

# 포화를 고려한 직류전동기의 유한시간 정정응답 제어

## Speed Control of DC Motor Using Deadbeat Response Method with Consideration of Saturation

劉完植 \* · 金榮石 \*\* · 大能繁 \*\*\*

(Wan - Sik You · Young Seok - Kim · Shigeru Okuma)

### 요 약

본 논문은 유한시간 정정응답 제어이론을 이용한 직류전동기의 속도제어에 대하여 논하였다. 유한시간 정정응답 제어계는 이산시간제어를 적용하므로 제어량 포화현상과 검출지연 문제가 발생하여 계통의 불안정을 초래한다. 이러한 문제를 해결하기 위하여 포화상태에서도 고속응답이 가능하도록 예측제어를 적용한 보상기를 제안한다.

실험 결과 지령치가 포화되지 않은 상태에서는 1샘플링시간으로 정정할 수 있었다. 지령치가 포화한 상태에서는 포화로 부터 벗어난 후 1샘플링시간으로 정정할 수 있었다. 또한 예측제어를 적용함으로써 과도시의 오우버슈우트가 억제된 고속정정이 가능함을 알 수 있었다.

### Abstract

This paper presents the speed control of DC motor based on deadbeat response method. Since the deadbeat response systems are characterized by the discrete time control, the instability of the systems caused by saturation and time lag problems is inevitable. In order to release these problems, we propose a compensator utilizing the predictive control so that the fast response can be also achieved in the saturation state.

Experimental results demonstrated that outputs are able to settle final values in one sampling time for unsaturated reference inputs. For saturated reference inputs, outputs take one sampling time after getting free from the saturation state. Further we are able to settle the fast response with suppressed overshoot by applying the predictive control.

\* 正會員：仁荷大 大學院 電氣工學科

\*\* 正會員：仁荷大 工大 電氣工學科 助教授 · 工博

\*\*\* 名古屋大 工學部 電子機械工學科 教授

接受日字：1991년 6月 4日

### 1. 서 론

산업의 급속한 발전과 더불어 산업용 로봇이나 컴퓨터 주변기기와 같은 정밀도를 요하는 분야가 늘어나게 됨에 따라 전동기에 있어서도 고성능의 제어특성이 요구되어지고 있다. 고정도 고속응답특성을 얻기 위한 제어법으로 PI 또는 PID 보상기를 적용하거나 진상-지상 보상기등을 적용한 연속치제어계가 적용되어 왔다.<sup>1)</sup> 그러나 이러한 제어방법은 루프이득에 매우 민감하고, 설계에 변경이 어려우며, 잡음의 영향을 받기 쉬운 단점이 있다. 그래서 어떤 특정한 것에 제한되지 않고 매우 다양한 제어법을 행할 수 있으며, 소프트웨어상에서 제어기의 회로정수나 알고리즘만을 수정함으로써 설계변경에 용이하게 대처할 수 있는 장점을 가진 마이크로프로세서를 이용한 이산치제어가 널리 행해지고 있다.<sup>2), 3)</sup> 이산치제어법 중 유한시간 정정응답제어는 최소의 샘플링시간으로 목표치에 정정되고 정상상태오차가 거의 없는 특성을 가지고 있는데, 이러한 유한시간 정정응답 제어를 이용한 직류전동기의 속도제어법에 대한 연구가 행하여 졌다.<sup>4), 5)</sup> 유한시간 정정응답제어에 있어서는 이산치제어에서 일반적으로 나타나는 문제점 즉, 제어량이 커짐에 따라 나타나는 포화현상, 제어 알고리즘을 연산하기 위해 소요되는 시간지연, 펄스엔코더의 사용에 의해 발생하는 검출지연 등의 문제점이 대두된다. 따라서 최근에는 이러한 시간지연, 검출지연의 문제점을 해결하기 위한 예측제어법 등이 제안되고 있다.<sup>6), 7)</sup> 그러나 제어량포화 현상은 제어계의 응답을 지연시키는 문제가 있음에도 불구하고 대부분의 연구는 제어량포화의 문제점에 대해서는 거의 언급하지 않고 있어 이에 대한 연구도 행해져야 하리라고 본다.

본 연구에서는 제어량포화를 고려한 유한시간 정정응답 제어를 이용하여 보상기를 설계함으로써 포화상태에서도 고속정정이 가능한 직류전동기의 속도제어를 행하고자 한다. 또한, 속도귀환신호로 사용하는 펄스엔코더의 출력펄스열은 일정시간의 평균속도가 되는 검출지연

이 발생하는데, 이의 해결을 위해 펄스엔코더의 출력펄스열을 일정시간 계수함으로써 얻어진 평균속도의 검출치를 복수개 사용하여 현시점에서의 순시속도를 예측하고 그 예측치를 속도귀환신호로써 사용하는 순시속도 예측법을 이용한 경우의 속도응답성에 대해서도 검토한다.

