

경사도 기법을 사용한 PID 제어기의 성능 개선에 관한 연구

하동호·정종대
한국기술교육대학교 전기공학과

A Study on the Performance Improvement of the Auto-Tuning PID Controller Using Gradient Method

Dong-Ho Ha, Jong-Dae Jung
Dept. of Electrical Engineering Korea University of Technology and Education

Abstract - In this paper, we proposed a simple neural network-based parameter tuning algorithm, which could find the gradients of a certain performance index in the PID parameter spaces. In this process, we had to know the dynamics between input and output of the plant, and we used the Back Propagation Neural network to identify them. To make the parameter updating fast and smooth, we constructed the performance index as the sum of past N-squared plant errors, and applied a batch mode algorithm to update parameters. We performed several experiments with a DC Motor to show the validity of the proposed algorithm.

는 비용함수(Cost function)의 제어기 매개변수들에 대한 변화율, 즉 경사도(gradient)를 구한 다음 이 비용함수 값이 감소하는 방향이 되도록 각 매개변수들을 조절하고자 한다. 즉, 제어기의 비용함수를 다음과 같이 N개의 과거오차를 누적한 것으로 정의하고 이 비용함수 값이 최소가 되도록 PID 제어기의 매개변수들을 동조시키고자 한다.

$$J_C = \frac{1}{2} \sum_{i=0}^{N-1} [r(k-i) - y_p(k-i)]^2 = \frac{1}{2} \sum_{i=0}^{N-1} e_p(k-i)^2 \quad (2)$$

$$e_p(k-i) = r(k-i) - y_p(k-i).$$

여기서 r 과 y_p 는 각각 목표입력과 공정의 출력이다. 식(2) 비용함수의 제어기 각 매개변수들에 대한 변화율은 다음과 같이 계산된다

$$\frac{\partial J_C}{\partial K} = \sum_{i=0}^{N-1} e_p(k-i) \frac{\partial e_p(k-i)}{\partial K} = - \sum_{i=0}^{N-1} e_p(k-i) \frac{\partial y_p(k-i)}{\partial u(k-i-1)} \frac{\partial u(k-i-1)}{\partial K(k-i-1)} \quad (3)$$

여기서 $K=[K_P, K_I, K_D]^T$ 이며, 매 N개의 과거 데이터가 채집될 때마다 식(3)에 의해 비용함수의 기울기 $\partial J/\partial K$ 가 구해지며 이 기울기 정보를 사용하여 식(4)와 같이 비용함수 값이 줄어드는 방향으로 매개변수 K 값들의 갱신이 이루어진다. 이 구조는 근본적으로는 [1]의 방법과 같으나 식(3)과 같이 정해진 구간 동안 비용함수의 매개변수에 대한 기울기 변화분을 합하여 평균을 취해 갱신에 사용함으로써 순간적인 외란이나 공정의 매개변수 변화, 오차변동 등에 더욱 강인하고 안정된 제어출력을 기대할 수 있게 된다.

$$\Delta K(k) = -(1-a_c) \frac{\partial J_C}{\partial K} + a_c \Delta K(k-1) \quad (4)$$

$$K(k) = K(k-1) + \eta_c \Delta K(k)$$

여기서 a_c 는 모멘텀 계수이고 η_c 는 제어기의 동조율이다.

그런데 경사도 계산 시 $\partial y_p/\partial u$ 는 바로 공정의 입력에 대한 출력의 감도로서 이 값은 직접 주어지지 않으므로 신경망에 의한 시스템 식별기를 사용하여 이 값을 추정하고자 한다. 신경망에 의한 시스템 식별기는 공정에 가해진 제어입력 $u(k-1)$, $u(k-2)$ 와 $y_p(k-1)$, $y_p(k-2)$ 를 이용하여 추정출력 $y_M(k)$ 를 만들어 낸 다음 이 값이 공정의 실제 출력 $y_p(k)$ 와 같아지도록 매개변수들(신경망에서는 가중치들)을 조절한다. 따라서 신경망이 제대로 훈련된다면 이 $y_M(k)$ 가 실제 공정의 출력 $y_p(k)$ 와 근사하게 될 것이고 직접 구할 수 없는 량인 $\partial y_p(k)/\partial u(k-1)$ 대신에 알고 있는 신경망 구조로부터 자연스럽게 얻을 수 있는 $\partial y_M(k)/\partial u(k-1)$ 을 사용할 수 있을 것이다.

