

## 적응 IP 제어를 이용한 유도전동기의 벡터제어

조지원<sup>\*</sup> 이정민 박수영 최규하 김한성  
 건국대학교 전기공학과

### Vector Control of Induction Motor using Adaptive IP Controller

Jee-Won Joe<sup>\*</sup> Jeong-Min Lee Su-Young Park Gyu-Ha Choe Han-Sung Kim  
 Department of Electrical Engineering Kon-Kuk University

#### Abstract

In vector control of induction motor, to improve stability of system generated by load and disturbance have used the speed controller. however this regulator is unquestionably the most common PI controller.

If it is tuned, it's performance become satisfactory. In spite of this, in system requiring more complete control algorithms, the ability of controller is loosed totally. Therefore, in this system, it is requiring to tune controller and to design more complex regulator. It may also be costly and time consuming to tune such regulator.

This paper proposes in vector control that implement more complete speed control using adaptive IP controller.

#### 1. 서론

인버터를 이용한 유도기의 가변속제어는 최근에 활발히 연구되어지고 있다. 이런 유도기의 가변속제어에서 고효율화, 신뢰화를 얻기 위해서는 벡터 또는 자속기준제어가 필수적이다. 반면 기존의 V/F제어 방식은 자속 성분 전류와 토오크 성분 전류가 분리되지 않으며 상호 간섭이 생겨 원하는 토오크를 정확히 발생시킬 수 없게된다. 벡터제어의 기본원리는 회전자자속을 일정하게 유지하고 고정자자속과 전류벡터사이의 각을 90°로 유지시키는 방법이다. 실행에 있어서는 전동기의 매개변수 특히 회전자 저항은 온도, 포피효과 등에 따라 상당히 넓게 변하게 된다. 이러한 현상은 회전자 자속과 전류벡터사이의 직교성을 상실하게 되어 제어성능에 손실을 입히게 된다. 실제적으로 고정자 저항이 제어루프에서 가정된 값보다 작게 된다면 자속의 부족이 발생하게 되어 토오크에서의 비례적인 현상을 나타내게 된다. 또한 벡터제어의 목적은 유도기를 타어자 직류전

동기와 같이 운전하는 것이므로 직류전동기 제어에서 문제되는 속도제어기에 대한 연구가 필요하다. 속도제어기는 서보 시스템의 성능을 좌우하므로 상당히 중요하고 보통은 PI나 PID제어기를 사용하고 이러한 제어기는 적절히 tuning 되었을 때는 만족스런 특성을 나타낼 수 있지만 원하는 특성을 얻기 위해 제어기의 매개변수를 시스템 특성에 맞게 적절히 결정하여야 하므로 시스템의 매개변수를 정확히 알아야 하고 부하의 변동 또는 외란에 의해 시스템의 매개변수들이 변하게 되면 제어기 매개변수를 그때마다 다시 조정해야 한다.

이러한 점은 시간과 경비면에서 불합리하므로 본 논문에서는 시스템의 매개변수를 순환최소 자승법으로 추정하여 이를 참 값이라고 가정하고 이로부터 제어기 매개변수를 구하는 방식인 IP 자기동조기 (STR, Self-Tuning Regulator)를 사용하고 있다. 또한 제어기는 시스템의 구을 원하는 곳으로 옮기는 극배치방식을 사용하고 있다. 이러한 형태의 제어기를 사용함으로써 시뮬레이션을 통하여 만족스런 특성을 얻을 수 있다는 것을 확인하였다.

#### 2. 유도기의 고정자 자속 기준 제어 [2]

벡터제어의 직접 방식에서의 고정자 및 회전자 자속은 다음과 같이 구해진다.

$$\lambda_m = \int (v_m - R_m i_m) dt \quad (2-1)$$

$$\lambda_r = \frac{L_r}{L_m} (\lambda - \sigma L_m i_m) dt \quad (2-2)$$

$$\sigma = 1 - \frac{L_m^2}{L_s L_r} \quad (2-3)$$

식 (2-1)에서 알 수 있듯이 고정자 자속 연산값의 정확도는  $R_m$ 의 정확도에 따라 변환된다. 일반적으로 고정자 저

량은 비교적 정확히 구할 수 있고 온도에 따른 변화도 적다.<sup>[11]</sup> 그러나 회전자 자속을 구하기 위해서는 누설인덕턴스의 양이 들어감으로서 정확도가 떨어지게 된다. 따라서 회전자 자속을 연산하여 벡터제어를 하는 방식보다는 고정자 자속을 이용하여 벡터제어를 수행하면 좀 더 정확한 벡터제어를 구현할 수 있다.

