

AC MOTOR의 속도제어 개선을 위한 신경망제어기의 설계

Design of Neuro Controller for Improving Velocity Control of AC Motor

실 재훈, 임 영도
동아대학교 전자공학과

요 약

본 논문에서는 신경회로망의 학습능력을 이용하여 AC 모터의 속도제어에 이용된 기존의 PI제어기의 문제점을 보완하고자 한다. 기존의 아날로그 PI제어기에서는 각 비례, 적분 파라미터를 개발자가 조정하여 고정하면 부하가 변동될 경우 적응성이 떨어지는 문제점을 안고 있었다. 본 논문에서 제시된 디지털 신경망제어기는 학습을 통해 새로운 환경에 적응가능하다는 점에 가정하여 설계하고 성능을 비교 평가하였다. 본 논문에서 사용된 신경회로망의 구조는 신경망중에서 가장 범용적으로 사용되는 다층 퍼셉트론 모델구조를 선택하였다. 신경망 제어기장치로는 인텔 8097 마이크로 콘트롤러를 이용하였다.

I. 서 론

신경망에 대한 높은 관심과 새로운 알고리즘의 발표에 따라 신경망이 여러 분야에서 응용되어 기존의 패턴인식분야 뿐만 아니라 제어분야에서도 성공적으로 적용되어 상당한 영역을 차지하고 있다. 특히 신경망은 자기 스스로를 조직화하는 학습의 형태를 가짐으로서 주위환경 변화에 따른 다양한 학습을 통해 적응이 가능하다는 큰 이점을 가지고 있다. 예를 들면 선형시스템 뿐만 아니라 비선형시스템에 대한 제어에서도 적절한 신경망모델을 선정, 학습시킴으로써 쉽게 접근 가능하도록 해주었다.

본 논문에서는 AC 모터의 역학적인 파라미터를 알지 못하는 상태에서 신경망을 제어기로 취하여 제어시켜 보았다. 본 논문에서 사용된 신경망모델은 다층 퍼셉트론 신경망모델이다. 다층 퍼셉트론 신경망모델은 신경망모델중 가장 범용적으로 사용되며, 본 논문에서도 기존 PI제어기를 대신하여 다층 퍼셉트론 신경망모델로 제어기를 설계한다.

본 논문의 연구 목적은 다음과 같다.

첫째로 기존의 DC 서보모터 응용분야보다 산업계에서 주로 사용하는 AC 모터에 다층 퍼셉트론

신경망모델을 적용시킴으로 다층 퍼셉트론 신경망모델 응용분야의 폭을 넓히고자 연구를 실시하였다.

둘째로 기존의 AC 모터에 사용된 제어기는 외부적인 영향, 즉 다양한 부하 변동 또는 외란에 대한 적응성이 떨어진다. 이에 본 논문에서는 다층 퍼셉트론 신경망모델의 역전파 학습을 통해 기존의 문제점을 해결하고자 한다.

끝으로 기존의 제어기는 아날로그 방식으로서 회로가 복잡할 뿐만 아니라 구성하기가 쉽지 않다. 이런 문제점을 본 논문에서는 인텔 8097 마이크로 컨트롤러를 사용하여 보다 간단한 디지털 방식의 회로로 구성하고자 한다.

II. 기존 PI제어기를 이용한 AC 모터의 속도제어

본 논문에서 사용된 기존의 PI제어기로 AC 모터를 제어하는 구조를 살펴보면, 그림 1과 같이 설정치와 귀환되는 실제출력치와의 오차를 PI제어기에 입력시켜 위상제어신호 U 를 출력한다. 이 위상제어신호 U 는 위상제어소자(TRIAC)에 입력되어 3상 Y결선으로 위상검출된 각 상의 위상을 제어신호에 따라 제어한다. 이렇게 제어된 3상의 위상신호는 모터에 가해져 설정치 속도를 얻도록 구성되며 이 때 모터는 TG에 의해 다시 모터의 실제출력치를 귀환시켜서 설정치와 비교한다.

