

## 자기동조PID 알고리즘을 이용한 다중 processor 방식의 DC 서보모터 속도제어기의 구현

## A Study on the Implementation of a DC Servo Motor Speed Controller Using Self-tuning PID Algorithm, with Multi-processor

Hanyang University

### < Abstract >

This paper presents a DC servo motor controller using self-tuning PID algorithm, which can support Multi-processor for the real time processing. Computer simulation as well as experiment using Multi-processor(8088) are implemented with self-tuning PID algorithm. Presented algorithm is used to compare the performance of the controller with that of the classical PID controller through computer simulation and experiment. The result which use the Self-Tuning algorithm show that motor output follows the reference input trajectory fairly well inspite of load disturbances and parameter variations.

1. 서 례

과거 20년 동안, 반도체 소자가 개발되고 값싼 디지털 제어기를 용이하게 구성할 수 있게 되므로써 제어이론은 눈부시게 발달되었다. 따라서 상태회귀판제어기 기법과 같은 현대 제어이론에 의한 진보된 형태의 제어기가 제시되었다. 그렇지만 일반 산업계에서는 PID제어기가 여전히 널리 사용되고 있다.<sup>1), 2)</sup> 그 이유는 PID제어기가 조절되는 파라미터가 2-3개에 불과 하므로 그 구조가 간단하고 강 범위 위한 프로세스에 대해서도 동작이 안정되어 효율적으로 제어 성능을 유지 할 수 있다는 것이다. 그러나 플랜트의 파라미터의 미지수는 제어환경이 변화하는 경우에는 PID제어기는 불편하고 검정이 둑으로 설계된 선형 제어기로는 만족 할만한 특성을 얻을 수 없다. 이와 같은 경우에는 제어이기로는 파라미터들을 온라인으로 자동 조절하는 자기동조제어기(self-tuning controller)가 다른 방식의 제어이기로 비하여 효과적이다. 이 구조는 매우 이상상형 제어 성능을 갖지만 PID제어기의 구조에 비하여 복잡하므로 산업 프로세스에 직접 적용하는데는 여러 가지 문제들이 발생된다. 그렇지만 이 제어기를 간단한 PID 알고리즘의 구조로 구성할 수 있으면 현실적으로 실현하기가 쉽게 될 것이다.

자기동조PID제어에 관한 이론은 여러 방법이 제시되었다.<sup>1), 2), 4)</sup> Seborg는 기존의 PID와 같은 간단한 구조를 가지면서 자기동조제어기의 특성을 그대로 유지하는 자기동조PID제어기를 제작하였다.<sup>2)</sup> 그러나 산량이 많은 자기동조제어기를 서보전동기 시스템에 많은 영향을 미치는 원인을 microprocessor의 처리속도 때문에 있다.<sup>5)</sup>

마지막으로, 전통적인 PID 조율법은 고정된 조건에서만 유효하지만, 실제 운영 환경은 종종 예상치 못한 변화로 인해 예측하기 어렵다. 예를 들어, 제어 시스템은 예상치 못한 외부 영향(예: 기후 변화)에 대처해야 하는 경우가 있다. 이러한 상황에서 전통적인 PID 조율법은 예상치 못한 변화로 인해 예측하기 어렵다. 예를 들어, 제어 시스템은 예상치 못한 외부 영향(예: 기후 변화)에 대처해야 하는 경우가 있다. 이러한 상황에서 전통적인 PID 조율법은 예상치 못한 변화로 인해 예측하기 어렵다.

