

An Enhanced Technologies of Intelligent HVAC PID Controller by Parameter Tuning based on Machine Learning

Jee Hyun Kim*, Young Im Cho**

Abstract

Design of an intelligent controller for efficient control in smart building is one of the effective technologies to reduce energy consumption by reducing response time with keeping comfortable level for inhabitants. In this paper, we focus on how to find major parameters in order to enhance the ability of HVAC(heating, ventilation, air conditioning) PID controller. For the purpose of that, we use machine learning technologies for tuning HVAC devices. We show the simulation results to illustrate the behavioral relation of whole system and each control parameter while learning process.

▶ Keyword: machine learning, PID controller, HVAC, comfortable, parameter tuning

1. Introduction

일반적으로 실내 공기는 대기 온도, 상대 습도, 압력 등에 의해 영향을 받는다. 대체로 실내 압력과 습도는 균일한 환경을 갖으나 최적의 실내 환경제어를 위해서는 실내 각 공간의 각 지점에 대한 대기 온도(air temperature) 데이터가 있어야 최적의 실내 환경제어가 가능하다.

이러한 실내의 온도변화는 히터나 에어컨으로 부터 나온 수평 또는 수직으로 배치 된 열(heat source) 또는 태양열, 방사능, 벽체, 창문 및 문 등으로부터 나온 열에 의해 변화되거나 제어될 수 있다 [1,2].

최적의 실내 환경제어를 위해서는 이외에도 실내 기후환경에 영향을 미치는 중요한 매개 변수인 실외 공기, 온도 및 압력 등도 포함되는데, 이러한 매개 변수 값들은 Zambretti 알고리즘[3] 등을 이용하여 높은 정밀도와 단기 예측 외부 기상 조건의 변화로 측정된다면, 미리 실외 환경변화에 반응할 수 있을 것이므로 이로 인해 실내 온도제어를 위한 비용을 감소시킬 수도 있다.

그러나 일반적으로는 제어방법에 의해 실내 환경제어를 하는데, PI(Proportional Integral) 제어로 실제 목표 값에 가깝게 하는 제어는 완벽하게 할 수 있으나 제어 응답의 속도를 위해 필요하게 된 것이 미분 동작제어 D(Differential)이다. 이것은 급격히 일어나

는 외란에 대해 편차를 보고, 앞의 편차와의 차가 큰 경우에는 조작량을 많이 하여 신속하게 반응하도록 하는 방법으로서 오차값의 변화에 따른 조작량을 결정하는 제어방법이다[4].

PID 퍼지 제어 시스템을 성공적으로 구현하는 방법은 지금까지 많은 연구가 있어왔으나 퍼지 제어시스템을 구현함으로써 해결할 수 있는데, 이를 위해서는 퍼지 멤버십함수를 정의해야 한다. 또한 실내의 온도 필드의 분포와 그 시간에 대한 변화량을 학습함으로써 제어 시스템은 여러 상황에 대한 일련의 관리 규칙 (환경 매개 변수 세트)을 생성할 수도 있다.

최적의 실내 환경에 대한 연구로는 '편안함(comfortable)'을 측정하는 국제표준이 있다[5]. 국제표준에 의하면 실내의 편안함은 실내의 소기후(microclimate)를 결정하는 매개변수의 조합으로 생성되는데, 이때의 매개변수들은 공기온도, 공기속도, 상대습도 등을 포함하는 실내온도와 몇 가지 변수들에 의해 생성되며 이를 HVAC(heating, ventilation, air conditioning) 라고 한다. 따라서 이러한 매개변수들을 어떻게 결정짓느냐가 스마트 환경에서 국제 표준규격에 따른 '편안함'을 결정하는 중요한 요소가 된다.

Veronika는 매개 변수의 최적 및 허용값과 실외 온도에 대한

*First Author: Jee-Hyun Kim, Corresponding Author: Young-Im Cho

*Jee-Hyun Kim (jhkim@seoil.ac.kr), Dept. of Software Engineering, Seoil University

**Young-Im Cho (yicho@gachon.ac.kr), Dept. of Computer Engineering, Gachon University

Received: 2017. 10. 31, Revised: 2017. 11. 26, Accepted: 2017. 12. 04.

The present research has been conducted by the Research Grant of Seoil University in 2017.

데이터를 제공한바 있다[6]. ‘편안함’을 결정짓는 값으로 Veronika는 따뜻한 계절에는 공기 온도 20~24°C, 상대 습도 30~60% 및 공기 흐름 속도 0.25m/s에 대한 임계값을 제시하였고, 추운 계절에는 공기의 최적값 온도를 20~22°C, 상대 습도 및 공기 흐름 속도의 임계값으로 30~45% 및 0.5m/s를 제시하였다. 만약 공장 등 생산 시설이 갖추어진 공간에서는 이러한 매개 변수값들은 일의 특성에 영향을 받으나, 인간은 대체적으로 0.5~1.5°C 범위내 온도변화와 동일한 온도에서 상대 습도의 변동이 10% 이내일 경우에는 신체에 특별한 영향을 미치지 않는다고 하였다.

따라서 Veronika가 제시한 온도, 습도 및 공기 흐름 등을 안정적으로 유지할 경우 인간은 매우 ‘편안함’을 느낄 수 있다는 것이다.

