의 문턱 값이다.

### 2.3 신경망 학습법칙

윤곽제어를 위한 신경망제어기의 학습법칙을 유도하기 위하여 각 축의 추적오차와 윤곽오차에 대한 항으로서 목적함수  $E$ 를 아래와 같이 정의한다.

$$E = E_x + E_y + \beta \cdot E_c \quad (7)$$

여기서  $\beta$ 는 윤곽오차항에 대한 가중치를 의미하고,  $E_x, E_y, E_c$ 는 다음과 같이 정의된다.

$$E_x = \frac{1}{2} \cdot (r_x - c_1)^2 \quad (8)$$

$$E_y = \frac{1}{2} \cdot (r_y - c_2)^2 \quad (9)$$

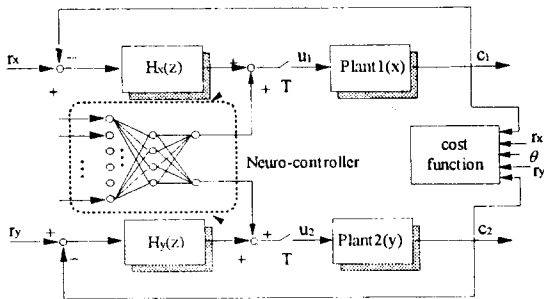


그림 3. 신경망 윤곽제어 시스템  
Fig. 3. Neuro-contouring control system

$$E_c = \frac{1}{2} \cdot \varepsilon_c^2 \quad (10)$$

여기서  $c_1$ 와  $c_2$ 는 각각 x와 y축의 출력이고, 각 축의 추적오차에 대한 항  $E_x$ 와  $E_y$ 는 각각 x와 y축상에서 정의된다. 또한  $E_c$ 는 윤곽오차에 대한 항으로 x-y 평면상에서 정의된다.

앞에서 정의된 목적함수 (7)을 최소화하도록 신경망제어기의 연결강도를 학습시킨다. 오차역전달 알고리즘을 이용하여  $j$ 번째 은닉노드와  $k$ 번째 출력노드 사이의 연결강도는 아래 식과 같이 조정된다.

$$w_{kj}(t+1) = w_{kj}(t) + \Delta w_{kj}(t) + \alpha \cdot \Delta w_{kj}(t-1) \quad (11)$$

여기서  $\Delta w_{kj}(t) = -\eta \cdot \frac{\partial E}{\partial w_{kj}}$  이 되고,  $\eta$ 는 학습률을 의미한다. 또한 (11)의 마지막 항은 모멘텀이며, 신경망의 수렴 속도를 향상시키기 위하여 사용된다. chain rule을 이용하여

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial c_k} \cdot \frac{\partial c_k}{\partial u_k} \cdot \frac{\partial u_k}{\partial w_{kj}} \quad (12)$$

으로 표현된다. 여기서

$$\frac{\partial u_k}{\partial w_{kj}} = \frac{\partial u_k^N}{\partial w_{kj}} = o_j \quad (13)$$

$$\delta_1 \equiv \frac{\partial E}{\partial c_1} = -(r_x - c_1) + \varepsilon_c \cdot \frac{\partial \varepsilon_c}{\partial c_1} \quad (14)$$

$$\delta_2 \equiv \frac{\partial E}{\partial c_2} = -(r_y - c_2) + \varepsilon_c \cdot \frac{\partial \varepsilon_c}{\partial c_2} \quad (15)$$

가 되고, (2)로부터  $\partial \varepsilon_c / \partial c_1$ 와  $\partial \varepsilon_c / \partial c_2$ 는  $\sin\theta$ 와  $-\cos\theta$ 가 된다. 또한 (12)에서 미분치  $\partial c_k / \partial u_k$ 는 플랜트의 동특성에 관계되는 항으로 정확히 계산하기가 힘들다. 이를 얻기 위해 플랜트의 특성을 모사하는 식별 신경망을 사용하기도 하지만, 여기서는  $\partial c_k / \partial u_k$ 를  $u_k$ 에 대한  $c_k$ 의 변화의 부호로 대치함으로써, 비교적 간단한 구조를 가지도록 하였다. 이러한 방법은 Zhang[10]의 연구에서도 좋은 결과를 보여 주고 있다. 최종적으로 신경망 연결강도의 변화량은

$$\Delta w_{kj} \simeq -\eta \cdot \delta_k \cdot \text{sign} \left( \frac{\partial c_k}{\partial u_k} \right) \cdot o_j \quad (16)$$

