International Journal of Fuzzy Logic and Intelligent Systems, vol. 2, no. 2, June 2002 pp. 146-152

Neural Fuzzy Mold Level Control for Continuous Steel Casting

Chang-Gyoon Lim*, **Yeong-Seob Kueon**** **and Yigon Kim*****

***Department of Computer Engineering, Yosu National University**
****Instrumentation & Control Research Team, Technical Research Laboratories, Pohang Iron & Steel Co., Ltd**
*****Department of Electrical Engineering, Yosu National University**

Abstract

Mold level control has been a major control task for continuous casting plants. The system involves nonlinearities such as stick-slip friction in the sliding gate, time-delay, friction force variations between molten steel and the inner wall of mold, and nozzle logging/unclogging. These complex problems should be solved to control mold level for steel cast. In this paper, we propose a neural fuzzy mold level control technique for solving these complex problems and give experiment studies to show the mold level control in continuous casting process.

Key words : Fuzzy control, Neural networks, Sliding mode control, PID control, Steel casting

Ⅰ. Introduction

In the steelmaking process, the continuous casting process converts molten steel into a solid shape of continuous slab. The continuous slab is cut into specified lengths to be supplied to the subsequent processes or direct sale. The purpose of mold level control is to decrease mold level fluctuations as much as possible since the fluctuation of mold level is one of the most significant factors deciding quality of the slab [1-2]. If mold level fluctuation occurs severely, the oscillation mark is formed onto the slab, mold powder and slag can be infiltrated into the slab, and breakouts can be occurred. The accurate control of mold level, however, is very difficult because the controlled system is nonlinear, time varying and submitted to various unexpected disturbances.

In order to solve the problem by more advanced feedback control with better disturbance rejection characteristics than conventional single-loop proportional-integral-derivative (PID) control, various methods have been developed for mold level control, such as predictive control, linear or nonlinear cascade control, repetitive learning for periodic disturbance, and disturbance observer [3-6].

Fuzzy logic control(FLC) has become an active and effective approach for mold level control. Usually it is implemented at the supervisory level for adjusting the low level conventional controller [7] or implemented in parallel with the conventional controller for dealing with large disturbances [8-9]. Attempt is also done to design a FLC for mold level control as a complete substitute for the conventional control, however, a better solution which achieves good steady-state performance as well as good robustness against disturbances is not found [8].

In this paper, we propose a neural fuzzy mold level control technique in which fuzzy rules and membership functions are prepared to overcome the above mentioned control difficulties. Section 2 of the paper describes process and modeling. Section 3 introduces the neural fuzzy mold level controller. Section 4 presents performance comparisons between the proposed controller and the conventional controller are given by simulation and implementation results. Section 5 gives the conclusion.

II. Process and Modeling Description

2.1 Problem Statements

Generally, the main problems of mold level control system were found to be the friction force variations between the inner wall of the mold and the molten steel. These are mainly from molten steel temperature and steel grade, the stick-slip friction in the sliding gate, and the clogging/erosion of the submerged entry nozzle (SEN). They are very laborious to be included in the mathematical model and controlled properly.

The dynamic system to be controlled can be modeled by *1st order ordinary differential equation*, although it possesses unmodeled dynamics, wave effects, and so on. Mold level should be controlled to within a few mm in accordance with specifications. A simple PI-D controller is not robust enough to satisfy the specifications since the dynamic system to be controlled and the operating conditions are changed during casting. As shown in Fig. 1, for example, mold level controlled by the existing PI-D controller is fluctuating as casting speed is changed.

This work was supported by the RRC program of MOST and KOSEF through the Research and Development Center for Facility Automation and Information System (FAIS) at Yosu National University.

