

# PC 기반의 순시속도 검출에 의한 서보 모터의 저속 제어

류재규\*, 박정일\*, 이석규\*\*

\*영남대학교 공과대학 전자공학과, \*\*영남대학교 공과대학 전기공학과

## PC-Based Low Speed Control of a Servo Motor Using Instantaneous Speed Detection

Jaekyu Ryu\*, Jungil Park\*, Suckgyu Lee\*\*

\*Department of Electronic Engineering, \*\*Department of Electrical Engineering, Yeungnam University

### ABSTRACT

The low speed control of a servo motor using instantaneous speed detection method is described. To estimate the instantaneous speed from the average speed, the speed estimator of the first or second order is used. We confirm that these estimators improve the speed control performance of a servo system with experiments.

### 1. 서론

현재 산업기에서 가장 많이 이용되고 있는 서보모터의 속도검출기로는 인크리멘탈 엔코더와 같은 디지털식 검출기가 많이 이용되고 있다. 이 엔코더는 가격이나 펄스의 카운트, 전송측면에서 보면 사용 가능한 엔코더의 펄스수에는 한계가 있다. 이 때문에 펄스수가 많지 않은 엔코더를 사용할 경우 특히 초저속에서는 펄스간격이 길어지게 되어서 순시 속도를 검출하는데 애로점이 발생한다.<sup>5)</sup> 그러므로 저속에서의 속도제어시에는 속도검출 지연시간이 증가해서 속도제어계가 불안정해지기 쉽다. 그래서 인크리멘탈 엔코더를 이용해서 초저속에서 안정하게 응답성이 좋은 속도제어계를 실현하기 위해서는 초저속영역에서도 검출지연이 적은 속도 검출치를 얻을 필요가 있다. 그래서 근래에 와서 순시속도를 검출하기 위한 여러 방법들이 연구 되고 있는데, 대표적으로 속도 간측기를 이용하는 방법이 연구되고 있다.<sup>4,5,6)</sup> 그러나 이 방식은 디지털적으로 구현이 될 때 계산량이 많기 때문에 DSP (Digital Signal Processor) 칩이 이용되고 있다.

[1]의 논문은 인크리멘탈 엔코더를 사용하는 서보모터의 속도를 검출하는 방법에 관한 논문으로서 저속이나 고속에 관계없이 높은 분해능을 갖는 속도를 검출하는 M/T 방식을 제안한 효시논문이다. 그러나 이 논문은 속도검출을 할 때 순간의 실제속도가 아니라 평균속도를 검출하게 되어 검출 지연이 존재하게 되며, 또 초저속제어를 수행하기 위해서는 Dead Time 이 기시간과 비슷 시켜야 저이 있다.<sup>2)</sup> [3]의 논문은 1차나 2차의 추정기를 이용하여 한 스텝 혹은 두 스텝 전의 샘플링시의 속도검출값을 이용하여 현재순간의 속도값을 추정하는 방법을 제시하였다. 그러나 이 논문은 고속제어의 진동억압제어에 대해서 고찰하였다.

본 논문에서는 복잡하지 않고 마이크로프로세서 레벨에서 구현이 가능한 방법을 모색하여 순시속도를 검출할 수 있는 방법을 제시한다. 이 방법은 1차나 2차의 추정기를 이용하여 순시속도를 추정하는 것이다. 대부분의 속도 제어계는 속도 Profile로 시다리꼴의 모양을 사용하고 있기 때문에 가속이나 감속이 되는 저속구간에서는 Ramp 지령이므로 1차의 추정기는 순시속도를 검출하는데 탁월한 효과를 갖게 된다.

그리고 실제 퍼스널컴퓨터에 인터페이스할 수 있는 하드웨어를 제작하고, 또 C 언어로 순시속도를 검출하고 제어하는 소프트웨어 알고리즘을 구성하여, 실시간으로 초저속에서 이 1,2차의 추정기가 속도제어에 우수한 효과를 나타냄을 실험을 통하여 보인다.