### 2. 유한시간 정정응답 제어

그림 1은 샘플치제어계의 블록선도이다. 이 시스템에서  $D(z)$ 는 보상기(마이크로프로세서)에 해당하는 부분이고,  $H$ 는 영차홀더 요소, 즉  $D/A$  변환기에 상당하며,  $G$ 는 제어대상을 나타낸다. 출력  $n_o$ 의 귀환신호를 속도지령치  $n_i$ 와 비교하여 오차신호  $e$ 를 만들고, 그것을 샘플링하여 샘플치  $e^*$ 를 얻는다. 단, 그림에서  $N_o, N_i, E$ 는 각각  $n_o, n_i, e$ 의 라플라스변환을 표시하고,  $E(z)^*$ 는  $e^*$ 의  $z$ 변환을 나타낸다. 이것을 통하여 전압지령의 샘플치  $m^*$ ( $z$ 변환하면  $M(z)$ )를 얻고,  $m^*$ 를 영차홀더한 파형  $m$ 을 제어대상인 전달함수  $G$ 에 가하여 출력  $n_o$ 를 얻는다. 그림 1에서 시스템의 폐루프 전달함수를  $W_c(z)$ 라 하면

$$W_c(z) = \frac{D(z)HG(z)}{1 + D(z)HG(z)} \tag{1}$$

이고, 식 (1)로 부터  $D(z)$ 는

$$D(z) = \frac{1}{HG(z)} \frac{W_c(z)}{1 - W_c(z)} \tag{2}$$

가 된다.  $D(z)$ 는  $W_c(z)$ 가 적절하게 결정되면 마이크로프로세서상에서 프로그램에 의해 정해진다.

여기서,  $H = \frac{1 - e^{-Ts}}{s}$ ,  $T =$  샘플링 주기

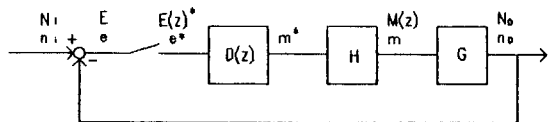


그림 1. 샘플치 제어계의 블록선도  
Fig. 1. Block diagram of the discrete control system

$$G = \frac{1}{1 + T_m \cdot s}, T_m = \text{직류전동기의}$$

기계적 시상수

그런데,  $W_c(z)$ 는 일반적으로 다음과 같이 표현된다.

$$W_c(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_m z^{-m} \quad (3)$$

만약, 이 시스템에 입력  $u(t)$ 가 주어진다면 출력  $y(t)$ 는  $t = nT$  ( $T$ : 샘플링 주기)에서,  $W_c(z) = Y(z)/U(z)$  이므로  $Y(z) = W_c(z)U(z)$ 가 되고, 이를 역 z변환하면

$$y(nT) = a_0 u(nT) + a_1 u[(n-1)T] + \dots + a_m u[(n-m)T] \quad (4)$$

가 된다. 이렇게  $W_c(z)$ 를  $z^{-1}$ 의 급수로 전개했을 때 유한개의 함수로 끝나도록 하면 유한회의 샘플링동작으로 목표치에 정정시킬 수 있다. 일반적으로, 제어대상의 차수가  $m$ 차인 경우 임의의 초기치로부터  $m$ 회의 샘플링동작으로 목표치에 정정시킬 수 있는데, 이것을 유한시간 정정응답(Deadbeat Response)이라 한다<sup>9)</sup>.

### 3. 시스템의 구성 및 보상기의 설계

#### 3.1 시스템의 구성

그림 2에 시스템 구성의 개략도를 표시한다. 펄스엔코더의 속도피환신호와 속도지령치와의 오차신호에 의해서 속도제어 연산부에서 유한시간 정정응답 제어를 실행하는 계산을 행한다. 마이크로프로세서는 8bit Z80을 사용하고 속도검출은 분해능 2000Pulse/rev의 펄스엔코더를 사용했다. 구동회로는 4상한 초퍼로 D/A 변환기의 출력전압이 1.75배 선형증폭되어 직류전동기에 인가된다.

