

Multiprocessor를 이용한 AC Servo Motor의 制御를 위한 Software Algorithm에 관한 研究

金斗京, 金一煥*

A Study on Software Algorithm for AC Servo Motor Control Using Multiprocessor

*Kim Doo-gyung, Kim Eel-hwan**

Summary

A number of motor driving systems based on the microprocessor have been developed.

A digital control system using a AC Servo Motor fed by PWM Sinusoidal inverter and the result of PWM generation, velocity control using a 8086 16 bit microprocessor are presented. The two pole proportional type control algorithm is implemented in this paper, which can compensated effectively the error between the reference and actual output by Rotary Encoder.

The experimental results show good performance of high speed with 2ms of sampling time.

Moreover enhancement of its performance is achieved by the algorithm containing constant speed, acceleration-deceleration, speed regulation, using interrupt, and Servo-Lock.

序 論

最近 microprocessor의 高性能化됨에 따라 그 應用技術이 광범위하게 使用되고 있다. 그 중에서도 mi-
理工大學 專任講師, 中央大學校 大學院 博士課程*

croprocessor에 의해 制御를 Digital化 함으로써 Analog로 할 때와 같은 주변환경의 영향에 의한 雜音에 약하다는 것과 측정기의 신호전달에서 오는 2차 및 非線形성에 문제가 많고 또한 Driving System이 複雜하다는 등 여러가지 문제점을 없애고 또한 각종 제

어 시스템을 集中관리하는 全 Digital System을 構成할 수 있다는 見解가 있다.

그러나 hardware의 으로는 많은 部分에서 Digital 化가 이루어져 있지만 이에 相當하는 Software Algorithm에 對해서는 만족할 만한 結果를 내지 못한 경우가 많다. 따라서 hardware의 Digital 化에 따르는 Software의 開發이 要求되고 있으나 이에 對한 技術 開發이 未洽한 實情이다.

本 研究에서는 이러한 實情을 감안하여 16 bit microprocessor를 使用하여 Motor의 구동을 위한 全 Digital System을 構成함으로써 精밀속도제어 기능

이 필요한 Software Algorithm을 설계하여 적용함으로써 multiprocessor에 의한 高精度의 Digital Servo Motor Driving System을 實現하고자 한다.

研究方法

1. System 構成

Fig. 1은 本 研究에서 全 System의 構成圖이다.

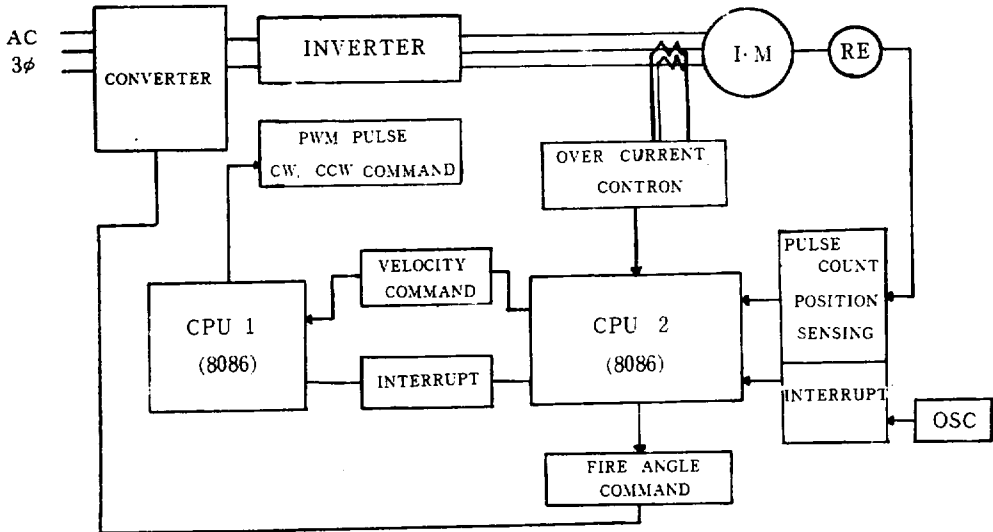


Fig.1. Configuration of driving system.