### 2. 순시 속도 검출

#### 2.1. 평균 속도 검출

평균속도를 검출하는 방법은 [2]의 논문에서 사용한 방

법을 사용하여 검출한다. 검출원리를 그림1에서 나타내고 이 때 측정된 회전속도(rpm)은 다음과 같이 쓸 수 있다.

$$N(\text{rpm}) = \frac{60f_c m_1}{Pm_2} \quad (1)$$

여기에서,  $m_2 = m_3 - m_4(i) + m_4(i-1)$  이다.

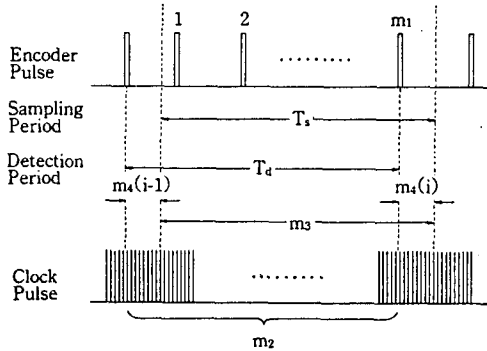


Fig. 1. Average speed detection using ideal M/T method.

## 2.2. 순시 속도 검출

순시속도는 다음에서 설명하는 바와 같이 한스텝 혹은 두스텝 전의 값을 이용하여 추정할 수 있다.<sup>3)</sup>

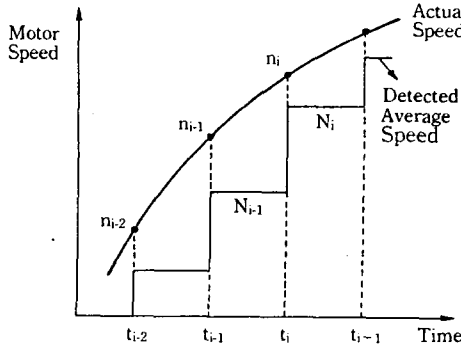


Fig. 2. The relation of the instantaneous and the average speed.

i번째 샘플링 구간에서의 검출속도는 다음과 같이 주어진다.

$$N_i = \frac{1}{T} \int_{t_{i-1}}^{t_i} n(t) dt \quad (2)$$

식(2)에서 검출속도  $N_i$ 는 그림2에서 알 수 있는 바와 같이 실제의 순시속도  $n_i$ 보다 검출구간의 약 반만큼 지연이 된 평균속도가 검출된다. 그래서 다음과 같이 순시속도를 추정한다.

1차 추정 : 모터속도  $n(t)$ 를 1차의 Ramp라고 가정하면 추정속도는 다음과 같다.

$$\hat{n}_i = \frac{3N_i - N_{i-1}}{2} \quad (3)$$

2차 추정 : 모터속도  $n(t)$ 를 2차의 Parabolic이라고 가정하면 추정속도는 다음과 같다.

$$\hat{n}_i = \frac{11N_i - 7N_{i-1} + 2N_{i-2}}{6} \quad (4)$$

## 2.3. 제안 순시속도 검출방법

### 2.3.1. 샘플링 구간내에 엔코더펄스가 1개이상 존재하는 경우

샘플링 구간내에 엔코더의 펄스가 1개 이상인 저속인 경우에 순시속도를 추정하는 원리를 그림3에서 설명한다. 실제 대부분의 서보제어에서 위치제어시 수행되는 속도제어의 속도 profile은 사다리꼴이기 때문에 위치제어로 들어가기 전의 속도제어는 1차 추정기를 이용하는 것이 효과적이다. 순시속도를 추정하는 방법은 먼저 식(1)에 의해 평균속도를 계산한 후, 다음과 같은 1차 추정기를 이용하여 순시속도를 추정한다.