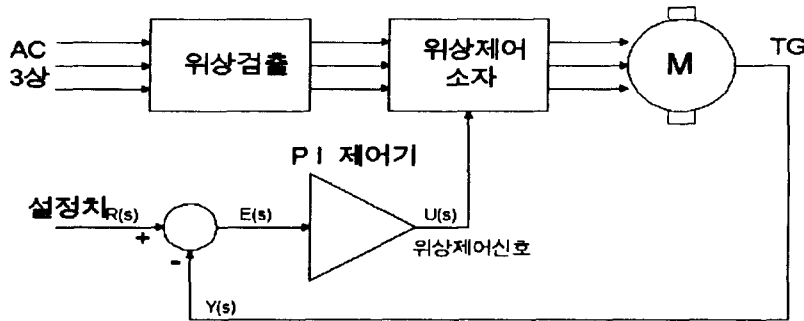


그림 1. PI제어기를 이용한 3상 AC 모터의 속도제어 계통도.

본 논문의 대상이 된 기존의 PI제어기를 사용함으로써 시스템에 맞는 적절한 비례, 적분 파라미터를 시행착오하여 설정하였다 하더라도 시스템이 바뀔 경우 또는 부하변동에 대하여 새로운 파라미터를 설정해 주어야 하는 문제점을 안고 있었다.

III. 신경망제어기를 이용한 AC 모터의 속도제어

본 절에서는 기존의 PI제어기를 대신하여 신경망을 이용한 제어기로 대체하려고 한다. 기존에 사용되었던 PI제어기는 개발자가 직접 비례, 적분 파라미터를 시스템에 맞도록 조정해 주어야 하는 번거움을 가지고 있다. 따라서 이러한 번거러움을 신경망의 학습을 통해 스스로가 시스템에 맞는 제어기로서 동작을 할 수 있도록 하고자 한다.

본 논문에서 설계된 신경망제어기는 설정치와 실제출력치와의 오차에 의해 신경망제어기 스스

로가 학습하여 대상 시스템에 적응시켜서 AC 모터로부터 원하는 속도를 제어하도록 이미 그림 1에서 보았던 것과 같이 구성한다. 신경망을 이용한 제어기의 제어계통도는 그림 2와 같다.

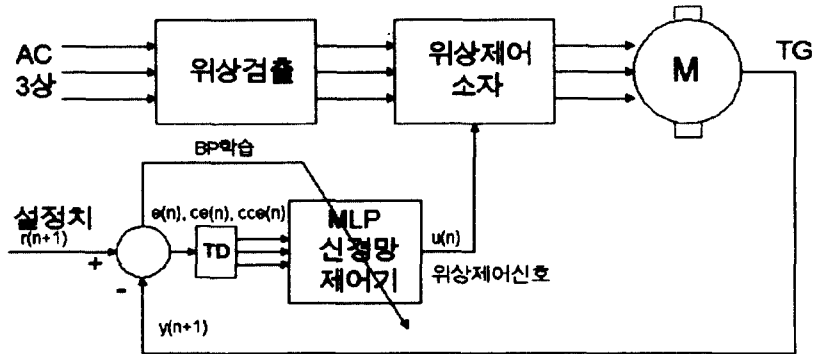


그림 2. 신경망제어기를 이용한 3상 AC 모터의 속도 제어계통도.

본 논문에서 설계된 신경망제어기의 내부구조는 그림 3과 같고 각 층의 뉴런의 개수와 뉴런의 활성화 출력치를 위한 임계함수는 다음과 같이 설계한다.

