

단일 루우프 제어기의 설계와 제작

이영일, 변대규, 권옥현

서울대학교 공과대학 제어계측공학과

Design and Implementation of One Loop Controller

Young Il Lee, Dae Gyu Byun, Wook Hyun Kwon

Dept. of Control and Instrumentation Eng.,

Seoul National University

Abstract

In this paper, the development of an One Loop Controller is described. The functions which are built in the One Loop Controller are summarized and the hardware implementation of these functions using 16 bit microprocessor is described. The developed One Loop Controller is applied to DC-Motor based position control system and its results are given.

1. 서론

단일루우프제어기(One Loop Controller: 이하 OLC)는 공정제어(Process Control)에 널리 쓰이는 DDC(Direct Digital Controller)로서, 그 이름이 시사하는 바와 같이 프로세스의 출력 하나에 대해 그림 1 과 같은 피드백제어 루우프를 형성하여 출력을 원하는 설정치로 유지하는 것을 기본목적으로 한다. OLC는 프로세스출력을 제어하는 외에, 설정치(SV), 출력값(PV), 제어값(MV)들을 아나로그나 디지털형태로 표시하고[11], 에러(e)가 적정선에서 유지되는 가를 감시하며[3], 출력값을 수동으로 조작할 수 있게 하는 등 프로세스출력 하나를 제어하고 감시하기에 편리한 구조를 갖는다. OLC는 아나로그제어기에 비해 제어내용의 변경이 용이하며, 여러가지 방법으로 설정치를 부여할 수 있다. 또한 여러가지 형태의 제어함수와 연산자들을 모듈형태로 제공하므로 사용자는 이들을 이용하여 대상시스템에 적합한 제어프로그램을 쉽게 작성할 수 있다

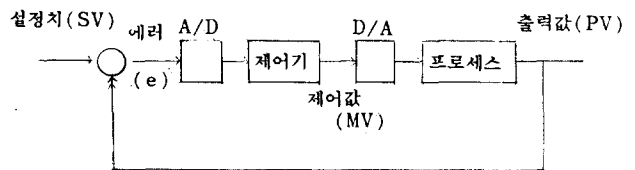


그림 1. 전형적인 제어루우프

제어프로그램은 별도의 프로그램용 기기나[3], OLC 자체에 부착된 프로그래머[10]를 이용하여 작성되어 제어에 필요한 다른 정보들과 함께 EPROM등의 불휘발성 기억소자에 저장되어 OLC에 장착된다. 공정제어에 있어서 가장 흔히 쓰이고 있는 제어기의 형태는 PID 제어기이다. 따라서 OLC의 제어함수중에서 가장 중요시 되는 것이 PID 함수로서, 2 자유도 PID[13]등 보다 효율적이고 신뢰도 높은 이산 PID 알고리즘을 개발하기 위한 연구가 이루어지고 있다. 특히 근래에 들어서는 PID 제어기의 자기동조(Auto Tuning)와 적응동조(Adaptive Tuning) 기능이 OLC가 갖추어야 할 기본적인 기능이 되어가는 추세이다[11].

본 연구에서는 다양한 구조의 제어알고리즘을 쉽게 프로그램화 할수 있는 제어언어를 가진 OLC를 개발하였는 바, 연구분야는 다음의 세가지로 나누어진다. 첫째가 아나로그 입출력과 디지털 입출력 그리고 사용자와의 인터페이스를 위한 주변기기들을 갖춘 하드웨어를 구성하는 것으로서, i8086 범용 마이크로프로세서를 사용하였다. 둘째로 마이크로프로세서를 잘모르는 사람도 사용할 수 있는 제어전용의 언어를 개발하는 것이다. 제어전용의

언어는 각종 제어알고리즘을 쉽게 사용할 수 있고, 제어시스템이 요구하는 엄밀한 시간한계를 지킬 수 있는 구조를 가져야 한다. 세제는 제어프로그램의 입력 및 수행, 자기진단, 각종 표시기능 등을 연결하여 OLC 전체의 동작을 관리하는 모니터 프로그램을 개발하는 것이다.