로 표현된다. 위와 유사하게  $i$ 번째 입력노드와  $j$ 번째 은닉노드 사이의 연결강도는

$$v_{ji}(t+1) = v_{ji}(t) + \Delta v_{ji}(t) + \alpha \cdot \Delta v_{ji}(t-1) \quad (17)$$

와 같이 조정되고, 여기서

$$\begin{aligned} \Delta v_{ji}(t) &= -\eta \cdot \left( \frac{\partial E}{\partial v_{ji}} \right) \\ &= -\eta \cdot \sum_{k=1}^2 \left( \frac{\partial E}{\partial c_k} \cdot \frac{\partial c_k}{\partial u_k} \cdot \frac{\partial u_k}{\partial o_j} \cdot \frac{\partial o_j}{\partial v_{ji}} \right) \\ &\simeq -\eta \cdot f'(q_j) \cdot z_i \cdot \sum_{k=1}^2 \delta_k \cdot \text{sign} \left( \frac{\partial c_k}{\partial u_k} \right) \cdot w_{kj} \end{aligned}$$

와 같이 표현된다.

### 3. 모의실험 및 검토

제안하는 신경망 윤곽제어 기법의 타당성을 검토하기 위하여 CNC 머신의 X, Y 서보축의 모델을 대상으로 모의실험을 수행하였다. 모의실험은 MATLAB을 이용하여 수행하였고, 각축의 플랜트 동특성은 마찰력을 포함하는 mass-force 시스템으로 모델링된다. 사용된 마찰력 모델은 정지마찰력과 쿨롱마찰력 또한 저속에서의 천이 특성을 포함하며, 각 축의 플랜트 모델은 아래와 같다. 또한 모의실험에 사용한 시스템 파라미터를 표 1에 나타내었다.

$$m \ddot{c} = u - F_j \quad (18)$$

$$F_j = B\dot{c} + f_c \cdot \text{sgn}(\dot{c}) + f_s \cdot e^{-\alpha \dot{c}} \cdot \text{sgn}(\dot{c}) \quad (19)$$

여기서  $\dot{c}$ 는 각축의 이송속도이고,  $B$ 는 점성마찰력 상수,  $f_c$ 는 쿨롱마찰력,  $f_s$ 는 정지마찰력을 의미한다.

먼저 양의 x축과 45° 기울어진 직선 경로를 2.7 m/min의 이송속도로 가공하는 경우에 대하여 모의실험을 수행하였고, 그 제어 성능을 비교하였다. 여기서 신경망제어기의 입력으로는 다음과 같은 값들이 사용되었고, 학습률은 0.05, 모멘텀은 0.15를 사용하였다.

$$[x_r(t) c_1(t) c_1(t-1) y_r(t) c_2(t) c_2(t-1) u_1^N(t) u_1^N(t)] \quad (20)$$

그림 5는 PID 제어기만을 사용하는 경우와 신경망

표 1. 모의실험에 사용한 파라미터

Table 1. The simulation parameters

	$m$	$B$	$f_c$	$f_s$	$\alpha$	P/I/D 이득	입력/은닉 노드수	샘플링 시간
X-axis	0.45	0.01	3	1	2	5/20/0.01	8/5	1ms
Y-axis	0.65	0.01	4	1.5	2			

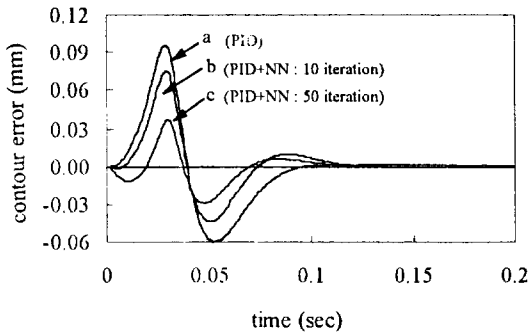


그림 5. 직선 경로에 대한 모의실험 결과  
Fig. 5. Simulation results for a linear contour

표 2. 직선 경로에 대한 윤곽오차의 비교  
Table 2. Comparison of contour errors for a linear contour

	PID	PID+ NN (10 iteration)	PID+ NN (50 iteration)
$\epsilon_{peak}$ (mm)	0.0961	0.0767	0.0321
$\epsilon_{ISE}$ (mm)	3.3009	2.3005	1.2825

제어기를 추가하는 경우에 대한 윤곽제어 결과를 보여 준다. 신경망제어기의 학습이 이루어짐에 따라 윤곽오차는 크게 감소함을 알 수 있다. 표 2에서는 각 경우에 대한 윤곽오차를 나타내었고, 여기서  $\epsilon_{peak}$ 와  $\epsilon_{ISE}$ 는 실제 윤곽오차의 최고치와 실제 윤곽오차의 제곱의 합을 의미한다.

