

# 적응제어 알고리즘과 80286 마이크로 프로세서를 이용한 DC 서보모터의 강인한 속도제어기의 구현

\*김 중석\* 이 건영\* 양 해원\*  
\*한양대학교 전기공학과

## An implementation of the speed controller for DC servomotor using adaptive control algorithm and 80286 $\mu$ -processor

\*Joong-Suk Kim\* Keon-Young Yi\* Hai-Won Yang\*  
\*Dept. of Electrical Eng. Han Yang Univ.

### ABSTRACT

This paper proposes a robust direct adaptive control system implementation using a 80286 microprocessor-based system for controlling the speed of a DC servo motor. In this paper, assuming that the unmodeled dynamics of the plant are sufficiently small in the low-frequency range, the plant as linear time-invariant system is the second relative degree, we construct the direct adaptive control system with the algorithm considering plant unmodeled dynamics and execute the experiment, and compare the characteristics with those of PI algorithm's.

It shows that an easy implementation of the built controller is due to the usage of software for the algorithm.

### 1 서론

최근 10년간, 고 성능의 16-bit 마이크로프로세서의 발전과 더불어 공학의 여러분야에서 컴퓨터는 과거 어느때 보다도 값싸고 편리하게 이용할 수 있게 되었다. 특히 수치제어용 공작기계, 사무자동화 기기 및 공장자동화 기기등의 여러분야에서 이용되는 DC 서보모터를 제어 하는데는 더욱 더 정밀하고 신뢰성이 있는 제어방법이 요구되고 이를 만족할 수 있는 제어기의 설계에는 신뢰도가 높으며 실시간 연산을 가능하게 해주는 컴퓨터의 활용을 당연시 하게 되었다. 또한 DC 서보모터운전시 전통적으로 PID제어기가 주로 사용되어 왔고 구조의 간단성과 안정된 동작점때문에 효율적인 제어성능을 유지할 수가 있었으나 이 제어기는 한 동작점에서 유효하나 매개변수의 변화에 따른 동작점의 변화에는 최적화 방법 (optimizing method), 시행오차법 (trial-and-error), 조정방법 (tuning rule)등의 방법으로 PID계수를 다시 조정하여야 하는 불편함이 있었다. 따라서 외란이나 부하변동 및 환경에 따른 제어기의 특성변화가 있게 되는 경우 이를 해결하기 위한 제어방법으로는 시스템의 특성을 잘 모르고 있다고 가정하고 시스템을 운전하면서 시스템의 특성에 관한 지식을 보완, 수정하고 적절한 대책을 강구해 가는 적응제어방법이 효과적인 제어특성을 나타낼 것으로 기대된다. 적응제어에는 두가지 방법이 있으며 하나는 자기동조기 (Self-tuning regulator)이고 또 하나는 기준모델 적응제어 (Model reference adaptive control) 방법이다. 자기동조기는 시스템의 특성을 추정하면서 추정된 매개변수를 바탕으로 적절한 제어기를 구성하는 것이며 기준모델 적응제어는 시스템에 적당한 제어기를 구성하여 원하는 시스템의 특성에 일치시키는 것이다.

기준모델 적응제어에 관한 이론은 여러가지 방법이 제시되었다. 그중에 Monopoli는 상대차수가 2차 이상인 시스템에 확장오차신호를 도입하였고[1] Ioannou와 Tsakalis가 플랜트의 비모형화 특성을 고려한 강인한 직접적용 제어기를 설계하였다[2].

본 논문에서는 Ioannou[2]의 알고리즘, 전동기 구동장치, 디지털 접속 장치 및 80286 마이크로프로세서 시스템을 결합한 제어기를 구성하여 DC 서보 모터의 실시간 제어가 가능하도록 구현하였다. 이를 통하여 기준모델 적응제어 알고리즘의 제어기를 쉽게 구현할 수 있음을 보여주었으며 그 결과를 PI제어기의 특성과 비교하여 보았다.