2. PWM pulse generation Algorithm

1) Inverter 入力 data와 出力 frequency와의 관계

Inverter 入力 data를 0~FFFF (16 bit data) 까지 變化시킬때 Inverter의 出力周波數와 出力電壓과는 比例關係가 되도록 해야 한다. 여기서 Inverter 入力 data를 μ 라 하고 CPU Clock frequency를 f_c , PWM 信號發生 Program에서 1cycle을 만드는데 소

요되는 CPU clock의 週期 T 狀態의 數를 Tw라 하면 Inverter 出力의 基本波의 周波數는 다음식 (1)로 주어진다.

$$f = \frac{1}{1 \text{ cycle} \cdot \text{수행시간}} = \frac{1}{T(\mu) \cdot \frac{1}{f_c}} = \frac{f_c}{T(\mu)} \text{ [Hz]} \quad (1)$$

여기서 Inverter 出力周波數의 最大값을 60[Hz] 最小값을 0[Hz]라 하고 f_c 를 5[MHz]라 하면 PWM 信號發生 Program의 1cycle 實行에 소요되는 CPU clock의 週期 T 狀態의 數 $T(\mu)$ 는 다음의 제약을 받는다. 즉 $T(\mu)$ 개의 Clock 週期내에 PWM 信號發

生 Program을 1회 수행하도록 作成해야 한다. 또 PWM 制御 Inverter 出力 電流波형을 正弦波로 만들기 위해서는 PWM 制御인 버터 出力 電壓波형을 Fig. 2와 같은 펄스를 CPU-1의 8255 B port에서 만들어 주어야 한다. 그리고 R相-S相 사이의 線間電壓은 (2) 식과 같이 正弦波로 되기 위해서는 R, S, T相의 相電壓은 式 (3), (4), (5) 와 같이 餘弦波로 되어야 한다.

$$V_{RS}=V_R-V_S=-\sqrt{3} V_m \sin(\omega t - \frac{\pi}{3}) \dots\dots\dots (2)$$

$$V_R=V_{m_s} \cos \omega t \dots\dots\dots (3)$$

$$V_S=V_{m_s} \cos(\omega t - \frac{2}{3} \pi) \dots\dots\dots (4)$$

$$V_T=V_{m_s} \cos(\omega t - \frac{4}{3} \pi) \dots\dots\dots (5)$$

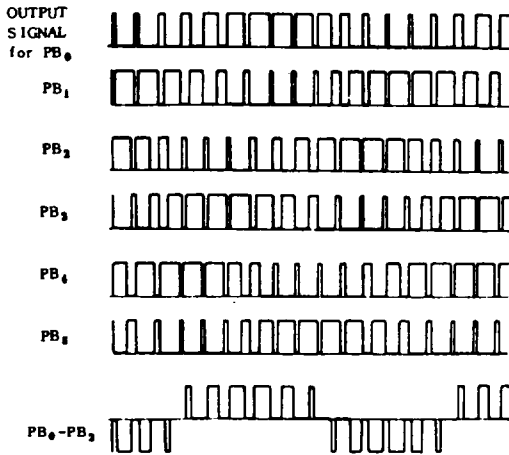


Fig.2. Waveform for sinusoidal PWM fulse at Bport of 8086 CPU-1

2) PWM 信號幅 算出

Fig. 2에서 PB₀, PB₂, PB₄의 각 PWM 波형을 基本波와 直流成分으로 나누어 나타낸 式은 다음 (6), (7), (8) 式과 같이 된다.

$$V_R=V_{m_s} \cos \omega t + (V_{m_s} + B) \dots\dots\dots (6)$$

$$V_S=V_{m_s} \cos(\omega t - \frac{2}{3} \pi) + (V_{m_s} + B) \dots\dots\dots (7)$$

$$V_T=V_{m_s} \cos(\omega t - \frac{4}{3} \pi) + (V_{m_s} + B) \dots\dots\dots (8)$$

윗 式 (6)~(8) 은 Fig. 3과 같이 나타내진다.

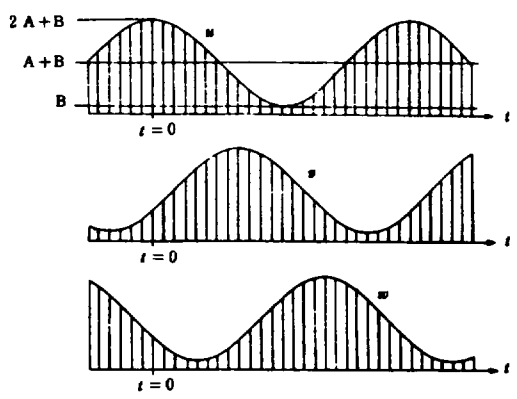


Fig.3. Synthesis of phase sinusoidal wave form and dc component.