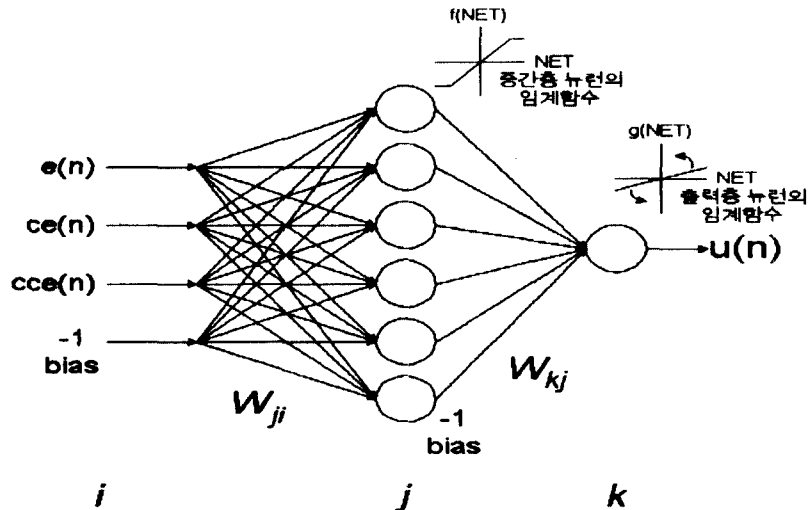


그림 3. 신경망제어기의 입력과 출력.

가. 신경망제어기에 입력되는 인자들은 그림 2와 같이 설정치와 실제출력치 비교에서 생기는 오차의 시간지연 성분 $e(n)$ 과 시간변화에 따른 오차변화 $ce(n)$ 과 오차변화의 변화치 $cce(n)$ 으로 구성된다. $e(n)$ 은 $e(n+1)$ 의 시간지연으로 $e(n)$ 은 식(1)로 나타내고 $ce(n)$ 와 $cce(n)$ 는 식(2)와 식(3)과 같다.

$$e(n) = TD [e(n+1)] = TD [r(n+1) - y(n+1)] \quad (1)$$

$$ce(n) = e(n) - e(n-1) \quad (2)$$

$$cce(n) = ce(n) - ce(n-1) \quad (3)$$

그리고 4번째 인자는 바이어스로서 중간층 뉴런의 입력에 첨가시켜서 보다 안정한 뉴런 출

력을 얻을 수 있도록 한다.

나. 신경망제어기의 중간층의 수와 각각의 중간층 뉴런의 개수는 시스템의 특성에 따라 설계자에 의해 융통성 있게 정할 수 있다. 그러므로 본 논문에서 중간층은 단일층으로 구성하고 중간층 뉴런의 개수는 5개로 한다. 그리고 입력층과 마찬가지로 하나의 바이어스를 첨가시키고 각 뉴런의 임계함수는 기울기가 1인 선형함수를 취한다.

다. 출력층 뉴런의 개수는 하나로서 제어신호 $u(n)$ 을 출력하도록 한다. 출력층 뉴런의 임계함수는 중간층과 같은 선형함수를 취하나 출력이득을 조정할 수 있도록 기울기를 설계자의 임의로 조정가능하도록 설계한다.

이렇게 설계된 신경망을 다음과 같이 순방향부분에 의해 제어신호를 얻고 역전파부분으로 학습을 하여 가중치를 수정한다.

순방향 부분은 먼저 중간층 뉴런의 출력을 구한다. 그리고 출력층 뉴런의 출력, 실제출력치를 구한다. 중간층 뉴런의 출력치 H_j 와 출력층 뉴런의 출력치 OUT_k 는 각각 식(4)와 식(5)와 같다.

$$H_j = f\left(\sum_{i=1}^1 W_{ji} \cdot X_i\right) \quad (4)$$

$$OUT_k = g\left(\sum_{j=1}^m W_{kj} \cdot H_j\right) \quad (5)$$

여기서, $f(\cdot)$ 와 $g(\cdot)$ 는 임계함수이다.

역전파 학습부분은 먼저 평가함수에 의해 설정치와 실제출력치의 오차를 먼저 구한다. 그리고 나서 델타규칙에 의해 출력층과 중간층 뉴런에 연결된 가중치 수정과 일반화된 델타규칙에 의해 중간층 뉴런과 입력들에 연결된 가중치를 수정한다. 이 때 평가함수는 식(6)과 같다.

$$E = \frac{1}{2} \sum_{k=1}^n (R_k - OUT_k)^2 \quad (6)$$

출력층 뉴런과 중간층 뉴런사이의 가중치는 식(7)과 식(8)에 의해 수정되며, 가중치의 수정은 음의 gradient 하강법에 의한다.