### II 플랜트의 모델과 제어목적

다음의 단일 입,출력 플랜트 (single-input single-output, SISO)를 생각하기로 한다.

$$\frac{y(s)}{u(s)} = G(s) = G_0(s) [ 1 + \mu \Delta_2(s) ] + \mu \Delta_1(s) \quad (1)$$

여기서  $G(s)$ 는 상대차수가 1 이상이며,  $\Delta_1(s), \Delta_2(s)$ 는 모두 양의 스칼라 파라미터  $\mu$  를 갖는다.

$$G_0(s) = k_p \frac{Z_0(s)}{R_0(s)} \text{ 모세 플랜트의 모형화된 부분이며,}$$

$\mu \Delta_1(s)$  와  $\mu \Delta_2(s)$ 는 각각 플랜트에 더해지고, 곱해지는 섭동으로서 플랜트의 모형화 되지않은 부분으로 일반적으로 저주파영역에서는 상당히 작은 값이 된다. 즉  $\mu$ 는 매우 작은 양의 값이다.

모형화된 부분에 대한 가정은 다음과 같다.

- A1 :  $Z_0(s)$ 는 monic이고, 안정한 다항식이며, 차수는  $m(s-n)$  이다.
- A2 :  $R_0(s)$ 는 monic(monic)이며, 차수는  $n$ 이다.
- A3 :  $K_p$ 의 부호와  $m, n$ 의 값은 같다.

이후로는  $K_p$ 의 부호를 알고 있으므로 편의상  $k_p > 0$  이라고 가정한다.

비 모형화 부분에 대한 가정은 다음과 같다

- A4 :  $\Delta_1(s)$ 는 상대차수 1 이상 안정한 전달함수이다.
  - A5 :  $\Delta_2(s)$ 는 안정한 전달함수이다.
  - A6 :  $\Delta_1(s-p)$  와  $\Delta_2(s-p)$ 의 극이 안정하기 위한 안정도 여유  $p > 0$  의 하한값  $p_0 > 0$  는 안다.
- 식 (1)로 부터 주어진 플랜트에 대한 적응제어 문제는 다음과 같이 기준모델로 놓을수 있다.

$$\frac{y_m(s)}{r(s)} = W_m(s) = k_m(s) \frac{1}{D_m(s)} \quad (2)$$

여기서  $D_m(s)$ 는 monic이고, 안정한 다항식으로 차수  $n^* = n - m$ 이며,  $r(t)$ 는 유계인 기준입력,  $y_m(t)$ 는 기준모델의 출력,  $k_m$ 은  $k_p$ 와 같은 부호를 갖는 상수이다.

식 (2)로부터 주어진 기준모델에 대하여 비모형화특성을 고려한 플랜트가 안정도를 유지하면서, 외란이 존재할 경우에도 제어입력이 강인하게 적용하여, 기준모델의 출력에 가능한 한 가깝게 따라가도록 하는 것이 본 논문에서의 제어 목적이다.

### III 적응제어 시스템의 구조

파라미터 적응식 및 제어식에 쓰이는  $(2n - 1)$ 차원의 보조벡터는 다음 관계식으로 부터 주어진다.

$$\dot{\omega}_1(t) = F \omega_1(t) + q u(t), \quad \dot{\omega}_2(t) = F \omega_2(t) + q y(t) \quad (3)$$

여기서  $F$  는 안정한 행렬이며,  $(F, q)$  는 가제어 쌍이다.  
 $\omega^T = [\omega_1^T, \omega_2^T, y]$ 이다.  
 플랜트에 대한 제어 입력은 다음과 같이 주어진다.

$$u(t) = O^T \omega + c_0 r \quad (4)$$

여기서  $O^T(t) = [O_1^T(t), O_2^T(t), O_3(t)]$  으로  $(2n - 1)$  차원의 제어기 파라미터 벡터이고,  $c_0(t)$  는 전향파라미터로써 스칼라이다.  
 본 논문에서는  $k_p$  는 모르는 경우를 다루고 있으며 그 적응식은 다음과 같다[2].