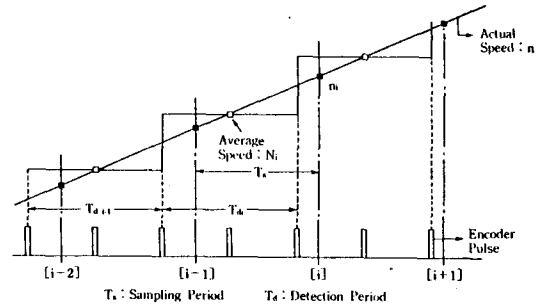


Fig. 3. Instantaneous speed detection at low speed region.

저속에서는 검출구간 끝에서의 순시속도와 샘플링 시점에서의 순시속도의 차이가 상당하므로 식(3)을 이용하여 순시속도를 추정하면 오차가 많이 발생한다. 그래서, i번째 샘플링 시각에서의 순시속도(■표시)는 평균속도(□표시)를 이용하여 다음과 같이 추정한다.

$$\hat{n}_i = \frac{N_i - N_{i-1}}{\frac{T_{di} + T_{di-1}}{2}} \left[ \frac{T_{di-1}}{2} + T_{di} + m_4(i) \right] + N_{i-1} \quad (5)$$

식(5)에서  $T_{di} \approx T_{di-1}$ 로 가정하면,

$$\hat{n}_i = (N_i - N_{i-1}) \cdot \left( 1.5 + \frac{m_4(i)}{m_2} \right) + N_{i-1} \quad (6)$$

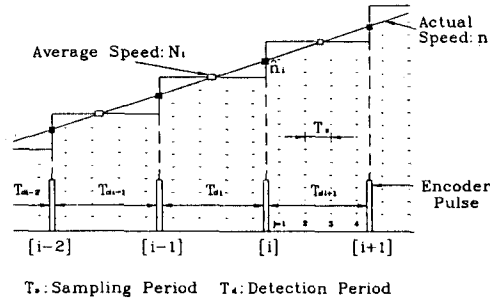
이 된다. 식(6)과 식(3)의 차이점은  $m_1(i)/m_2$ 의 양만큼 순시속도 오차가 발생한다. 이 양은 최악의 경우 1이므로 식(3)를 이용해서 저속에서 순시속도를 추정하는 것은 많은 오차를 발생할 가능성이 있다.

### 2.3.2. 샘플링 구간내에 엔코더펄스가 1개이하인 경우

속도가 초저속이어서 1샘플링 구간내에 엔코더 펄스가 카운트 되지 않을 경우를 그림4을 이용하여 설명한다. 그림4의 경우는 1차의 추정기를 사용하여 순시속도를 검출하는 원리를 나타낸 것이다. 여기에서  $j$ 는 엔코더 펄스가 나타날 때까지의 샘플링 구간 갯수를 카운트하기 위한 것이다. (1)식을 이용하여 먼저 평균속도를 검출할 수 있는데 이 때의  $m_2$ 는 다음식의 값으로 된다.

$$m_2 = j \cdot m_3 - m_4(i) + m_4(i-1) \quad (7)$$

평균속도  $N_i$ 를 이용해서  $j$ 가 1, 2, ... 인 샘플링 순간에서의 순시속도를 1차로 확장 추정할 수 있지만 검출구간  $T_{di}$ 에 비해서 추정해야 할 구간이 길기 때문에 불확실한 추정이 될 가능성이 많다. 그래서 그림4와 같은 초저속인 경우에는 (3)식이나 (4)식을 이용하여  $[i]$ 번째 엔코더 펄스 시점에서의 순시속도를 추정하여  $[i+1]$ 번째 구간 ( $j=1, 2, \dots$ )까지 같은 속도로 간주한다.



\*  $j = 1$  (low speed)

$$\hat{n}_i = \frac{3N_i - N_{i-1}}{2} + \frac{m_4(i)}{m_2} (N_i - N_{i-1})$$

$j > 1$  (very low speed)

$$\hat{n}_i = \frac{3N_i - N_{i-1}}{2} : \text{first-order estimation}$$

or

$$\hat{n}_i = \frac{11N_i - 7N_{i-1} + 2N_{i-2}}{6} : \text{second-order estimation}$$

Fig. 4. Instantaneous speed detection at very low speed region.