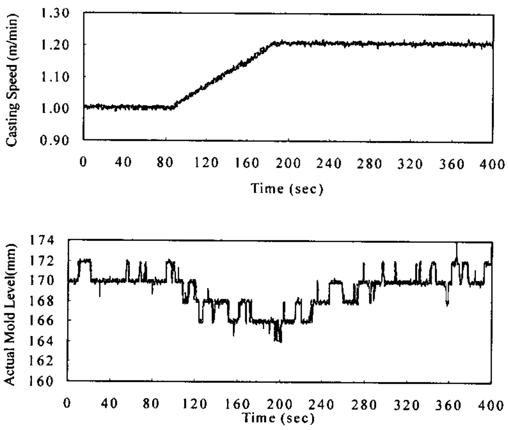

Fig. 1. Casting speed variations vs mold level fluctuations

2.2 The Continuous Casting Process

A schematic picture of the #2 continuous casting process at Pohang works of POSCO is shown in Fig. 2. The three subsystems, which make up the whole process, are the ladle, the tundish, and the mold. Molten steel arrives in a refractory lined ladle. The molten steel is first poured from the ladle into a refractory lined tundish, which works as a secondary reservoir which keeps a constant supply of molten steel as an empty ladle is exchanged for a full one. Molten steel flows from the tundish into the mold through SEN then passes through the primary and the secondary cooling zones and is cast into continuous slab [9-10]. The mold level is measured by means of radiation detectors. Its compensation is realized by the hydraulically activated sliding gate. Mold level fluctuation should be reduced because this will result in deterioration of the quality of cast slabs.

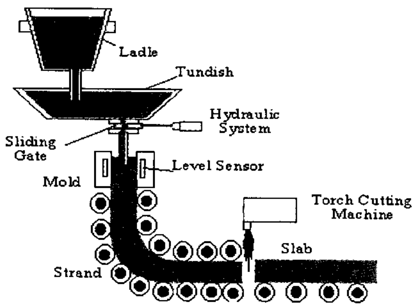

Fig. 2. A schematic picture of the continuous casting process

III. Controller Design by Neural Fuzzy Control

3.1 The statement of Control Problem

In this paper we consider systems having a state model nonlinear in the state vector and linear in the control vector. The general multivariable control problem will be described by the following system of state equation:

$$\begin{aligned} f(x,u,z)&=0 \\ y&=g(x,u) \\ u&=h(w,y) \end{aligned} \tag{1}$$

where **f**, **g**, and **h** are vector-valued functions.

A mathematical model is an nth order ordinary differential equation of the type [11].

$$\frac{d^n y^{(t)}}{dt^n} = w\left[t, y(t), \cdots\cdots, \frac{d^{n-1} y^{(t)}}{dt^{n-1}}, u(t)\right] \tag{2}$$

where t is the time parameter, u is a general nonlinear function for control, and y is the system output. The auxiliary functions are defined as following:

$$\begin{aligned} x_1(t) &= y(t) \\ x_2(t) &= y(t) \\ x_n(t) &= \frac{d^{n-1}y(t)}{dt^{n-1}} \end{aligned} \tag{3}$$

Now, the single nth-order equation Eq. (2) can be rewritten with a system of n first-order equations:

$$\begin{aligned} x_1(t) &= x_2(t) \\ x_2(t) &= x_3(t) \\ x_3(t) &= x_4(t) \\ &\mathrm{M} \\ x_{n-1}(t) &= x_{(t)} \\ x_n(t) &= w[t, x_1(t), x_2(t), \cdots\cdots, x_n(t), u(t)] \end{aligned} \tag{4}$$

The system state vector at time t is defined by:

$$x(t) = [x_1(t), x_2(t), x_3(t), \cdots\cdots, x_n(t)]^T \tag{5}$$

The dynamic system to be controlled can be modeled by 1^{st} order ordinary differential equation, although it possesses unmodeled dynamics, wave effects, and so on. So, Eq. (4) can be combined into a 1^{st} order ordinary differential equation:

$$x_1(t) = f[t, x(t), u(t)] \tag{6}$$

In this case, the output can be represented by:

$$y(t) = [1, 0, 0, \cdots\cdots, 0]x(t) \tag{7}$$

3.2 Sliding Mode Control(SMC)

SMC has received great attention because of its simplicity, robustness to various perturbations from modeling inaccuracies and disturbances, and guaranteed transient performance. To describe the basic concepts of sliding mode control, a nonlinear system represented by the state equation is considered:

$$x = f(x,t) + g(x,t)u \tag{8}$$

where the state vector of the system $x \in X$, an open set of $\mathcal{R}^n$, f,g: $\mathcal{R}^+ \blacksquare x \to \mathcal{R}^n$, and u: $x \to U \subset \mathcal{R}$ is the control input function. Then a sliding hyperplane S in the state space can be defined by:

$$S=\{x\in\Re^{n} \mid s(x,t)=0\} \tag{9}$$

where $s(x,t)\in\Re$ is called the switching function.