## 2.제어 알고리즘

## 2.1 자기동조 제어기

제어하고자 하는 플랜트를 다음과 같이 가정한다.

$$A(q^{-1})y(t) = q^{-k}B(q^{-1})u(t) + C(q^{-1})\xi(t) \quad (1)$$

$u(t)$ ,  $y(t)$ 는 플랜트의 입출력이고  $q^{-1}$ 은 단위시간 지연 연산자이다.  
여기서

$$A(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_n q^{-n} \quad (2)$$

$$B(q^{-1}) = b^0 + b_1 q^{-1} + \dots + b_m q^{-m} \quad (3)$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_l q^{-l} \quad (4)$$

$A(q^{-1})$ 와  $B(q^{-1})$ 은 서로 소이며  $C(q^{-1})$ 의 근은 모두 단위원 내에 있다고 가정한다. 또한 다항식들의 계수는 미지이다. 제어 입력  $u(t)$ 를 얻기 위하여 다음과 같은  
부조선형  $f(t)$ 를 도입한다.

$$\phi(t) = P(q^{-1})y(t) + Q(q^{-1})u(t-k) - R(q^{-1})W(t-k) \quad (5)$$

$$P(q^{-1}) = \frac{P_n(q^{-1})}{P_{n-1}(q^{-1})} \quad (6)$$

$$\frac{C(q^{-1})P_B(q^{-1})}{P_A(q^{-1})} = E(q^{-1})A(q^{-1}) + \frac{q^{-k}F(q^{-1})}{P_A(q^{-1})} \quad (7)$$

식 (1), (5), (7)로 부터

$$\phi(t+k) = \frac{F(q^{-1})}{P_a(q^{-1})C(q^{-1})}y(t) + \left[ \frac{E(q^{-1})B(q^{-1})}{C(q^{-1})} \right]$$

$$+ Q(q^{-1}) - u(t) = Rv(t) + E(t+k) \quad (8)$$

가 된다. 식 (8)의 예측 모델은 다음과 같이 표현된다.

$$\phi(t+k) = \phi^*(t+k|t) + \tilde{\phi}(t+k|t) \quad (9)$$

$\hat{\phi}(t+k)$ 은 예측오차를 나타내고, 이때의  $k$ 단 예측치는

$$\phi^k(t+k|t) = \frac{F(q^{-1})}{Pd(q^{-1})C(q^{-1})}y(t) + \left[ \frac{E(q^{-1})B(q^{-1})}{C(q^{-1})} \right. \\ \left. + Q(q^{-1}) \right] u(t) - R w(t) \quad (10)$$

이 경우에 제어식은 식(10)의  $\phi^k(t+k|t)$ 가 영이 되는  $u(t)$ 이다. 따라서 제어식은 다음과 같이 구해진다.

$$u(t) = \frac{C(q^{-1})R(q^{-1})w(t) - F(q^{-1})y(t)/Pd(q^{-1})}{E(q^{-1})B(q^{-1}) + C(q^{-1})Q(q^{-1})} \quad (11)$$

$A(q^{-1}), B(q^{-1})$  및  $C(q^{-1})$ 가 미지이면 예측모델의 식(8)을 사용하여 자기동조 제어기의 제어식을 구성해야 한다.

식(5)에서 시점  $t$ 의 신호  $Q(q^{-1})u(t-k)$ 와  $R(q^{-1})w(t-k)$ 는 알 수 있는 값이므로  $\phi(t)$ 를 예측하는 문제는 단순히

$$\phi y(t) = P(q^{-1})y(t) \quad (12)$$

을 예측하는 문제로 된다. 식(1)과 식(7)에서

$$\phi y(t+k) = \frac{F(q^{-1})}{Pd(q^{-1})C(q^{-1})}y(t) + \frac{E(q^{-1})B(q^{-1})}{C(q^{-1})}u(t) \\ + E_k(t+k) \quad (13)$$

이 때 절터핀 출력  $y_s(t)$ 를 다음과 같이 정의한다.

$$y_s(t) = \frac{y(t)}{Pd(q^{-1})} \quad (14)$$

이 경우  $G(q^{-1}) = E(q^{-1})B(q^{-1})$ 이라 하고  $C(q^{-1}) = 1$ 이면 식(13)은 다음과 같이 된다.

$$\phi y(t+k) = F(q^{-1})y_s(t) + G(q^{-1})u(t) + \epsilon(t+k) \quad (15)$$

여기서  $\epsilon(t+k)$ 는 잡음  $\xi(t)$ 의 이동평균이다.  $F(q^{-1})$ 와  $G(q^{-1})$ 의 추정치는 최소자승법을 사용하여 얻을 수 있다.

