

뉴로-퍼지 알고리즘을 이용한 유도전동기 속도 제어

임태우^o, 이동운, 안태천

원광대학교 제어계측공학과 지능정보시스템 연구실

Speed Control of Induction Motor Using Neuro-fuzzy Algorithm

Tae-woo Lim^o, Dong-yoon Lee, Tae-chon Ahn

Dept. of Control and Instrumentation Engineering Wonkwang University

Abstract - 논문에서는 유도전동기의 동특성을 효율적으로 제어하기 위해서 고정자 전류를 토크성분 전류와 자속성분 전류로 나누어 독립적으로 제어하는 우수한 백터제어 기법을 사용하고 있는 유도전동기 제어 시스템에 광범위한 영역에서도 제어성능과 부하특성을 향상시킬 수 있게 하는 지능형 뉴로-퍼지 알고리즘에 의한 고성능의 속도 제어기를 설계한다. 그리고 제안한 뉴로-퍼지 속도 제어기의 우수성을 시뮬레이션과 실제 시스템에의 적용을 통하여 입증한다.

1. 서 론

전동기의 가변속 구동은 많은 산업 분야에서 그 필요성이 증대되고 있다. 이러한 요구를 위한 과거에는 제어가 쉬운 직류 전동기가 널리 사용되어 왔다. 교류 전동기는 기계적으로 견고하고 직류전동기에 비하여 유지·보수가 간단한 반면에 종래에는 제어의 어려움과 주파수 변환 장치의 가격으로 인하여 가변속 구동에 널리 사용되지 못하였다.

공정분야에서의 유도전동기는 PID 제어기 적용이 많은 상태이었다. PID제어기는 공정분야에서 많이 사용되는 알고리즘으로 근래에 우수하고 견고한 제어이론들이 개발되었음에도 불구하고 PID 제어가 가장 많이 사용되고 있는 이유는 제어기의 구조가 간단하며 제어대상의 지연시간이 크거나 비선형 특성을 갖지 않는다면 만족스러운 제어 목적을 달성할 수 있기 때문이다. 본 논문에서는 지능형 뉴로-퍼지 알고리즘에 의한 속도 제어기를 제안하여 유도전동기의 속도제어 시스템을 제안한다.

2. 본 론

2.1 뉴로-퍼지

뉴로-퍼지 네트워크는 퍼지제어기의 연산과정을 수행할 수 있어야 하므로 각 계층 뉴런의 활성화함수는 이러한 조건을 만족하도록 선정되어야 한다. 또한, 계층의 학습으로 뉴로-퍼지 네트워크내의 모든 파라미터들을 조정할 수 있어야 하며, 학습으로 조정할 파라미터의 수를 최소화시켜 학습이 효율적으로 이루어져야 한다. 또한, 학습이 완료된 후 실험에서 사용하기 때문에 전체 뉴런수가 적어지도록 네트워크를 구성하여 연산시간이 짧아져야 한다. 이러한 문제점들은 퍼지제어기의 모든 연산 기능을 수행하도록 뉴로-퍼지 네트워크의 구조와 각 계층의 활성화함수를 정의함으로써 해결될 수 있다.

퍼지제어기에서 전건부 언어변수를 5-7정도로 선정하는 것이 일반적이며, 본 연구에서는 계층 1은 2개의 입력변수를 각각 3개와 5개의 퍼지집합으로 변환하기 위하여 6개와 10개의 뉴런이 있으며, 계층 2-4는 각 퍼지집합에 대하여 제어규칙의 추론연산을 수행하도록 각각 9개와 25개의 뉴런으로 구성되며, 각 계층의 뉴런 수는

표 2-1과 같다.