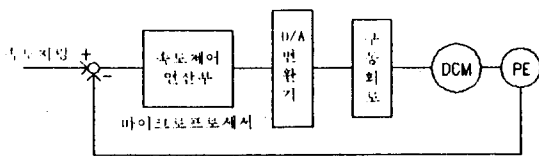


그림 2. 시스템 구성의 개략도

Fig. 2. A schematic diagram of the system

#### 3.2 보상기의 설계

속도제어의 경우 제어대상은 1차로 간주할 수 있으므로,  $W_c(z) = a_0 + a_1 z^{-1}$ 이 된다. 그런데

$$HG(z) = Z[HG] = z^{-1} \frac{1 - e^{-T/T_m}}{1 - z^{-1} e^{-T/T_m}} \quad (5)$$

이 되어 첫 항이  $z^{-1}$ 로 시작되므로 시스템이 물리적으로 실현 가능하기 위해서는  $W_c(z)$ 의 첫 항은  $HG(z)$ 의 첫 항과 차수가 같아야하고 제어입력은 계단입력이므로

$$W_c(z) = z^{-1} \quad (6)$$

이 된다. 식 (6)를 식(2)에 대입하면  $D(z)$ 는

$$D(z) = \frac{1}{HG(z)} \cdot \frac{Z^{-1}}{1 - Z^{-1}} \quad (7)$$

이 되고, 식 (5), 식 (7)로부터

$$D(z) = \frac{1 - z^{-1} e^{-T/T_m}}{1 - e^{-T/T_m}} \frac{1}{1 - z^{-1}} \quad (8)$$

이다. 또,

$$D(z) = \frac{M(z)}{E(z)} \quad (9)$$

로 나타낼 수 있으므로, 식 (8), 식 (9)로부터

$$\frac{M(z)}{E(z)} = \frac{1 - z^{-1} e^{-T/T_m}}{1 - e^{-T/T_m}} \frac{1}{1 - z^{-1}} \quad (10)$$

가 되며, 이를 정리하면 다음과 같다.

$$(1 - e^{-T/T_m})[M(z) - Z^{-1}M(z)] = E(z) - e^{-T/T_m} z^{-1} E(z) \quad (11)$$

식 (11)에 본 시스템의 전체이득  $K$ 를 대입하고 역 z변환하면 다음 식과 같다.

$$m[nT] = \frac{1}{K(1 - e^{-T/T_m})} e[nT] - \frac{e^{-T/T_m}}{K(1 - e^{-T/T_m})} e[(n-1)T] + m[(n-1)T] \quad (n=0, 1, 2) \quad (12)$$

그런데, 식 (12)를 이용하면 1 샘플링 정정이 가능하나, 그 가능한 범위는 한정된다.

지금, 직류전동기를 정지상태로부터 회전시킨다면, 식 (12)에서  $m[-T] = e[-T] = 0$ 이므로, 최초의 출력  $m[0]$ 는

$$m[0] = \frac{1}{K(1 - e^{-T/T_m})} e[0] \quad (13)$$

이고, 본 시스템에서 1 샘플링시간에 정정할 수 있는 속도는  $m[0]$  값이 256을 넘지않는 범위내로 제한된다. 따라서 속도지령치  $n_i$ 가 포화 속도 이상인 경우는 직류전동기에 인가하는 전압이 포화하여 1 샘플링으로 정정할 수 없게

된다. 이 경우는 미리 전압의 포화를 고려해서 1 샘플링시간으로 정정시켜주는 새로운 보상기  $D(z)$ 가 필요하게 된다.

**3.3 전압의 포화를 고려한 보상기  $D(z)$ 의 설계**

전절에서 보인것처럼 속도지령치가 포화속도 이상이면 전압이 포화되므로 전압이 포화상태로 부터 벗어난 후 목표치에 대하여 1 샘플링 정정시킬 수 있는  $D(z)$ 를 설계한다.

그림 3으로 부터, 우선 A점에서 초기전압이 포화상태에서 벗어났다고 하면, 그 때까지는 제한전압이 인가되고 있으며, 이 상태로는 D점에 도달하지만 실제로는 C점에 도달하도록 해야 한다. 현재  $t=iT$  상태에 있다고 하고, E점으로 향하는 속도를 내는 전압  $\Delta M$ 을 A점에서 인가하면

$$(1 - e^{-T/T_m})\Delta M = -\Delta n \tag{14}$$

이 되며, 단,

$$\Delta M = M(z) - z^{-1}M(z) \tag{15}$$

$$\Delta n = \overline{BD} = \overline{AF}e^{-T/T_m} \tag{16}$$

이고, 지금 A점에 있는 상태이므로

$$\overline{AF} = z^{-1}E(z) - E(z) \tag{17}$$

가 된다. 식 (16), (17)로 부터

$$\Delta n = [z^{-1}E(z) - E(z)] e^{-T/T_m} \tag{18}$$

이 된다. 식 (18)을 식 (14)에 대입하여 M을 구하면 속도가 B점으로 향하게 되지만 실제로는 C점으로 향해야 하므로 식 (16)의  $\Delta n$ 으로