이하 본 논문의 구성은 다음과 같다. 2절에서는 OLC의 하드웨어와 모니터 프로그램의 구조에 대해서 언급하고, 제어언어의 형태와 수행방법은 3절과 4절에서 설명한다. 5절에서는 본 OLC에 설치된 PID 자기동조방법에 대해 설명한다. 그리고 6절에서는 모터를 이용한 위치제어 시스템에 본 OLC를 적용실험한 결과를 보였다.

2. OLC의 구조

그림 2는 개발된 OLC의 개념도이며, 그림 3은 설정치, 제어값, 출력값을 표시하는 전면패널의 실물사진이다. OLC의 동작을 위해서 필요한 정보들을 기본 파라미터(Basic Parameter), 입력 파라미터(Input parameter), PID 파라미터의 세 가지로 나누었다. 입력 파라미터에 의해 입력이 처리되면 기본 파라미터 및 PID 파라미터에 지정된 방식으로 제어프로그램을 수행하고, 각종 값들을 표시한다. 사용자는 측면패널과 전면패널을 통해 세 가지 파라미터를 조작하고, 제어프로그램을

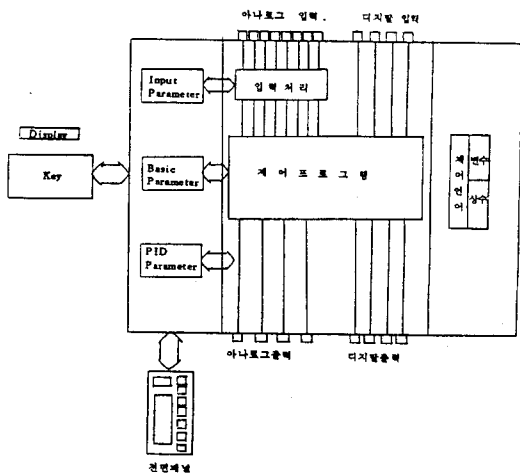


그림 2. OLC의 개념도



그림 3. 전면패널 사진

작성하며, 각종 값들을 감시할 수 있다. OLC의 하드웨어는 CPU부와 입출력부로 나누어진다. CPU부는 시스템 ROM, RAM, KEY/DISPLAY 접속부, 타이머 등으로 이루어진다. 다음은 CPU부에 사용된 부품과 기능을 요약해 놓은 것이다.

i8086	CPU
2764	System ROM (16 Kbyte)
6264	파라미터, 제어프로그램 저장, 백업전지 포함
8259	인터럽트 제어
8253	샘플링 시간 점검 Watch Dog Timer
8279	KEY/DISPLAY 접속
8251	Serial 통신

입출력부는 OLC에 있어서 제어를 위한 정보들 읽어 들이고 제어결과를 외부로 출력하는 부분이다. 입출력부는 아날로그 입출력부와 디지털 입출력부로 나눌 수 있다. 아날로그 입출력에 있어서는 잡음 없는 안정된 전원의 공급이 정확한 입출력 변환에 필수적이다. 따라서 아날로그 입출력부는 CPU부와 별도의 전원을 사용하고 있으며, OPTO COUPLER를 통하여 CPU부와 신호를 주고 받는다. OLC의 모니터 프로그램은 KEY입력을 처리하여 사용자로부터 제어프로그램과 파라미터를 받아 들이고, 샘플링 시간을 점검하여 필요한 시기에 입력을 읽어 들여 제어프로그램을 수행 시킨다. 또한 설정치, 프로세스출력 등을 표시하고, 자기진단 기능에 의해 에러 발생 여부를 점검하고, 에러가 발생 하였을 때 이를 적절히 처리한다. 위와 같은 기능을 수행하기 위해 모니터 프로그램은 주흐름(Main Flow)과

인터럽트 흐름(Interrupt Flow)의 이원적인 프로그램 기법을 사용한다. 즉 1 msec 마다 수행되는 인터럽트 흐름에 의해 시간준수가 요구되는 모든 사항들이 점검되어 각각의 해당하는 Flag와 Data 버퍼가 결정되면 주흐름은 그 Flag와 Data에 따라 입출력 동작을 하고, 제어프로그램을 수행하며, Key입력을 처리한다.