1. 서 론

본 논문에서는 인공 신경망의 학습기능을 이용하여 전적으로 미지인 공정에 대해서도 설정된 제어목표가 달성될 수 있는 제어기 구조를 제안하고자 한다. 또한 외란과 공정의 동특성이 변화더라도 능동적으로 대처할 수 있게 스스로 학습을 수행하면서 제어기의 매개변수들을 온라인으로 조정하는 방법을 제안하고자 한다.

본 논문에서 시도하고자 하는 방법은 기존의 PID 제어기 구조를 그대로 유지하고 있으면서 미지 공정의 동특성을 식별(identification)하기 위한 하나의 신경망만을 사용한다. 이 신경망 식별기는 공정의 상태변수나 매개변수 전체를 추정하는 것이 아니라 단지 입출력 사이의 감도만을 추정하면 되므로 전체 제어기의 구조를 단순화 할 수 있고 계산량의 부담도 경감된다는 이점이 있다. 이 구조는 [1]에서 소개되어 있다. 하지만 [1]에서 제안된 방법에서는 매개변수 추정방향이 현재의 오차에 민감하게 반응하여 학습이 반복될 수록 제어공정의 응답 속도가 빨라져 공정의 출력이 점점 불안해지는 현상을 보였다. 따라서 본 논문에서는 [1]에서 사용된 구조를 수정하여 보다 강인한 제어 및 매개변수 동조가 가능한 제어시스템을 제안하고자 한다.

2. PID 제어기의 매개변수 동조 알고리즘

$$u(k) = u(k-1) + K_P[e(k) - e(k-1)] + K_I T_0 e(k) + K_D[e(k) - 2e(k-1) + e(k-2)] / T_0 \quad (1)$$

여기서 T_0 는 샘플링 타임이다. 식(1)을 속도형 PID 제어기라 하며, 이 제어기는 디지털 제어시스템에 일반적으로 사용하는 제어알고리즘으로서, 본 논문에서도 속도형 디지털 PID 제어식을 적용하였다. 식(1)에서 K_P, K_I, K_D 들이 PID 제어기의 매개변수들이다.

본 논문에서는 제어시스템의 추종오차 $e_p(k)$ 로 구성되

3. 신경망을 사용한 공정의 모델링

비용함수의 제어기 매개변수들에 대한 기울기 $\partial J/\partial K$ 를 구하기 위해서는 공정의 입출력 감도 $\partial y_p/\partial u$ 를 알아야 하는데 공정이 미지라고 가정하였으므로 신경망을 사용하여 공정의 입출력 동특성을 학습하게 하고 학습된 신경망의 입출력 감도 $\partial y_M/\partial u$ 를 계산하여 $\partial y_p/\partial u$ 대신 사용하고자 한다[1][2]. 이는 신경망이 올바르게 학습될 경우 그의 입출력 감도 $\partial y_M/\partial u$ 와 실제 공정의 입출력 감도 $\partial y_p/\partial u$ 가 같게 될 것이라는 가정에 기인하고 있다.

신경망 식별기의 구조는 입력층과 출력층 그리고 하나의 은닉층을 가지며, 입력으로는 지연된 제어입력들과 공정출력들을 인가함으로써 플랜트의 동특성을 잘 내포할 수 있도록 하였다[3].

