

전기로 제어를 위한 전문가 시스템

명노직

허옥열

인하대학교 공과대학 전기공학과

Expert system for electrical furnace

No-Jik Myoung
Dept. of Electrical Eng., Inha Univ.

Uk-Youl Huh

ABSTRACT.

In this paper, An expert system for electric furnace with time delay is proposed. The expert system uses Fuzzy control theory. The conventional controller uses Auto-Tuning control theory. From experiment, we can obtain that the response of expert system is superior to the response of the conventional controller. In this experiment, the expert controller is implemented with the IBM PC. The 8751 One chip processor controlling the electric furnace is used.

1. 서론.

디지털 컴퓨터의 개발이후에 많은 제어이론들이 개발되었다. 초기 Ziegler-Nichols[1] 는 실험적데이터를 이용해 PID 계수를 조정하는 방법을 제안하였다. 이 방법에 의해서는 비교적 좋은 결과를 얻기는 했으나 비례상수를 정하는 문제가 하나의 단점으로 나타났다. 그후 이 기본 아이디어를 이용 비례기 대신에 리레이를 삽입하여 인계이득과 임계주기를 자동으로 구하자는 자동동조이론이 Astrom[2] 에 의해 제안되었다. 이 방법은 실제 제어기를 구성하기 쉽고 프로그램이 간단하기 때문에 최근들어 많은 연구가 진행중에 있다. 그러나 이 방법 역시 리레이의 크기를 구한다거나 히스테리시스의 크기를 구하는 문제에 부딪치게 된다.

이와 같이 기존의 제어이론들은 모델링, 수학적인 어려운 접근 등에 의해 실제 대상물에 적용하기가 수월하지가 않다. 이에 AI 에서 사용되던 방법들을 이용 전문가가 가지고 있는 지식을 구현하자는 전문가 시스템의 연구가 시작되었다. Sripada[3] 는 퍼지집합을 이용한 제어기를 구성하여 전문가 시스템이 기존의 PID 제어기보다 우수함을 보였고, Porter[4] 는 Expert Tuner 를 구성하여 Ziegler-Nichols[1] 가 제안한 방법보다 우수한 Tuner 를 구성하였다. Tomohiro Takagi[5] 는 전문가의 지식을 Kalman Filter 를 이용해 프로그램할수 있는 방법을 제안하였다.

본 논문에서는 Tomohiro Takagi[5] 가 제안한 동정(Identification) 방법을 이용, 퍼지제어기를 구현하고 비교 제어기로 자동동조 PI 제어기를 구현하여 퍼지제어기의 우수성을 보이고자 한다.

일반적으로 전기를 모델링하는데는 여러가지 방법이 있으나 가장 간단한 모델로서 다음과 같은 형태의 일차함수형태로 모델링할수있다.

$$G(s) = \frac{K * e^{-Td*s}}{(\tau*s + 1)} \quad (1)$$

여기서

- K : 개루우프 이득
- Td : 시간지연 (Time delay)
- τ : 시정수 (Time Constant)

기존제어이론에서도 시간지연을 고려한 제어기를 구성하는 연구가 많은 진척을 보이고 있으나 자동동조 제어기의 경우 시간지연이 많은 시스템에 대해서는 정확한 계수 추정이 어렵다. 또한 시스템의 계수들이 변화하는 비선형성을 갖기 때문에 제어에 많은 문제점들이 있다. 이를 극복하기위한 한 방법으로 본 논문에서는 퍼지제어기법을 이용하여 시간지연에 좀더 강인한 제어기를 구성하고자 한다.

2. 제어이론

2-1. 퍼지 제어기 (Fuzzy Controller)

초기 Zadeh[6] 가 일반적인 집합이론보다 좀더 확장된 퍼지집합론을 제안하였다. 이를 제어에 이용하자는 이론들이 최근들어 많이 나오고 있다. Gupta[7] 는 퍼지집합에 대해 자세한 정리를 해주고 있다.

퍼지제어도 일반적인 인공지능과 마찬가지로 지식베이스(Knowledge Base)와 추론기구(Inference Engine)로 구성되어 있다. 지식베이스로는 몇개의 부분집합을 이용한 데이터 베이스와 제어에 필요한 규칙들로 구성된다. 추론기구로서는 Composition rule 이 사용되고 있다.