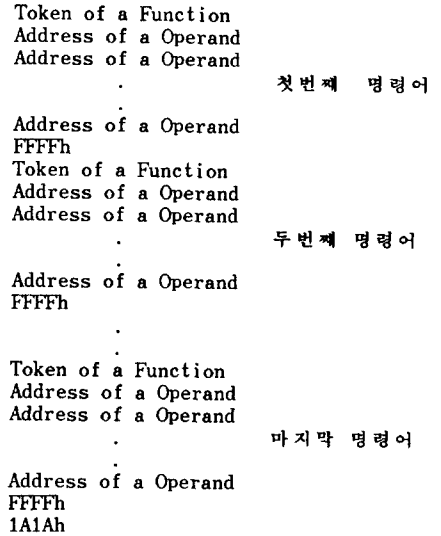
3. 제어언어

디지털 제어장치의 효율성과 신뢰성은 소프트웨어에 의해서 좌우된다. 따라서 복잡한 프로그래밍 작업이 필요하고, 이때 주로 사용되는 언어는 마이크로프로세서의 기본언어인 어셈블리(Assembly)어이다. 그러나 이 언어는 마이크로프로세서의 내부구조를 잘 모르는 사람이 사용하기 힘든 것으로서 보다 사용하기 쉬운 고급언어의 필요성이 대두된다.

본 OLC의 제어프로그램 작성에 사용되는 제어언어는 블록선도로 표시된 제어시스템을 쉽게 프로그램화 할 수 있는 구조를 가진 BOL(Bolck Oriented Language) 이다. BOL은 블록선도에 나타난 각 블록의 전달함수를 하나의 명령어(Function)로 실현할 수 있다. 명령어를 사용하는 기본구조는 다음과 같다.

명령어 Operand1, Operand2, ... , Operand n
(Mnemonic)

명령어는 그 기능에 따라 산술연산, 논리연산, 입출력동작, 제어동작의 네가지로 나눌 수 있다. 본 OLC의 제어언어는 43개의 명령어를 가지고 있다. 표 1 은 각 명령어와 해당하는 토큰을 나타내고 있다. 제어동작의 PID를 제외한 명령어의 수행에 필요한 정보는 오퍼란드로 주어진다. 제어프로그램은 위와 같은 구조의 명령어를 순서적으로 사용하여 작성한다. 제어프로그램의 저장형태는 다음과 같다.



셋플링 시간이 되면 첫번째 명령어의 토큰을 페취(Fetch)해서 거기에 해당하는 루틴으로 점프해서 FFFFh를 만날때 까지 오퍼란드를 가져오면서 명령어를 수행한다. 첫번째 명령어의 수행이 끝나면 다음 토큰을 페취 해와서 거기에 해당하는 루틴으로 점프하는 과정을 1A1Ah를 만날 때까지 반복한다.

4. OLC의 실수연산

OLC에서 사용되는 수는 실수, 정수, ON/OFF의 세가지가 있다. 여기서는 실수의 표현방식과

표 1. 명령어의 토큰(Token)