표 2-1. 각 계층의 뉴런수

		입력	계층	계층	계층	계층	출력
		층	1	2	3	4	층
뉴런 수	전건부 (5개)	2	10	25	25	25	1

또한, 각 계층의 활성화 함수는 제어규칙의 추론연산을 수행하도록 설정하였다. 이러한 뉴로-퍼지 네트워크의 구조는 피드포워드 신경회로망의 구조와 동일하게 되며, 각 계층의 활성화함수를 미분 가능한 것으로 하면 오차역전파 알고리즘을 이용하여 최적의 제어가 구현되도록 전건부와 후건부의 각 파라미터들을 조정할 수 있게 된다. 제어규칙은

$$\text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } B_1 \text{ then } f = P_1x_1 + Q_1x_2 + R_1$$

$$\text{If } x_1 \text{ is } A_2 \text{ and } x_2 \text{ is } B_2 \text{ then } f = P_2x_1 + Q_2x_2 + R_2 \quad (2-1)$$

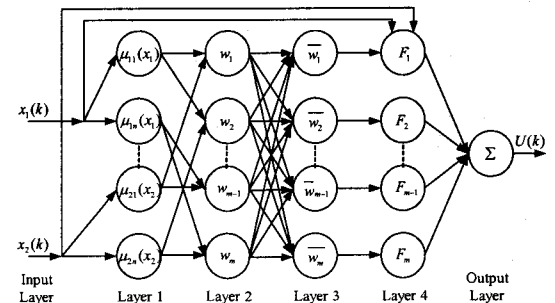
⋮

$$\text{If } x_1 \text{ is } A_m \text{ and } x_2 \text{ is } B_m \text{ then } f = P_mx_1 + Q_mx_2 + R_m$$

형태로 구성되어 있다. 이러한 제어규칙은 퍼지제어와 선형 제어의 특성을 조합한 제어를 구성할 수 있다.

식 (2-1)의 전건부 퍼지집합 A_m 과 B_m 의 소속함수 $\mu_{1m}(x_1)$ 과 $\mu_{2m}(x_2)$ 는 범종형으로 계산되며, 최적의 소속함수는 범종형 소속함수의 파라미터들을 조정함으로써 결정할 수 있다.

그림 2-1은 뉴로-퍼지 네트워크의 구조로서 표 2-1과 같은 구조를 가지며, 각 계층의 구조는 다음과 같다.



m : 전체 Rule 갯수 n : 각 입력에 대한 멤버십 함수 갯수

그림 2-1. 뉴로-퍼지 네트워크의 구조

입력 층은 입력 x_1 과 x_2 의 전달기능을 한다.

Layer 1은 입력 x_1 과 x_2 에 대하여 전건부 퍼지집합 A_m 과 B_m 의 소속함수 $\mu_{1m}(x_1)$ 과 $\mu_{2m}(x_2)$ 를 입력받으며, 각 뉴런들은 식 (2-2)와 같은 활성화함수를 가진다.

$$\mu_{1m}(x_1) = \exp\left(-\frac{(x_1 - b_{1m})^2}{2a_{1m}}\right) \quad (2-2)$$

$$\mu_{2m}(x_2) = \exp\left(-\frac{(x_2 - b_{2m})^2}{2a_{2m}}\right)$$

식 (2-2)의 정상화강법에 의하여 Layer 1의 각 뉴런 퍼지집합에 대한 소속함수가 계산된다. 식 (2-2)에서 a_{1m} 와 a_{2m} , 그리고 b_{1m} 와 b_{2m} 는 각 소속함수의 중심값을 결정해 주는 파라미터와 그리고 각 소속함수의 기울기로 제어 성능에 영향을 미치는 전건부 파라미터들이다. 이들은 시스템의 오차가 감소하도록 학습에 의해 조정되므로, 퍼지 제어기의 설계과정에서 나타나는 반복시행 없이 최적의 제어를 구현할 수 있으며, 최대 값 1, 최소 값 0인 범중형 소속함수를 계산하며, 학습으로 조정할 파라미터는 2개이고 입력의 모든 영역에서 미분 가능하여 오차역전과 알고리즘의 사용이 가능하게 된다.