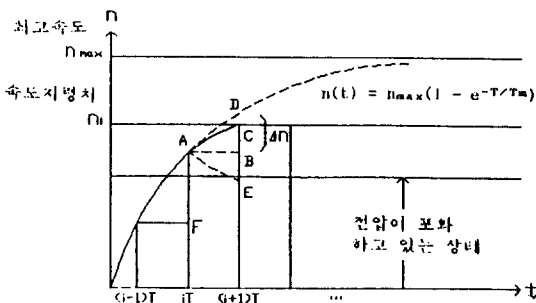


그림 3. 전압 포화상태로 부터 벗어난 후 1 샘플링 시간내에 목표치에 정정시키는 파형  
Fig. 3. Waveform to settle final value in one sampling time after getting free from the saturation state

부터  $BC=E(z)$ 를 빼주면

$$(1 - e^{-T/T_m})[M(z) - z^{-1}M(z)] = E(z) - [z^{-1}E(z) - E(z)]e^{-T/T_m} \tag{19}$$

가 된다. 여기에 전체이득을 고려하고 역 z변환하면 다음과 같다.

$$m[nT] = \frac{1 - e^{-T/T_m}}{K(1 - e^{-T/T_m})} e[nT] - \frac{e^{-T/T_m}}{K(1 - e^{-T/T_m})} e[(n-1)T] + m[(n-1)T] \tag{20}$$

여기서 주의해야 할 것은 식 (20)은  $t=0$ 인 상태, 즉 기동시에는 이용할 수 없다는 사실이다. 왜냐하면, 식 (19)를 구하는 조건이  $z^{-1}E(z)$ 와  $E(z)$ 가 지수함수적으로 변화한다는 조건하에서 구했기 때문이다. 따라서,  $t=0$ 인 상태에 한해서 식 (12)를 이용해야 한다. 이와 같이, 기동시에는 식 (12)를 이용하고, 그 이후에는 식 (20)을 이용하면 프로그램 작성상 복잡하게 되므로 식 (20)을 기동시에도 사용할 수 있도록 변형시키면 다음과 같다. 방법은 식 (18)의  $[z^{-1}E(z) - E(z)]$ 의 항을 실속도  $N(z)$ 로 표시하도록 한다. 즉,

$$\Delta n = [N(z) - z^{-1}N(z)] e^{-T/T_m} \tag{21}$$

이 되고 식 (21)를 식 (19)에 대입하면

$$(1 - e^{-T/T_m})[M(z) - z^{-1}M(z)] = E(z) - [N(z) - z^{-1}N(z)] e^{-T/T_m} \tag{22}$$

가 된다. 또, 식 (20)은

$$m[nT] = \frac{1}{K(1 - e^{-T/T_m})} e[nT] - \frac{e^{-T/T_m}}{K(1 - e^{-T/T_m})} \{no[nT] - no[(n-1)T]\} + m[(n-1)T] \tag{23}$$

이 된다. 식 (23)은 식 (20)과 동가이며,  $t=0$ 인 상태에서도 사용할 수 있다. 이것은  $e$ 가  $t=0$ 에서 계단적으로 변화하는데 비해  $no$ 는  $n=0$ 인 상태에서 지수함수적으로 변화하기 때문이다.

**3.4 예측제어를 적용한 보상기의 설계**

본 논문에서는 속도귀환신호로써 펄스엔코더의 출력펄스열을 일정시간 계수한 것을 사용하기 때문에 그 값은 일정시간 동안의 평균속도가 되고, 속도가 변화하고 있는 상태에서는 제어하고 있는 시점에서의 실제속도와는 다른 일종의 검출지연이 발생하는데, 이의 해결을 위

해 예측제어를 적용한 보상기를 설계한다.

전동기의 실제속도  $n(t)$ 가 그림 4와 같이 변화하고 있다면, 이때 펄스엔코더의 출력펄스열을 일정시간 마다 계수하면 속도검출치  $N$ 은 그림 4와 같다. 즉  $i$  번째의 샘플링시점에서 검출된 속도검출치  $N_i$ 는 양자화오차를 무시하면 다음과 같이 나타낼 수 있다.

$$N_i = \frac{1}{T} \int_{(i-1)T}^{iT} n(t) dt \quad (24)$$

식 (24)에서 나타냈듯이 속도검출치  $N_i$ 는  $(i-1)T$ 에서  $iT$ 까지의 속도의 평균치를 의미하고 있다. 그러므로 그림 4에서와 같이 속도가 변화하고 있는 시점에서는  $iT$ 시의 순시속도  $n_i$ 와 검출된 평균치  $N_i$ 는 다르다. 따라서 검출치  $N_i$ 를 속도귀환신호로 사용하면 속응성이 결여된다. 이점을 개선하기 위하여 여기서는 검출된 평균속도  $N_i$ 와  $N_{i-1}$ 을 사용하여 순시속도  $n_i$ 를 예측하고 그 값을 속도귀환신호로 사용하는 방법을 고찰한다.