### 2.3.3. 순시속도 추정 순서도

앞 두절의 저속인 경우와 초저속인 경우에 평균속도를 검출하여 순시속도를 추정하는 경우를 순서도로 나타내면 그림5와 같다. 순서도에서  $j$ 는 초저속인 경우에 엔코더 펄스사이에서 몇개의 샘플링 구간이 들어가는 가를 카운트하는 인자이다. 샘플링 구간내에 여러개의 엔코더 펄스가 카운트되는 저속의 경우에는 (6)식을 이용하여 순시속도를 추정하고, 1 샘플링 구간내에 엔코더 펄스가 카운트되지 않는 초저속인 경우에는 (3)식이나 (4)식을 이용하여 속도를 추정한다.

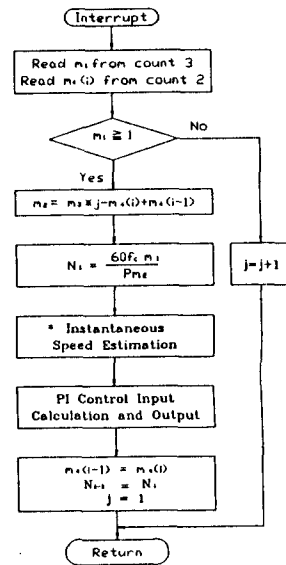


Fig. 5. Instantaneous speed estimation flowchart.

### 2.4. 순시속도 추정의 효과 검토

저속에서의 순시속도의 추정의 효과를 살펴보기 위해서 1차직선으로 감소하는 속도 Profile에 대해서 추정한 시뮬레이션 결과를 그림6에 나타내었다. 그림의 아래쪽에 검은 눈금은 엔코더 펄스의 발생시점이며 샘플링 시간은 10 msec이다. □로 표시된 것은 (1)식으로 검출한 평균속도이며, ▽로 표시된 것은 (3)식을 이용하여 1차 추정한 속도이며, △로 표시된 것은 본 논문에서 제안한 (6)식을 이용하여 1차 확장 추정한 결과이다. 그림에서 보는 바와 같이 확장 추정한 방식이 실제 순시 속도에 가장 근접함을 알 수 있다.

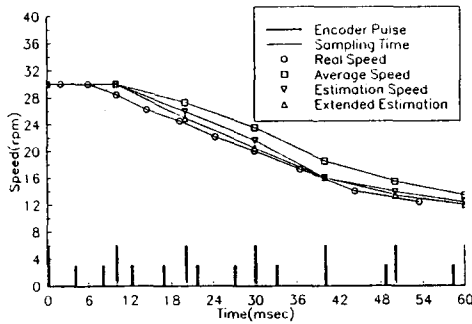


Fig. 6. Comparison of the various speed detection method.

### 3. 실험 및 고찰

#### 3.1. 하드웨어 구성

앞에서 설명한 대로 순시속도를 검출해서 저속제어를 하기 위하여 그림7과 같은 하드웨어를 구성할 수 있다.

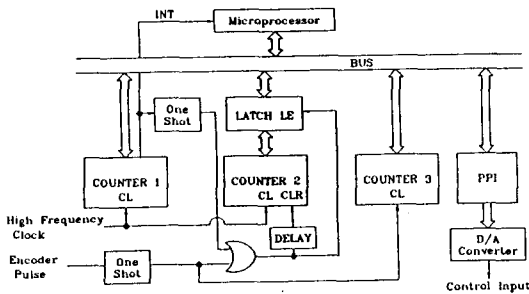


Fig. 7. Digital hardware block diagram of the low speed control.

실험을 하기 위한 하드웨어는 TTL소자와 LSI소자(8253, 8255)를 이용하여 구성하였으며 IBM-PC/AT의 slot에 장착할 수 있도록 카운터들과 래치, 8255등을 PC에서 이용하지 않고 있는 I/O 번지로 할당하였다.