$$W_{kj}(n) = W_{kj}(n-1) + \Delta W_{kj}(n) \quad (7)$$

$$\begin{aligned} \Delta W_{kj}(n) &= -\eta \frac{\partial E}{\partial W_{kj}} + \alpha \Delta W_{kj}(n-1) \\ &= \eta \delta_k H_j + \alpha \Delta W_{kj}(n-1) \\ &= \eta (R_k - OUT_k) g'(NET_k) H_j + \alpha \Delta W_{kj}(n-1) \end{aligned} \quad (8)$$

중간층 뉴런과 입력들사이의 가중치는 식(9)와 식(10)에 의해 수정되어진다.

$$W_{ji}(n) = W_{ji}(n-1) + \Delta W_{ji}(n) \quad (9)$$

$$\begin{aligned} \Delta W_{ji}(n) &= -\eta \frac{\partial E}{\partial W_{ji}} + \alpha \Delta W_{ji}(n-1) \\ &= \eta \delta_j X_i + \alpha \Delta W_{ji}(n-1) \\ &= \eta f'(NET_j) \sum_{k=1}^n (\delta_k W_{kj}) X_i + \alpha \Delta W_{ji}(n-1) \end{aligned} \quad (10)$$

여기서, R_k : 설정치,

$$NET_k = \sum_{j=1}^m W_{kj} \cdot H_j, \quad NET_j = \sum_{i=1}^n W_{ji} \cdot X_i,$$

η : 학습을 또는 학습속도계수,

δ_k, δ_j : 출력층의 오차항, 중간층의 오차항,

α : 모멘트항에서의 모멘트 계수.

IV. 실험 및 결과

본 절에서는 앞에서 설계된 신경망제어기를 18097 마이크로 콘트롤러에 어셈블리로 프로그램하여 구성하고 대상 시스템을 접속한다. 그리고 나서 기존의 PI제어기와 설계된 신경망제어기의 성능을 평가 하고자 한다.

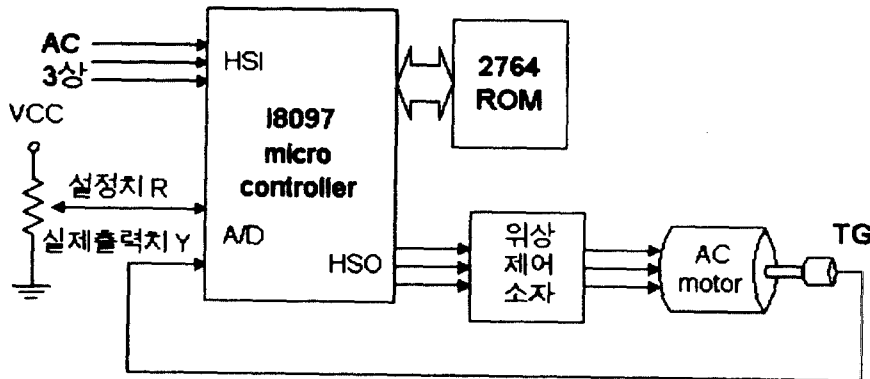


그림 4. 신경망제어기의 하드웨어 구성도.

신경망제어기의 하드웨어 설계는 다음과 같다.

- 가. 먼저 220V 3상 AC를 검출하기 위하여 마이크로 콘트롤러의 HSI(High Speed Input)핀을 이용한다. 이렇게 하여 3상의 위상을 얻는다.
- 나. 오차 계산을 위한 설정치 입력, 실제출력치 입력은 각각 A/D 컨버터의 2채널로 입력받도록 한다. 여기서 입력되는 모터의 실제출력치는 모터의 TG에 의해 구해진다.
- 다. 입력된 설정치와 실제출력치의 오차에 의해 마이크로 콘트롤러에 프로그램된 신경망에서 계산된 각상의 제어신호를 HSO(High Speed Output)핀을 사용하여 출력한다. 이 때 마이크로 콘트롤러로부터 출력된 각상의 제어신호는 위상제어소자에서 위상을 제어하여 제어된 3상의 위상을 AC 모터에 입력한다.

라. 신경망 프로그램은 ROM을 사용하여 전용 제어기로 동작하도록 설계한다.