본 논문은 여러 가지 국제표준 규격에 따라 인간이 편안함을 느끼는 최적의 실내 환경제어를 위해 HVAC PID 제어기를 설계하고자 한다. 앞서 제시한 인간의 편안함 온도, 습도, 공기흐름속도를 유지하려면 PID제어기에서 파라미터 P(gain), I(process delay), D(derivative)의 값을 정의해야 한다. 기존에는 대부분 최초 I값을 길게 주고 테스트를 시작하다가, process delay가 발생하면 P값을 줄이고 변화가 보이지 않으면 P값을 올리도록 하여 단 한번의 시도에도 바로 목표치에 바로 도달하도록 최적의 P값을 찾는 방법을 사용하여 왔다. 즉, 초기 I값을 점점 줄여서 최단 최적의 응답시간을 찾는 방법인데, 이 방법은 전문가 또는 인간의 지속적 제어관리가 필요한 문제점이 있다.

따라서 본 논문에서는 PID 제어기에서 최적의 파라미터 P(gain), I(process delay), D(derivative) 값을 정의하기 위해 머신러닝 학습방법을 적용하여 환경에 가장 적합한 파라미터값을 정의하는 제어 알고리즘을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 PID 제어기의 일반적 제어방법과 기존 방법의 문제점을 제시하고, 3장에서는 제안하는 지능적인 HAVC PID 제어 알고리즘을 제시하고자 한다. 4장에서는 시뮬레이션 결과를 제시하고 마지막 5장에서 결론으로 마무리하고자 한다.

II. Related Study

1. Definition of comfortable

‘편안함(thermal comfortable or comfortable)’은 몇 가지 환경을 바탕으로 연구하기 위해 제안된 복잡한 용어로서, 환경적 요소가 공기 온도(air temperature), 습도(humidity), 공기 교환 속도(air exchange rate) 및 평균 복사 온도(mean radiant temperature) 등과 통합되어 발생되는 용어이다.

이러한 편안함은 국제표준으로 정의되었는데 *PMV* (predicted mean vote)라고 하는 보편적인 방법에 의해 온도를 측정한다[7]. *PMV*는 다음 식(1)과 같이 나타낸다.

$$PMV = f(t_a, t_{mr}, v, p_a, M, I_d) \quad (1)$$

여기서 t_a 는 공기 온도이고 t_{mr} 는 평균 복사 온도이며, v 는 상대적 공기속도(relative air velocity)이고 p_a 는 공기 수증기압(air vapor pressure)이며, M 은 액티비티 레벨(activity level)이고 I_d 은 의복단열(clothing insulation)을 의미한다. 일반적으로 *PMV*는 의복과 액티비티 레벨을 고려한 환경변수의 조합을 측정하여 나타낸다.

*PMV*에 의해 ISO 7730-1984 표준은 일상적으로 앉아서 활동하는 데에 가장 적절한 온도를 여름과 겨울로 나누어서 다음 Table 1과 같이 제시하였다. 겨울의 실내 최적온도는 22°C이고 여름의 실내 최적온도는 24.5°C를 나타내고 있다. 여기서 $1\text{ clo} = 0.155^\circ\text{C}/W$ (W : watt)를 나타낸다. 의복단열을 나타내는 *clo*는 0(안입은 상태)에서부터 4(완전히 두껍게 껴입은 상태)까지 나타내며 한 레벨당 0.155°C 정도의 온도를 유지한다는 것을 나타낸다. 즉, 옷을 두껍게 입으면 0.155°C의 두배만큼의 온도상승 효과가 있다는 것을 의미한다. Watt는 액티비티 측정단위로 사용된다.

Table 1. Recommended Operative Temperature for Sedentary Activity according to ISO 7730-1984 Standard

Season	Clothing Insulation (clo)	Activity Level (met)	Operative Temperature (°C)	Optimal Operative Temperature (°C)
Winter	1.0	1.2	20-24	22
Summer	0.5	1.2	23-26	24.5

Table 2는 ASHRAE가 제안한 최적 실내환경을 나타낸 표로서 집안이나 빌딩의 거주자에게 추천하는 환경이다[7]. 최적 온도는 동일하게 겨울은 22°C이고 여름의 실내 최적온도는 24.5°C를 나타내고 있다.

Table 2. Recommended Operative Temperature Sets for Inhabitants based on ASHRAE 55-1992 Standard

Season	Clothing Insulation (clo)	Activity Level (met)	Operative Temperature (°C)	Optimal Operative Temperature (°C)
Winter	0.9	1.2	20-23.5	22
Summer	0.5	1.2	23-26	24.5

Fig.1.은 ISO 7730에서 추천하는 편안함(comfortable), 보통(moderate), 건조한 불편함(uncomfortable dry) 그리고 습기찬 불편함(uncomfortable damp)등을 나타내는 온도와 습도 간의 상관관계 그림이다[5].

Fig.1.은 실내의 습도(humidity)는 실내 온도(indoor temperature)와 공기 습도레벨(air humidity level)에 영향을 준다는 것을 나타낸다. 그림에서 60%~80%의 범위의 습도가

편안함을 느끼게 함을 알 수 있다. 겨울 최적온도인 22°C인 경우 70%가 최적의 상태 습도환경임을 나타낸다.