$k$ 단 예측치는

$$\hat{\phi} y(t) = X^T(t-k)\hat{\Theta}(t) \quad (16)$$

$$X^T(t) = [y_s(t-k), y_s(t-k-1), y_s(t-k-2) \\ ; u(t-k), u(t-k-1)]$$

$$\hat{\Theta}(t) = [f_0(t), f_1(t), f_2(t); g_0(t), g_1(t)]$$

미지의 매개변수  $\Theta$ 는 다음과 같이 축차형태로 추정될 수 있다.

$$\hat{\Theta}(t) = \hat{\Theta}(t-1) + P(t)X(t-k)[y_s(t) - \phi y(t-1)] \quad (17)$$

$$P(t) = \frac{1}{\mu} [P(t-1) \\ - \frac{P(t-1)X(t-k)X^T(t-k)P(t-1)}{\mu + X^T(t-k)P(t-1)X(t-k)}] \quad (18)$$

$F(q^{-1}), G(q^{-1})$  파라미터의 수를 각각  $L, J$ 라고 하면 다음과 같이 된다.

$$L = \deg A(q^{-1}) + \deg Pd(q^{-1}) \quad (19)$$

$$J = \deg B(q^{-1}) + \deg E(q^{-1}) + 1 \\ = \deg B(q^{-1}) + k \quad (20)$$

$G(q^{-1}) = E(q^{-1})B(q^{-1})$ ,  $C = 1$  그리고  $F(q^{-1})$ ,  $G(q^{-1})$  을 식(11)에 대입하면 식(21)이 구해진다.

$$u(t) = \frac{R(q^{-1})w(t) - F(q^{-1})y(t)/Pd(q^{-1})}{G(q^{-1}) + Q(q^{-1})} \quad (21)$$

## 2.2 PID구조를 갖는 자기동조 제어기

여기서는 제어하고자 하는 이산시스템을 설정한 후 이를 제어하기 위해 연속시간형 PID제어기 구조로부터 이산시간형 PID구조를 유도한다. 표준형 연속시간 PID제어기는 다음과 같이 정의된다.

$$U(t) = K_p(1 + \frac{1}{Ti s} + Td s)E(s) \quad (22)$$

$$E(s) = W(s) - Y(s) \quad (23)$$

$U(s), E(s), W(s)$  및  $Y(s)$ 는 각각 입력, 오차, 목표치 및 출력의 라플라스 변환이고,  $K_p$ 는 비례이동,  $Ti$ 는 적분(동작)시간,  $Td$ 는 미분동작시간이다. 그림1은 일반적인 PID제어기의 구조를 나타내고 있다.

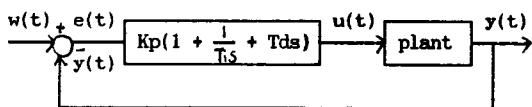


그림1 일반적인 PID제어기의 구조

Fig.1 Structure of a general PID controller

식(21)을 이산시간영역에서 표현하면 다음과 같다.

$$u(t) = K_p[1 + \frac{1}{Ti} \cdot \frac{T_s}{1-q^{-1}} + \frac{T_d}{T_s} (1-q^{-1})]e(t) \quad (24)$$

여기서  $T_s$ 는 샘플구간이다. 식(24)의 제어입력은 플랫트에 가해지기 때문에 위치형(position form)이라고 한다. 제어입력의 1단변화분은  $\Delta u(t) = u(t) - u(t-1)$ 이므로 속도형(velocity form)이라고 한다. 이 때(24)식은 다음과 같이 된다.

$$\Delta u(t) = K_p[1-q^{-1} + \frac{T_s}{Ti} + \frac{T_d}{T_s} (1-2q^{-1}+q^{-2})]e(t) \quad (25)$$

$$\text{여기서 } e(t) = w(t) - y(t) \quad (26)$$

목표치는 적분형에만 나타나고,  $y_s(t)$ 는  $y(t)$ 를 절터링한 것이라면 속도형 PID제어기는 다음과 같이 쓸 수 있다.