$$\dot{\sigma}(t) = - \frac{\bar{\Gamma} \bar{\epsilon}_1 \bar{\zeta}}{m^2} - \bar{\Gamma} \sigma \bar{O}$$

$$\sigma = \begin{cases} 0 & \text{if } \|\bar{O}\| < M_0 \\ \sigma_0 \left( \frac{\|\bar{O}\|}{M_0} - 1 \right) & \text{if } M_0 < \|\bar{O}\| < 2M_0 \\ \sigma_0 & \text{if } \|\bar{O}\| > 2M_0 \end{cases} \quad (5)$$

여기서  $\gamma > 0$ ,  $\sigma_0 (> 0)$  는 상수이고

$$\bar{\Gamma} = \text{diag}(\Gamma, \gamma, \gamma), \quad \Gamma = \Gamma^T > 0,$$

$\bar{O}(t) = [O^T, c_0, \varphi_0]^T$  는  $\bar{O}^* = [O^{*T}, c_0^*, \varphi_0^*]^T$  의 추정치이다.

$$c_0^* = k_m / k_p, \quad \varphi_0^* = k_p / k_m,$$

$$\epsilon_1 = y - y_m + \varphi_0 \zeta, \quad \zeta = O^T \zeta - \nu + c_0 y_m - \nu_0,$$

$$\nu_0 = W_m(s) c_0 r \text{ 이고}$$

$$\bar{\zeta} = [\zeta^T, y_m, \zeta]^T, \quad \zeta = W_m(s) \nu \text{ 이며}$$

$$m = -\delta_0 m + \delta_1(|u| + |y| + 1), \quad m(0) > \delta_1 / \delta_0 \text{ 이다.}$$

제어기 파라미터 벡터의 놈(norm)  $M_0 \geq 10^*$ 로 정의되며 이 알고리즘에서 필요로 하는 사전정보가 된다.

**IV 제어시스템의 하드웨어, 소프트웨어 구성**

본 절에서는 제어대상인 DC 서보모터 구동시스템의 하드웨어와 실험 방법에 관하여 다루는데 제어기 전체의 구성은 그림 1.과 같다.

본 논문의 PI알고리즘 및 기준모델 적용제어 알고리즘을 DC 서보모터의 속도제어에 적용하기 위해 80287이 부착된 16 bit 12 Mhz IBM PC/AT를 이용하였다. 본 실험에 사용하기 위해 특별히 제작된 제어기는 제어 알고리즘 소프트웨어만 수정하면 다양한방식의 제어기가 되는 구조를 갖기위해 82C54 Programmable interval timer를 이용하여 모터와 관련된 신호를 발생시키며 8Mhz의 시스템 클럭과 동기되어 동작하는 PC Bus 형식의 Card로 설계되었다. 또한 제어기에서 실시간 구현을 위한 인터럽트를 초당 1000회 발생 시키고 직접적용제어기의 경우 매 30 msec마다, PI제어기의 경우 매 5 msec 마다 한번씩 Plant 출력을 샘플링하여 제어입력을 발생시킨다.

모터구동회로는 컴퓨터 외부에 따로 설치하였고 별도의 Power Supply(최대 50V)를 통해 모터를 구동한다. 모터를 동작시키는 원리는 PWM방식을 이용하였으며, 이를 위해서 PWM Pulse Generator 회로 및 pulse 폭 조정을 위한(period는 H/W level) PC Interface(write)를 제공한다.

본 실험에서 컴퓨터는 user-interface 및 알고리즘 수행을 위한 실시간 계산을 담당한다. 속도정보는 모터에 부착된 Encoder를 통해서 나오는 단위 시간당 펄스수로부터 측정하였고 출력은 82C54 counter 값을 이용하여 PWM펄스의

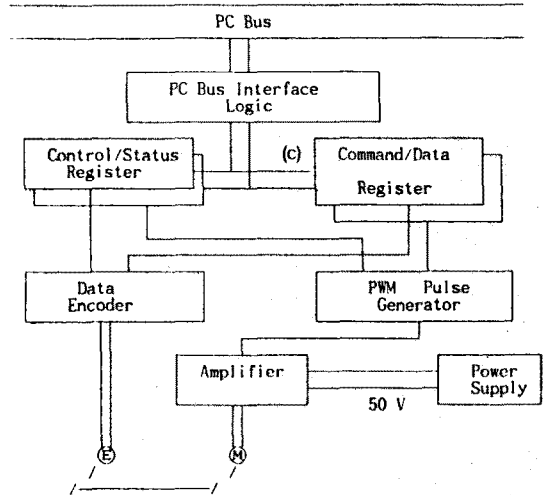


그림 1. 하드웨어 블록 선도

on-time을 조정하므로써 모터에 가해지는 전류를 조절하는 방식이다.