회전자 자속 기준 제어 시스템의 경우 모든 변수는 회전자 자속을 기준축으로 하여 표현된다.

$$(1+\sigma_{Trp}) \lambda_{dr} = L_m i_{ds} + \omega_{s1} Tr \lambda_{qr} \quad (2-4)$$

$$(1+\sigma_{Trp}) \lambda_{qr} = L_m i_{qs} - \omega_{s1} Tr \lambda_{dr} \quad (2-5)$$

$$\lambda_{qs} = \frac{L_m}{L_r} \lambda_{qr} + \sigma L_s i_{qs} \quad (2-6)$$

$$\lambda_{ds} = \frac{L_m}{L_r} \lambda_{dr} + \sigma L_s i_{ds} \quad (2-7)$$

$$T_o = \frac{3}{2} \frac{P}{2} \frac{L_m}{L_r} (\lambda_{ds} \lambda_{qs} - \lambda_{qs} \lambda_{ds}) \quad (2-8)$$

고정자 자속 기준 제어 시스템에서는 고정자 자속을 기준축으로하여 모든 변수를 표현하는 것이 보다 편리하다.

고정자 자속 기준축 상의 유도기 표현식은 회전자 자속을 소개하고 고정자 자속의 q축 성분  $\lambda_{qs} = 0$  으로 놓음으로써 얻을 수 있다.

$$(1+\sigma_{Tr}) L_s i_{qs} - \omega_{s1} Tr (\lambda_{ds} - \sigma L_s i_{ds}) = 0 \quad (2-9)$$

$$(1+\sigma_{Trp}) \lambda_{dr} = (1+\sigma_{Trp}) L_s i_{ds} - \omega_{s1} Tr \sigma L_s \lambda_{qs} \quad (2-10)$$

$$T_o = \frac{3}{2} \frac{P}{2} \lambda_{ds} \lambda_{qs} \quad (2-11)$$

그런데, 식에서 알 수 있듯이 고정자 전류  $i_{qs}$ 와 고정자 자속  $\lambda_{ds}$ 가 간섭되어 있다. 즉 토크 성분 전류의 변화에 따라 자속도 변하게 된다. 따라서 과도 상태에서  $\lambda_{qs}$ 가 변화하면  $\lambda_{ds}$ 가 변하지 않는데도 불구하고 자속이 변하게 된다. 이러한 단점을 극복하기 위하여 다음과 같은 분리기(decoupler)가 고안되었다.<sup>[12]</sup>

$$i_{ds} = G_f(p) (\lambda_{ds} - \lambda_{qs}) + i_{dq} \quad (2-12)$$

$$(1+\sigma_{Trp}) \lambda_{dr} = (1+\sigma_{Trp}) L_s G_f(p) (\lambda_{ds} - \lambda_{qs}) + (1+\sigma_{Trp}) L_s i_{dq} - \omega_{s1} Tr \sigma L_s i_{qs} \quad (2-13)$$

$\lambda_{dr}$ 와  $\lambda_{qr}$ 를 분리해 내기 위해 식(13)의 우변의 2항과 3항의 합이 0이 되면 된다.

$$(1+\sigma_{Trp}) L_s i_{dq} - \omega_{s1} Tr \sigma L_s i_{qs} \quad (2-14)$$

$$i_{dq} = \frac{\omega_{s1} Tr \sigma L_s i_{qs}}{(1 + \sigma_{Trp}) L_s} \quad (2-15)$$

$$\omega_{s1} = \frac{(1 + \sigma_{Trp}) L_s i_{dq}}{Tr (\lambda_{ds} - \sigma L_s i_{ds})} \quad (2-16)$$

$$i_{dq} = \frac{\sigma L_s (i_{qs})^2}{\lambda_{ds} - \sigma L_s i_{ds}} \quad (2-17)$$