Layer 2는 퍼지제어 규칙의 IF 부분의 연산을 수행하여, 계층 1의 각 뉴런들의 출력을 조합하여 전건부 조건을 구성하게 된다. 계층 2의 뉴런은 식 (2-3)과 같이 입력된 두 신호를 곱하여 출력하므로, 퍼지논리의 min연산을 수행하며, 각 규칙들의 활성화강도(firing strength) 또는 가중치(weight) w_m 를 출력하게 된다.

$$w_m = \mu_{1m}(x_1)\mu_{2m}(x_2) \quad (2-3)$$

Layer 3은 전체 규칙의 활성화강도의 합에 대한 각 규칙의 활성화강도의 비율을 계산하여, 각 규칙의 정규화 활성화강도(normalized firing strength) $\overline{w_m}$ 를 출력하며, 식 (2-4)의 기능을 갖는 뉴런들로 구성되어 있다.

$$\overline{w_m} = \frac{w_m}{\sum_{m=1}^n w_i} \quad (2-4)$$

Layer 4는 해당 규칙의 후건부를 구성하며, 각 뉴런들은 식 (2-5)와 같은 함수기능을 갖는다.

$$F_m = \overline{w_m}(P_mx_1 + Q_mx_2 + R_m) \quad (2-5)$$

식 (2-5)에서 P_m, Q_m, R_m 는 상태 평가형 제어규칙의 후건부 파라미터(consequent parameter)들로서 시스템의 오차가 감소하는 방향으로 학습에 의해 조정된다. 출력 층은 식 (2-6)과 같이 입력되는 값들을 합하여 퍼지제어기의 출력을 만들어 준다.

$$U(k) = \sum_{m=1}^n F_i \quad (2-6)$$

식 (2-2)-식 (2-6)에서 m은 x_1 과 x_2 에 대한 전건부 언어변수의 수에 의한 전체 Rule의 개수가 되며, 본 논문에서 5개의 두 가지의 경우를 고려한다.

그림2-3은 제어규칙의 추론방법을 나타낸 것이다. 이러한 추론의 연산과정은 뉴로-퍼지 네트워크의 내부에서 이루어지며, 추론방법의 연산과 무게중심법의 비퍼지화 연산을 수행하게 된다. 따라서, 뉴로-퍼지 네트워크는 퍼지제어기와 동일한 성능을 가지게 되며, 퍼지논리적 구조와 신경회로망 학습능력을 융합한 구조가 된다. 그림 2-3의 추론방법은 두 개의 입력에 따른 소속함수의 연산 과정을 통하여 3개와 5개의 언어변수에 대한 퍼지집합을

계산하고, 퍼지규칙을 이용하여 각 규칙의 출력을 계산한다. 각 규칙의 출력은 비퍼지화에 의해서 토크성분 전류로 계산된다.

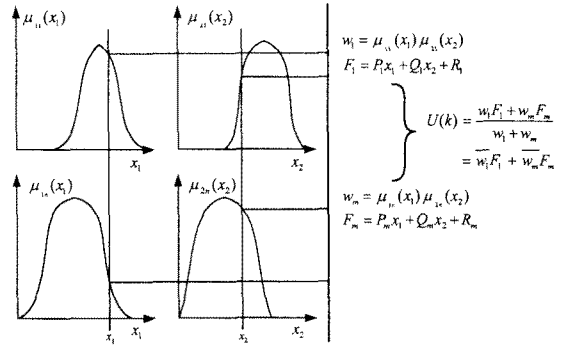


그림 2-3. 제어규칙의 추론방법

뉴로-퍼지 네트워크의 학습은 그림 2-3의 좌측의 범중형 소속함수 중심값 b_{1m}, b_{2m} 와 소속함수의 폭을 결정하는 a_{1m}, a_{2m} 인 전건부 파라미터와 후건부 파라미터인 P_m, Q_m, R_m 를 조정하여 제어목적에 적합한 출력 $U(k)$ 가 계산 되도록 한다.