소프트웨어는 C언어를 사용하였고 인터럽트 서비스 루틴을 이용하여 Encoder 값을 샘플링하여 모터의 rpm을 계산한다. 이 속도정보는 제어알고리즘 수행루틴에 이용되며 이때 사용된 알고리즘[2]은 강인한 직접적용 제어기법이며, 수치계산을 위해 Runge-Kutta method with Gill's coefficient법을 이용하였다. 또한 실시간 처리가능 및 제어입력 데이터를 하드웨어에 보내는 기능을 갖는다. 모터구동 회로에 보내는 제어입력은 모터에 가해지는 전압과 PWM펄스주기에 의하여 계산되며 이때 PWM펄스의 주기는 Power Transistor의 switching 특성 및 모터의 특성을 고려하여 결정하였다.

소프트웨어의 전체 블록선도는 그림 2.와 같다.

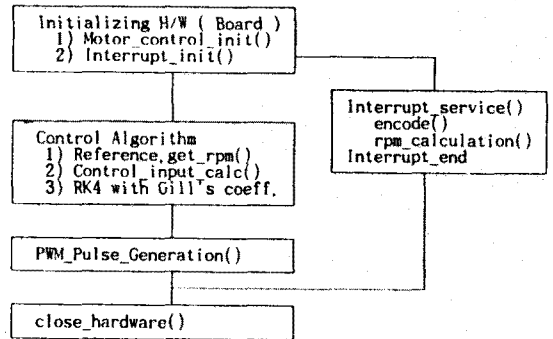


그림 2. 소프트웨어 블록 선도

**V 실험**

실험에 있어서 플랜트는 아래의 조건을 만족한다는 가정을 갖는다.

- S1: 고 주파수 이득  $k_p$ 의 부호는 안다.
- S2: 플랜트의 최고차수는 2차이다.
- S3: 플랜트의 상대차수는 2차이다.
- S4: 영점은 좌반면에 존재한다.

실험에서의 기준모델은 다음과 같이 설정하였다.

$$\frac{y_m(s)}{r(s)} = \frac{1}{(s+1)(s+2)} \quad (6)$$

본실험에서 사용한 플랜트의 사양은 정격출력 200W, 유기전압정수  $2.25 \times 10^{-2}$  V/A, 토크정수  $2.2 \times 10^{-4}$  N·m/A, 전기자 관성  $2.52 \times 10^{-7}$  kgm<sup>2</sup>, 전기자 저항 2.38 Ω, 기계적 시상수 12.68 msec, 전기적 시상수 1.6 msec인 DC 서보 모터이다.

직접적용제어기 실험에서 사용한 적용칙은 (5)식이며 설계파라미터와 기준입력은 (3)~(7)에서  $F = -1.7$ ,  $q = 1$ ,  $\delta_0 = 0.7$ ,  $\delta_1 = 0.3$ ,  $M_0 = 5$ ,  $\sigma_0 = 0.1$ ,  $\gamma = 2$ ,  $r(t) = 3600$  과  $r(t) = 3600\sin(0.1t)$  를 사용하였다.

PI의 경우에서는 Transient-response method을 사용하여 구한  $K_p = 0.13$ 과  $T_i = 0.0114$ 가 이용되었으며 위에서와 같은 기준입력을 적용하였다. PI 제어기 구현을 위해 직접적용 제어기의 알고리즘 부분을 PI 알고리즘으로 대체하였다.