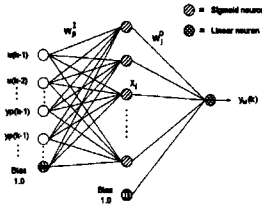


그림1. 신경망 식별기의 구조

신경망 식별기의 추정오차를 줄여 나가기 위하여 다음과 같은 2차형 비용함수를 정의하고 이 비용함수를 줄여 나가는 방향으로 학습하게 된다.

$$J_M = \frac{1}{2} [y_p(k) - y_M(k)]^2 = \frac{1}{2} e_M(k)^2 \quad (5)$$

$$e_M(k) = y_p(k) - y_M(k)$$

신경망의 학습은 다음과 같은 오류역전파(Error back propagation) 학습 알고리즘에 의해 수행된다[4][5][6].

$$\Delta W^L(k) = -(1 - \alpha_M) \frac{\partial J_M}{\partial W^L(k-1)} + \alpha_M \Delta W^L(k-1)$$

$$W^L(k) = W^L(k-1) + \eta_M \Delta W^L(k) \quad (6)$$

여기서 L 은 I 와 O 이고 η_M 은 신경망의 학습률, α_M 은 모멘텀 계수, W^I 와 W^O 는 각각 신경망 입력층과 출력층의 가중치이다.

이제 미지공정의 입출력간 변화를 $\partial y_p(k)/\partial u(k-1)$ 은 다음과 같이 근사화 하여 구할 수 있다.

$$\frac{\partial y_p(k)}{\partial u(k-1)} \cong \frac{\partial y_M(k)}{\partial u(k-1)} = \sum_{j=1}^n W_j^O f_j'(S_j^H) W_j^I \quad (7)$$

여기서 f_j' 는 은닉층 뉴런의 활성화 함수이며, S_j^H 는 은닉층 뉴런에 대한 입력이다.

다음 그림 3.2에 신경망에 의한 시스템 식별기와 PID 제어기를 결합한 종합적인 제어 계통도를 보인다.

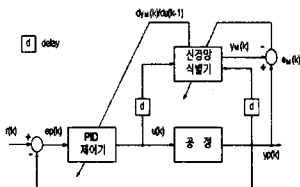


그림2. 종합적인 제어 계통도

4. 시뮬레이션 및 검토

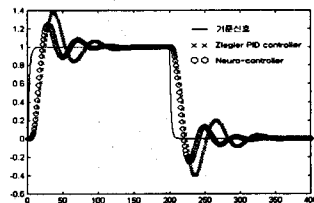
이 장에서는 시뮬레이션을 통해 위에서 제안한 PID 제어기 동조알고리즘을 임의의 공정에 적용하여 결과를 알아보도록 한다. 시뮬레이션에 사용된 공정은 Zhen-Yu Zhao등[7]이 사용했던 4차 공정을 사용하였다.

$$G(s) = \frac{27}{(s+1)(s+3)^3} \quad (8)$$

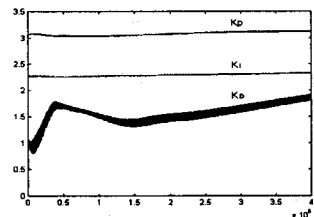
시뮬레이션에서는 이 공정들을 샘플링 타임 $T_0 = 0.05$ 초로 이산화 하였다. 시뮬레이션에 사용된 신경망은 4개의 입력층 뉴런과 11개의 은닉층 뉴런을 가지며 학습 시 η_M 의 값은 0.07, α_M 은 0.15를 사용하였고 연결가중치의 초기값으로는 -1과 +1사이의 임의의 값을 사용하였다. 여기서 입력벡터는 $I(k) = [u(k-1), u(k-2), y_p(k-1), y_p(k-2), 1.0]$ 가 사용되었고 마지막 성분 1.0은 바이어스항을 위한 입력요소이다.