따라서 Fig. 3의 電壓진폭 變化를 Fig. 2의 Pulse 폭(時間定數)와 같이 等價의으로 바꾸어 주어야 한다. 이를 위하여 式 (4), (5), (6)를 이용하여 pulse 폭의 時間定數로 計算한 實行結果는 Table 1과 같다. Table 1의 結果값을 分析하면,

- (1) $\omega t = 0^\circ \sim 360^\circ$ 사이를 12°씩 분할하여 각각을 區間이라 하고 각 區間의 中心位置에서 計算하여 R, S, T값을 구한다.
- (2) 全體를 6 MODE로 나누고 MODE 1에서의 크기는 $R > S > T$ 順으로 한다.
- (3) R, S, T 각각의 1 pulse 주기 동안에 4번의 서로 다른 Inverter 出力 Transistor의 Switching이 발생하며 이것을 Switching 狀態라 한다. PB₀~PB₄까지의 bit를 Switching 狀態 data와 같이 出力한 후 그 狀態대로 일정시간 (t_1, t_2, t_3, t_4) 유지시킨 후 다른 스위칭 狀態 (s_1, s_2, s_3, s_4)로 바뀌게 한다. 이 때 ($t_1 + t_2 + t_3 + t_4$) = 118로 바뀌게 한다. 이 때 이 값의 計算 근거는 4) 절에서 論한다.
- (4) 이상에서 서술한 t_i 과 s_1, t_2 와 $s_2 \dots t_4$ 와 s_4 등을 1개조로 하여 CPUL. memory의 $\phi 2\phi 1$ 번지에서 $\phi 2F\phi H$ 사이에 기억시켜 둔다.

3) PWM pulse generation Program.

PWM pulse generation은 다음과 같은 조건하에서 이루어진다는 가정하에 Program을 作成한다.

Table 1. The time constraint and switching states of PWM inverter.

Mode	Section	Phase	Calculated data			Time data of 1 pulse period				R.S.T Switching Mode
			u	v	w	t ₁	t ₂	t ₃	t ₄	
〈1〉	1	6	117	35	25	25	11	81	1	R > S > T
	2	18	114	47	16	16	31	67	4	t ₁ :00010101= 15H
	3	30	109	59	9	9	50	50	9	t ₂ :00100101= 25H
	4	42	102	71	4	4	67	31	16	t ₃ :00101001= 29H
	5	54	93	83	1	1	81	11	25	t ₄ :00101010= 2AH
〈2〉	6	66	83	93	1	1	81	11	25	S > R > T
	7	78	71	102	4	4	67	31	16	t ₁ :00010101= 15H
	8	90	59	109	9	9	50	50	9	t ₂ :00100101= 25H
	9	102	47	114	16	16	31	67	4	t ₃ :00100110= 26H
	10	114	35	117	25	25	11	81	1	t ₄ :00101010= 2AH
〈3〉	11	126	25	117	35	25	11	81	1	S > T > R
	12	138	16	114	47	16	31	67	1	t ₁ :00010101= 15H
	13	150	9	109	59	9	50	50	4	t ₂ :00010110= 16H
	14	162	4	102	71	4	67	31	9	t ₃ :00100110= 26H
	15	174	1	93	83	1	81	11	16	t ₄ :00101010= 2AH
〈4〉	16	186	1	83	93	1	81	11	25	T > S > R
	17	198	4	71	102	4	67	31	25	t ₁ :00010101= 15H
	18	210	9	59	109	9	50	50	9	t ₂ :00010110= 16H
	19	222	16	47	114	16	31	67	4	t ₃ :00011010= 1AH
	20	234	25	35	117	25	11	81	1	t ₄ :00101010= 2AH
〈5〉	21	246	35	25	117	25	11	81	1	T > R > S
	22	258	47	16	114	16	31	67	4	t ₁ :00010101= 15H
	23	270	59	9	109	9	50	50	9	t ₂ :00011001= 19H
	24	282	71	4	102	4	67	31	16	t ₃ :00011010= 1AH
	25	294	83	1	93	1	81	11	25	t ₄ :00101010= 2AH
〈6〉	26	306	93	1	83	1	81	11	25	R > T > S
	27	318	102	4	71	4	67	31	16	t ₁ :00010101= 15H
	28	330	109	9	59	9	50	50	9	t ₂ :00011001= 19H
	29	342	114	16	47	16	31	67	4	t ₃ :00101001= 29H
	30	354	117	25	35	25	11	81	1	t ₄ :00101010= 2AH