$$\Delta u(t) = K_p [e(t) - e(t-1) + \frac{T_s}{Ti} e(t) \\ + \frac{T_d}{T_s} [-y_s(t) + 2y_s(t-1) - y_s(t-2)]] \quad (27)$$

식(27)은 다음과 같이 정리 될 수 있다.

$$\Delta u(t) = K_p[(1+\frac{T_s}{Ti})-q^{-1}]w(t) - [(1+\frac{T_s}{Ti}) + \frac{T_d}{T_s}] \quad (28)$$

$$- (1 + \frac{2}{T_s})q^{-1} + \frac{T_d}{T_s} q^{-2} y_s(t) \quad (28)$$

자기동조제어기가 PID구조를 갖기 위해서는 제어기는 적분동작을 해야하고 다항식  $F(q^{-1})$ 의 차수는 2이어야 한다. 식(19)에서 다항식  $F(q^{-1})$ 의 차수가 2라는 것은  $L = 3$ 과 등가이다. 적절한 출력필터의 차수는  $\deg P_d(q^{-1}) = 1$  또는  $\deg P_d(q^{-1}) = 0$ 이다. 그러면 식(19)로부터  $L = 3$ 이라는 것은 2자나 3자 의 프로세스모델을 선택하는 것과 동가임을 알 수 있다. 즉  $n=2$  또는  $n=3$ 이다. 따라서

$$P_d(q^{-1}) = \frac{1+q^{-1}P_{d1}}{1+P_{d1}} \quad (29)$$

로 가정한다. 다음  $R(q^{-1})$ 는 설계시 임의로 지정할 수 있으므로

$$R(q^{-1})w(t) = [f_0(t) - f_2(t) + (f_1(t) + 2f_2(t))q^{-1}]w(t) \quad (30)$$

로 둔다.  $Q(q^{-1})$ 도 마찬가지로 설계자가 임의로 지정할 수 있으므로 자기동조제어작에 적분동작이 들어가도록 다음과 같이 선택한다.

$$G(q^{-1}) + Q(q^{-1}) = \frac{1-q^{-1}}{v} \quad (31)$$

여기서  $v$ 는 자기동조제어기를 동조시키는데 편리한 수단을 제공해 주는 설계파라미터이다. 식(31)과 (30)을 식(21)에 대입하면 다음의 PID구조를 갖는 자기동조제어기가 얻어진다.

$$u(t) = v[(f_0(t) - f_2(t)) + (f_1(t) + 2f_2(t))q^{-1}]w(t) - (f_0(t) + f_1(t)q^{-1} + f_2(t)q^{-2})y_r(t) \quad (32)$$

PID제어기와 대응되는 식은 식(28)과 식(32)을 서로 비교하여 얻어진다.

$$K_p = -v(f_1 + 2f_2) \quad (33)$$

$$T_d = \frac{vf_2 T_s}{K_p} \quad (34)$$

$$T_i = \frac{K_p T_s}{v(f_0 + f_1 + f_2)} \quad (35)$$

식(30)과 (31)을 만족하도록 시변의  $Q(q^{-1})$ 와  $R(q^{-1})$ 을 갖는 예측모델 식(15)을 사용하는데  $Q(q^{-1}), R(q^{-1})$ 를 명확하게 계산할 필요는 없다. 왜냐하면  $\phi(t)$ 에서 와는 달리  $\phi y(t)$ 에 있어서는  $Q(q^{-1})$  및  $R(q^{-1})$ 이 특별히 정의되지 않았기 때문이다. 식(35)에서  $v$ 는 종래의 제어기 이득  $K_p$ 와 같은 효력을 갖는다. 따라서  $v$ 의 큰 값은 부족제동응답, 작으면 과제동응답의 결과가 얻어진다.