2 샘플링시점 사이의 속도  $n(t)$ 가 1차함수로 변화하고 있다고 가정하면

$$n(t) = at + b \quad (25)$$

가 되고,  $(i-1)$ ,  $i$  번째의 샘플링시점에서의 속도검출치  $N_{i-1}$ ,  $N_i$ 는 각각 시간  $\{(i-2)T + (i-1)T\}/2$ ,  $\{(i-1)T + iT\}/2$ 에서의 순시속도와 같으므로 식 (25)에 의해 다음과 같다.

$$\begin{aligned} N_{i-1} &= a\{(i-2)T + (i-1)T\}/2 + b \\ &= a(2i-3)T/2 + b \end{aligned} \quad (26)$$

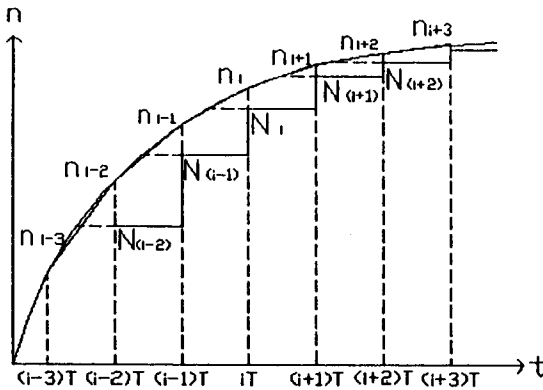


그림 4. 순시속도 예측법

Fig. 4. Estimation of current speed

$$\begin{aligned} N_i &= a\{(i-1)T + iT\}/2 + b \\ &= a(2i-1)T/2 + b \end{aligned} \quad (27)$$

여기서  $iT$  시점에서의 순시속도  $n_i$ 는 식 (25)에 의해 다음과 같이 된다.

$$n_i = a \cdot iT + b \quad (28)$$

식 (26), (27), (28)으로 부터  $a$ ,  $b$ 를 소거하면 1차식 근사를 이용한 순시속도의 예측치  $n_i^*$ 는 다음과 같이 된다.

$$n_i^* = \frac{3N_i - N_{i-1}}{2} \quad (29)$$

#### 4. 실험 결과 및 검토

실험에 사용된 시스템의 상수와 전체이득  $K$ 는 표 1과 같다.

실험은 컴퓨터 모의실험과 실제실험 두 가지를 행하였고, 비교를 위하여 컴퓨터 모의실험 파형과 실제실험 파형을 병행해서 보이도록 한다. 각 그림에서 상부는 전압파형으로 D/A 변환기의 출력파형이고, 하부는 속도파형으로 펄스엔코더의 출력을  $f/v$  변환한 것이다.

그림 5와 그림 6은 속도지령치가 103[rpm]인 때 응답이다. 이와 같이 속도지령치가 제한속도 166[rpm] 이하인 경우는 식 (12)에 의해 1 샘플링시간으로 정정할 수 있다. 그림 7과 그림 8은 속도지령치가 550[rpm]인 경우이며, 제대로 유한시간 정정응답이 되어있지 않음을 볼 수 있다(1 샘플링시간보다도 훨씬 뒤진 정정시간이 되어 있다). 이것은 전압이 포화하고

표 1. 시스템의 상수 및 전체이득

Table 1. Constant and total gain

샘플링시간	T	2.9 msec
기계적 시정수	$T_m$	20.9 msec
전기자 저항	$R_a$	13.3 ohm
관성	J	0.29 G · cm · sec <sup>2</sup>
역기전력 상수	$K_E$	13.8 V/kRPM
토크 상수	$K_T$	1.4 kg · cm
D/A 변환기 이득	A	10/256
구동회로 이득	B	1.75
펄스엔코더 이득	C	96.7/kRPM
전체이득	K	$A * B * C * 1/K_E = 0.4790$

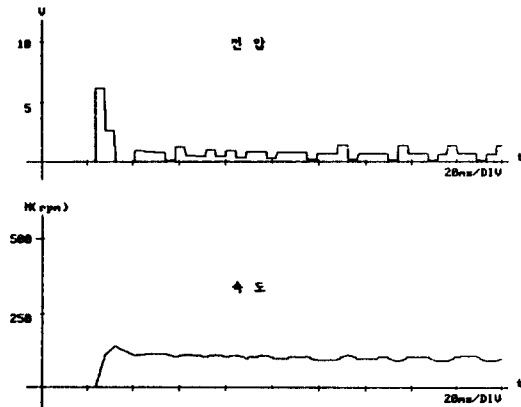


그림 5. 식 (12)를 이용한 경우의 컴퓨터 모의실험 (전압이 포화하지 않은 경우)  
 Fig. 5. Computer simulation for using equation (12) (For unsaturated reference input)