- (1) 1주기당 30개의 pulse를 出力한다.
- (2) 주파수가 전압의 실효치 v는 비례관계에 있다. 즉 v/f 비를 일정하게 유지할 수 있도록 한다.
- (3) 주파수 f는 入力 Port O[Hz]~60[Hz]의 범위까지 지령하는 것으로 한다.
- (4) Inverter의 TR₀와 TR₁, TR₂와 TR₃, TR₄와 TR₅는 同時의 動作하는 구간이 있으면 回路의 단락을 유발시켜 TR을 손상시킬 위험이 있으므로 어떤 경우에도 同時에 동작하지 않도록 스위칭 상태를 出力시키는 Program을 作成한다. 이상의 조건을 만족시키는 3相 正弦波 PWM 信號를 發生시키기 위하여 1週期를 30개의 pulse로 세분하고 1週期狀*를 Fig.

4와 같이 4개의 상태로 나누고 각 상태가 바뀌는 境界에서 出力 Power Tr의 同時動作을 피하기 위하여 數(μs)의 짧은 不動時間(dead-time)을 준다. 또 각각의 4개 狀態를 그대로 유지시키는 時間 data (PWM) 信號 pulse폭의 時間定數가 주어져야 한다. 따라서 3相 正弦波 PWM 信號를 만들기 위해서는 각 狀態의 狀態 data와 서로 다른 時間 data 4개와 필요하며 총계 (4+4) × = 240 項目의 data를 필요로 한다. 이 data를 Look-up Table로 해서 CPUL의 0200 H~φ2FφH번지 사이의 memory에 기억 시켜두고 Program에 따라 순차적으로 入출하여 使用한다. 이상의 정보를 기초로 하여 入버터 制御 Program을 作

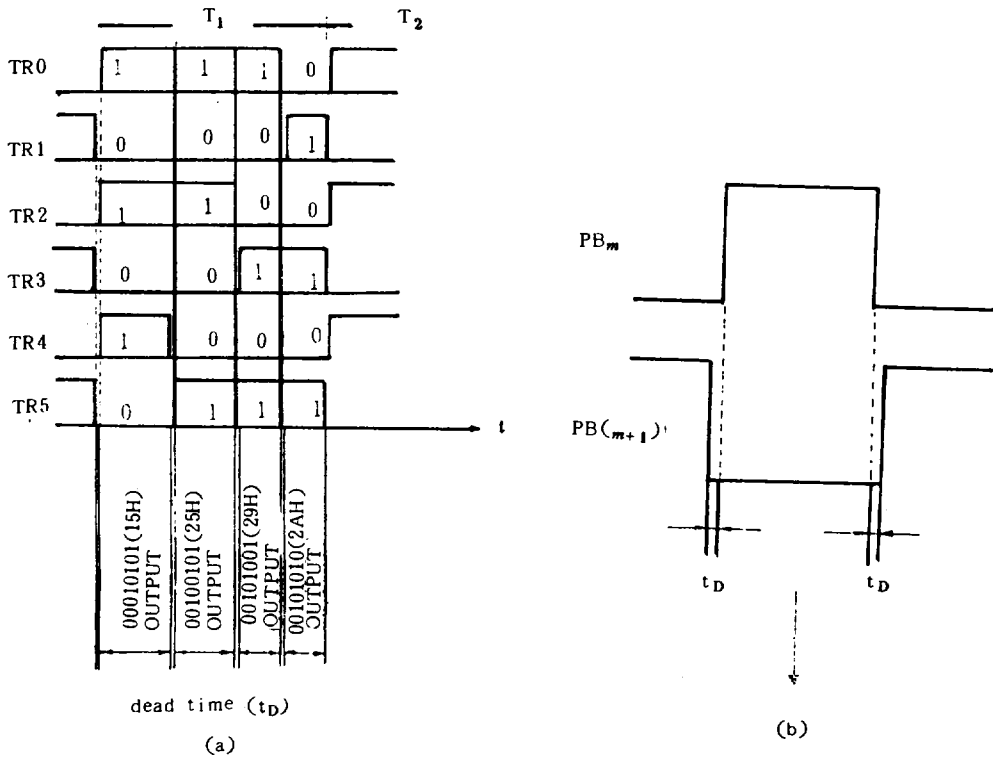


Fig.4. The pattern of inverter voltage waveforms.

(a) The inverter voltage waveform for one pulse

(b) The deadtime of inverter voltage waveforms

成하면 다음과 같이 나타낼 수 있다. 즉, 인버터 제어 프로그램의 전압파형 모형도는 Fig. 4 flowchart는 Fig. 5와 같다.

따라서 정현파 PWM인버터 제어프로그램은 Table 2와 같다.