속도를 읽어들이어서 순시속도를 추정하고 제어입력을 계산, 인가하는 모든 알고리즘은 C 언어를 사용하여 작성하였으며 compile한 수행파일을 수행시켜 저속제어를 수행하였다. 샘플링 시간은 10 msec로 하였는데 이것은 카운터1(8253)로 들어오는 PC의 1 MHz의 클럭 주파수를 분주하여 10 msec마다 인터럽트를 걸어 샘플링을 하였다. 엔코더의 펄스간격을 측정하기 위한 카운터2로 들어가는 높은 주파수도 1 MHz의 PC 클럭을 이용하였다.

모터는 ELECTRO-CRAFT 회사의 DC 서보 모터를 이용하였

는데, 엔코더의 1회전당 펄스수가 800개의 것을 이용하였다.

카운터1은 높은 입력 주파수를 분주하여 원하는 일정 시각마다 마이크로 프로세서에 인터럽트를 걸어 샘플링 시점을 발생시키기 위한 카운터이다.

카운터2는 그림3에서의  $m_i(i)$ 와  $m_i(i-1)$ 의 폭을 측정하기 위한 것이다. 엔코더 신호와 인터럽트 신호의 상승에서 짧은 기간의 펄스를 발생시켜 이 신호로 카운터2의 값을 래치에 래치 시킴과 동시에 클리어 시킨다. 이렇게 카운트된 카운터 값을 읽어 들이기 위해서 CPU는 인터럽트가 걸리면 첫순간에 래치의 값을 읽어들이면 샘플링 이전의 엔코더 펄스와 샘플링 펄스 사이의 폭이 측정되게 된다.

카운터3는 샘플링 구간 동안의 엔코더 펄스 수를 헤아리기 위한  $m_i$  카운터이다.

카운터1과 카운터3은 8253을 이용하였으며 카운터2는 TTL을 사용하여 16비트의 동기카운터로 구성하였다.

#### 3.1.1. 순시속도 검출

위와 같이 측정된 값으로 (1)식을 이용하여 평균속도를 먼저 검출하고 그림5의 프로그램으로 순시속도를 추정하였다.

#### 3.1.2. 제어입력 인가

제어입력은 PC에서 PI(Proportional-Integral) 제어입력을 계산하여 PPI(8255)를 통하여 8비트의 D/A 변환기로 출력하였다.

### 3.2. 결과 고찰

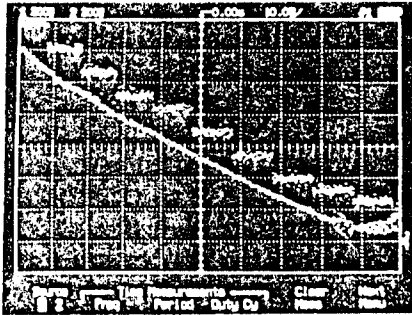
#### 3.2.1. 각종 방식의 속도 검출실험

그림8은 그림6에서 각종 방식의 속도 검출을 시뮬레이션한 것과 같이 실험으로 검출한 속도이다.

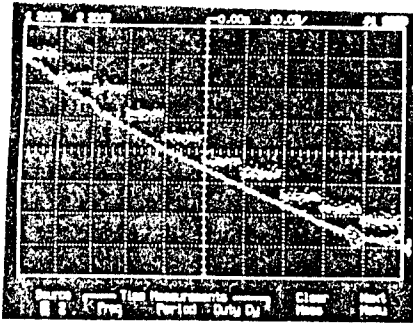
모터에는 타코제너레이터가 부착되어 있어서 실제의 순시속도를 알 수가 있다. 먼저 속도제어를 수행하기 전에 이 타코제너레이터의 출력과 본 논문에서 다루고 있는 각종 디지털 속도 검출방식을 이용하여 10 msec마다 속도를 검출하여 D/A 변환기로 출력하여 얼마나 순시속도에 가까운가를 비교하였다. 초저속의 경우에는 타코제너레이터의 출력에 잡음이 많이 섞여 비교가 곤란하여 60rpm에서 200rpm내에서 비교를 하였다. 오실로스코프의 가로축 1 눈금은 10 msec이다.