The main purpose of control problem is to obtain the state x for tracking a desire state x_d in the presence of model uncertainties and disturbances. So, we define a sliding surface that satisfies Eq. (9) with the tracking error, $e=x-x_d$, as:

$$s(x,t)+(\frac{d}{dt}+\lambda)^{n-1}e; \quad \lambda\geq 0 \tag{10}$$

The second order system with $n=2$ would be:

$$s(x,t)=(\frac{d}{dt}+\lambda)e=\dot{e}+\lambda e \tag{11}$$

The slope of a sliding hyperplane is an important factor to decide control performance in the sliding mode controller. Therefore, we introduce fuzzy sliding mode control, which changes the slope of a sliding hyperplane to increase system performance.

3.2 Design of Neural Fuzzy Controller

Neural fuzzy controllers are basically fuzzy inference systems that are implemented under the structure of neural networks. A n-dimensional feature vector is applied at the input to the fuzzy inference systems, composed of a set of fuzzy rules, in the training phase. When the input variables are represented by $x=(x_1,x_2,\cdots\cdots,x_n)$ and the output is expressed by y, the inference rule of fuzzy reasoning will be expressed as follows:

Rule i : If x_1 is A_{i1} **and** x_2 is A_{i2} **and** …… **and** x_n is A_{in} **Then** y is w_1 (12)

where i is the index of the rule, A_{ij}, is a fuzzy set for i-th rule and j-th linguistic variable and wi is a consequent real number.

Membership function for Neuro-fuzzy controllers is defined as Gaussian membership function. The definition of the membership function is given by Eq. (13).

$$A_j(x)=\exp\left[-\frac{\|x-\nu_j\|^2}{2\cdot\sigma_j^2}\right] \tag{13}$$

For the application of the knowledge base we need to define fuzzy inference. The membership value of the ith rule can be calculated by:

$$\mu_1=A_{ij}(x_1)\cdot tA_{i2}(x_2)\cdot A_{i3}(x_3)\cdot\cdots\cdots\cdot A_{in}(x_n) \tag{14}$$

The consequence parts of the rules are crisp values. Therefore, the final output of fuzzy reasoning corresponding to a given input is given by:

$$\lambda=\frac{\sum_{i=1}^{c}\mu_{ik}w_i}{\sum_{i=1}^{c}\mu_{ik}} \tag{15}$$

A promising approach for obtaining the benefits of both fuzzy systems and neural networks and solving their respective problems is to combine them into an integrated system. Such a system has the following properties: learning capabilities, optimization capabilities, connection structures, human-style IF-THEN rules, and the use of knowledge derived from experts. The neural fuzzy controller can express fuzzy reasoning through the connection weights between the antecedent part and the consequent part of fuzzy IF-THEN rules. So, the system can learn and adapt rules with low-level computational power for better total system performance.

The structure of neural fuzzy networks is similar to the neural networks. We apply a neural fuzzy controller to sliding mode to control mold level in continuous casting process. This model creates fuzzy rules from training data in the learning phase. The basic idea in using fuzzy C-means clustering is to define the number of rules in the system. It consists of one input layer, one hidden layer, and one output layer. Fig. 3 shows the structure of the neural fuzzy system used in the paper.

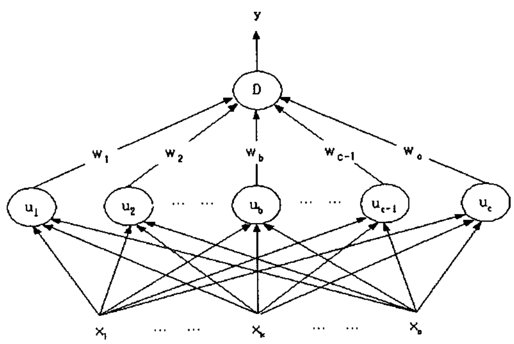

Fig. 3. The structure of neural fuzzy system

The method consists of two phases: First, they use a fuzzy C-means algorithm to decide the number of rules from a given training data set and prepare the initial parameters such as weights, mean (center), and variance (width) with respect to A_{ij} defined Eq. (13) for the system. After the first phase, the optimal number of fuzzy partitions is determined based on cluster validity. Then, they use the self-tuning algorithm to identify parameters.