I/O	Function	Arithmetic Function	Logic Function	Control Function
SRT	: 1	MOV : 15	AND : 29	PID : 43
LIN	: 2	ADD : 16	OR : 30	ROC : 44
TMC	: 3	SUB : 17	XOR : 31	PRO : 45
PRC	: 4	MUL : 18	NEG : 32	RAT : 46
ASV	: 5	DIV : 19	GT : 33	LLG : 47
APV	: 6	ABS : 20	LT : 34	DIF : 48
FIL	: 7	INV : 21	GE : 35	DLY : 49
AOT	: 8	MAX : 22	LE : 36	: 50
HLA	: 9	MIN : 23	SEL : 37	: 51
LIM	: 10	MEN : 24	SCH : 38	: 52
DOT	: 11	MED : 25	TIM 1 : 39	: 53
	: 12	SUM : 26	TIM 2 : 40	: 54
	: 13	: 27	: 41	: 55
	: 14	: 28	: 42	: 56

연산방법에 대해서 설명한다. OLC 내에서 실수는 16 bit의 고정소숫점 형태로 표현 된다. 따라서 다룰수 있는 실수의 크기와 정도(Resolution)에 한계가 있다. 소숫점이 어느 bit 사이에 있다고 보느냐에 따라 다룰수 있는 실수의 크기와 정도는 서로 반비례적으로 변한다. 소숫점이 10째와 11번째 bit 사이에 있다고 볼때에 다룰수 있는 실수의 크기와 정도는 다음과 같다.

범 위 (Range)	:	-32— 31.999
정 도 (Resolution)	:	1/1024
저장형태	:	
실 수	저장형태	
-32	8000h	
0.5	200h	

모든 산술연산의 결과도 위와같은 형태로 저장 되어야 한다. 따라서 곱셈과 나눗셈, 자승근은 다음과 같이 구해진다.

A, B : 고정 소숫점 형태로 저장된 수

C : A, B 의 연산결과로 저장될 수

곱셈 : $C = A * B / 1024$

나눗셈 : $C = 1024 * A / B$

자승근 : $C = 1024 * A$

고정소숫점 형태로 숫자를 표시하고 연산하는 것은 부동소숫점 형태를 사용하는 것에 비해 수의 크기와 연산의 정확도에 있어서 많은 제약이 있지만, 디지털 제어기의 경우, 제한된 bit의 A/D 와 D/A 변환기에 의해서 생기는 입력력 신호의 크기와 정도의 한계를 고려한다면 고정소숫점 연산과 부동소숫점 연산은 결과적으로 큰 차이가 없으리라 생각된다. 또한 고정소숫점 연산을 사용할 경우 연산시간이 절약 되므로 보다 빠른 시스템에 적용이 가능하다. OLC 의 입력을 통해 들어오는 신호의 물리량은 대상시스템에 따라 각각씩이므로 이를 0 에서 4 까지의 수로 정규화(Normalize)하여 표현한다. 또한

입력신호의 최소값을 0 % , 최대값을 100 %로 표현하여 입력된 신호가 그것이 가질수 있는 영역상에서 어떤 위치에 있는가를 알 수 있게한다.

5. PID 제어기의 동조

PID 제어기는 공정제어(Process Control)에 널리 쓰이고 있으며 경험에 의해 그 유용성이 증명 되었다. PID 제어기의 적절한 동조(Tuning)를 위한 체계적인 방법들이 Ziegler-Nichols(1942) 이후로 많이 제안 되었다. 근래에 공정제어에 DDC가 채용 되면서 동조방법을 자동화한 자기동조(Auto Tuning) 방법이 제시되었고 실제로 자기동조 기능을 가진 DDC가 상품화 되어 나오기도 했다[11].

PID 동조방법으로 가장 널리 쓰이는 Ziegler-Nichols[7] 방법은 P 제어물 이용하여 프로세스를 준안정의 진동을 일으키도록 해야하므로 그 과정을 마이크로프로세서를 사용하여 자동화 하기에 난점이 있었다. 1983년 Astrom과 Hagglund[8]는 릴레이를 이용하여 이 난점을 해결 하고 위상이득 또는 비레이득을 보장하는 동조방법을 제안 하였다. 본 논문에서는 Astrom의 릴레이 방법을 사용하여 프로세스를 식별하고, 그 정보를 이용해서 디지털 PID 제어기에 대해 비레이득 또는 위상이득을 보장하는 동조방법을 개발하였다. 그림 2와 같은 이산제어시스템을 생각한다. Astrom의 릴레이 방법의 의해 $\text{Arg}(G_p(j\omega_1)) = -180$ 인 ω_1 과 $\left| G_p(j\omega_1) \right|$ 을 알고 있다고 볼때, 비레이득 A_m 을 가지기 위해서는 다음식이 만족 되어야 한다.