### 3. 하드웨어 설계 및 실험

본 연구에서는 계산속도를 높이기 위하여 8088 CPU 2개를 가지고 다중 processor 제어기를 설계하여 계산함을 분배하였다. 또한 연산속도를 높이기 위하여 8087 (coprocessor)도 사용하였다.

#### 3.1 병렬통신회로

그림 2 와 같이 2개의 processor간의 병렬통신에는 8255(PPI)와 LS123의 저항값과 캐퍼시티값을 조절하여 사용하였다.

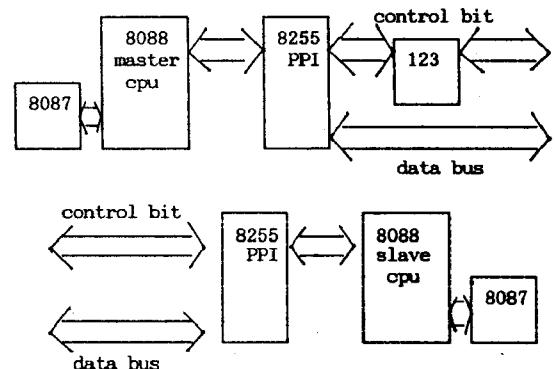


그림 2 병렬통신회로  
Fig.2 Parallel communication circuit.

### 3.2 전동기 제어시스템

전동기 속도는 전동기에 부착된 encoder로 측정되어 LS123을 이용한 단위시간당 폴스의 밀도로 제어입력을 조절하였다.

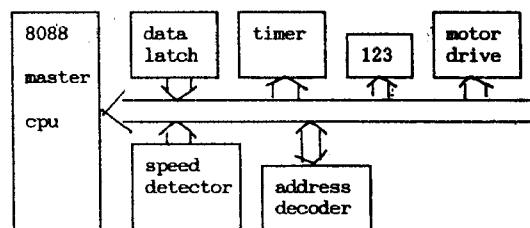
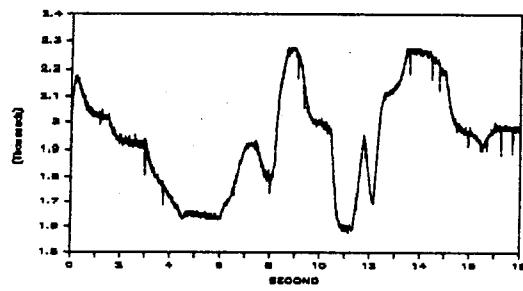


그림 3 전동기 제어시스템  
Fig.3 Motor control system

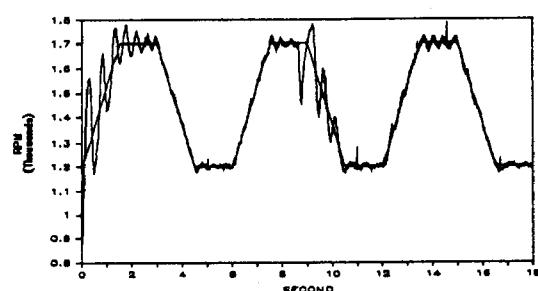
### 4. 실험 결과

본 연구에서는 고전적인 PID 알고리즘과 자기동조 PID 알고리즘을 DC서보전동기 속도제어에 적용하였다. 또한 주어진 서보전동기에 걸친 적인 부하변동이 생겼을 때의 각 알고리즘의 부하복복 상태를 보의 실험을 통해 확인하고 실험을 통하여 입증하였다. 기준입력을 사다리꼴 입력을 사용하였고 고전PID의 이득은 부하변동이 없을 때의 최적값을 선정하였다. 그림 4는 부하를 가감시킬 수 있는 실험장치를 사용하여 9초에서 부터 부하를 가했을 경우의 PID제어방식에 대한 응답을 보여준다. 그림 5는 PID경우와 같은 부하조건의 자기동조 PID제어 방식에 대한 결과이다.