They also used a Nonsymmetrical Gaussian Function Network (UGFN) instead of normal Gaussian function to identify the parameters [12]. The model takes advantage of unsupervised learning for determining the number of rules and of supervised learning for tuning the parameters of the model using training data. The numbers of rules are determined by utilizing fuzzy C-means clustering methods [13].

Once the structure of the network is identified, we need to train it to tune the parameters to generate the best solution. It can be done using the supervised method. Tuning parameters can be updated using gradient methods by minimizing the error function. A detailed algorithm of this approach can be found in [12].

IV. Experiment Results

4.1 The Process Model for Mold Level Control

As a primary step towards comprehension the control problems, a mathematical model of the process was developed. Now, we define dynamic model of the system

$$\frac{d}{dt}\left[\int_{h(t)}^{0} A_m(z)dz\right] = M_{in}(t) - M_{out}(t) \qquad (16)$$

where $A_m(h)$ is the cross sectional area of the mold, $M_{\in}(t)$ is the inflow of molten steel which is defined the effective flow area times the height of molten steel, $l(t)$, and $M_{out}(t)$ is the outflow which is defined the mold size times the casting speed [2]. The dynamics of (16) can be rewritten:

$$A_m(h) = M_{\in}(t) - M_{out}(t) = A_v c\sqrt{sgl(t)} - A_m V_{cs} \qquad (17)$$

where c is the coefficient of discharge, A_v is an effective flow area, and V_{cs} is the casting speed. Here, we assume that the outflow speed from the exit of SEN is equal to the passing speed of molten steel at the sliding gate, then $l(t)$ can be approximated as the molten steel head in the tundish.

The main consideration is how to convert effective flow area, $A_v(mm^2)$, into the linear displacement of the sliding gate $x_v(mm$, stroke). Due to the geometry of the overlapping circular orifices in the sliding gate, the exact transformation is nonlinear as shown in Eq. (18) and Fig. 4. In this paper, we used a look-up table to convert effective flow area into the stroke of the sliding gate.

$$A_v = 1600\pi + 3200\sin^{-1}\left[\frac{x_v - 120}{80}\right] + \frac{x_v - 120}{2}\sqrt{6400 - (x_v - 120)^2} \qquad (18)$$

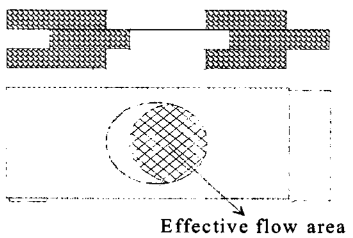

Fig. 4. Sliding gate arrangement

4.2 Step Input Response Test

As mentioned the previous section, the model consists of two phases: First, using a fuzzy c-means algorithm it decides the number of rules from a given training data set and prepares the initial parameters such as weights, mean (center), and variance (width) with respect to A_{ij} for the system. After the first phase, the optimal number of fuzzy partitions is determined based on cluster validity. Then, the network uses the self-tuning algorithm to identify parameters. Nonsymmetrical Gaussian Function Network (UGFN) is used to identify the parameters [12]. To use UGFN Eq. (13) can be written as:

$$\mu_{A_i}(x_j) = \begin{cases} \exp\left(\dfrac{-(x_j - \nu_{ij})^2}{2(\sigma^i_{jl})^2}\right) \text{if } x_j \le \nu_{ij} \\ \exp\left(\dfrac{-(x_j - \nu_{ij})^2}{2(\sigma^i_{jr})^2}\right) \text{if } x_j \le \nu_{ij} \end{cases} \quad j = 1, 2, \cdots, m \qquad (19)$$

By training the system we are given a neural fuzzy control system with the following eleven fuzzy rules. The obtained fuzzy rules are used for switching the slope of a sliding hyperplane. e and Δe are linguistic input variables for the system. The eleven fuzzy rules for the control system are generated with learning constant 0.0001 and 100 iterations. The final If-Then rules derived from the training are as follows:

Rule 1: **If** e is A_{11} **and** Δe is A_{12} **Then** y_1 is 0.028781;
Rule 2: **If** e is A_{21} **and** Δe is A_{22} **Then** y_2 is -0.616071;
Rule 3: **If** e is A_{31} **and** Δe is A_{32} **Then** y_3 is -0.401069;
Rule 4: **If** e is A_{41} **and** Δe is A_{42} **Then** y_4 is -3.243907;
Rule 5: **If** e is A_{51} **and** Δe is A_{52} **Then** y_5 is -0.173598;
Rule 6: **If** e is A_{61} **and** Δe is A_{62} **Then** y_6 is 1.959666;
Rule 7: **If** e is A_{71} **and** Δe is A_{72} **Then** y_7 is -0.483297;
Rule 8: **If** e is A_{81} **and** Δe is A_{82} **Then** y_8 is 0.234432;
Rule 9: **If** e is A_{91} **and** Δe is A_{92} **Then** y_9 is 1.661174;
Rule 10: **If** e is A_{101} **and** Δe is A_{102} **Then** y_{10} is 1.892862;
Rule 11: **If** e is A_{111} **and** Δe is A_{112} **Then** y_{11} is -0.123439;

where A_{11}, ..., A_{112} are the linguistic labels characterized by the Nonsymmetrical Gaussian Function Network. The final membership functions of a eleven-rule fuzzy controller that has about the similar performance as the nine-rule fuzzy controller are given in Appendix.

In the first experiment, the step response test is involved moving mold level set values from 160 to 170 or from 170 to 160 etc. These values are changed in every 10 seconds. Other values such as mold size, tundish weight, and casting speed were collected from the continuous caster. Fig. 5 shows the performance evaluation of neural fuzzy and conventional sliding mode controllers. We compared the proposed system with conventional sliding mode controller. The experiments show that a small quantity (100-sec) of prepared step input data could yield a significant improvement in reaching stable point.

It is quite easy to differentiate performances between two approaches in Fig. 6, which shows a detail result of experiment at intervals between 7.4 an 15 sec. in Fig. 5. A significant improvement in system behavior is also apparent. It should be noted that the neural fuzzy controller gives a better recovery time to reach stable point. The result of Fig. 6 reports that the proposed technique takes about 1.2 second to reach final sliding surface but conventional sliding mode controller requires about 3 second in the step response test. In this experiment we ignore overshoot because we only concern how fast the system come back to the stable point to improve the surface quality of the cast slabs and safety in the production of the slab caster.

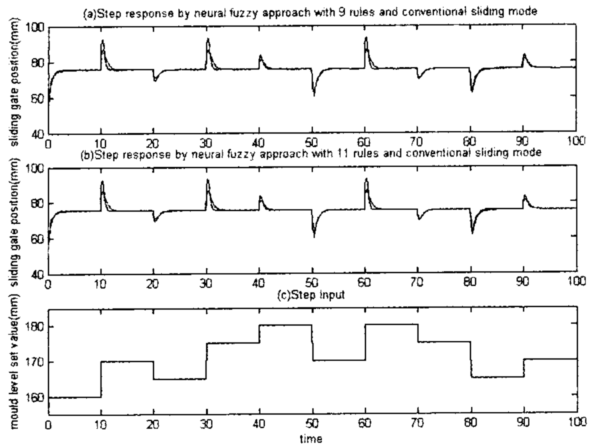

Fig. 5. Comparison result of the step response test

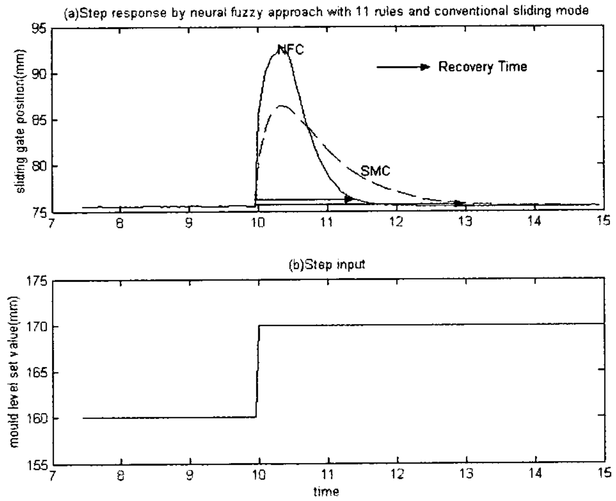

Fig. 6 . Enlargement section between 7.4 and 15 sec. in Fig. 5

4.3 Normal Operation Test

Test data were selected from POSCO #2 continuous caster to evaluate neural fuzzy and conventional sliding mode control technique. The selected controllers as shown in Fig. 7 adjust the sliding gate to control mold level. Before testing, control parameters were tuned on the computer simulation by using the data collected from the process.