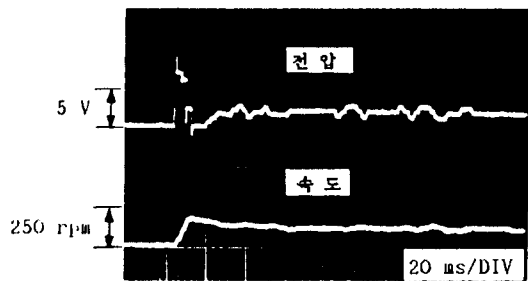


그림 6. 식 (12)를 이용한 경우의 속도응답 (전압이 포화하지 않은 경우)  
 Fig. 6. Speed response for using equation (12) (For unsaturated reference input)

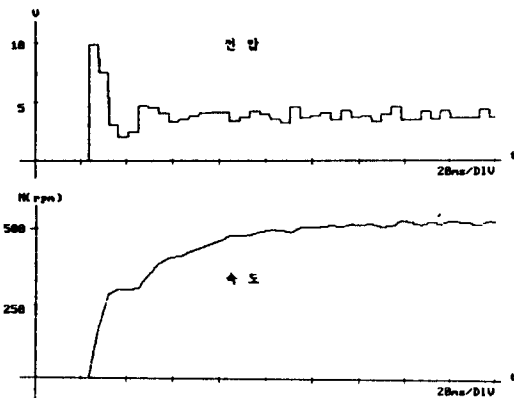


그림 7. 식 (12)를 이용한 경우의 컴퓨터 모의실험 (전압이 포화하고 있는 경우)  
 Fig. 7. Computer simulation for using equation (12) (For saturated reference input)

포화를 고려한 직류전동기의 유한시간 점정 응답제어 있기 때문이다. 따라서, 포화를 고려한 보상기가 설계되어야 한다. 그림 9와 그림 10은 포화

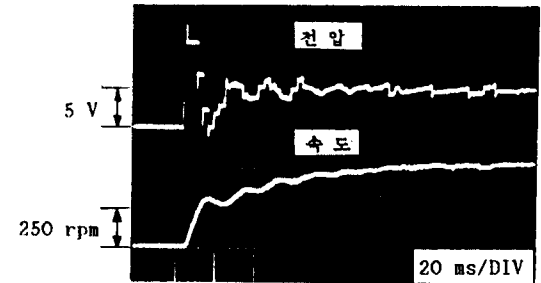


그림 8. 식 (12)를 이용한 경우의 속도응답 (전압이 포화하고 있는 경우)  
 Fig. 8. Speed response for using equation (12) (For saturated reference input)

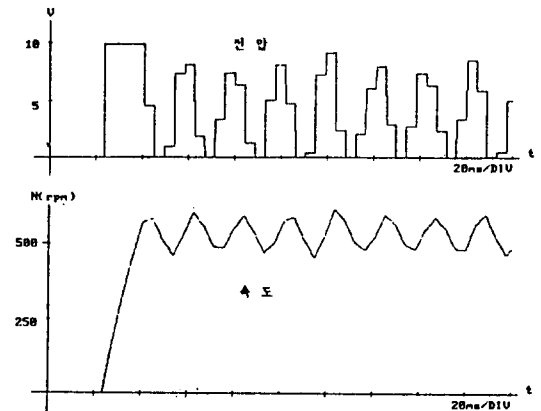


그림 9. 식 (23)를 이용한 경우의 컴퓨터 모의실험 (전압이 포화하고 있는 경우)  
 Fig. 9. Computer simulation for using equation (23) (For saturated reference input)

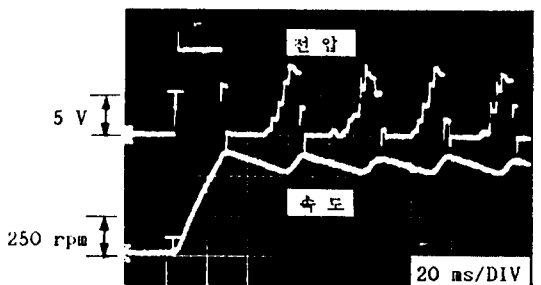


그림 10. 식 (23)을 이용한 경우의 속도응답 (전압이 포화하고 있는 경우)  
 Fig. 10. Speed response for using equation (23) (For saturated reference input)

를 고려한 식 (23)을 이용한 경우의 응답이다. 이것은 그림 8에 비해 속응성은 비교적 양호하지만 정상시에 진동이 발생하고 있다. 그 이유는 식 (12)보다 식 (23)의 이득이 비교적 크기 때문이다. 이것을 개선하려면 전체이득을 작게 해야하나 그렇게 되면 속응성이 나쁘게 된다. 그림 10에서 본 것처럼 식 (23)을 이용하면 정상시에 진동이 생기므로, 전압이 포화되어 있는 상태에서 부터 벗어날 때까지의 과도