부하 및 부하변동을 위하여는 다른 모터 하나를 커플링하여 그 모터에 입력단에 스위치를 통하여 또 다른 모터를 연결하는 방법을 이용하였다. 부하변동특성 실험시에는 부하축 모터의 입력단 스위치를 개폐하는 방식을 택하였다.

알고리즘부분을 소프트웨어로 처리하므로써 각종 알고리즘을 본 논문에서 구성된 하드웨어에 쉽게 적용할 수가 있음을 알 수 있다.

### VI 실험결과

두 제어기 모두 원하는 출력을 추종하는데 우수한 성능을 나타냈으며 직접적용제어기의 경우가 PI 제어기의 경우보다 적용속도가 다소 느렸으나 이는 샘플링시간 및 적용이득의 영향으로 생각된다. 그러나 정상상태에서는 PI 제어기에 비해 안정된 출력을 보여주며 이는 모터와 부하사이의 커플링 또는 측정오차등으로 인한 고주파 성분의 외란에 대해 직접적용제어기가 강인성을 보여주고 있음을 알 수 있다.

그림에서의  $U_p$ 는 10배 확대된 전압값을 나타내며 최대값은 50V이다.

#### 1. 부하변동이 없는 경우

그림 3.과 4.는 부하시 계단입력을 기한경우의 실험결과를 보여준다. 과도상태에서 적용속도의 차이를 보이고 있으며 정상상태에서는 적용제어기가 보다 안정한 특성을 보이고 있다. 그림 4.에서의 경우가 그림 3.에서보다 약간 큰 과도상태 오차가 존재하나 빠른 적용능력을 보여준다. 플랜트의 출력이 기준모델의 출력을 잘 추종하고 있음을 알 수있다.

그림 5. 와 6.은 정현입력에 대한 결과로 두 경우 모두 모델을 잘 추종하여 기변속도에 대해서도 원만한 제어 특성을 보이고 있음을 알 수 있다. 과도및 정상상태에서의 특성은 계단입력과 유사하다.

#### 2. 부하변동이 있는 경우

부하변동이 존재하는 경우의 실험결과가 그림 7. ~ 10. 에 나타나있다. 계단 입력 및 정현입력에 대해 부하변동의 증가와 동시에  $U_p$ 가 증가하여 기준모델의 출력을 추종하며 추가된 부하가 사라지면  $U_p$ 가 다시 감소하여 원하는 출력으로 복귀하는 적용특성을 보이고 있다. 그림 7. ~ 10.의 확대된 부분에서 보면 두 알고리즘의 적용능력을 비교해 볼 수가 있다. 적용제어기의 부하변동에 따른 적용속도는 PI제어기에 비해 느리나 정상상태에서는 각종 고주파 외란에 대해 강인한 적용특성을 나타내고 있다.

### VII 결론

본 연구는 80286 마이크로프로세서를 이용하여 직접 적용제어 알고리즘의 제어기를 구성하고 이를 DC 서보모터의 속도제어에 이용하였다. 제어기의 하드웨어에 80286 마이크로 프로세서를 이용하여 계산량이 많은 직접적용제어를 원만하게 실시간에 구현하였으며, 이 제어기는 외란 및 부하변동이 존재 하는 경우에 대해서도 기준모델의 출력을 잘 추종하는 강인성을 보여주었다. 또한 제어알고리즘을 RK4 방법으로 구성하므로써 연속시간계 해석이론을 그대로 적용하여 기존의 이산 시간계 방식보다 제어알고리즘 선택의 폭을 넓게하여 적용제어 이론의 실제 적용을 용이하게 하였다.

그러나 아직 PI제어기의 경우보다 좋지 못한 과도응답특성을 나타내는데 이를 개선하기 위하여 샘플링시간 및 적용이득에 관한 연구가 이루어져야 할 것이다.