지금까지 국제표준 규격에 의해 겨울과 여름철의 최적의 실내온도와 습도와의 상관관계를 알 수 있었다. 여기서 관건은 PID제어기가 이러한 온도와 습도와의 상관관계를 어떻게 잘 유지하면서 제어하느냐 인데, PID제어의 최적의 P, I, D 파라미터 값을 세팅하는 것이 중요하다. 본 논문에서는 이것을 위해 머신러닝 방법을 사용하고자 한다.

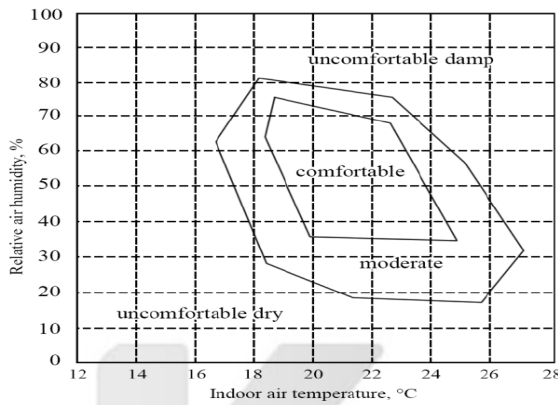


Fig. 1. The comfortable data of the premises, given on the basis of international European standards and recommendations ISO 7730

2. PID Controller

PID 제어는 전통적인 선형제어 방법이다. PID 제어기는 다음 식(2)와 같은 알고리즘으로 나타낼 수 있다[4].

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \left(\frac{de(t)}{dt} \right) \quad (2)$$

여기서 u 는 PID 제어 시그널이고 e 는 시스템 오차(시스템의 setpoint 예상치와 실제 출력값 간의 차이)이다. 제어 시그널은 $P(e$ 에 비례), $I(e$ 의 적분값에 비례) 및 $D(e$ 의 미분값에 비례)의 합에 의해 결정된다.

그동안 대부분의 제어관련 연구는 단일 PID보다는 기본구조는 PID제어와 같으나 내부 기능이 휴리스틱 규칙을 제어하는 기능을 갖는 PID 퍼지제어기가 지능적 제어기로서 많이 연구되어 왔다[8]. 여기서 ‘지능적’이라 함은 인간의 간섭없이 시스템이 스스로 결정하여 문제를 해결하는 능력을 말한다. 또한 스스로 튜닝하거나 인공지능 인식기술, 최적화 방법들이 포함된 제어기 연구도 있었으며[9], 적응형 퍼지규칙 조절이나 학습기능을 강화시키는 제어기 연구도 많이 있었다[10].

Fig. 2는 전형적인 PID 제어기의 구조이나 setpoint 설정 후 그림과 같이 P(gain), I(process delay), D(derivative) 값을 차례대로 식에 의해 결정하면서 다음 프로세스로 진행하는데, 본 논문에서는 여기서 퍼지 제어방법을 사용한다는 점이 특징이다.

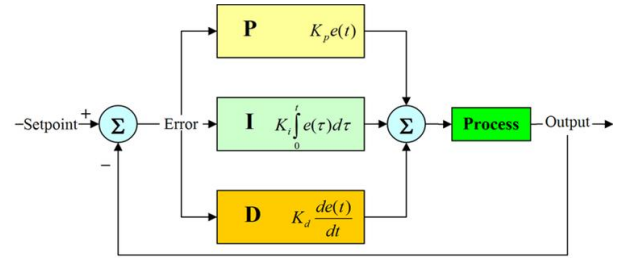


Fig. 2. Typical Block Diagram of PID Controller

3. Problem Definition

PID 컨트롤러를 사용하는 가장 큰 문제는 세 가지 핵심 파라미터인 k_p , k_i , k_d 를 최적으로 설정하는 일이다.

기존의 파라미터 설정방법으로는 대부분 최초 I값을 길게 주고 테스트를 시작하다가, process delay가 발생하면 P값을 줄이고 변화가 보이지 않으면 P값을 올리도록 하여 단 한번의 시도에도 바로 목표치에 바로 도달하도록 최적의 P값을 찾는 방법을 사용하여 왔다. 즉, 초기 길게 준 I값을 점점 줄여서 최단 최적의 응답시간을 찾는 방법인데, 이 방법은 전문가의 지속적 제어관리와 간섭이 필요한 문제점이 있어서 지능적 제어관리가 어려운 문제점이 있다. 따라서 보다 정확하고 단기간에 최적의 실내환경에 도달하기 위한 지능적인 PID 제어기의 연구개발이 필요하다.

본 논문에서는 단기간에 최적의 실내환경(온도, 습도환경)에 도달할 뿐 아니라 지능적으로 인간의 개입이나 간섭 없이 세 개의 파라미터 값을 찾는 알고리즘을 제안하고자 한다. 이를 위해 본 논문에서는 머신러닝의 학습알고리즘을 이용하여 인간 개입 없이 최단 시간 내에 가능하여 지능적인 제어가 가능하도록 하는 알고리즘을 제안하고자 한다.