본 논문에서는 부분집합 A 의 Membership Function 은 A(x) 로 표현하고 제안(proposition) "x is A AND y is B" 의 참값은

$$| x \text{ is A AND } y \text{ is B} | = A(x) \wedge B(x) \quad (2-1-1)$$

로 표현한다.

본 논문에서는 다음과 같은 제어에 필요한 몇개의 규칙과 몇개의 부분집합으로 구성하였다.

$$R: \text{ IF } x_1 \text{ is } A_1 \text{ AND } x_2 \text{ is } A_2 \text{ THEN } u = G(x_1, x_2) \quad (2-1-2)$$

여기서

- x_1 : 현재의 Error (Reference Input - Plant) 값
- x_2 : Error 의 변화 (Error(k) - Error(k-1)) 값
- A_1, A_2 : 부분집합

G : u 값을 구해주는 수식
 $G(x_1, x_2) = x_1 * g_1 + x_2 * g_2$

위의 AND 는 두개의 값중에서 작은값을 취해준다
 위와같은 규칙들을 여러개 구성하고 이때 구해지는
 각각의 u 값에 Composition rule 을 적용하여 최종의
 u 값을 구한다.

최종의 u 가 구해지는 과정을 보면 다음과 같다.
 1) 부분집합들을 구하고 각각의 제어변수에 대응되는
 Membership Function 을 구한다.

여러개의 제어규칙을 구성한다.
 2) 각각의 $u^i = G(x_1, x_2)$ 들을 구한다.
 u^i 의 참값은

$$|u=u^i| = A^i_1(x_1) \wedge A^i_2(x_2) \quad (2-1-3)$$

3) 최종값 u 는

$$u = \frac{\sum |u=u^i| * u^i}{\sum |u=u^i|} \quad (2-1-4)$$

에 의해 구해진다.

이제 a_1 과 a_2 를 구하는 문제이다.
 본 논문에서는 Tomohiro Takagi[5] 가 제안한 방법을
 사용하여 이들을 구하고자 한다. 제어변수를 x_1 과 x_2
 로 하고 부분집합을 Large, Small, Zero 로 한다. 이를
 이용 다음과 같은 제어 규칙을 정한다.

$$\begin{aligned} R^0 : & \text{ IF } x_1 \text{ is Zero THEN } u^0 = 0 \\ R^1 : & \text{ IF } x_1 \text{ is Small AND } x_2 \text{ is Small} \\ & \text{ THEN } u^1 = a^1_1 * x_1 + a^1_2 * x_2 \\ R^2 : & \text{ IF } x_1 \text{ is Small AND } x_2 \text{ is Large} \\ & \text{ THEN } u^2 = a^2_1 * x_1 + a^2_2 * x_2 \\ & \vdots \\ & \vdots \\ R^n : & \text{ IF } x_1 \text{ is Large AND } x_2 \text{ is Large} \\ & \text{ THEN } u^n = a^n_1 * x_1 + a^n_2 * x_2 \end{aligned} \quad (2-1-5)$$

이때의 플랜트 입력 u 는

$$u = \frac{\sum_{i=0}^n ((A^i_1(x_1) \wedge A^i_2(x_2)) * u^i)}{\sum_{i=0}^n (A^i_1(x_1) \wedge A^i_2(x_2))} \quad (2-1-6)$$

계산을 간단하게 하기위해 새로운 변수

$$K_i = \frac{(A^i_1(x_1) \wedge A^i_2(x_2))}{\sum_{i=0}^n (A^i_1(x_1) \wedge A^i_2(x_2))} \quad (2-1-7)$$

를 사용하면 플랜트 입력 u 는

$$\begin{aligned} u &= \sum_{i=0}^n K_i * (a^i_1 * x_1 + a^i_2 * x_2) \\ &= \sum_{i=0}^n (a^i_1 * x_1 * K_i + a^i_2 * x_2 * K_i) \end{aligned} \quad (2-1-8)$$

x_{1j}, x_{2j}, u^i_j 가 주어졌을때 계수 a^i_{1j}, a^i_{2j} 를 최소
 자승법에 의해 구할수 있다.