$i_{qs}$ 의 변화에 따라 식(2-17)의  $i_{dq}$ 를 식(2-12)과 같은 방법으로  $i_{ds}$ 에 첨가하여 구하면 토크의 변화에 따른 자속의 간섭효과를 제거할 수 있다.<sup>[13]</sup>

$$|\lambda_s| = \sqrt{(\lambda_{ds} \lambda_{ds} + \lambda_{qs} \lambda_{qs})} \quad (2-18)$$

동기좌표계에서 정지 좌표계로 변환에 필요한 단위 벡터는

$$\cos(\theta) = \lambda_{ds} / |\lambda_s| \quad (2-19)$$

$$\sin(\theta) = \lambda_{qs} / |\lambda_s| \quad (2-20)$$

로 표현된다.  
동기 각속도는

$$\omega_s = (\lambda_{qs} \lambda_{ds} - \lambda_{ds} \lambda_{qs}) / |\lambda_s|^2 \quad (2-21)$$

이 되고 여기에서 추정된 회전자 각속도는

$$\omega_r = \omega_s - \omega_{s1} \quad (2-22)$$

로 구해진다.

이 추정된 회전자 속도로부터 궤환신호를 얻게된다. 이러한 이유로 이런 벡터제어방식을 속도검출기가 없는 형태가 된다.

### 3. 적용 IP제어기를 이용한 유도 전동기 벡터제어

#### A. 유도전동기 모델링

제어 대상인 유도 전동기의 전달함수는

즉, 동기좌표계 q축 전류인  $I_{qs}(s)$ 에 대한 추정  $\omega_r(s)$ 의 전달함수는

$$\frac{\omega_r(s)}{Iqs(s)} = \frac{K'}{Js + B} = G(s) \quad (3-1)$$

또한, 이 전달함수는 디지털 제어를 위해 z-domain 전달함수의 형태로 바꾸어야 한다. 그러므로 zero-order holder를 포함한 z-domain 전달함수는

$$\frac{\omega_r(z^{-1})}{Iqs(z^{-1})} = Z \left( \frac{(1-e^{-sT})}{s} G(s) \right) \quad (3-2)$$

$$= (1-z^{-1}) Z \left( \frac{K'}{s(Js+B)} \right) = \frac{bz^{-1}}{1+az^{-1}} \quad (3-3)$$

$$a1 = -e^{-BT/J} \\ b1 = K' / B (1 - e^{-BT/J})$$

로 표현된다. 시스템을 차분방정식의 형태로 바꾸면

$$A(z^{-1}) \omega_r(t) = B(z^{-1}) Iqs(t) + C(z^{-1}) e(t) \quad (3-4)$$

여기서 A(z<sup>-1</sup>), B(z<sup>-1</sup>)와 C(z<sup>-1</sup>)는 계수가 알려지지 않은 다항식이고 C(z<sup>-1</sup>)은 이것의 평균이 0인 잡음에 해당한다. 적응제어를 하기 위해서는 제어대상에 대한 정보가 필수적이므로 플랜트에 대한 매개변수를 추정하기 위해 순환최소자승법을 이용하였다. 순환최소자승법은 다음과 같다.

**B. 순환 최소 자승법 ( Recursive Least Square )**

$$\theta(t) = \theta(t-1) + K(t) ( y(t) - \phi(t)^T \theta(t-1) ) \quad (3-5)$$

$$K(t) = P(t) \phi(t) = P(t-1) \phi(t) ( \lambda I + \phi(t)^T P(t-1) \phi(t) )^{-1} \quad (3-6)$$

$$P(t) = [ P(t-1) - P(t-1) \phi(t) [ 1 + \phi(t)^T P(t-1) \phi(t) ]^{-1} \phi(t)^T P(t-1) ] / \lambda \quad (3-7)$$

$$= [ 1 - K(t) \phi(t)^T ] P(t-1) / \lambda \quad (3-8)$$

λ : 망각 인자 ( forgetting factor or discounting factor )

실제적인 λ의 선정은 0.99 ~ 0.999 사이<sup>[5]</sup>

K(t) : weighting factor  
P(0) : initial variance  
보통 10<sup>6</sup> ~ 10<sup>11</sup> 정도

여기서 구한 플랜트의 매개변수를 가지고 제어를 설계하게 된다.