2.2 뉴로-퍼지 제어기의 학습

유도전동기 구동 시스템에 뉴로-퍼지 제어기를 사용하기 위하여, 뉴로-퍼지 네트워크는 오차역전과 알고리즘으로 지도제격을 학습하여 뉴로-퍼지 제어기로 구현한다. 학습이 완료된 뉴로-퍼지 제어기는 외란이나 시스템의 파라미터 변화에 강인성을 가질 뿐만 아니라 오버슈트와 응답시간을 줄일 수 있게 되며, 토크성분 전류의 맥동을 감소시켜 전류 고조파성분과 운전 소음을 감소시켜 준다.

뉴로-퍼지 네트워크의 각 파라미터들은 매 샘플링 시간 시스템의 오차를 역전과 시켜주는 시제역전과(temporal back propagation)알고리즘에 의해 조정되며, 최적의 제어기가 구성되도록 해 준다. 각 샘플링 순간의 시스템 평가 함수는

$$E_k = \frac{1}{2} (x_1(k+1) - x_d(k+1))^2 \quad (2-7)$$

가 되며, $x_1(k+1)$ 은 k 번째 상태값 $x_1(k)$ 과 $x_2(k)$ 에 의하여 뉴로-퍼지 제어기로부터 출력된 제어 입력 $U(k)$ 에 의한 유도 전동기의 각 위치제적의 k+1번째 값이며, $x_d(k+1)$ 은 지도 제적의 k+1번째 값이다. 유도 전동기의 토크 동특성 식은

$$T_e = J \frac{d\omega_m}{dt} + B\omega_m + T_l \quad (2-8)$$

$$\theta_m = \int \omega_m dt \quad (2-9)$$

이며, J 는 회전관성계수, B 는 마찰계수, T_l 은 부하토크이다. 식 (2-8)과 (2-9)를 상태변수 식으로 표현하면 식 (2-10)과 같다.

$$\dot{x}_1 = \dot{\theta} = \omega_m = x_2 \quad (2-10)$$

$$\dot{x}_2 = \dot{\omega}_m = (T_e - T_l - B\omega_m) \frac{1}{J}$$

유도 전동기의 출력 토크는 $T_e = K_t i_\infty i_{\dot{\theta}}$ 이며, 본

연구에서 사용하는 일정 자속 제어방식에서는 $i_{\alpha} = \text{constant}$ 이므로, 출력 토크 $T_e = K' i_{\alpha}$ 로 표현되며, i_{α} 를 U 로 하여 식 (4-10)을 표현하면 식 (2-11)과 같아 된다.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{B}{J} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{K'}{J} \end{bmatrix} U + \begin{bmatrix} 0 \\ -\frac{1}{J} \end{bmatrix} T_i \quad (2-11)$$

부하 토크 항을 무시하면

$$\begin{aligned} \dot{x}(t) &= Ax(t) + BU(t) \\ y(t) &= Cx(t) \end{aligned} \quad (2-12)$$

가 되며 식(2-12)에서 A, B, C 는 식 (2-13)과 같다.

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{B}{J} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ -\frac{K'}{J} \end{bmatrix}, \quad C = [1 \ 0] \quad (2-13)$$

또한 디지털 제어를 위해서는 이산치 상태 방정식으로 변환하는 과정이 필요하며 샘플링 주기를 T 로 하면

$$U(kT) = U(kT) = \text{constant} : (kT \leq t < (k+1)T) \quad (2-14)$$

이 되므로, 이산화된 상태 방정식은

$$\begin{aligned} x((k+1)T) &= \bar{A}(T)x(kT) + \bar{B}(T)U(kT) \\ y(kT) &= Dx(kT) \end{aligned} \quad (2-15)$$

로 표현되며, 식 (4-15)에서 \bar{A} 와 \bar{B} 및 D 는

$$\begin{aligned} \bar{A}(T) &= e^{AT} \\ \bar{B}(T) &= \int_0^T e^{A(T-\tau)} B d\tau \\ D &= C \end{aligned} \quad (2-16)$$