다음으로 반지름이 10 mm인 원의 경로를 5 m/

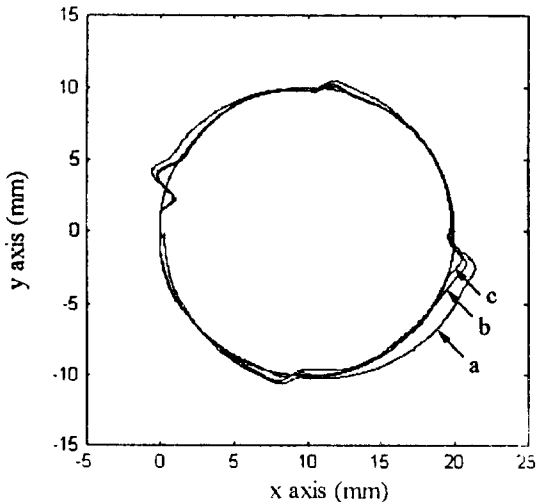


그림 6. 원호 경로에 대한 모의실험 결과  
Fig. 6. Simulation results for a circular contour

표 3. 원호 경로에 대한 윤곽오차의 비교  
Table 3. Comparison of contour errors for a circular contour

	PID	PID+ NN (10 iteration)	PID+ NN (50 iteration)
$\epsilon_{peak}$ (mm)	0.0724	0.0445	0.0373
$\epsilon_{ISE}$ (mm)	17.8285	7.4691	5.1139

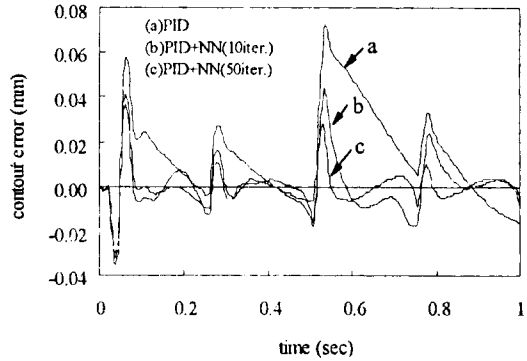


그림 7. 원호 경로에 대한 윤곽오차  
Fig. 7. Contour errors for a circular contour

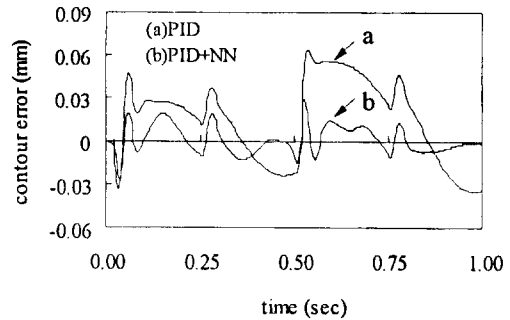


그림 8. 절삭력 외란이 존재하는 경우의 모의실험 결과  
Fig. 8. Simulation results with a cutting force

min의 이송속도로 가공하는 경우에 대한 모의실험을 수행하였다. 그림 6에는 각 경우 원의 기준 경로에 대한 응답 특성을 X-Y 평면상에 나타내었다. 여기서 제어 성능의 비교를 보다 편리하게 하기 위하여 각각의 경우에 대하여 매 샘플링마다의 오차를 10배 증폭시켜 나타내었다. 또한 각 경우에 대한 실제 윤곽오차를 표 3과 그림 7에 나타내었다. 직선의 경로에서와 마찬가지로 신경망 윤곽제어기의 사용으로 윤곽오차는 크게 감소함을 알 수 있다. 또한 그림 8은 신경망제어기가 충분히 학습된 후 학습 시 고려되지 않은 비선형 절삭력이 두 축에 추가하였을 경우의 원호경로의 이송 제어 결과를 보여 준다. 여

기서 학습 기능을 가지는 신경망 윤곽제어기의 사용으로 절삭력과 같은 외부 외란이 존재하는 경우에도 좋은 윤곽제어 성능을 유지할 수 있음을 알 수 있다.