그림8의 결과는 모터의 회전속도가 최고 200rpm 이었으며 최저가 60rpm 이었다.

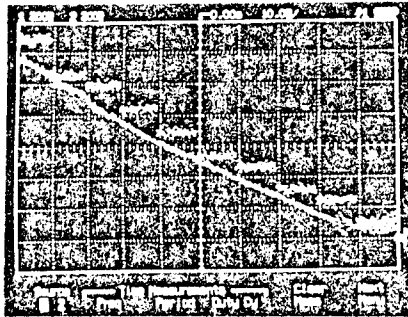
그림8의 (a)는 순시속도를 추정하지 않은 평균속도와 타코제너레이터의 값을 비교한 것이며, (b)는 (3)식의 1차 추정기를 이용하여 순시속도를 추정한 것이며, (c)는 (6)식의 확장된 1차 추정기를 이용하여 추정한 것이다. 예상대로 순시속도를 추정하지 않은 평균속도는 순시속도에 비해서 10 msec정도의 검출지연이 존재한다는 것을 알 수 있다. 아울러 (6)식으로 추정한 방식이 순시속도에 가장 근접함을 알 수 있다.



(a) average detection



(b) detection using 1st order estimator

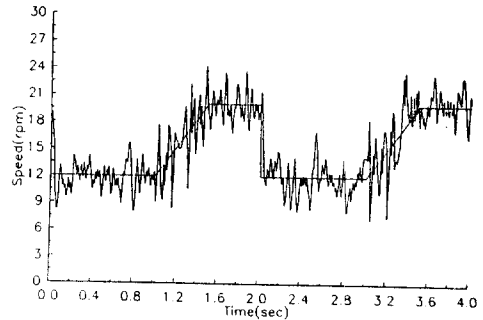


(c) detection using extended 1st order estimator

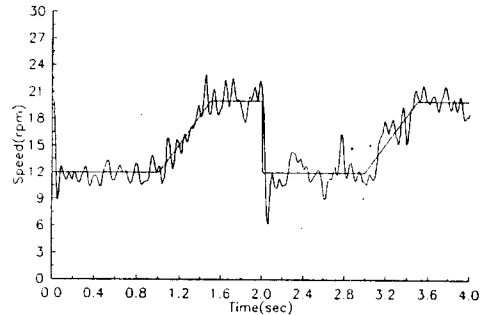
Fig. 8. Comparison between the real speed and the detection speed.

### 3.2.2. 저속제어 실험

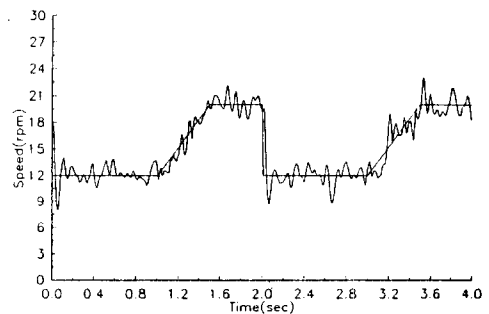
그림9는 12rpm과 20rpm사이를 속도제한한 결과이다. 결과에서 볼 수 있듯이 (6)식을 사용하여 순시속도 추정한 결과를 이용한 (c)의 경우가 가장 좋은 결과를 나타낼 수 있다. 또 그림10은 엔코더의 분해능인 7.5rpm보다 저속인 5rpm의 일정속도로 속도제어를 수행한 결과이다. 여기에서도 (4)식을 이용하여 2차로 순시속도를 추정한 결과를 이용한 (c)의 경우가 가장 좋은 결과를 얻었다. 대체로 실험결과 데이터의 리플이 심한 것은 샘플링 기간이 10msec로 너무 길고, 엔코더의 회전당 펄스수가 800개로 너무 적고, 모터에 연결된 관성모멘트가 너무 크며, D/A 변환기의 분해능(8비트)이 낮는데 기인한 것 같다. 초저속제어를 더 건실하게 수행하려면 이러한 점을 고려할 필요가 있을 것 같다.