4) CPU 出力 周期 T狀態의 數

式 (6) ~ (8) 에 있는 Vms값과 B값을 구하는 방법은 다음과 같이 한다. T狀態의 數를 구하기 위해서는 기계어의 각 명령을 처리하는 데 필요한 T 狀態의 數를 조사해야 한다. 프로그램에서 Loop 4를 4회 수행하는 T狀態의 數는 다음과 같다.

$$1구간의 T狀態의 數 = 19(h + n) + 522 \quad (9)$$

여기서 h는 4개의 CPU1의 8255 port A에서 주파수 지령으로 읽어들이는 0~FFFF data이다. 또 Loop 4를 30회 반복하여야 인버터 出力이 1주기를 제어한 셈이 되므로 식(9)에 30을 곱하여 1 cycle당 전 T狀態의 數 T(u)를 다음과 같이 나타낸다. 즉,

$$T(u) = 30\{19(h + n) + 522\} \quad (10)$$

CPU를 구동하는 Clock 주파수를 f_c 라 하면 1 T狀態 實行에 요하는 시간은 $\frac{1}{f_c}$ 로 구해지고 正弦波 1 cycle를 발생시키는데 소요되는 시간은 다음 式으로 주어진다.

$$1 \text{ cycle 수행시간} = 30\{19(h + n) + 522\} / f_c \quad (11)$$

여기서 주파수 지령 port를 I(000000000000000001B)