$$X = \begin{bmatrix} x_{11} * K_{11} & x_{11} * K_{21} & \dots & x_{21} * K_{11} & x_{21} * K_{n1} \\ x_{12} * K_{12} & x_{12} * K_{22} & \dots & x_{22} * K_{12} & x_{22} * K_{n2} \\ x_{13} * K_{11} & x_{13} * K_{23} & \dots & x_{23} * K_{13} & x_{23} * K_{n3} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{1m} * K_{1m} & x_{1m} * K_{2m} & \dots & x_{2m} * K_{1m} & x_{2m} * K_{nm} \end{bmatrix} \quad (2-1-9)$$

$$K_{ij} = \frac{A_{i1}(x_{1j}) \wedge A_{i2}(x_{2j})}{\sum A_{i1}(x_{1j}) \wedge A_{i2}(x_{2j})} \quad (2-1-10)$$

$$Y = [y_1, \dots, y_m]^T \quad (2-1-11)$$

$$P = [a^1_0, \dots, a^n_0, a^1_1, \dots, a^n_1, a^1_2, \dots, a^n_2]^T \quad (2-1-12)$$

이에 P 는 다음과 같이 구해진다.

$$P = (X^T X)^{-1} X^T Y \quad (2-1-13)$$

그러나 위와 같은 경우 데이터의 수가 작을 경우에
 적합한 방법이라 할수있다, 그러나 데이터가 많고 특히
 실험 데이터 자체에 오차를 포함하고 있다면 정확한
 계수를 구한다는 것이 매우 어렵다. 그래서 Tomohoro
 Takagi [5] 는 Stable state Kalman Filter 를 사용하였
 는데 그 구조를 보면 다음과 같다.

$$P_{i+1} = P_i + S_{i+1} * x_{i+1} * (u_{i+1} - x_{i+1} * P_i) \quad (2-1-14)$$

$$S_{i+1} = S_i - \frac{S_i * x_i + x_{i+1} * S_i}{1 + x_{i+1} * S_i * x_i^T} \quad (2-1-15)$$

$$P = P_m \quad (2-1-16)$$

이때 P 와 S 의 초기치는 다음과 같이 정한다.

$$P_0 = 0 \quad (2-1-17)$$

$$S_0 = \alpha * I \quad (2-1-18)$$

α 는 큰수, I 는 단위 행렬

위와 같은 수식을 이용하여 P 를 구하고 이를 이용
 하여 퍼지제어기를 구성한다.

2-2. 자동동조 제어기 (Auto-Tuner)

Astrom [2] 은 임계이득과 임계 주기를 구하기 위해
 동조기에 리레이를 삽입하여 자동동조기를 구성하였다
 이의 블록선도는 그림 1 과 같다.

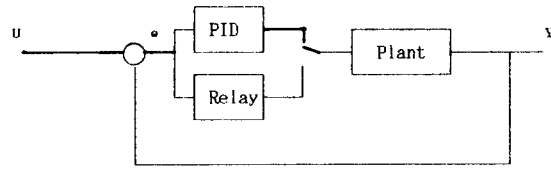


그림 1. 자동동조기의 블록선도

이때 리레이의 크기를 d 로 하고 편차(Error) 의 크
 기를 e 라 하면 임계이득 K_c 는

$$K_c = \frac{4 * d}{\pi * e} \quad (2-2-1)$$

이다.

임계 주기는 편차가 0(Zero) 를 지나는 시간을 측정함으로써 구할수 있다.
구해진 임계이득과 임계주기를 이용해 PID 의 계수를 구하면 다음과 같다

PI 의 경우

$$K_p = K_c / 2.2 \quad K_i = T_c / 1.2 \quad (2-2-2)$$

PID 의 경우

$$K_p = K_c / 1.7 \quad K_i = T_c / 2 \quad K_d = T_c / 8.0 \quad (2-2-3)$$

본 논문에서는 위에서 구한 PI 계수를 이용 PI 제어기를 구성하였다.