이렇게 설계된 신경망제어기와 기존 PI제어기를 성능 평가한다. 실험은 무부하시와 부하시로 나누어 실시한다. 성능평가결과는 그림 5 ~ 그림 8과 같다.

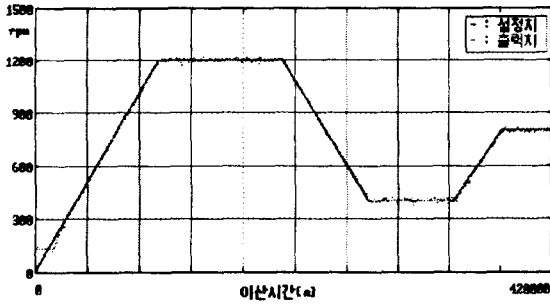


그림 5. 무부하시 PI제어기의 출력

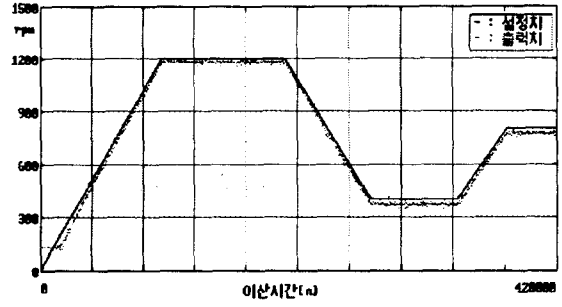


그림 6. 부하시 PI제어기의 출력

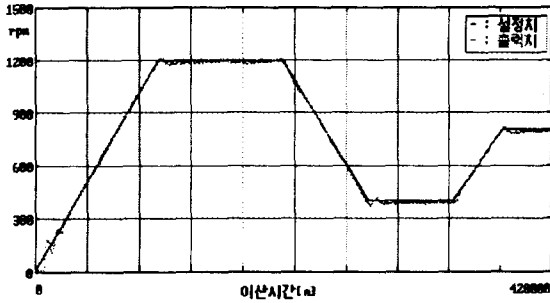


그림 7. 무부하시 신경망제어기 출력

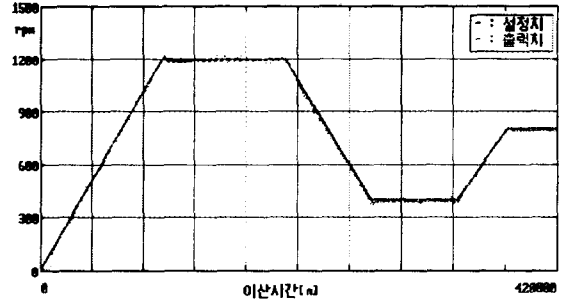


그림 8. 부하시 신경망제어기 출력

V. 결 론

본 논문에서는 기존 PI제어기에서 문제시 되는 부하변동에 따른 적응성에 대하여 이 문제점을 보완하기 위해 신경망제어기를 설계하여 제안하였다. 실험의 성능평가 결과 무부하시는 두 제어기 모두 다 안정적인 제어가 되지만 부하가 변동되었을 경우 PI제어기는 설정치 속도보다도 속도가 떨어져서 제어됨을 보여주었으나 신경망제어기는 부하변동에 적응하여 설정치대로 속도가 제어되었다. 본 논문에서 제안된 AC 모터제어를 위한 신경망제어기는 부하변동이 심한 산업체의 시스템 적용에 적절할 것으로 본다.

VI. 참고문헌

- [1] Jacek M. Zurada, "Multilayer Feedforward Networks," *Introduction to Artificial Neural Systems*, West publishing Company, pp. 175- 206, 1992.
- [2] Katsuhiko Ogata, "Pulse transfer function of a digital PID controller," *Discrete time control systems*, Prentice-Hall Inc., pp. 199-207, 1987.
- [3] K. J. Astrom and B. Wittenmark, "Adaptive control : Self-Tuning Control," *Computer-controlled systems*, Prentice-Hall Inc. pp. 438-440, 1990
- [4] M. M. Nelson and W. T. Illingworth, "How do Neural Networks Work?," *A practical guide to Neural Nets*, Addison-Wesley Publishing Company, Inc., pp. 36-58, March. 1991.