III. Machine Learning based Enhanced Intelligent HVAC PID Controller

본 절에서는 지능적인 PID 제어가 되도록 하기 위해 머신러닝 기반으로 세 개의 파라미터 값을 세팅함으로써 단시간 내에 최적의 실내환경에 도달하도록 하는 향상된 지능적인 HVAC PID 제어알고리즘을 제안하고자 한다.

Fig.3은 본 논문에서 제안하는 향상된 지능적인 HVAC 제어 알고리즘의 전체적인 그림이다. 본 알고리즘은 크게 두 파트로 구성되어 있다. PID 제어기 부분과 머신러닝 부분으로 구성되어 있다.

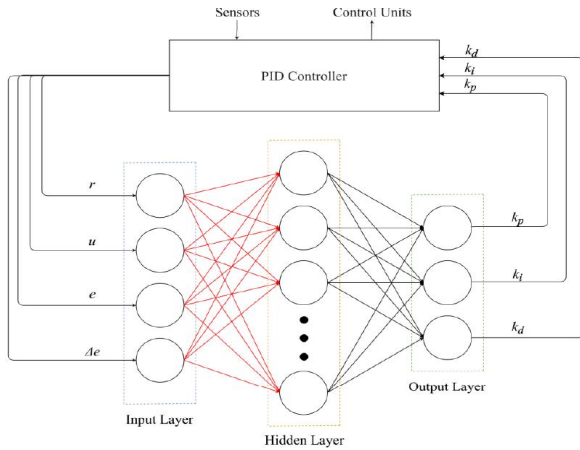


Fig. 3. Proposed Enhanced Intelligent HVAC PID Controller

1. PID Controller Part

지능적인 PID 제어를 위해 본 논문에서는 PID 퍼지제어기로 설계하여 휴리스틱 용어에 대해 퍼지 멤버십 함수를 설정 한다[1].

본 논문에서는 1년간 (2016.5월~2017.4월) 겨울과 여름철 동안 본 연구실내의 온도를 측정하였으며 그 결과 ‘온도’에 대한 멤버십 함수를 다음 Table 3과 같이 얻을 수 있었다.

Table 3. Temperature Values between Cold and Warm Seasons

Members hip Function	Cold Season			Warm Season		
	min, t, °C	max, t, °C	ave, t, °C	min, t, °C	max, t, °C	ave, t, °C
Very Cold	14.89	16.71	15.80	15.13	17.53	16.33
Cold	15.19	19.01	17.09	15.37	21.84	18.61
comfortable	16.87	24.28	20.57	20.16	26.12	23.14
Hot	21.76	28.26	25.01	22.92	30.24	26.58
Very Hot	27.40	30.90	29.15	29.02	32.18	30.60

1년 동안 관측된 데이터로부터 각 온도 용어(예, cold, hot...)에 대한 멤버십 함수는 Taleghani[11]의 실내 환경을 의미하는 용어표시를 이용하여 위의 Table 3과 같이 멤버십 함수를 정의하였다.

본 연구에서는 본 실험실 환경에 맞는 편안한(comfortable) 온도 조건을 찾은 것이며 이는 ISO 7730-1984 표준(Table 1) 이나 ASHRAE 55-1992 표준(Table 2)과도 부합되는 온도 조건이다.

이러한 온도조건과 해당되는 습도조건 (Fig.1)을 맞추어 줄 수 있는 최적의 k_p , k_i , k_d 를 찾아 세팅하고자 하며, 찾는 과정은 앞서 정의한 식 (2)에 의해 다음 식(3)과 같이 전개된다.

$$u_i = u_{i-1} + k_p \times (e_i - e_{i-1}) + k_i \times e_i + k_d \times (e_i - 2 \times e_{i-1} + e_{i-2}) \quad (3)$$

여기서 u 는 PID 제어기의 출력값이며, k_p 는 비례상수이고, k_i 는 적분상수이며, k_d 는 미분상수이다. 또한 e_i 는 시스템 오

류로서 다음 식(4)와 같이 표현된다.

$$e_i = y_i - r_i \quad (4)$$

여기서 y_i 는 시스템의 실제적 출력값이고 r_i 는 시스템의 예상되는 출력값이다.

2. Machine Learning Part

Fig. 3에서 보듯이 $r, u, e, \Delta e$ 는 머신러닝 부분의 입력값으로 입력되고, k_p, k_i, k_d 가 출력값이 된다. 본 논문에서는 중간층이 하나로 구성된 세 층의 오류역전과 학습알고리즘을 사용하여 지도학습 방법의 머신러닝을 이용하고자 한다. 여기서 Δe 는 시스템 변화율로서 다음 식(5)에 의해 구한다.

$$\Delta e = e_i - e_{i-1} \quad (5)$$

입력층의 출력값에 의해 중간층 각 뉴런의 입력값은 다음 식 (6)에 의해 구한다.

$$input_i = \sum_{j=1}^4 w_{ij} I_j \quad (6)$$

여기서 I_j 는 4개의 입력층($1 \leq j \leq 4$)을 의미하고, w_{ij} 는 두 층간의 가중치를 의미한다. 여기서 가중치는 [0,1]의 임의의 값으로 초기화 하였다. 중간층의 출력값인 out_i 는 다음 식 (7)과 같이 중간층의 활성화함수에 의해 결정된다.

$$out_i = f(input_i) \quad (7)$$

활성화함수는 여러 가지가 있으나 오류역전과 학습알고리즘에서는 식(8)과 같이 대칭형 시그모이드 함수를 사용하였다.