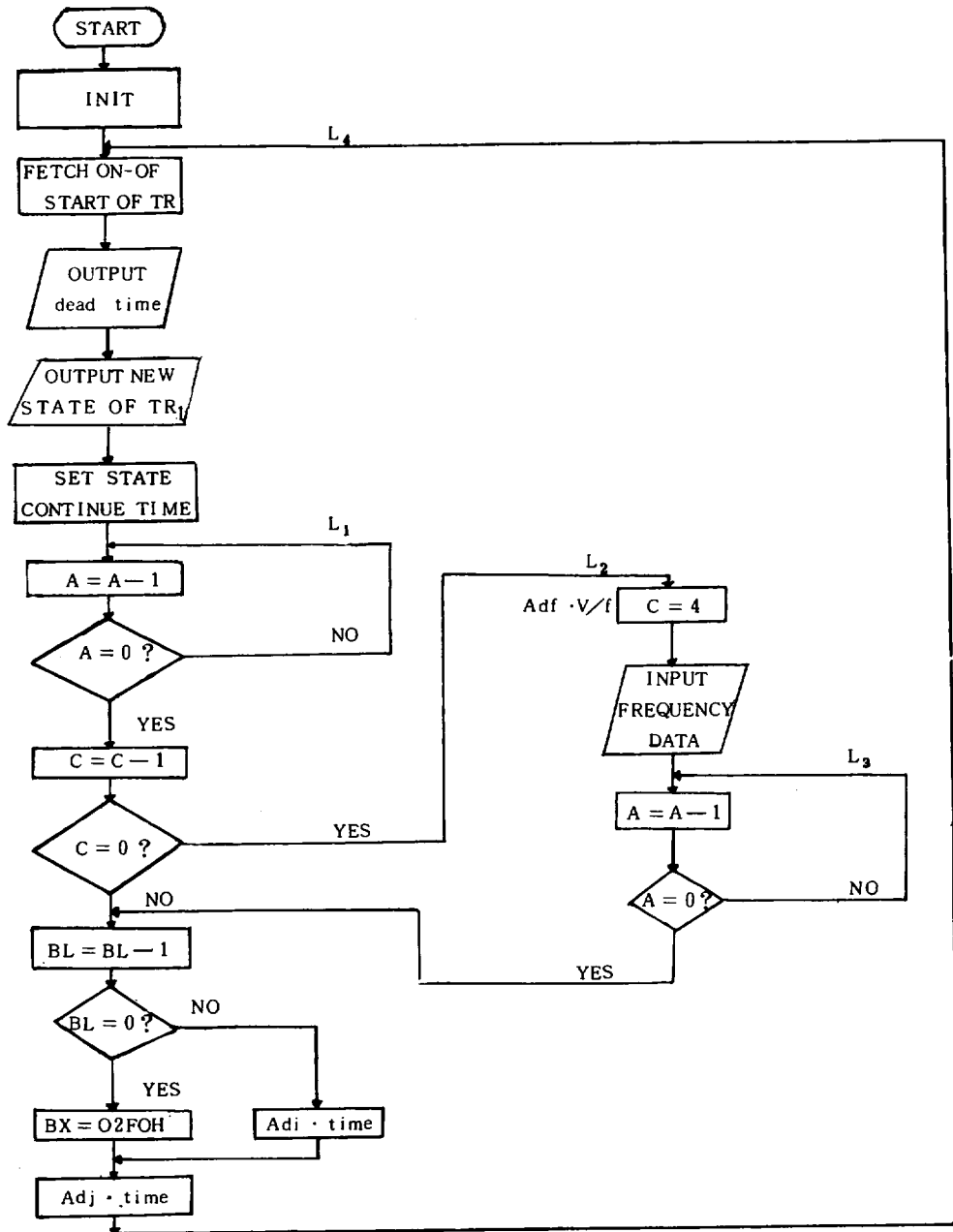


Fig.5. Frowchart for sinusoidal PWM control program.

Table 2. The control program for sinusoidal PWM inverter.

```

;SINUSODIAL PWM WAVEFORM

                ORG     100H

FRQCY EQU     18H
DRIVE EQU     1AH
CONT EQU     1EH

;
START:

0100 B0 90      MOV     AL,90H
0104 E6 1E      OUT     CONT,AL
0104 B2 00      MOV     DL,00H
0106 B1 04      MOV     CL,04H
0108 BB F0 02   MOV     BX,2F0H

                LOOP1:
0108 8A C2      MOV     AL,DL
010D 8A 17      MOV     DL,(BX)
010F 22 C2      AND     AL,DL
0111 E6 1A      OUT     DRIVE,AL
0113 8A C2      MOV     AL,DL
0115 E6 1A      OUT     DRIVE,AL
0117 FE CB      DEC     BL
0119 8A 07      MOV     AL,(BX)

                LOOP2:
011B FE CB      DEC     AL
011D 75 FC      JNZ    LOOP2
011F FE C9      DEC     CL
0121 75 08      JNZ    NEW

                LOOP3:
0123 B1 04      MOV     CL,04H
0125 E4 18      IN     FRQCY

                LOOP4:
0127 FE CB      DEC     AL
0129 75 FC      JNZ    LOOP4

NEW:
012B FE CB      DEC     BL
012D 74 03      JZ     RESET
012F EB 04 90   JMP     RETURN

RESET:
0132 BB F0 02   MOV     BX,2F0H

RETURN:
0135 EB D4      JMP    LOOP 1
    
```

로 하고 8086 CPU1 Clock으로, f_c 는 5MHz로 한다. 따라서 h값은 다음과 같이 한다.

$$\begin{aligned}
 h &= \frac{1}{19} \left\{ \frac{f_c}{60} - 19n - 522 \right\} \\
 &= \frac{1}{19} - \left\{ \frac{5000000}{60} - 19 - 522 \right\} = 118 \quad (12)
 \end{aligned}$$

즉, h는 $t_1 + t_2 + t_3 + t_4$ 의 합이므로 펄스폭에서 ON, OFF쪽의 최소단위를 1로 하면 $B=1$ 이 되고 그림 2)로부터 $2V_{ms} + B'$ 가 정현파 1주기 동안의 최대값이므로 $2V_{ms} + B = (118-1)$ 로 되어야 하며 $B=1$ 로 한 것은 $t_1 \sim t_4$ 의 최소단위가 1이기 때문이며 따라서 $V_{ms} = 58$ 로 된다. 이 경우 인버터 주파수 f는 다음 식으로 주어진다.

$$f = 5 \times 10^{-4} / \{30\{19(118+n) + 522\}\} \quad (13)$$

여기서 n은 CPU2의 8255 port에서 16 bit의 data값이다.

實驗方法

1. 速度調節 Program

이 프로그램은 전동기가 설정된 속도로 일정하게 회전하도록 속도를 조절하는 Program으로 Interrupt에 의해서 일정시간마다 수행하도록 한다. 정상 상태에서 전동기의 원하는 속도가 x이고 Sampling 주기가 T[s], 1회전당 Encoder 出力 펄스 K.이라 하면 설정된 속도를 유지하기 위해 일정 Sampling 시간 동안 출력되는 Counter값 즉 Encoder 出力 pulse의 증가(감소) 값은 다음 식으로 나타낸다.