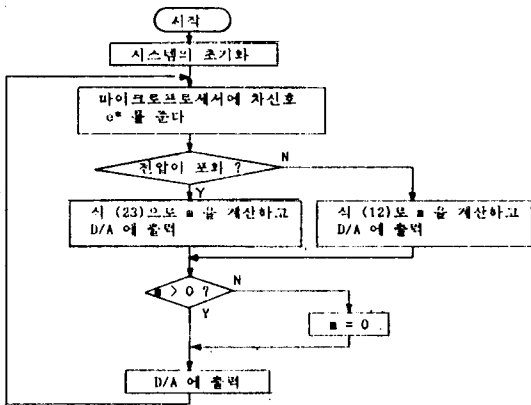


그림 11. 과도시에는 식(23)을 이용하고 정상시에는 식 (12)를 이용한 알고리즘

Fig. 11. Algorithm using equation (23) on the transient state and equation (12) on the steady state

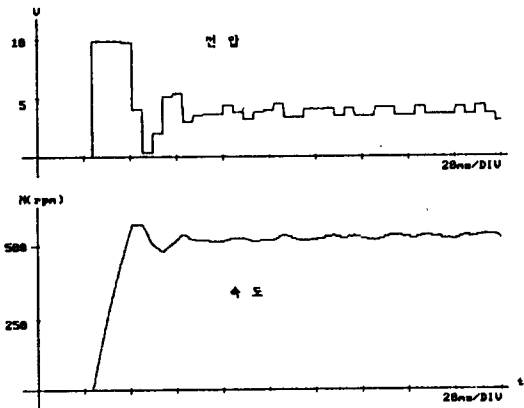


그림 12. 그림 11의 알고리즘을 이용한 컴퓨터 모의실험

Fig. 12. Computer simulation for using the algorithm of fig. 11

시에는 식 (23)을 사용하고, 그 이후 정상시에는 식 (12)를 이용하도록 한다. 그림 11은 이 경우의 알고리즘이며, 그림 12와 그림 13은 실험 결과이다. 그림 13은 그림 8에 비해 속응성이,