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (8)$$

중간층의 출력뉴런을 모두 계산한 후 출력층의 입력은 다음 식(9)와 같이 계산된다.

$$input_i = \sum_{j=1}^4 w_{ij} out_j \quad (9)$$

출력층의 출력값은 PID의 중요한 세 가지 파라미터가 된다. 출력층의 출력뉴런들은 다음 식(10)에 의해 계산된다.

$$out_i = g(input_i) \quad (10)$$

여기서 $g(x)$ 는 입력층과 출력층 뉴런들 간의 관계를 설명하는 출력층의 활성화 함수이며, 본 논문에서는 식(11)과 같이

영이 아닌(non-negative) 시그모이드 함수를 사용하였다.

$$g(x) = \frac{1}{2}(1 + \tanh x) = \frac{e^x}{e^x + e^{-x}} \quad (11)$$

본 논문에서 제안하는 머신러닝 알고리즘은 사람의 간섭이나 개입 없이 스스로 PID 제어기의 파라미터들을 제어할 수 있으며 시스템을 단시간에 최적의 실내환경으로 변화시키기 때문에 전체적인 시스템 비용(cost 즉, 컴퓨팅시간, CPU소모량, 전기소모시간 등을 의미)을 줄일 수 있다.

그러나 머신러닝에서 Δe 값이 갑자기 커지거나 발산하는 경우가 발생하여 학습이 이루어지지 않는 단점이 있으므로 본 논문에서는 다음 식(12)와 같은 가중치 조절과정을 수행한다. 일반적으로 local minima에서 학습할 수 있는 문제점을 해결하기 위해 모멘텀 값을 추가하였다.

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} + \alpha \Delta w_{ij} \quad (12)$$

여기서 η 는 학습률이고 α 는 모멘텀 값이며, E 는 다음 식(13)과 같이 정의한다.

$$E = \frac{1}{2}(r - y)^2 = \frac{1}{2}e^2 \quad (13)$$

가중치를 갱신하는 과정은 식(14)와 같으며, 입력층과 중간층 사이에서는 식(15)와 식(16)을 이용한다. 모든 층간의 연결 값인 가중치는 $[0,1]$ 의 임의의 값으로 초기화 하였고 학습이 진행되면서 식(14)에 의해 가중치 값이 갱신된다. 여기서 δ_j 는 출력층에 대한 중간층의 오차함수를 의미한다.

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k) \quad (14)$$

$$\Delta w_{ij}(k) = \eta \delta_i out_j(k) + \alpha \Delta w_{ij}(k-1) \quad (15)$$

$$\delta_i = \nabla f(input_i) \times \sum_{j=1}^3 \delta_j w_{ij} \quad (16)$$

중간층과 출력층 사이의 가중치 w_{ji} 갱신과정은 식(17)과 식(18)을 이용한다.

$$f(x) = \frac{1 - f^2(x)}{2} \quad (17)$$

$$\delta_i = e \times \frac{\partial y}{\partial u} \times \frac{\partial u}{\partial out_j} \times \nabla g(input_i) \quad (18)$$

여기서 out_j 는 출력층으로 부터의 출력값들 즉, 세 개의 파라미터 값 k_p, k_i, k_d 를 의미하고, ∇g 는 다음 식(19)에 의해 유도된 활성화함수이다.

$$\nabla g(x) = g(x) \times (1 - \nabla g(x)) \quad (19)$$

이상에서 설명한 바와 같이 본 논문에서는 HVAC PID 제어기의 성능을 향상시키기 위해, 먼저 PID 퍼지 제어기로부터 내부의 편안한 환경(온도, 습도) 조건을 찾을 수 있고, 학습과정을 통해 PID의 세 개 파라미터 값의 최적조건을 찾을 수 있으며 이로부터 단시간 내에 최적의 실내환경제어를 할 수 있는 장점이 있다.

다음 장에서는 본 논문에서 제안한 알고리즘을 실제 구현을 통해 성능향상 정도를 비교분석 해보고자 한다.

IV. Simulation and Discussions

1. Simulation Environment

본 논문의 시뮬레이션 환경은 다음과 같다. 실외환경이 덥거나 춥고 건조한 상황에서 최적의 온도와 습기공급을 위해 학습된 ‘편안함’ 레벨에 따라 온도는 히터가 습기는 가습기가 공급하게 된다.

실험실의 구성환경은 Fig. 4와 같으며, 가로, 세로, 높이가 $2.5m \times 3.5m \times 2.5m$ 인 크기로 구성되어 있다. 여기서 실험을 위해 온도, 습도, 에어컨 상태를 인식하는 세가지 종류의 센서들을 사용하였다.

시뮬레이션을 위해 사용된 소프트웨어 라이브러리와 도구들은 다음과 같다.