여기서 제어기 매개변수의 초기치는 제어시스템이 불안정영역에 있지 않도록 적당한 값을 정해 주어야 하는데 본 시뮬레이션에서는 Ziegler-Nichols 방법으로 결정되는 PID 이득 값을 초기치로 사용하였다. 공정에 대하여 사용된 매개변수는 $K_P=3.072$, $K_I=2.272$, $K_D=1.038$ 이다. 또한 η_c 의 값은 0.1, α_c 는 0.05의 값을 사용하였으며, 일괄처리를 위한 채집자료의 개수 N 은 30으로 하였다.

그림 3은 1000epoch 학습 후 제안한 방식의 제어기와 Ziegler-Nichols방식의 제어기에 의한 계단응답을 비교한 것으로서 제안한 방식의 출력응답이 상승시간, 오버슈트, 세틀링타임 면에서 우수함을 알 수 있다. 최대 epoch수는 1000이다. 여기서 한 epoch이라 함은 한 주기의 구형과 응답을 400 샘플로 채집하여 구성한 총 학습 데이터를 한 번 학습한 것을 의미한다.



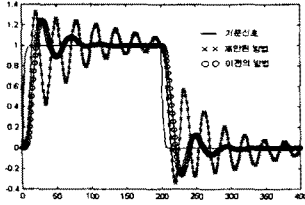
(a) 제안한 방식과 Ziegler-Nichols방식의 제어기와의 계단응답 비교



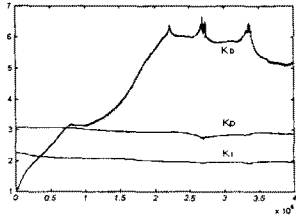
(b) 학습 진행에 따른 PID 이득의 변화
그림3. 4차 공정에 대한 시뮬레이션 결과

그림 4는 제안된 방법과 이전 방법과의 출력을 비교하는 그림이다. 기존의 $N=1$ 인 방법과 제안된 방법에 의해

$N=30$ 일 때의 제안응답 출력 그리고 학습 진행에 따른 각각의 매개변수 변화양상을 보여준다.



(a) 제안한 방식과 이전 방식의 제어기와의 제안응답 비교

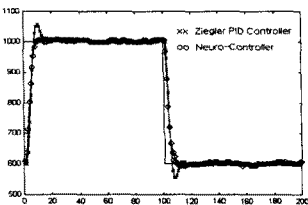


(b) 학습 진행에 따른 PID 이득의 변화(이전 방법)
그림4. 제안된 방법과 이전 방법과의 제안응답 비교

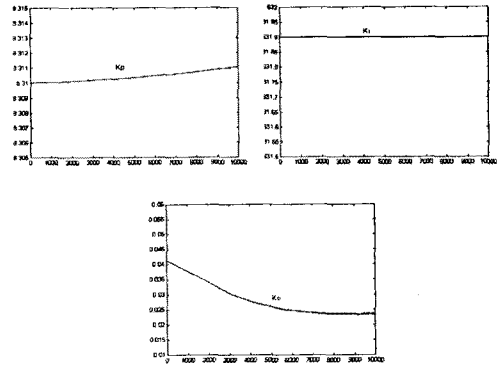
그림 4의 결과는 매 샘플링마다의 오차 $e_p^2(k)$ 에 의존하여 제어기 매개변수들을 갱신하는 경우(이전의 방법), 지속적인 학습에 의해 계속 응답의 상승시간을 단축시키기 위해 과도한 제어입력이 발생되었고, 매개변수의 수렴양상이 제안된 방법(그림 3(b))에 비해 느리고 변화가 심함을 알 수 있다.

5. 실험 및 검토

이 절에서는 제안한 신경망 식별기에 의한 PID동조 알고리즘을 실제 직류전동기의 속도제어에 적용시키는 실험을 수행하였다. 실험에서 사용한 K_p , K_i , K_d 의 초기치는 모터의 오픈루프 응답을 사용하는 Ziegler-Nichols의 이득 계산 방법으로 결정된 값을 사용하였다. 여기서 사용한 값은 $K_p=8.31$, $K_i=631.9$, $K_d=0.041$ 이며, 동조율 η_c 는 1.0×10^{-10} 으로 하였고 모멘텀 α_c 는 사용하지 않았다. 신경망 식별기의 구조는 $n_i=4$, $n_h=7$ 로 하였으며, 사용된 입력벡터는 $I(k) = [u(k-1), u(k-2), y(k-1), y(k-2), 1.0]$ 이다. 학습률 η_M 과 모멘텀 α_M 의 값은 각각 0.08, 0.15이고 입출력층 가중치의 초기치는 -1과 1사이의 임의의 값을 사용하였다. 그리고 누적데이터 개수 N 은 7을 선택하였다.