**C. 적응 IP제어기**

적응 제어기에는 두가지 형태가 존재하는데 하나는 MRAC ( Model Reference Adaptive Controller )와 자기동조기 ( STR : Self-Tuning Regulator )가 있다. 여기에서의 제어기는 그림 3.1과 같은 자기동조기를 사용하고 있으며 시스템의 매개변수를 추정하여 이를 이용해 제어기 매개변수를 구하는 간접 방식을 채택하고 있다.

앞의 순환 최소 자승법으로 구한 플랜트의 매개변수를 값으로 가정하고 그림 3.2의 IP제어기를 고려한 시스템 페루우프 전달함수는

$$\frac{\omega_r}{\omega_r^{ref}} = \frac{K(s) B(s)}{s A(s)} \quad (3-9)$$

$$= \frac{B(s)}{1 + K_p \frac{B(s)}{s A(s)} + \frac{K_i B(s)}{s A(s)}}$$

로 표현된다. 또한 Z-영역 전달함수는

$$\frac{\omega_r(z^{-1})}{\omega_r^{ref}(z^{-1})} = \frac{K_i \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}} \frac{B(z^{-1})}{A(z^{-1})}}{1 + K_p \frac{B(z^{-1})}{A(z^{-1})} + K_i \frac{T}{2} \frac{1+z^{-1}}{1-z^{-1}} \frac{B(z^{-1})}{A(z^{-1})}}$$

T : sampling time (3-10)

로 된다.

이 페루우프 시스템에서 입력에 대한 출력의 만족스러운 응답을 얻기 위하여 극을 요구되는 위치로 재배치하는 극배치 방식을 이용하게 된다. 적응 제어기를 설계하기 위해 우선 극을 재배치할 플랜트의 모델을 다음과 같이 설정한다.

$$A_m(z^{-1}) = 1 - 2e^{-\zeta\omega T} \cos \omega T \sqrt{1 - \zeta^2} z^{-1} + e^{-2\zeta\omega T} z^{-2} \quad (3-11)$$

ζ : relative damping  
ω : 주파수

여기에서 ζ는 0.7, 상승시간 t<sub>r</sub>은 0.1[Sec], 샘플링 시간은 1[μSec]로 정하였다.

IP제어기는 PI제어기와 유사한 특성을 가지고 있다. 우선 두 제어기는 모든 정상상태 오차를 0으로 만들고 같은 페루우르구를 발생시키므로 특성상 유사하다. 그러나 IP제어기는 PI제어기와는 달리 제어기에 의해 페루우르 구를 발생시키지 않는다. PI제어기에서의 이터 영점은 시스템 페루우르 구점이 적절히 위치된다 해도 예리한 상승시간과 오버슈트 (overshoot)를 발생하게 된다. 이런 영점은 구점이 고정되더라도 전동기 매개변수에 의존하므로 만족스럽게 배치되지 못하게 된다.

그러므로 극비치영태의 적용제어에서는 IP제어기가 PI제어기보다 좋은 성능을 발생시키게 된다.

4. 시뮬레이션 과정

시뮬레이션 흐름도는 그림 4.1과 같고 여기에서 전류제어의 샘플링 시간은 50[ $\mu$ Sec]이고 나머지 부분은 1[ $\mu$ Sec]이다. 속도제어기에서 토오크 기준치  $T_{ref}$ 를 내보내면 벡터 제어기에서 3상 기준전류  $i_{abc}^*$ 를 계산하고 인버터 부분에서 전압  $v_{ds}, v_{qs}$ 를 계산하고, 전동기에서는 실제전류  $i_{abc}$ 를 발생시킨다. 실제전류  $i_{abc}$ 와 전압으로부터 단위 벡터를 발생하고 또한 회전자 자속을 추정하여 궤환한다.

전류제어기에서의 히스테리시스벤드는 0.01[A]이다.

그림 4.2에서는 기존의 고정된 PI제어기를 사용하였을 경우의 실제 전동기의 속도응답을 나타내고 있으며 그림 4.3에서는 IP제어기를 사용하였을 경우의 1400[rpm]에 대한 속도응답을 나타내고 있다.