$$1/T * D(\exp(j\omega_1 T)) G(j\omega_1) = 1/A_m \exp(-180^\circ) \quad (1)$$

$$G(j\omega) = \frac{1 - \exp(Ts)}{S} G_p(S) \quad \left. \begin{matrix} T = \text{샘플링 시간} \\ S = j\omega \end{matrix} \right\}$$

다음과 같은 이산 PID식을 이용하면

$$m(k) = m(k-1) + k_p (e(k) - e(k-1)) + k_i e(k) + k_d (e(k) - 2e(k-1) + e(k-2)) \quad (2)$$

$$D(Z) = \frac{M(Z)}{E(Z)} = k_p + k_i \frac{T * Z}{Z - 1} + k_d \frac{Z - 1}{T * Z}$$

로 주어진다.

$w_1 T < 1$ 인 경우에 $\exp(jw_1 T) = \frac{1+0.5jw_1 T}{1-0.5jw_1 T}$ 의

근사식을 이용하여 식 (1)을 다시 쓰면 몇가지 과정을 거쳐 다음과 같은 관계식을 얻을 수 있다.

$$k_p + \frac{k_i T}{2} + \frac{k_d (T/2) w_1^2}{1+(T w_1 / 2)^2} = \frac{\sqrt{1+(0.5 w_1 T)^2}}{A_m |G_p(jw)|} \cos \frac{w_1 T}{2}$$

$$k_d \frac{w_1}{1+(T w_1 / 2)^2} - \frac{k_i}{w_1} = \frac{\sqrt{1+(0.5 w_1 T)^2}}{A_m |G_p(jw)|} \sin \frac{w_1 T}{2} \quad (3)$$

정상상태오차를 없애도록 k_i 를 적당히 잡으면 k_p , k_d 는 (3)에서 구할 수 있다.

6. 실험결과

OLC의 성능실험을 위해서 간단한 제어시스템을 구성하였다. 대상 시스템은 DC-서어보 모터를 이용한 위치제어시스템으로서 구성된 제어루우프는 그림 4와 같다. 표 2는 자기동조 기능에 의해 측정된 w_1 과 $|G_p(jw_1)|$ 그리고 Ziegler-Nichols 방법과 샘플링 시간이 10 msec 일때 식 (3)에서 계산된 k_p, k_i, k_d 값이며, 그림 5는 릴레이 제어를 통해 PID 계수가 결정되고, 곧이어 식 (3)에 의해 설정된 계수에 의해 PID제어가 행해지는 과정에서의 출력 위치를 그린 것이다.

7. 결론

본 연구에서 개발된 OLC는 8점의 아나로그 입력, 4점의 아나로그 출력, 그리고 각각 4점씩의 디지털 입출력을 가지고 2개의 PID 루우프를 구성할 수 있다. 모든 연산은 16 bit의 정수연산 형태로 행하였으며, 제어함수들을 부동 소숫점연산에 손색없이 수행할 수 있었다. 또한 연산시간이 부동소숫점 연산에 비해 줄어들어서 모터와 같은 비교적 빠른 시스템도 제어할 수 있었다. PID 제어기의 자기동조 방법에 대해서도 연구가 이루어진 바, 시스템에 대한 사전지식이 전혀 필요없는 Astrom[8]의 릴레이 제어를 이용하여 시스템의 극한주파수(Ultimate Frequency)와 극한이득(Ultimate Gain)을 측정하고 이에 대해 Ziegler-Nichols[7]의 동조식을 사용한 결과 진동이