로 계산된다. 식(2-16)을 이용하여 \bar{A} 와 \bar{B} 를 구하면,

$$\bar{A} = \begin{bmatrix} 1 & \frac{J}{B}(1 - e^{-\frac{B}{J}T}) \\ 0 & e^{-\frac{B}{J}T} \end{bmatrix} \quad (2-17)$$

$$\bar{B} = -\frac{K'}{B} \begin{bmatrix} T - \frac{J}{B} + \frac{J}{B} e^{-\frac{B}{J}T} \\ 1 - e^{-\frac{B}{J}T} \end{bmatrix}$$

가 된다. 유도 전동기의 다음 상태는

$$\begin{bmatrix} x_1((k+1)T) \\ x_2((k+1)T) \end{bmatrix} = \bar{A} \begin{bmatrix} x_1(kT) \\ x_2(kT) \end{bmatrix} + \bar{B}U(kT) \quad (2-18)$$

이므로, 유도 전동기의 k+1번째 샘플링 값 $x_1((k+1)T)$ 와 $x_2((k+1)T)$ 는

$$\begin{bmatrix} x_1((k+1)T) \\ x_2((k+1)T) \end{bmatrix} = \begin{bmatrix} 1 & \frac{J}{B}(1 - e^{-\frac{B}{J}T}) \\ 0 & e^{-\frac{B}{J}T} \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \end{bmatrix} - \frac{K'}{B} \begin{bmatrix} T - \frac{J}{B} + \frac{J}{B} e^{-\frac{B}{J}T} \\ 1 - e^{-\frac{B}{J}T} \end{bmatrix} U(kT) \quad (2-19)$$

가 되며, 따라서 k+1 번째 샘플링시간의 플랜트를 통과한 오차 $\delta_{E,k}$ 는

$$\begin{aligned} \delta_{E,k} &= -\frac{\partial E_k}{\partial U(kT)} \\ &= -\frac{\partial}{\partial U(kT)} \frac{1}{2} (x_1((k+1)T) - x_d((k+1)T))^2 \\ &= -\frac{K'}{B} (T - \frac{J}{B} + \frac{J}{B} e^{-\frac{B}{J}T}) (x_1((k+1)T) - x_d((k+1)T)) \end{aligned} \quad (2-20)$$

로 되며

$$-\frac{K'}{B} (T - \frac{J}{B} + \frac{J}{B} e^{-\frac{B}{J}T}) = \text{constant} = K \quad (2-21)$$

이므로 $\delta_{E,k}$ 는 식(2-22)와 같이 계산된다.

$$\delta_{E,k} = K [x_1((k+1)T) - x_d((k+1)T)] \quad (2-22)$$

식(2-22)를 뉴로-퍼지 제어기의 학습 오차량으로 사용함으로써 계산시간이 단축되며, 네트워크 구조가 간단히 구성될 수 있어서 실시간 제어를 가능하게 해준다. 뉴로-퍼지 제어기와 플랜트를 직접 연결한 구조에서 플랜트를 통과하는 역전과 오차는 식(2-22)으로부터 토크 성분 전류 U 와 각위치 x_1 의 1차 비례로 하여 학습을 실시한다. 즉 “토크성분 전류의 증가에 의해서 각속도가 증가하고 따라서 각 위치가 증가한다”는 플랜트에 대한 최소한의 지식만으로 학습을 실시한다.

매 샘플링마다 시스템의 오차를 역전과 시켜서 뉴로-퍼지 제어기 내의 파라미터들을 조정한다. 학습에 사용된 각 파라미터의 δ 값들은 오차역전과 알고리즘에서의 오차량 계산과정을 이용하여 아래 식으로 구한다. 샘플링 시간 k+1 의 플랜트를 통과한 $\delta_{E,k}$ 는 식(2-23)으로 하여 학습에 사용한다.

$$\delta_{E,k} = -\frac{\partial E_k}{\partial U} \propto K [x_1((k+1)T) - x_d((k+1)T)] \quad (2-23)$$

학습은 뉴로-퍼지 제어기 내의 전진부와 후진부의 각 파라미터 초기화로부터 시작되며, 뉴로-퍼지 제어기에 출력에 의한 유도전동기의 출력과 지도 케적간의 오차를 계산하고, 각 파라미터들은 시계역전과 알고리즘에 의하여 오차가 감소하는 방향으로 조정된다.