실제적으로 궤환신호로 사용하고 있는 추정속도는 오버슈트가 상당히 감소된 것을 알 수 있으며 그러므로 실제 전동기의 속도에서도 적은 양의 오버슈트를 나타내게 된다. 그러므로 앞의 내용에서도 지적한 IP제어기의 특성이 유용함을 증명하게 된다. 더욱 더 자기동조기 형태의 IP제어기를 사용하였을 경우 그림 4.4에서의 같이 앞의 응답특성에서 오버슈트가 없게 개선된 사실을 알 수 있다. 속도명령이 주어졌을 때 직류기와 동가적으로 d-축 제어전류는 일정해야 한다. 이것에 대한 파형은 그림 4.5에 나타나고 토오크와 Q축, D축전압의 파형은 그림 4.6와 4.7, 4.8에 나타났다. 또한 고정자와 회전자의 d-축 전류와 q-축 전류는 그림 4.9와 4.10에서의 같이 정상상태에서는 안정된 값을 갖게 된다.

그림 4.11는 적용제어에서의 최소순화자속을 이용한 플랜트의 매개변수 추정값들이다. 이 값들은 마르계 정상상태에 도달한다는 것을 알 수 있다. 속도를 추정하기 위해 고정자 자속에 대한 정보는 필수적이다. 그러나 자속을 구하기 위해 적분과정이 필요하게 되어 적분에서의 초기값에 의한 오차와 음절 때문에 적분기가 포화되는 현상을 발생하게 된다.

이러한 현상은 정확한 자속의 계산에 기초하여 속도를 추정하는 알고리즘에서는 시스템 전체의 성능을 저하시키게 된다. 이런 점을 개선하기 위해 순수적분기 대신에 1차 지연된 필터를 설계해 사용하였다.

5. 결론

벡터제어시 일반적인 속도제어기인 PI제어기를 사용했을 때 발생하는 속도 응답특성의 오버슈트특성을 적용 IP제어기를 사용함으로써 해서 제거하고 속도점출기 없이 유도기를 운전하였다. 또한 적용제어기를 사용함에 의해 전동기 매개변수에 의해 발생하는 문제를 해결하였다.

앞서 전기한 이론적 고찰내용이 시뮬레이션과 잘 일치됨을 확인 할 수 있었다. 또한 이 연구는 앞으로 유도기 가변속제어에서 새로운 제어이론을 도입하기 위해 준비단계로서의 역할을 수행할 수 있을 것으로 사료된다. 앞으로 좀 더 강인한 응답특성을 요구하는 시스템에 대해서는 제어기의 차수를 높이는 방향으로 연구되어야 할 것이다.

참고 문헌

- [1] X.Xu, e.a., "Stator Flux Orientation Control in Field Weakening Region", IEEE-IAS Annual Meeting, 1988.
- [2] X.Xu, Rik.De Doncker, Donald W.Novotny, "A Stator Flux Oriented Induction Machine Drive", PESC 1988 Conference Record, pp 870-876
- [3] X.Xu, D.W.Novotny, "Implementation of Direct Stator Flux Orientation Control on a Versatile DSP Based System", IEEE-IAS Annual Meeting, 1990
- [4] Raymond B.Sepc, J.H.Lang, "Real-Time Adaptive of the Permanent-Magnet Synchronous Motor", IEEE-IAS Annual Meeting, 1990
- [5] A.O.Csorlo, D.Q.Mayne, "Deterministic convergence of a self-tuning regulator with variable forgetting factor", IEE-PROC, Vol.128, 1981
- [6] P.K.Nandam, P.C.Sen, "Analog and Digital speed control of DC Drives Using PI and IP Control Techniques", IEEE Transaction Industrial Electronics, vol.1E-34, no.2, May 1987
- [7] Åström, K.J. and B.Wittenmark "Self-tuning controllers based on pole-zero placement" Proc.IEE, 127, 120-130.