$$N = \frac{K_n X}{60} \text{ (counts)} \dots\dots\dots (14)$$

本 연구에서는 高速 回轉時에도 처리가 가능하도록 하기 위하여 Interrupt 주기, 즉, Sampling 주기 T를 2[ms]로 하였고, Encoder 出力 pulse K.은 3,000 pulse/Rev이다. 예를 들어 900 rpm으로 회전하여야 할 때 Sampling 주기마다 출력되어야 할 pulse數는

$$N = \frac{3000 \times 900 \times 0.002}{60} = 180[\text{counts}] \dots (15)$$

이 된다. 따라서 Interrupt에 의한 속도 조절, Progra

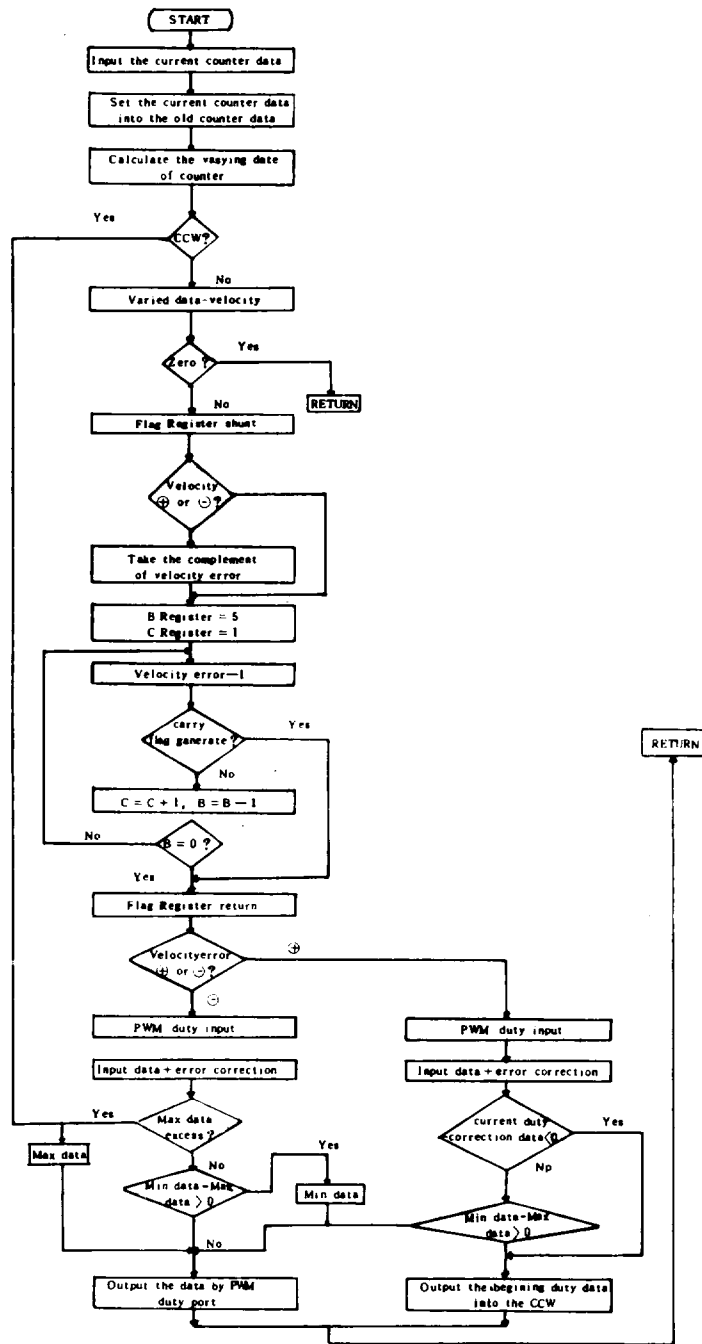


Fig.6 Flowchart of speed regulation program.

m에서는 $N = N(t) - N(t - T)$ 의 값이 180이 되도록 하면 900rpm의 속도가 유지된다. 설정 속도조절 Program의 flowchart는 Fig. 6과 같다.

2. 정속 구동 Program

전동기를 가감속 없이 一定한 속도로 구동하는 데에는 방향과 속도만 설정하면 속도유지는 Interrupt Program에서 수행하게 된다. 정속구동 Program의 flowchart는 Fig. 7과 같다.

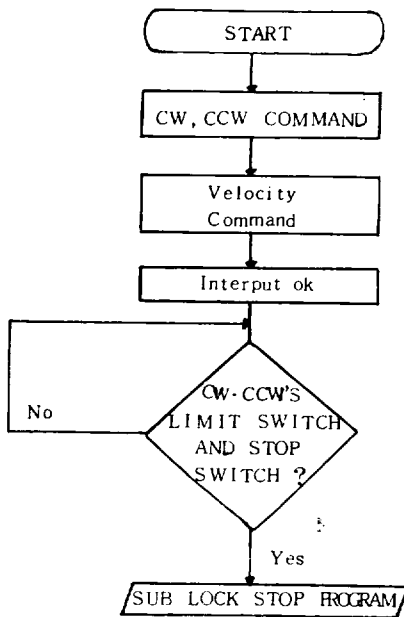


Fig.7. Fiowchart of constant speed drive.