이러한 과정은 전체 지도케적에 대하여 반복 실시하게 되며, 학습이 완료된 후 시스템의 최대오차와 허용오차 범위 이하로 감소하거나 증가 또는 발진하면 각 파라미터의 학습률과 지도케적을 변화시켜서 다음 단계 학습을 실시한다. 각 단계 학습의 결과케적과 조정된 파라미터들은 다음 단계 학습의 지도케적과 초기 파라미터로 사용하며, 학습은 정상상태 오차가 정해진 규격 이내로 될 때까지 실시한다.

2.3 시뮬레이션

제안된 알고리즘의 성능을 알아보기 위하여 시뮬레이션은 부하의 변화 없이 일정하며, 유도전동기의 속도는 1200rpm에서 1500rpm으로 가변을 하는 형태의 변화를 주어 시뮬레이션 하였다. 또한 백터제어로서의 속도특성과 PID 속도제어기에 의한 방법과 본 논문에서 제안한 뉴로-퍼지 속도제어기의 성능을 비교를 하였다.

3. 결 론

그림2-4 ~ 그림2-6에서 벡터제어, PID 속도제어기, 본 논문에서 제안한 뉴로-퍼지 속도제어기에 의한 계-의 변화와 전류의 변화 특성을 보인다. 이상의 내용을 바탕으로 하여 전체적인 성능을 비교한 결과 다음과 같은 결과를 얻을 수 있다.

1) 부하가 있을 경우에는 전건부 언어변수가 5개인 뉴로-퍼지 제어기가 전건부 언어변수가 벡터 제어기법과, PID속도제어법 보다 오버슈트가 작고 전체적인 성능에서 우수함을 알 수 있다.

2) 뉴로-퍼지 제어기의 전건부 및 후건부의 파라미터를 Look-up 테이블로 만들어 사용함으로써, 뉴로-퍼지 제어기의 사용에 의해 3상 유도전동기를 제어할 수 있으며, 우수한 성능을 나타냄을 알 수 있다.

제안된 뉴로-퍼지 제어기는 빠른 응답 특성과 정밀도를 요하는 분야에서 서보 제어용으로 효과적으로 적용될 수 있을 것이며, 최근 연구가 활발히 진행되고 있는 전기자동차 등에 동력용으로 응용될 수 있을 것으로 기대된다.

[참 고 문 헌]

- [1] F. Blaschke, "The principle of field orientation as applied to the new transvector closed loop control system for rotating field machines", Siemens Review, Vol. 34, pp. 217-221, 1972.
- [2] S. Beierke, "Enhanced fuzzy controlled AC motor using DSP", Embedded Systems Conference, pp. 545-553, 1995.
- [3] C. H. Tsa, H. C. Lu, "Design and implementation of a DSP-based grey-fuzzy controller for induction motor drive", Electric Machines & Power Systems, Vol. 28, pp. 373-384, 2000.
- [4] Y. Kinpara, M. Koyama, "Speed sensorless vector control method of induction motor including a low speed region", Trans. of the Inst. of Electrical Engineers of Japan D - A Publication of Industry Applications Society, Vol.120-D, 223-229, 2000
- [5] C. Y. Huang, T. C. Chen, C. L. Huang, "A microcomputer-based induction motor drive system using current and torque control", IEEE Transactions on Energy Conversion, Vol. 14, pp. 874-880, 1999.
- [6] B. K. Bose, "Power electronics and AC drivers", Prentice-Hall, 1986.
- [7] P. C. Krause, "Analysis of electric machinery", McGraw-Hill Book, 1987.
- [8] M. Ouhrouche, "Vector control of an induction motor with on-line rotor resistance identification ", Proc. of the 1999 IEEE Canadian Conf. on Elec. and Computer Eng., Vol. 2 , pp. 1121-1125 , 1999.
- [9] G. Minas, J. S. Martins, C. Couto, "A microcontroller based voltage space vector modulator suitable for induction motor drives", Proc. of the IEEE International Symp. on Industrial Electronics, Vol. 2, pp. 469-473, 1999.
- [10] A. Tewari, A. Tripathi, S. P. Das, "A fuzzy logic controller based indirect field-oriented induction motor drive system", Proc. of the International Conf. on Industrial Tech. 2000, pp. 359-364, 2000.
- [11] G. C. D. Sousa, B. K. Bose, J. G. Cleland, "Fuzzy logic based on-line efficiency optimization control of an indirect vector controlled induction motor drive", IEEE Trans. on Industrial Electronics, Vol. 42 No. 2, 1995.
- [12] M. F. Tsai, Y. Y. Tzou, "An adaptive current control scheme for high performance AC drives", Proc. of the 23rd International Conf. on Industrial Elec., Control and Instrumentation, Vol. 1, 1997.
- [13] J. Holtz, "Pulsewidth modulation a survey", Conf. Rec. of IEEE PESC, pp. 11-18, 1992.