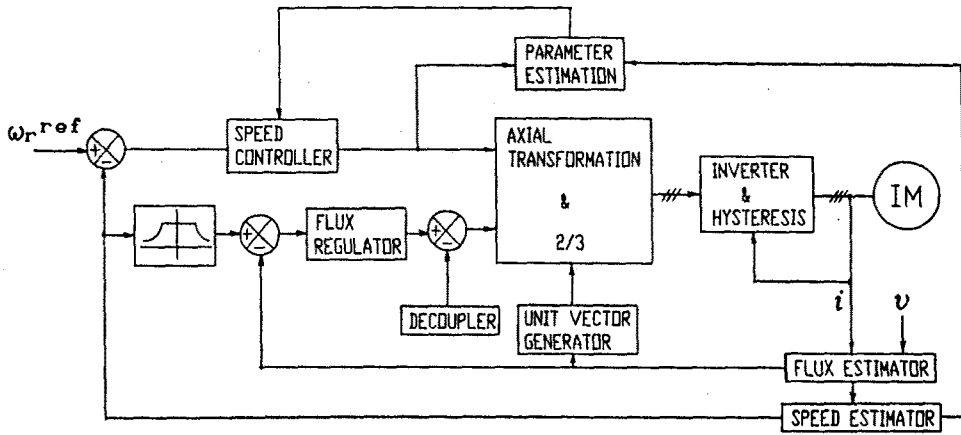


그림 3.1 전체 시스템 블록

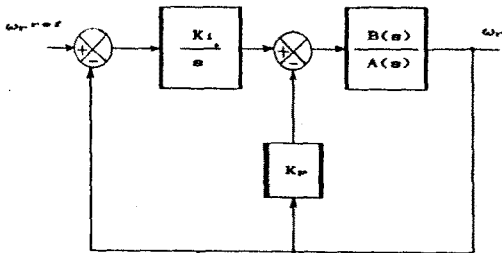


그림 3.2 IP 제어기

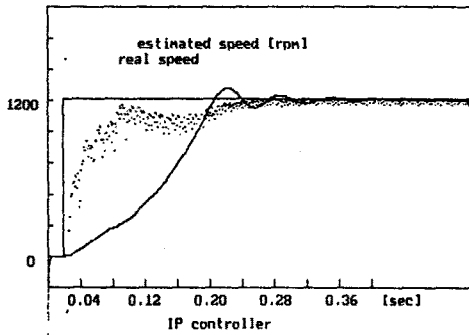


그림 4.3

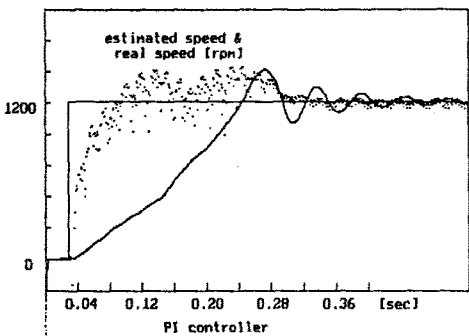


그림 4.2

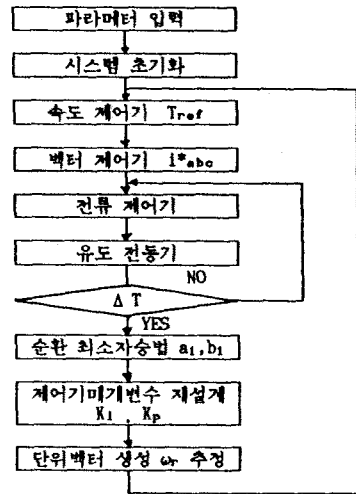


그림 4.1 시뮬레이션 흐름도

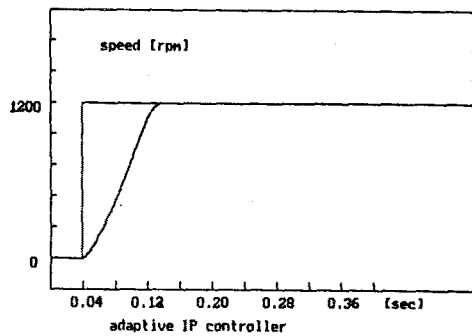


그림 4.4

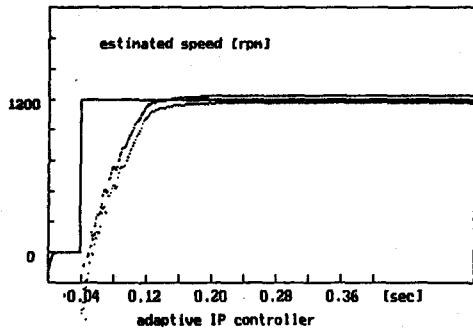


그림 4.4(a)