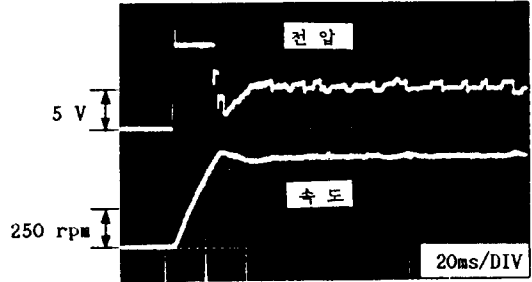


그림 13. 그림 11의 알고리즘을 이용한 경우의 속도응답

Fig. 13. Response for using the algorithm of fig. 11

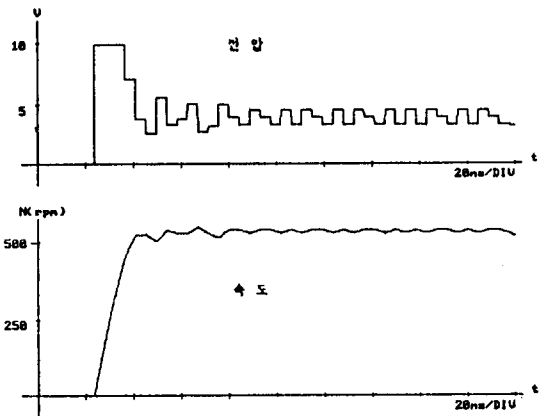


그림 14. 예측제어의 컴퓨터 모의실험

Fig. 14. Computer simulation for the feedforward control

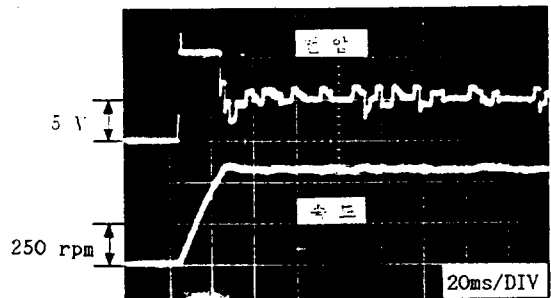


그림 15. 예측제어에 의한 속도응답

Fig. 15. Speed response for using feedforward control

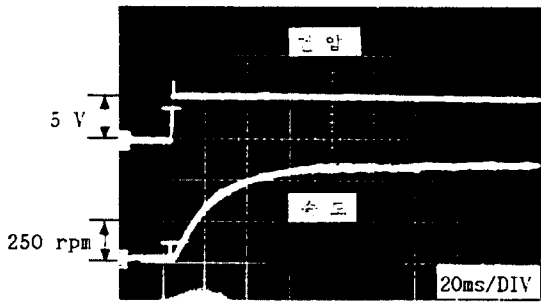


그림 16. 제어되지 않은 경우의 응답

Fig. 16. Speed response for uncontrolled step input

그림 10에 비해 진동이 개선되었음을 알 수 있다. 그림 14와 그림 15는 예측제어법을 적용하여 지령치 550[rpm]으로 실험한 경우의 응답이다. 그림에서 보면 알 수 있듯이 과도상태에서의 오버슈트(overshoot)가 양호하게 제어되어 고속으로 정정하고 있다. 정상상태에서 전압의 맥동이 보이고 있는데 이것은 앞에서 보인 그림들에서의 정상상태의 전압맥동과 마찬가지로 마이크로프로세서가 연산에 소모하는 시간에서 생기는 오차, 속도귀환에서 생기는 오차, 소수점 부분을 무시한 양자화오차 등에 의해서 생기는 것이다. 이러한 오차는 고속 대용량 마이크로프로세서의 채용과 속도귀환신호의 정밀도를 높이기 때문에 해결할 수 있다.

이상의 실험결과와 비교를 위하여 참고로 제어하지 않은 상태에서의 응답을 보인다. 그림 16은 정상상태속도 550[rpm]에 해당하는 제어하지 않은 일정전압을 계단적으로 가했을 때의 파형으로 그림 15는 그림 16에 비하여 응답속도가 매우 향상되었음을 알 수 있다.

## 5. 결론

이상으로 유한시간 정정응답 제어이론을 이용하여 직류전동기의 속도를 최소한의 시간내에 목표치에 정정시키는 제어 알고리즘을 개발했으며, 이 알고리즘을 적용한 컴퓨터 모의실험과 실제실험을 행함으로써 고속 응답특성을 얻을 수 있음을 확인하였다.

본 논문에서 얻은 결과로는,

1) 전압이 포화되지 않을 때는 1 샘플링시간으로 목표치에 정정할 수 있었으며,

포화를 고려한 직류전동기의 유한시간 정정 응답제어

2) 전압 포화시에는 미리 전압의 포화를 고려한 보상기를 적용함으로써, 유한회의 샘플링에 의해 목표치에 정정시킬 수 있음을 확인할 수 있었고,

3) 순시속도 예측법을 적용함으로써 검출 지연에 의해 발생하는 문제점을 해결할 수 있다.

금후의 과제로는 이 이론을 적용하여 속도성이 우수하고 정밀도가 높은 위치제어 시스템의 개발 등이 있다.

## 참고 문헌

- 1) THADIAPPAN KRISHNAN and BELLAMKONDA RAMASWAMI, "A Fast-Response DC Motor Speed Control System", IEEE Trans. Ind. Appl., Vol. IA-10, No. 5, pp634-651, Sep./Oct. 1974.
- 2) NICHOLAS J. KRIKELIS, SPILOS D. FASSOIS, "Microprocessor", Implementation of PID Controller and Lead-lag Compensators", IEEE Trans. Ind. Electron., Vol. IE-31, pp79-85, February 1984.
- 3) PRADEEP K. NANDAM, PARESH C. SEN, "Analog and Digital Speed Control of DC Drivers using Proportional-Integral and Integral-Proportional Control Techniques", IEEE Trans. Ind. Electron., Vol. IE-34, pp227-233, May 1987.
- 4) JIANG JING-PING, RICHARD S. MARLEAU, "Digitally Controlled DC Drive Motors", IEEE Trans. Ind. Appl., Vol. IA-18, No. 6, pp728-735, Nov. / Dec. 1982.
- 5) J. M. Ho, "The Study of Predictive Observer in a Motor Drive System", IECON'87, pp189-196, 1987.
- 6) YOSHIHARU HARADA, HIROSHI MURATA, SETUO SAGARA, "有限整正應答装置によるフィードフォワード制御とその直流電動機速度制御系への適用", 日本電氣學會論文誌 B, Vol. 106, No. 2, pp174-180, 1986.
- 7) FUMIO HARASHIMA, SELJI KONDO, "Robust State Observer and Variable Sampling Technique for Performance Improvement of a Motor Drive System", IEEE Trans. Ind. Electron., Vol. IE-30, No. 4, pp325-329, Nov. 1983.
- 8) Katsuhiko Ogata, "Discrete-Time Control System" Prentice-Hall, INC., 1987
- 9) H. SERAJI, "Deadbeat Control of Discrete-time System using Output Feedback", INT. J. CONTROL, Vol 21, No. 2, pp213-223, 1975.