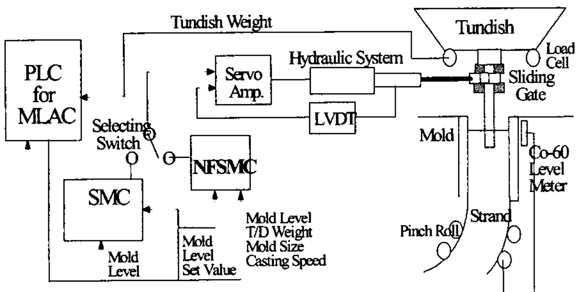

Fig. 7. Controller connections for the experiment

The comparison is given between the proposed control strategy and the conventional sliding mode control for system response against casting speed variation. Fig. 8 shows the result that the proposed approach and sliding mode controller give similar performance under casting speed. Although we just cannot simply compare neural fuzzy approach and sliding mode controller, as we have pointed out, the proposed approach has better recovery time to reach stable area.

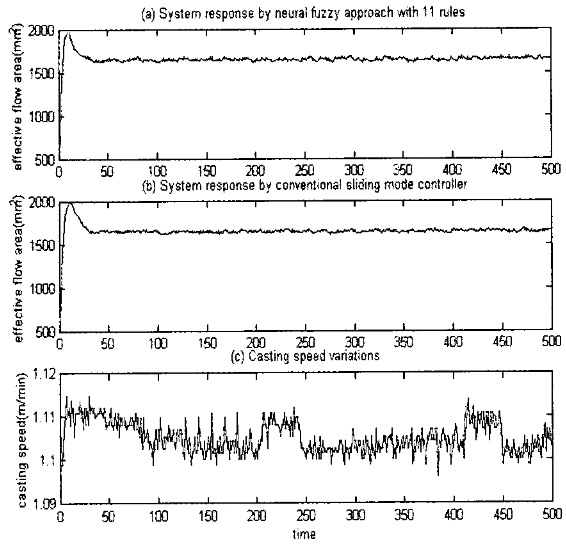

Fig. 8. System performances of neural fuzzy and sliding mode control under casting speed

V. Conclusions

We present a new neural fuzzy control approach for mold level control. It is a sliding mode type controller with a simplified scheme, in which the controller decreases mold level fluctuations as quick as possible. By employing UGFN, a new method can refine fuzzy rules and membership functions for switching the slop of hyperplane. The obtained fuzzy rules and membership functions constituted the knowledge base for mold level control in continuous casting process. It is also shown that a neural fuzzy controller can be applied for a realization of reaching stable point quickly. Even if the number of rule is small, the proposed controller reaches stable point fast compared to the conventional sliding mode controller.

If we can rephrase the above-mentioned, the proposed approach produces less damage on the slab. We come to the conclusion that the proposed approach is viewed as a useful alternative ways of dealing with these mold level control problem. Although use of the proposed approach has succeeded in step input response and normal operation, further investigations are need in terms of variations such as tundish weight and casting speed, and so on. While no further investigation has yet been done on the use of the fuzzy sliding mode control in continuous casting process, we need more work to get optimal number of fuzzy rules. As a result, proposed approach can lead to the improvement of the final cast products.

Appendix : Membership Functions

$$\mu_{A_{11}}(e)=\begin{cases}\exp\left(\dfrac{-(e-5.853318)^2}{2(0.884043)^2}\right) & \text{if } e\le 5.853318\\ \exp\left(\dfrac{-(e-5.853318)^2}{2(0.828852)^2}\right) & \text{if } e>5.853318\end{cases}$$

$$\mu_{A_{12}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-(-0.374690))^2}{2(0.005260)^2}\right) & \text{if } \Delta e\le -0.374690\\ \exp\left(\dfrac{-(\Delta e-(-0.374690))^2}{2(30617.484)^2}\right) & \text{if } \Delta e>-0.374690\end{cases}$$

$$\mu_{A_{21}}(e)=\begin{cases}\exp\left(\dfrac{-(e-(-2.849877))^2}{2(1.420781)^2}\right) & \text{if } e\le -2.849877\\ \exp\left(\dfrac{-(e-(-2.849877))^2}{2(-0.945910)^2}\right) & \text{if } e>-2.849877\end{cases}$$