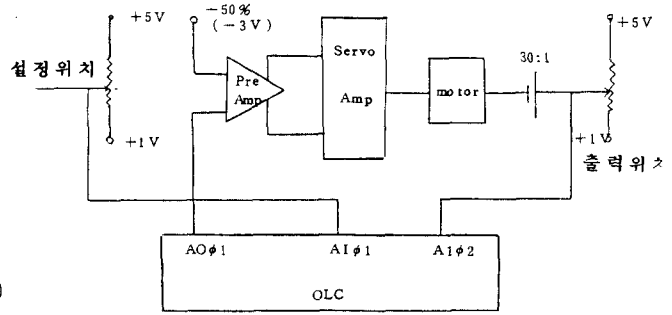


그림 4. 위치제어 시스템

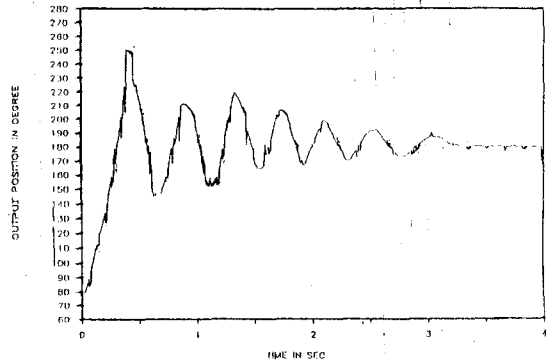


그림 5. 출력 위치

표 2. PID 계수

	식 (3)	Ziegler-Nichols
k_p	1.46	1.6
k_i	0.2	1
k_d	0.003	0.25

심하게 나타났다. 즉 이산식의 PID 제어기에 대해서는 Ziegler-Nichols[7]가 제안한 제어기의 계수보다 적은 값을 사용하여야 했다. 특히 샘플링 간격이 커질수록 적은 값을 써야 했다. 본 논문에서는 이산 PID 제어기에 적용 가능한 새로운 동조식을 유도하였다. 이를 이용하여 구한 계수를 Ziegler-Nichols의 것과 비교해 본 결과 보다 타당한 값을 얻을 수 있었다. 새로운 동조식에 대해서는 많은 시뮬레이션을 통하여 다른 방식과 비교해 보아야 할 것이며, 제어언어의 여러 제어함수와 OLC 하드웨어의 신뢰성도 많은 실험을 거쳐 개량되어야 할 것으로 본다.

참 고 문 헌

1. 변대규, "제어언어에 관한 연구," 서울대학교 공과대학 제어계측공학과 석사학위논문 1985.
2. 서병철, 박호건, "제어기기의 동조에 관한 연구," 전자학회 논문집, Vol 21., No. 3, 1984.
3. Toshiba One Loop Controller, Technical Manual.
4. Product Data, Yamatake-Honeywell.
5. F.G. Shinskey, Process Control systems. McGraw-Hill, 1979.
6. C.L. Phillips and C.T. Huang, "Bode Design of Digital Controllers," IEEE Trans. Education, Vol. E-26, No. 2, MAY 1983.
7. J.G. Ziegler, N.B. Nichols and N.Y. Rochester, "Optimum Settings for Automatic Controllers," Trans. ASME, November 1942.
8. K.J. Astrom and T. Hagglund, "Automatic Tuning of Simple Regulators with Specifications on Phase and Amplitude Margins," Automatica, MAY 1984.
9. K.J. Astrom and B. Wittenmark, Computer Controlled System. Prentice-hall, 1984.
10. Teleperm Compact Controller F, Training Textbook, Fuji.
11. 760 Series Single Station Micro Controller, EXACT Tuning, Foxboro.
12. Technical Information, TI 200-222, Foxboro.
13. PID 제어방식의 구성, 특성과 적용, 월간 자동화기술, 1987, 2월.