#### 4. 결 론

이 연구에서는 CNC 공작기계의 X-Y 두 서보축을 대상으로 윤곽제어 성능을 개선시키기 위한 신경망 윤곽제어 알고리즘을 제안하였다. 기존에 사용하고 있는 각 축의 PID 형태의 제어기에 신경망제어기를 부가하여, 두 축 상호간에 미치는 영향을 신경망의 학습 능력을 이용하여 보상하였다. 이를 위하여 원호, 인볼루트 곡선등 일반적인 비선형 곡선 경로에 대해서도 실제 윤곽오차를 정확히 계산해 낼 수 있는 윤곽오차 모델을 제안하였다. 또한 이러한 윤곽오차에 대한 항을 포함하는 성능지수를 정의하고, 신경망제어기를 위한 온라인 학습법칙을 유도하였다. 모의실험을 통하여 신경망 윤곽제어기의 사용으로 시스템이 비선형 특성을 가지거나 외부 환경이 변화하는 경우에도 좋은 윤곽제어 성능을 유지할 수 있음을 확인하였다.

#### 참고문헌

- [1] T.C. Chiu and M. Tomizuka, "Coordinated Position Control of multi-axis Mechanical Systems," *ASME Dynamic Systems and Control*, DSC-Vol. 55-1, pp. 159-166, 1994.
- [2] Y. Koren and C.C. Lo, "Advanced Controllers for Feed Drives," *Annals of the CIRP*, Vol. 41, no. 2, pp. 689-699, 1992.
- [3] M. Tomizuka, "Zero Phase Error Tracking Algorithm for Digital Control," *ASME Trans. J. of Dynamic Systems, Measurement and Control*, Vol. 109, pp. 65-68, 1987.
- [4] M. Weck and G. Ye, "Sharp Corner Tracking Using the IKF Control Strategy," *Annals of the CIRP*, Vol. 39, No. 1, pp. 437-441, 1990.
- [5] Y. Koren, "Cross-coupled Biaxial Computer Control for Manufacturing Systems," *ASME Trans. J. of Dynamic Systems, Measurement and Control*, Vol. 102, NO. 4, Dec., pp. 265-272, 1980.
- [6] Y. Koren and C.C. Lo, "Variable-gain Cross-cou-

pling Controller for Contouring," *Annals of the CIRP*, Vol. 104, pp. 371-374, Aug., 1991.

- [7] P.K. Kulkarni and K. Srinivasan, "Cross-coupled Control of Biaxial Feed Drive Servomechanisms," *ASME Trans. J. of Dynamic Systems, Measurement and Control*, Vol. 112, No. 2, June, pp. 225-232, 1990.
- [8] K.S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," *IEEE Trans. Neural Networks*, Vol. 1, pp. 4-27, Mar., 1990.
- [9] J. de Villiers and E. Barnard, "Backpropagation Neural Nets with One and Two Hidden Layers," *IEEE Trans. Neural Networks*, Vol. 4, No. 1, pp. 136-141, Jan., 1993.
- [10] Y. Zhang, P. Sen, and G.E. Hearn, "An On-line Trained Adaptive Neural Controller," *IEEE Control Systems*, pp. 67-75, Oct., 1995.



**이현철 (Hyun Chul Lee)**

1991년: 경북대학교 전자공학과 (공학사)  
 1993년: 경북대학교 대학원 전자공학과(공학석사)  
 1995년~현재: 경북대학교 대학원 전자공학과 박사과정  
 1993년~현재: 서울대학교 제어계측신기술연구소 연구원



**주정홍 (Jeong Hong Joo)**

1995년: 창원대학교 전자공학과 (공학사)  
 1996년~현재: 경북대학교 대학원 전자공학과 석사과정

**전기준 (Gi Joon Jeon)**

1969년: 서울대학교 금속공학과 (공학사)  
 1978년: 미국 Houston대 전기공학과 (공학석사)  
 1983년: 미국 Houston대 전기공학과 (공학박사)  
 1971년~1976년: 국방과학연구소 연구원  
 1983년~현재: 경북대학교 전자전기공학부 교수