- PyCharm IDE
- Python 3.5 라이브러리와 프로그래밍 도구들
- Tensorflow, matplotlib, numpy etc. Library for Python

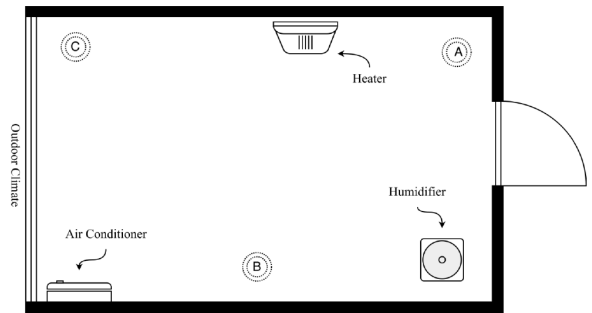


Fig. 4. Experimental Environment with Three Sensors
(A: Heating sensor, B: Humidifier sensor, C: Air conditioner sensor)

2. Simulation Comparison

기존의 PID 제어기는 사용자의 경험에 영향을 받기 때문에 사용자의 선택에 따라 다를 수 있는 단점이 있다. 그러나 대체적으로 k_p 는 $\{-0.3, 0.30\}$ 범위에 있고, k_i 는 $\{-0.06, 0.06\}$ 범위에 있으며, k_d 는 $\{-4, 4\}$ 범위에 존재한다.

Fig. 5는 수동으로 PID 제어를 했을 경우의 목표값(편안함)에 도달하는 과정을 그림으로 나타낸 것이며 이때의 세 개의 파라미터 값은 $k_p = 0.30, k_i = 0, k_d = 1.30$ 이다. 이를 위해 약 132번의 반복과정이 진행되었다.

따라서 기존 방법에 의한 시뮬레이션 결과에 의해 편안함 수준에 도달하기 위해서는 132번의 반복과정이 필요하고 이때의 최적의 파라미터 값은 $k_p = 0.30, k_i = 0, k_d = 1.30$ 임을 알 수 있다.

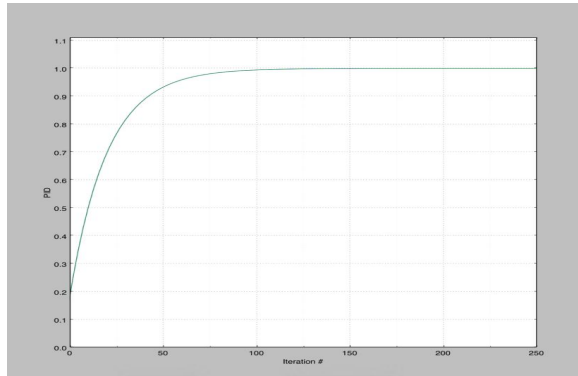


Fig. 5. Result of Traditional (manual) PID controller

한편, 본 논문에서 제안한 알고리즘에 의한 PID 제어인 경우 Fig. 6에서와 같이 매우 빠르게 목표값(편안함)에 도달하였다.

본 논문에서는 시뮬레이션을 위해 $\eta = 0.277, \alpha = 0.038$ 을 사용하였고, 머신러닝에서 뉴런의 값들은 $[-0.5, +0.5]$ 범위의 값을 사용하였다. 또한 오버슈트(overshoot) 즉, 목표 값에 비해서 최고로 오차가 커지는 부분이 얼마인가를 보는 것을 말하는데, 제안한 알고리즘에서는 오버슈트 값이 4%로 매우 작게 나타났다. 이 그림에서는 그래프의 왼쪽 윗부분에서 최초로 보이는 피크 포인트를 오버슈트라고 한다.

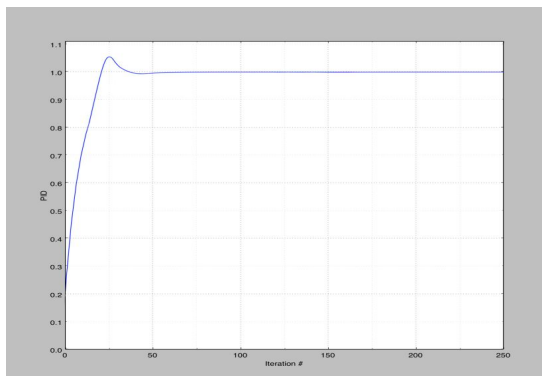


Fig. 6. Result of Our proposed PID controller

시간 경과에 따른 PID 파라미터 k_p, k_i, k_d 및 auto의 자동 조절 및 튜닝 프로세스 결과는 Fig. 7에 제시하였다. 초기 반복 값이 0일 경우 $k_p = 0.25, k_i = 0.23, k_d = 0.05$ 로 세팅되었으나 시간이 지나면서 k_p, k_i, k_d 값이 빠르고 신속하게 변경되었다. 이는 머신러닝에서 네트워크 기반으로 계속 최적의 값으로 학

습하는 과정이 수반되기 때문이다.

본 알고리즘을 사용한 결과 최종적으로 세 개의 파라미터값 k_p, k_i, k_d 는 반복이 최장 57번이 되는 시점에서 다음과 같이 결정되어 그 상태를 유지하게 된다. Fig. 7은 최적의 파라미터 값을 찾는 반복과정을 그래프로 나타낸 것이다.

$$k_p = 0.33, k_i = 0.42, k_d = 0.14$$

이러한 이유는 본 알고리즘이 기존의 PID 제어기와는 별도로 머신러닝에 의해 사람의 간섭 없이 최적의 제어를 수행하고 그 최적값을 오랫동안 유지한다는 데 있다.