(a) M method

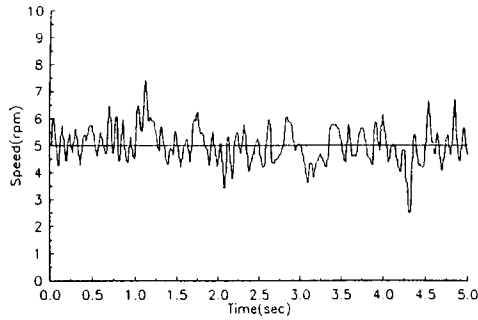


(b) average M/T method with eq. (1)

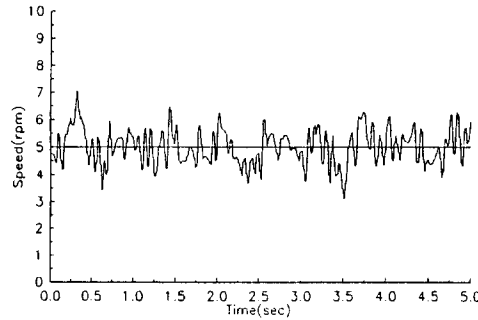


(c) extended 1st order estimation with eq. (6)

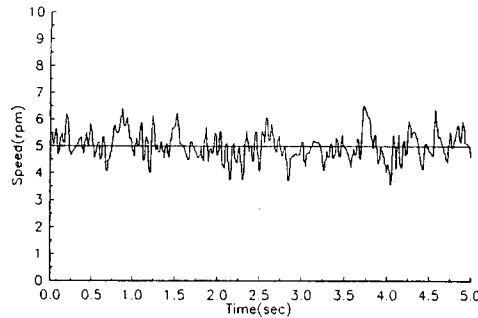
Fig. 9. Experimental results at low speed control.



(a) average M/T method with eq. (1)



(b) 1st order estimation with eq. (3)



(c) 2nd order estimation with eq. (4)

Fig.10. Experimental results at very low speed control.

#### 4. 결 론

본 논문에서는 엔코더를 사용하는 서보 시스템에서 순시속도 검출에 의한 저속제어를 수행하였다. 저속에서의 순시속도를 검출하기 위하여 확장된 1차 추정기를 제안하였으며, 엔코더 분해능 이하의 초저속에서는 1,2차의 추정기를 이용하여 순시속도를 검출하여 속도제어를 수행하

는 것이 평균속도 검출에 의한 속도제어 보다도 우수한 성능을 나타낸다는 것을 실험을 통하여 확인하였다.

본 논문에서 제시한 하드웨어는 간단하여 마이크로 프로세서 레벨에서 구현 가능하기 때문에 향후 서보제어계의 성능향상에 도움이 되리라고 본다.

#### 참 고 문 헌

- [1] T.Ohmae et al., "A Microprocessor-Controlled High Accuracy Wide-Range Speed Regulator for Motor Drives," *IEEE Trans. Ind. Elec.*, vol. IE-29, no. 3, pp.207-211, Aug.1982.
- [2] 채상락, 박정일, 이석규, "서보모터의 고정도 속도 검출을 위한 M/T 방식의 하드웨어 구현," *한국자동제어 학술회의*, 1993.
- [3] K.Saito et al., "A Microprocessor-Controlled Speed Regulator with Instantaneous Speed Estimation for Motor Drives," *IEEE Trans. Ind. Elec.*, vol. IE-35, no.1, pp.95-99, 1988.
- [4] K.Fujita et al., "Instantaneous Speed Detection with Parameter Identification for ac Servo Systems," *IEEE Trans. Ind. Appl.*, vol. IA-28, no.4, pp.864-872, July/August 1992.
- [5] 土手康彦, *ロバスト 高速 サーボ 制御技術*, トリケッス 出版部, pp.251-266, 1991.
- [6] 渡邊, "速度推定 オブザーバを用いた デジタル サーボ," *電氣學會 論文誌 D*, vol.107, no.12, pp.1468-1474, 1987.