3. 제어 시스템 구성

본 논문에서는 실험을 위해 8751 을 이용한 하드웨어(Hardware)와 IBM PC 에서 동작하는 제어 프로그램을 구성하였다. 제어 프로그램은 C 언어를 사용하였고 그래픽을 위해 Halo 88 라이브러리를 이용하였다. 실제 온도의 변화를 온라인으로 볼수 있도록 온라인 제어프로그램을 구성하였다. 퍼지제어와 자동동조제어기가 온라인 제어 프로그램에 함께 포함되었다. 하드웨어로는 8751을 기본으로 한 전기로 제어기를 구성하였다. 전기로에 전원을 가해주는 회로로는 SSR(Solid State Relay)를 사용하였고 온도를 입력위해 ADC 회로와 Pt100 Ω 을 사용하였다 8751은 ADC의 값을 읽어 IBM PC로 보내주고, 또 IBM PC에서 계산된값을 가지고 SSR을 ON,OFF 해주는 역할을 한다. IBM PC와 8751과는 직렬통신에 의해 데이터가 오고가게 하였다. 일정샘플링을 위해 제어 프로그램내에서 타이머 인터럽트를 사용하였다. 온라인 제어기의 신호흐름도는 그림 2와 같다. Timer Interrupt Routine 은 매 샘플링마다 8751로 동기신호를 보낸다. 이동기신호에 맞추어 8751에서는 온도를 읽어 직렬통신을 이용해 온도를 IBM PC로 보낸다. 이때 Serial Interrupt Routine 이 구동되어 그림 2에 나타나 있는 흐름을 따라 제어를 하게된다. 위와 같은 과정이 매 샘플링 시간마다 일어나게 된다. 제어기 전체 블록은 그림 3과 같다.

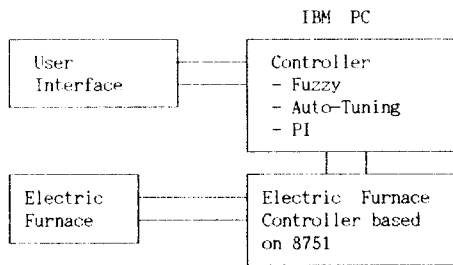


그림 3. 전체 블록선도

4. 실험결과 및 고찰

자동동조제어기와 퍼지제어기 모두다 모델링과정을 생략하고 제어기를 구성하기 때문에 본 논문에서는 모의 실험을 위해 임의의 시스템을 통한 모의실험을 해보았다. 임의의 시스템은 다음과 같은 시스템으로 하였다.

$$G(s) = \frac{1}{150 * s + 1} * e^{-30*s} \quad (4)$$

그림 4은 동조과정을 보이고 있다. 이때 입력으로는 50을 주고 리레이의 크기는 30으로 하였다. 초기 입력 25, 리레이크기 15로 동조를 하고 이때 구해진

PI 계수를 PI 제어기를 구동하여 출력을 보고 이때의 출력이 만족하지않을 경우 다시 동조하는 기법을 사용하였다, 다시 동조할때의 입력은 50, 리레이 크기는 30으로 하였다. 그림 5는 그림 4의 동조과정에서 얻어진 PI 계수 값으로 모의 실험하였을경우의 출력과 제어대상에 가해지는 입력을 나타냈다. 그림 6는 위의 제어대상을 퍼지제어를 이용하여 출력을 구한 그래프이다. PI 제어기보다는 썩좋은 결과는 아니지만 제어대상에 가해지는 입력의 크기가 작고 과도응답특성이 좋다고 볼수있다. 자동동조에 의한 결과는 더이상의 좋은 결과를 기대하기는 어렵다는 것이 이것이외의 모의실험에서 얻어지고 있다.