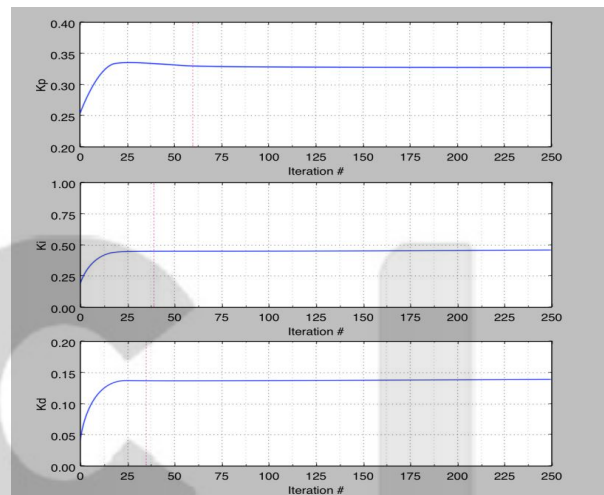


Fig. 7. Resulting Parameters k_p, k_i, k_d by Our Proposed PID Controller

세 개의 파라미터들의 변화하는 과정을 표로 나타내면 Fig. 8과 같다. 이와 같이 변화하는 이유는 본 알고리즘이 기존의 PID 제어기와는 별도로 머신러닝에 의해 사람의 간섭 없이 최적의 제어를 수행하고 최적값을 빠르게 찾아서 그 값을 유지한다는 데 있다. 이러한 결과로 인간의 ‘편안함’의 수준을 빠른 시간안에 느낄 수 있는 것이다.

Iteration #	k_p	k_i	k_d	u
1	0.23	0.23	0.05	0.21
2	0.25591818	0.27924454	0.07332636	0.271141
3	0.27811884	0.31572579	0.09060695	0.333162
4	0.30134303	0.34275177	0.10340873	0.396123
5	0.32290245	0.3627731	0.11289252	0.459198
6	0.331625425	0.37760527	0.11991829	0.5138911
7	0.33117011	0.38859321	0.1251231	0.569714286
8	0.32775436	0.39673328	0.12897892	0.619666667
9	0.325312233	0.40276359	0.13183538	0.663619048
...
56	0.33068999	0.41991038	0.14019989	0.999383254
57	0.33078212	0.41992311	0.14019999	0.999999982

Fig. 8. Process of Getting Parameters k_p, k_i, k_d by Our Proposed PID Controller

이상의 시뮬레이션을 통해 기존의 PID 제어방법과 제안하는 머신러닝을 이용한 PID 방법 간의 성능차이가 발생하였다. 이를 종합적으로 비교하면 다음 Table 4와 같다.

Table 4. Comparison between Traditional PID Controller and Our Proposed PID Controller

Indexes	Traditional PID Controller	Our Proposed Controller	Enhancement	
			Compare	Difference
No. of Iteration	132	57	2.3158 times	75
Ave. Error	0.1346	1.05344	2.5191 times	0.0812
Ave. Change Rate(%)	0.5948	1.4107	2.3716 times	0.8159
Max. Change Rate(%)	3.4267	6.3075	-	2.8808
Min. Change Rate(%)	0.0101	-0.7509	-	-0.761

Table 4에서 비교를 위해 제시한 각 지표(indexes)들을 설명하면 다음과 같다.

향상정도(enhancement)를 나타내는 부분에서 ‘비교(compare)’는 두 방법들간의 차이를 배수로 표시한 것으로서 몇배 향상되었는지를 나타내는 것이고, ‘차이(difference)’는 두 방법들 간의 절대적인 값의 차이를 나타낸 것이다.

No. of Iteration은 PID 제어기가 최적의 편안함 수준에 도달하기 위해 반복해야 하는 횟수를 말한다. 본 논문에서 제안한 알고리즘은 기존의 방법보다 약 2.3배나 빠르게 도달하였다.

평균오차(ave. error)는 실제값과 예상값 간의 평균 변동값을 말하며 기존 방법보다 약 2.5배 정도 오차가 적게 나타나서 빠르게 안정적으로 최적값을 찾아주었음을 알 수 있다.

평균 변화율(ave. change rate)은 반복횟수 사이의 출력값의 변화량의 평균값을 말한다. 좋은 알고리즘이란 반복횟수 사이에 변화량이 많지 않아서 효과적으로 학습속도를 높이는 알고리즘을 말한다. 제안하는 알고리즘은 학습속도 향상을 위해 모델변수를 사용함으로써 기존대비 2.3배 정도의 학습속도 향상을 가져왔다.

최대 변화율(max. change rate)은 반복횟수 사이 해당하는 파라미터 변화량의 최대값을 의미하고, 최소 변화율(min. change rate)은 반복횟수 사이 해당하는 파라미터 변화량의 최소값을 의미한다. 따라서 값들 간 몇배인지를 표시하는 것은 무의미하여 제외하였다.

제안하는 방법에서는 출력값이 1.0 이상인 오버슈트 값을 제어하고 학습하기 위해 음의 값으로 학습하는 방법을 사용하였다.