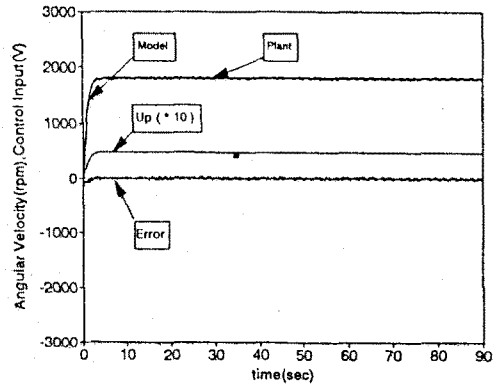


그림 3. 부하시 계단입력에 대한 PI 알고리즘의 실험 결과

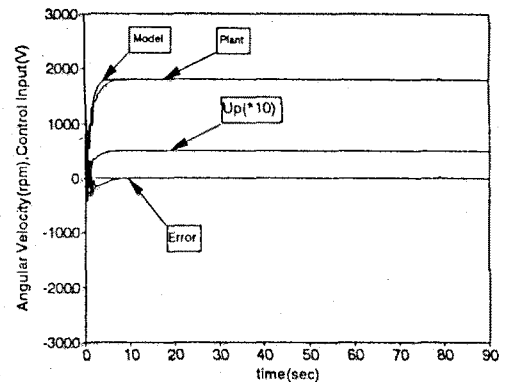


그림 4. 부하시 계단입력에 대한 적용제어 알고리즘의 실험 결과

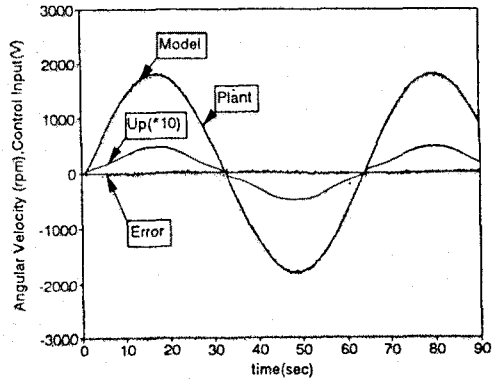


그림 5. 부하시 정현입력에 대한 PI 알고리즘의 실험 결과

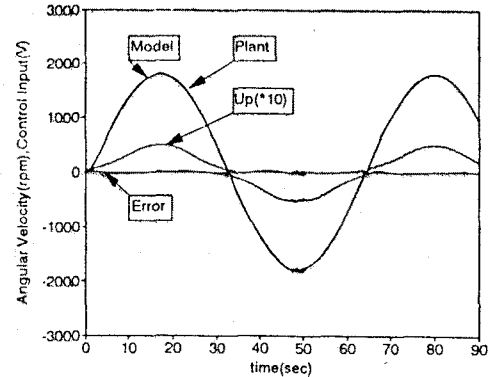


그림 6. 부하시 정현입력에 대한 적용제어 알고리즘의 실험 결과

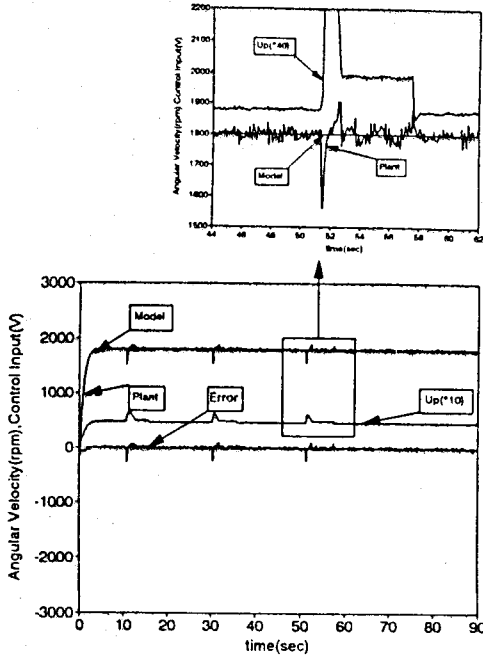


그림 7. 부하변동시 계단입력에 대한 PI 알고리즘의 실험결과

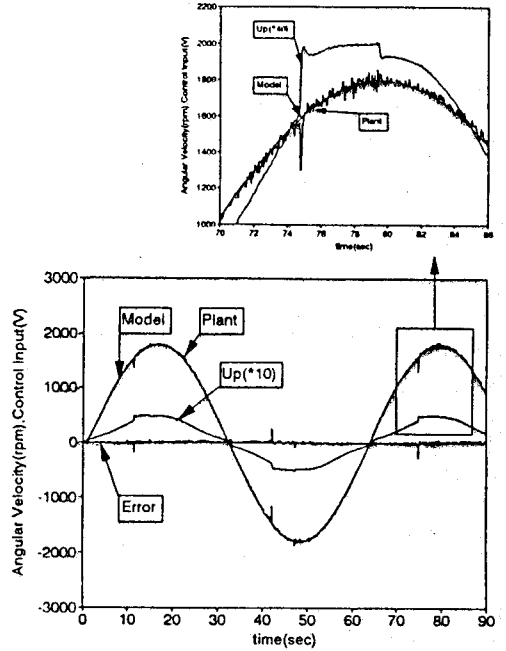


그림 9. 부하변동시 정현입력에 대한 PI 알고리즘의 실험결과

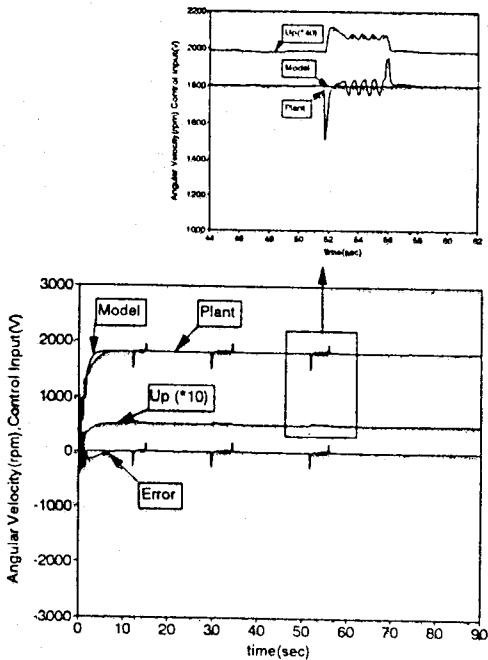


그림 8. 부하변동시 계단입력에 대한 적응제어 알고리즘의 실험결과