(a) 제안된 방법의 200epoch 학습 후 출력과 Ziegler-Nichols 방식의 제어기에 의한 출력과의 비교



(b) 학습 진행에 따른 PID 이득의 변화
그림5. 제안한 방식과 Ziegler-Nichols방식의 제어기와의 제안응답 비교

그림5는 제안된 방법을 DC전동기의 속도제어의 응용했을 때에 의한 제어 출력과 Ziegler-Nichols방법에 의한 제어 출력을 보여주고 있다. 제안된 방법에 의한 200epoch 학습 후 출력이 Ziegler-Nichols방법에 의한 출력보다 오버슈트, 언더 슈트, 세틀링타임 면에서 현저하게 개선된 것을 알 수 있다.

3. 결 론

본 논문에서는 PID 제어기 설계에 있어서 가장 문제가 되는 제어기 매개변수의 설정을 위한 방법으로 과거의 몇 구간에 대한 오차의 제곱을 누적시킨 비용함수를 설정하고 그 값이 감소하도록 제어기의 매개변수를 조정하는 방법을 제안하였다. 제안된 방법을 3차 공정과 4차 공정에 대해 시뮬레이션 하였고, 또한 DC 전동기의 속도제어에 적용하여 실험을 해보았다. 시뮬레이션과 실험을 통해서도 확인하였지만 제안된 신경망 제어기는 기존의 PID제어기의 구조를 그대로 유지하면서 적은 알고리즘 추가에 의해 Ziegler-Nichols 방법과 단지 $e_p^2(k)$ 만을 비용함수로 하는 기존 방법들에 의한 제어기보다 오버슈트, 언더슈트, 세틀링타임 면에서 개선된 결과를 얻었다. 또한 외란에 대해서도 제안된 방법을 사용하였을 경우 제어기 매개변수의 동조양상이 비교적 안정적임을 확인하였다.

앞으로의 과제는 제어기 및 식별기의 수렴성을 수학적으로 입증하는 것과 수렴성을 확보하면서 좀 더 빠른 학습을 가능하게 하는 최적 학습률을 결정하는 문제가 될 것이다.

(참 고 문 헌)

- [1] 정중대, "신경망에 의한 PID 제어기의 자동동조 및 직류 전동기 속도제어에의 응용", 한국 퍼지 및 지능 시스템 학회 논문지, Vol. 6, No. 2, 1996.
- [2] Chao-Chee Ku, Kwang Y. Lee, "Diagonal recurrent neural networks for dynamic systems control", IEEE Trans. Neural Networks, Vol. 6, No. 1, Jan, 1995.
- [3] 이권순, 이영진, 조현철, "신경회로망 예측 PID 제어시스템에 의한 다층 건축물의 진동제어", 한국 퍼지 및 지능 시스템 학회 논문지, Vol. 8, No. 7, 1998.
- [4] David M. Skapura, James A. Freeman, *Neural Networks*, Addison Wesley, 1992.
- [5] Howard B. Demuth, Mark Beale, Martin T. Hagan, *Neural Network Design*, PWS Publishing, 1996.
- [6] 이상원, *학습하는 기계 신경망*, Ohm사, 1995.
- [7] Masayoshi Tomizuka, Zhen-Yu Zhao, "Fuzzy gain scheduling of PID controllers", IEEE Trans. Systems, Man, and Cybernetics, Vol 23, No. 5, Sep. 1993.