V. Conclusions

본 논문에서는 기존의 PID 제어기에서 세 개의 주요 파라메

터인 k_p, k_i, k_d 값을 신속하게 찾는 것이 어려운 점을 해결하고자 머신러닝 기반으로 지능적 최적값 결정이 가능한 HVAC PID 제어기의 알고리즘을 제안하였다.

기존의 PID 제어에서는 인간의 개입이나 조작이 필요하고, PID 파라미터 조절과정이 느리며, 오류 및 오버슈트 발생이 자주 일어나기 때문에 전문가가 제어기를 조정하지 않으면 환경 변화에 적응할 수 없다는 문제점이 있다.

본 알고리즘의 기존의 PID 제어기에 비해 다음과 같은 장점을 갖는다.

- 인간 개입을 필요로 하지 않음
- PID 파라미터 튜닝의 빠른 상승 속도 및 안정화 속도
- 적은 값의 오버슈트
- 적은 수준의 고정 오차
- 환경 변화에 대한 안정성 및 적응성

또한, 시간이 지연되거나 실내에 많은 사람들이 존재하여 측정이 어려워지거나 잘못 측정되는 경우 등에서도 강건하게 지능적 제어가 가능할 것이다.

본 연구는 머신러닝에 의해 가장 최적의 파라미터값을 찾아서 실내환경에 적용할 수 있는 방법을 연구한 논문이다. 그러나 실제 세계에서 실내환경에 적용하기 위해서는 기계적인 처리 즉, 실내 조절기구나 시스템과 연계되어 동작하게 되어야 하며, 이 경우 사람이 관여하거나 간섭하지 않더라도 바로 실내의 최적의 환경을 유지하게 될 것이다.

본 논문은 실내 최적환경제어를 위해 세 개의 PID 파라미터를 조정하는데 초점을 두고 연구를 진행하였다. 앞으로는 다양한 실내의 환경과 스마트 시티 환경에 폭넓게 적용할 수 있는 범용의 인공지능 기반 PID 제어기가 개발된다면 인간의 편안함 수준에 신속하고 오류없이 도달하도록 할 수 있을 것이다.

REFERENCES

- [1] Taleghani, M., Tenpierik, M., Kurvers, S., “A review into thermal comfortable in buildings”, Renewable and Sustainable Energy Reviews, Vol.26, pp.201-215, October 2013.
- [2] L. Chuan, A. Ukil, “Modeling and Validation of lectrical Load Profiling in Residential Buildings in Singapore”, IEEE Transactions on Power Systems, Vol.30, No.3, pp.2800-2809, 2015.
- [3] Sandra Rodríguez-Valenzuela, Juan A. Holgado-Terriza, José M. Gutiérrez-Guerrero and Jesús L. Muros-Cobos, “Distributed Service-Based Approach for Sensor Data Fusion in IoT Environments”, Sensors, Vol.14, No.10, pp.19200-19228, October 2014.
- [4] Karl Johan Astrom, “PID control”, Control System Design, 2002.

- [5] SO/FDIS 7730:2005, International Standard, Ergonomics of the thermal environment — Analytical determination and interpretation of thermal comfortable using calculation of the PMV and PPD indices and local thermal comfortable criteria, 2005.
- [6] Veronika Földváry, Hana Pustayová Bukovianska, Dušan Petráš. "Analysis of Energy Performance and Indoor Climate Conditions of the Slovak Housing Stock Before and After its Renovation", 6th International Building Physics Conference, Vol.78, Nov. 2015, pp.2184-2189. IBPC 2015.
- [7] Epstein Y, Moran DS. "Thermal comfortable and the heat stress indices", Industrial Health, Vol.44, pp.388-398. April 13, 2006.
- [8] Qiao, W. Z., & Mizumoto, M. "PID type fuzzy parameters adaptive method", Fuzzy sets and Systems, Vol.78, pp. 23-35, 1996.
- [9] Hung, L. C., Lin, H. P., & Chung, H. Y. "Design of self-tuning fuzzy sliding mode control for TORA system", Expert Systems with Applications, Vol.32, pp. 201-212, 2007.
- [10] Woo, Z. W., Chung, H. Y., & Lin, J. J. "A PID type fuzzy controller with self-tuning scaling factors", Fuzzy Sets and Systems, Vol.115, pp. 321-326, 2000.
- [11] B. Omarov, A. Altayeva, Zh. Suleimenov, Young Im Cho. "Design of Fuzzy Logic based Controller for Energy Efficient Operation in Smart Buildings", 2017 IEEE International Conference on Robotic Computing, April 2017.

Authors



Jee Hyun Kim received a Doctor of Computer Science from Dankook University, Korea, in 2004, her M.B.A degree in the Information Management from Dankook University in 1994, her B.S degree in Mathematics from Ewha Womans

University in 1978. Her major is Software Engineering. She is a professor in the Department of Software Engineering in Seoil University. Her research interests are Web Engineering, Big data, Quality Management, Information Retrieval etc.



Young Im Cho received her B.S., M.Sc., and Ph.D from the Department of Computer Science, Korea University, Korea, in 1988, 1990 and 1994, respectively. Professor Cho is a professor at Gachon University. Her research interest includes AI, Big data,

information retrieval, smart city etc.