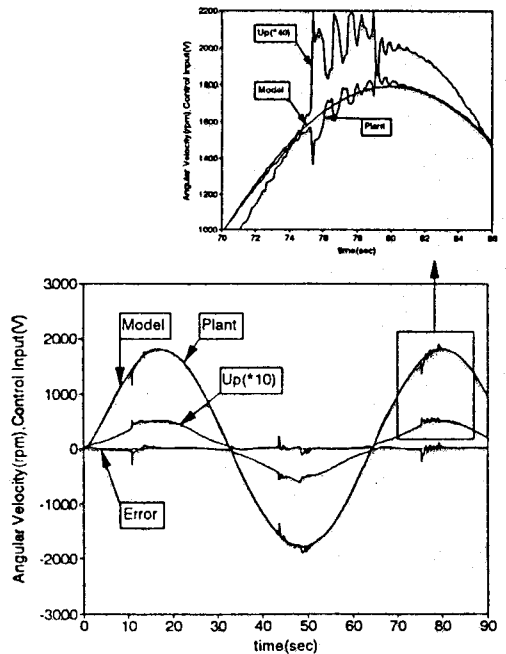


그림 10. 부하변동시 정현입력에 대한 적응제어 알고리즘의 실험결과

참고문헌

1. R.V. Monopoli, "Model reference adaptive control with an augmented error signal," IEEE Trans. Automat. Contr., Vol. AC-19, pp.474-484, 1974.
2. P. A. Ioannou and K. S. Tsakalis, "A Robust Direct Adaptive Controller," IEEE Trans. Automat. Contr., Vol.AC-31, No.11, pp.1033-1043, 1986.
3. B.C. KUO, Incremental motion control, SRI Publishing company, 1978.
4. T. KENJU, Power electronics for the microprocessor age, Oxford science publications, 1990.