$$\mu_{A_{22}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-0.089975)^2}{2(-1.137601)^2}\right) & \text{if } \Delta e\le 0.089975\\ \exp\left(\dfrac{-(\Delta e-0.089975)^2}{2(0.067973)^2}\right) & \text{if } \Delta e>0.089975\end{cases}$$

$$\mu_{A_{31}}(e)=\begin{cases}\exp\left(\dfrac{-(e-(-5.355424))^2}{2(1.249046)^2}\right) & \text{if } e\le -5.355424\\ \exp\left(\dfrac{-(e-(-5.355424))^2}{2(1.037229)^2}\right) & \text{if } e>-5.355424\end{cases}$$

$$\mu_{A_{32}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-0.143541)^2}{2(-1.973942)^2}\right) & \text{if } \Delta e\le 0.143541\\ \exp\left(\dfrac{-(\Delta e-0.143541)^2}{2(0.054439)^2}\right) & \text{if } \Delta e>0.143541\end{cases}$$

$$\mu_{A_{41}}(e)=\begin{cases}\exp\left(\dfrac{-(e-(-0.048784))^2}{2(1.943681)^2}\right) & \text{if } e\le -0.048784\\ \exp\left(\dfrac{-(e-(-0.048784))^2}{2(1.575320)^2}\right) & \text{if } e>-0.048784\end{cases}$$

$$\mu_{A_{42}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-0.064426)^2}{2(-4.419355)^2}\right) & \text{if } \Delta e\le 0.064426\\ \exp\left(\dfrac{-(\Delta e-0.064426)^2}{2(0.110076)^2}\right) & \text{if } \Delta e>0.064426\end{cases}$$

$$\mu_{A_{51}}(e)=\begin{cases}\exp\left(\dfrac{-(e-(-9.381106))^2}{2(0.754038)^2}\right) & \text{if } e\le -9.381106\\ \exp\left(\dfrac{-(e-(-9.381106))^2}{2(1.206335)^2}\right) & \text{if } e>-9.381106\end{cases}$$

$$\mu_{A_{52}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-0.190612)^2}{2(2.046102)^2}\right) & \text{if } \Delta e\le 0.190612\\ \exp\left(\dfrac{-(\Delta e-0.190612)^2}{2(0.025018)^2}\right) & \text{if } \Delta e>0.190612\end{cases}$$

$$\mu_{A_{61}}(e)=\begin{cases}\exp\left(\dfrac{-(e-(-0.047700))^2}{2(-1.661920)^2}\right) & \text{if } e\le -0.047700\\ \exp\left(\dfrac{-(e-(-0.047700))^2}{2(-1.284288)^2}\right) & \text{if } e>-0.047700\end{cases}$$

$$\mu_{A_{62}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-0.010169)^2}{2(-0.478710)^2}\right) & \text{if } \Delta e\le 0.010169\\ \exp\left(\dfrac{-(\Delta e-0.010169)^2}{2(0.279354)^2}\right) & \text{if } \Delta e>0.010169\end{cases}$$

$$\mu_{A_{71}}(e)=\begin{cases}\exp\left(\dfrac{-(e-1.942530)^2}{2(1.399365)^2}\right) & \text{if } e\le 1.942530\\ \exp\left(\dfrac{-(e-1.942530)^2}{2(1.094973)^2}\right) & \text{if } e>1.942530\end{cases}$$

$$\mu_{A_{72}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-0.186591)^2}{2(-0.739700)^2}\right) & \text{if } \Delta e\le 0.186591\\ \exp\left(\dfrac{-(\Delta e-0.186591)^2}{2(0.034745)^2}\right) & \text{if } \Delta e>0.186591\end{cases}$$

$$\mu_{A_{81}}(e)=\begin{cases}\exp\left(\dfrac{-(e-9.416042)^2}{2(0.882172)^2}\right) & \text{if } e\le 9.416042\\ \exp\left(\dfrac{-(e-9.416042)^2}{2(0.492911)^2}\right) & \text{if } e>9.416042\end{cases}$$

$$\mu_{A_{82}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-0.159880)^2}{2(-2.790712)^2}\right) & \text{if } \Delta e\le 0.159880\\ \exp\left(\dfrac{-(\Delta e-0.159880)^2}{2(-0.062377)^2}\right) & \text{if } \Delta e>0.159880\end{cases}$$