結果 및 考察

2장에서와 같이 System을 構成하고 제어 Program을 使用해서 全 System을 동작시켰다. Fig. 8과 Fig. 9는 1주기당 PWM 파형을 10개와 30개로 했을 때의 전류와 전압 파형이다.

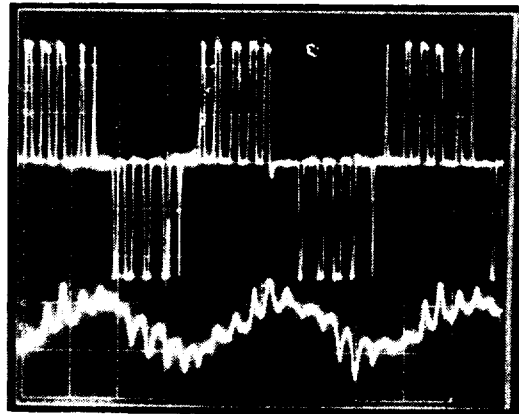


Fig.8. Inverter output waveform.
(U) Voltage waveform
(D) Curent waveform

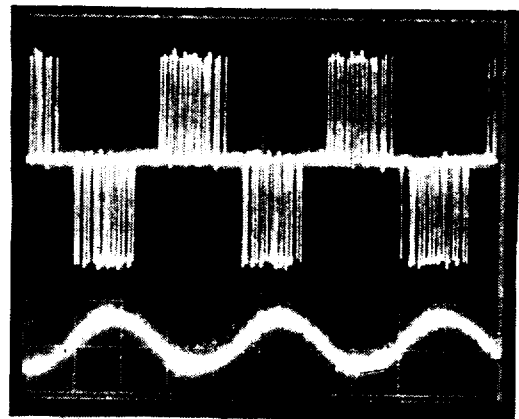


Fig.9. Inverter output waveform.
(U) Voltage waveform
(D) Current waveform

摘 要

本 研究에서는 multiprocessor에 의한 Digital化 AC Servo Driving System을 위해 속도제어를 중심으로 간략화된 구동 hardware 및 Software Algorithm을 構成하였다.

本 연구에서는 다음과 같은 結論을 얻었다.

1. Tachogenerator, A/D Converter, D/A Converter 등의 Analog 소자를 쓰지 않고 全 部分을 Digital 회로로 구성하고 속도 검출기로 2相의 Rotary Encoder만을 사용하여 방향 속도 위치 등의 검출에서 우수한 특성을 얻었다.

2. hardware의 간략화로 제어기의 크기를 줄일 수 있고 신뢰성이 높으며 탄력적인 적용을 할 수 있게 되었다.

3. Software에 의한 Digital Control 방식이기 때

문에 수치적으로 입력된 속도에 대해 정확한 속도로 회전한다.

4. 인버터 제어용 PWM pulse를 Software적으로 하므로써 hardware 부분을 줄일 수 있었다.

따라서 本 研究를 통해 기존의 Analog 제어방식에 서는 볼 수 없는 우수한 특징을 많이 얻을 수 있었고 이러한 특징들을 이용하여 고성능 System을 具現하는 데 도움이 될 것으로 期待된다.

引用 文 獻

Divan D.M, G.Hancock, Hope G.S and Barton T.H, 1984, "Microprogrammable sequential controller", Proc. IEE, Vol 1131, Pt. E, No 6, pp. 201-208.

Lim A.K and Koespel Wiw, 1977, "A microprocessor speed control system." IEEE Trans, ELEC, Contr, Instrum, Vol IECI-24, pp. 241-247.

Mota Philippe, Jean Pierre Rogon, 1984, "Digital position servo system: A state variable

feedback system," IEEE Trans. Ind. App. Vol. IA-30, pp 1473-1481.

Nordy C, and Gabriel R, Leonhard W, 1980, "Field-orientated control of a standard AC-Motor using microprocessor", IEEE Trans Ind App Vol IA-16, No 2, pp. 186-192. MAR./APR.

見城尙志, 1981, 少形モータ基礎とマイコン制御, 第7章, 統合電子出版社.