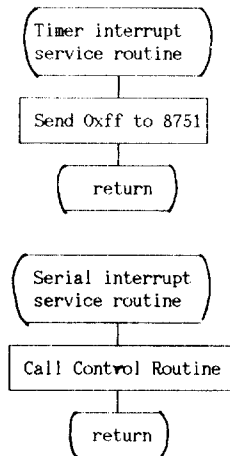
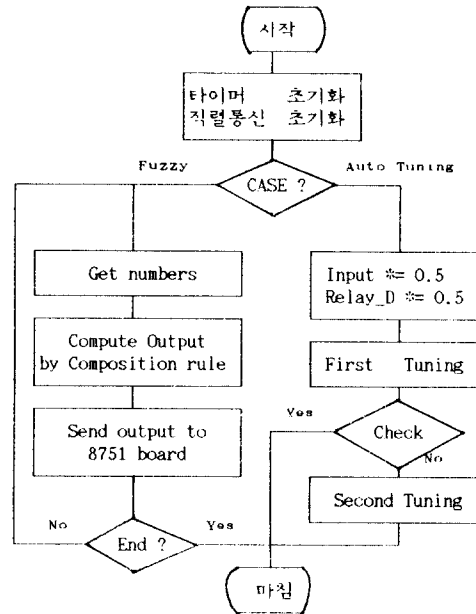


그림 2. 신호 흐름도

5. 결론

이상과 같이 모의 실험을 통한 퍼지제어와 자동동조 제어기와의 비교를 해보았다. 본 논문에서는 위의 이론을 실제 플랜트에 적용하고자 한다. 현재 실험은 진행중에 있다. 실제 실험에서는 전기로제어 전문가를 구할

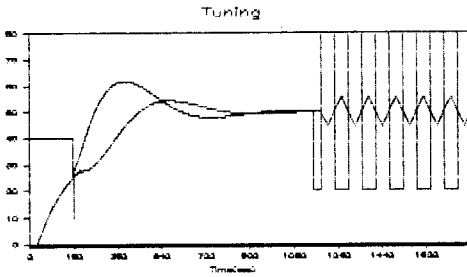


그림 4 동조 과정

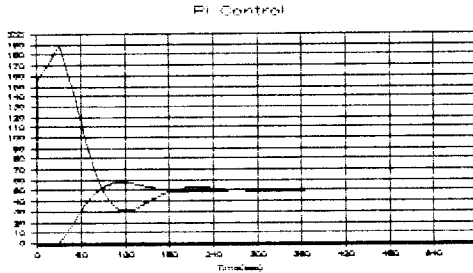


그림 5 PI 제어기의 응답

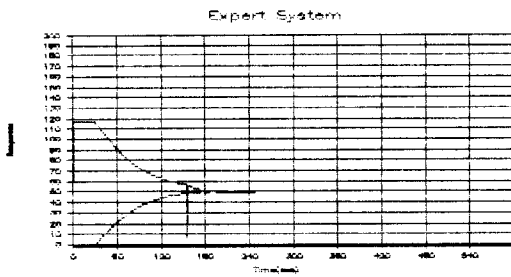


그림 6 퍼지제어기의 응답

수 없었던 관계로 시행오차를 통해 규칙을 구성하는 방법을 사용하였다. 모의 실험에서도 나타나듯이 더 좋은 규칙을 구성한다면 더 좋은 결과를 얻을수 있다. 위의 자동동조제어기에 퍼지기법을 이용한 전문가기법을 이용한 동조기를 구성하는 방법에 대한 연구도 현재 진행중에 있다.

6. 참고 문헌

- [1] Ziegler, J.G., N.B. Nichols and N.Y. Rochester, "Optimum Settings for Automatic Controllers," Trans on ASME. 64, 759, 1942.
- [2] Astrom, K.J. and Hagglund, "Automatic tuning of simple regulators with specifications on phase and amplitude margins" Automatica, 20, 645 - 651, 1984.
- [3] N.R. Sripada, and D.G. Fisher, and A.J. Morris "AI application for process regulation and servo control." IEE Proceedings, Vol. 134, Pt D, No. 4. 1987.
- [4] B. Porter, and A.H. Jones, and C.B. Mckeown "Real-time expert tuners for PI controllers. IEE Proceedings", Vol. 134, Pt D, No. 4. 1987.

- [5] Tomohiro Takagi and Michio Sugeno, "Fuzzy identification of systems and its applications to modeling and control", IEEE Trans on SMC-15, No 1, p116, 1987.
- [6] Lotfi A. Zadeh, "Outline of a New Approach to the Analysis of complex systems and decision process", IEEE Trans on SMC-3, No 1, p28, 1973.
- [7] Rammohan K. Ragade and Madan M. Gupta, "Fuzzy set theory : introduction." Fuzzy automata and decision process, North-Holland, New York, 1977.