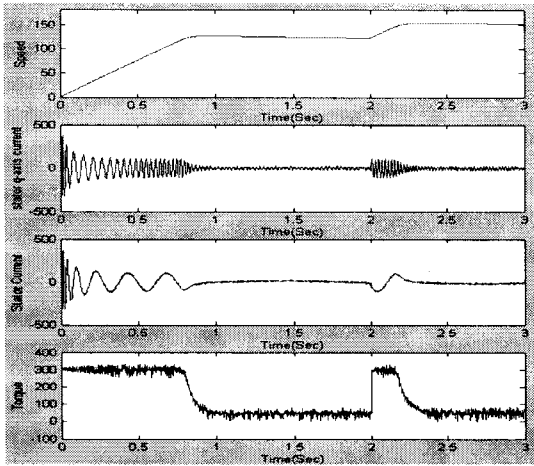


그림 2-2 벡터제어 3sec 동안 시뮬레이션

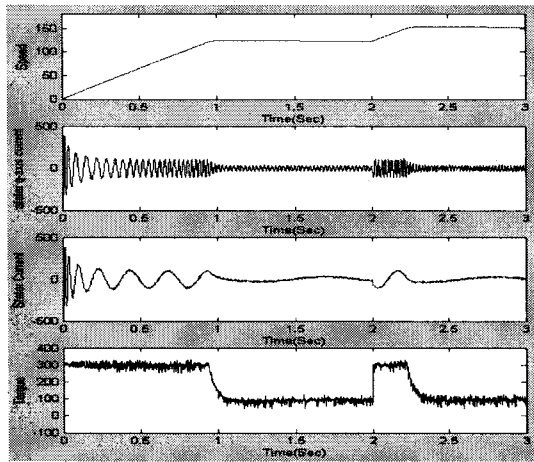


그림 2-3 PID 속도제어기에 의한 3sec 시뮬레이션

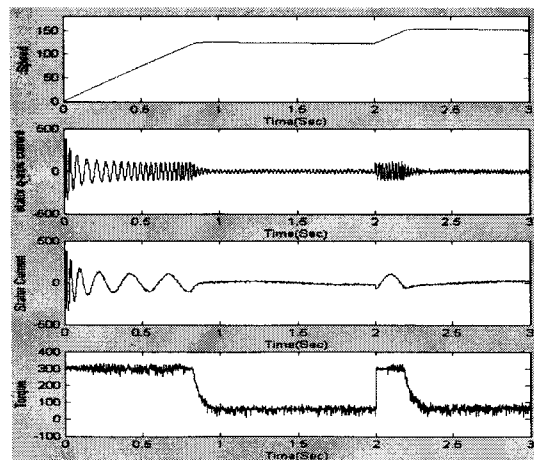


그림 2-4 뉴로-퍼지 속도제어기에 의한 3sec 시뮬레이션