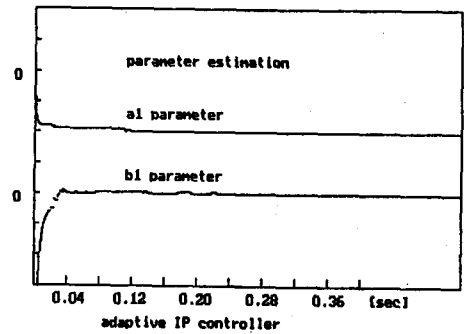


그림 4.11

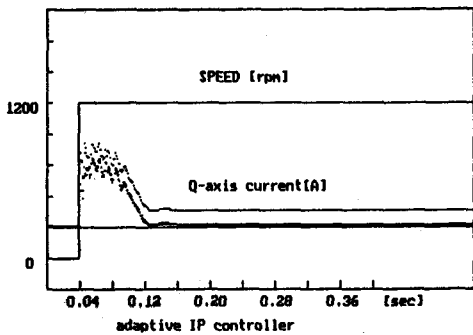


그림 4.6

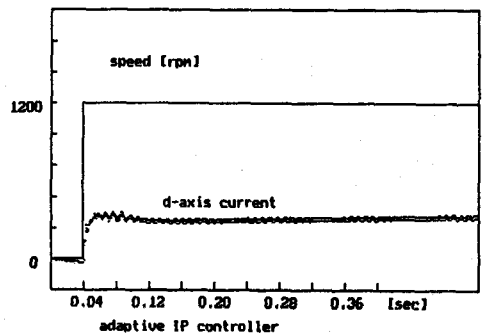


그림 4.5

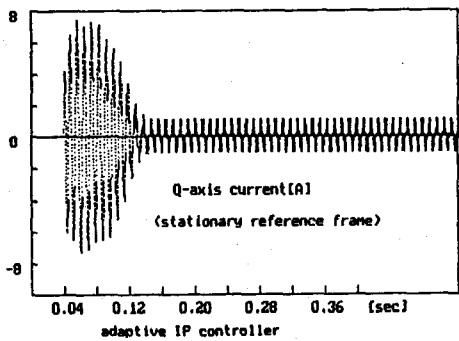


그림 4.10

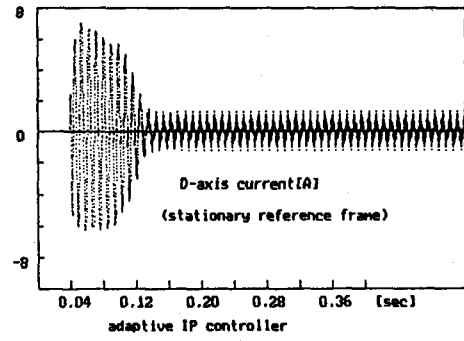


그림 4.9

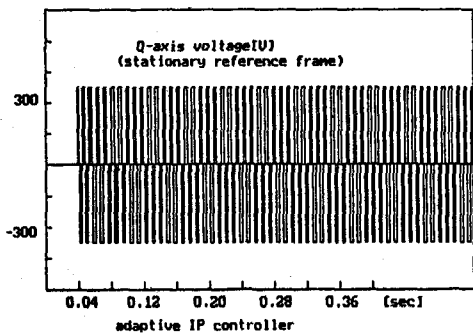


그림 4.7

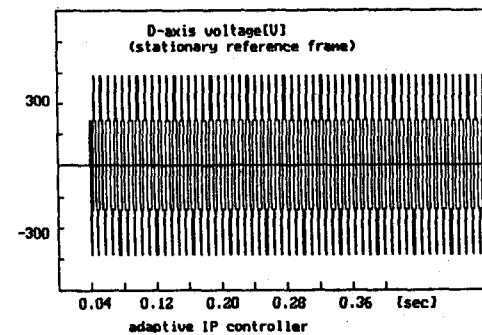


그림 4.8