아래 그림에서 알 수 있듯이 초기 상태에서는 고전적인 PID제어기가 자기동조 PID제어기보다 상당히 빨리 기준입력에 수렴해감을 볼 수 있지만 부하변동에 의해 모델의 미개변수가 영향을 받았을 경우, 고전PID 제어기보다 자기동조 PID제어기가 부하변동에 따른 회복 속도가 우수함을 보이고 있다.



(a)



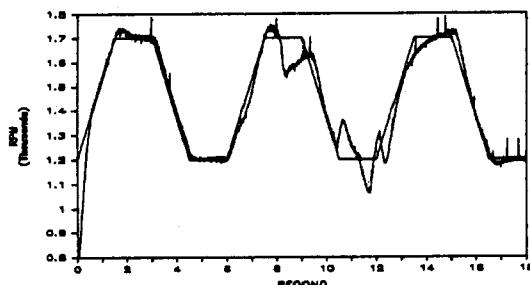
(b)

그림 4. PID제어방식

a) 입력 b) 출력

Fig.4 PID control method

a) input b) output



(b)

그림 5. 자기동조 PID제어방식

a) 입력 b) 출력

Fig.5 Self-tuning PID control method

a) input b) output

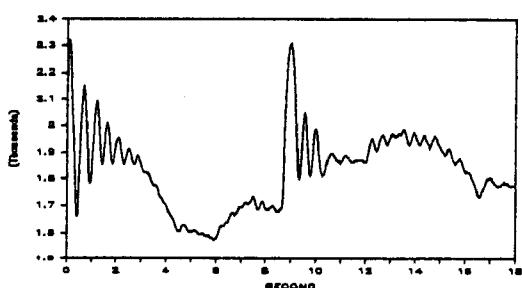
## 5. 결론

본 연구는 기존의 PID제어기와 같은 간단한 구조를 가지면서 자기동조 제어기의 특성을 그대로 유지하는 자기동조 PID 알고리즘을 DC서보전동기 속도제어에 적용하였다. 제어기의 하드웨어 설계시에는 다중processor를 사용하여 1개의 processor를 사용하는 기존의 제어기 보다 계산속도를 높였으며, 따라서 계산정밀도는 적은 적응제어 알고리즘인 자기동조 PID 알고리즘을 무리 없이 구현할 수 있었다. 그리고 부하가 가해졌을 경우에 자기동조 PID제어기가 매개변수를 추정하지 않고 전작인 PID제어방식보다 수렴속도 및 부하에 대한 회복능력에 있어서 훨씬 우수함을 실제 실험을 통해 입증하였다.

다중 processor 제어기 설계시 중요한 사항은 통신시간을 줄이는 것이다. 16bit, 32bit의 processor 및 통신기능이 강화된 IC를 채택하면 제어기의 부품수를 줄여 신뢰성을 높이는 한편 시스템의 스피드와 효율을 극대화 할 수 있을 것이다.

## 6. 참고문헌

- P.J. Gawthrop, "Self-tuning PID controllers; Algorithm and Implementation," IEEE Trans. Automat. Contr., Vol.AC-31, No.3, pp.201-209, 1986
- F.Cameron, D.E Seborg, "A self-tuning controller with a PID structure," Int.J.Cont., Vol.38, No. 32, pp.401-417, 1983.
- B.Wittenmark, "Self-tuning PID-Controllers Based on Pole-placement," Dept. of Auto. Contr., Lund Inst. of Tech., Lund, Sweden, Rep., No. LUF2/(TFRT-7179/0371/), 1979.
- D.W.Clarke, "Implementation of self-tuning controllers", in C.J. Harris and S.A. Billings (Eds) Self-tuning and adaptive control. Peter Peregrinus. 1981.
- 양해원, 강향수, "극ベ치 PID자기동조 제어방식에 의한 DC서보전동기 속도제어에 관한 연구." Trans. KIEE, Vol 37. No.9 SEP.1988
- B.C.Kuo, Incremental motion control, SRL Publishing company 1978.
- Ray Duncan, Advanced MS-DOS, Microsoft 1986.



(a)