$$\mu_{A_{91}}(e)=\begin{cases}\exp\left(\dfrac{-(e-1.366906)^2}{2(-6808.264)^2}\right) & \text{if } e\le 1.366906\\ \exp\left(\dfrac{-(e-1.366906)^2}{2(4.203948)^2}\right) & \text{if } e>1.366906\end{cases}$$

$$\mu_{A_{92}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-(-0.215590))^2}{2(-3.983949)^2}\right) & \text{if } \Delta e\le -0.215590\\ \exp\left(\dfrac{-(\Delta e-(-0.215590))^2}{2(1.461605)^2}\right) & \text{if } \Delta e>-0.215590\end{cases}$$

$$\mu_{A_{101}}(e)=\begin{cases}\exp\left(\dfrac{-(e-0.183392)^2}{2(-28.06681)^2}\right) & \text{if } e\le 0.183392\\ \exp\left(\dfrac{-(e-0.183392)^2}{2(-3.919199)^2}\right) & \text{if } e>0.183392\end{cases}$$

$$\mu_{A_{102}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-(-0.140218))^2}{2(0.376912)^2}\right) & \text{if } \Delta e\le -0.140218\\ \exp\left(\dfrac{-(\Delta e-(-0.140218))^2}{2(0.169457)^2}\right) & \text{if } \Delta e> -0.140218\end{cases}$$

$$\mu_{A_{111}}(e)=\begin{cases}\exp\left(\dfrac{-(e-3.962338)^2}{2(-0.889211)^2}\right) & \text{if } e\le 3.962338\\ \exp\left(\dfrac{-(e-3.962338)^2}{2(0.934247)^2}\right) & \text{if } e> 3.962338\end{cases}$$

$$\mu_{A_{112}}(\Delta e)=\begin{cases}\exp\left(\dfrac{-(\Delta e-0.136390)^2}{2(-1.794278)^2}\right) & \text{if } \Delta e\le 0.136390\\ \exp\left(\dfrac{-(\Delta e-0.136390)^2}{2(0.068178)^2}\right) & \text{if } \Delta e> 0.136390\end{cases}$$

References

[1] G. Bocher, R. Obermann, and G. Krueger, "Slab quality improvement by means of advanced mold level control systems, *MP T International*," Vol. 16, No. 3, pp. 50-55, 1993.

[2] S. F.Graebe, G. Elsey, and G. C. Goodwin, "Accurate Control of mold Level in Continuous Bloom Casting," *Control '92*. Perth 2-4, 1992.

[3] R. De Keyser, " Improved mold-level„ control in a continuous steel casting line," *Control Engineering Practice*, Vol. 5, No. 2, pp. 231-237, 1997.

[4] S. F Graebe, G. C. Goodwin, and G. Elsley, "Control design and implementation in continuous steel casting," *IEEE Control Systems Magazine*, Vol. 15, No. 4, pp. 64-71, 1995.

[5] T. J. Manayathara, T. C. Tsao, J. Bentsman, and D. Ross, "Rejection of unknown periodic load disturbances in continuous steel casting process using learning repetitive control approach," *IEEE Transactions on control system technology*, vol. 4, No. 3, pp. 259-265, 1996.

[6] K. Asano, , T. Kaji, H. Aoki, M. Ibaraki, and S. Moriwaki, "Robust molten steel level control for continuous casting," *Proc. 35th IEEE Conf.* on Dec. and Contr., Vol. 2, pp. 1245-1250, 1996.

[7] J. Paiuk, A. Janini, M. Remorino, and O. Frola, "The automatic mold level control for a continuous casting process, practical implementation of different control algorithms," in *Proc. IFAC Automat. in Mining, Mineral,* and Metal Processing, Buenos Aires, Argentina, pp.205-208, 1989.

[8] N. Kiupel, P. M. Frank, and J. Wochnik, "Improvement of mold-level control using fuzzy logic," *Engineering Applications of Artificial Intelligence*, Vol.7, No.5, pp. 493-499, 1994.

[9] T. Hesketh, , D. J. Clements, and R. Williams, "Adaptive mold Level Control for Continuous Steel Slab Casting," *Automatica*, Vol. 29, No. 4, 1993.

[10] T. Kurokawa and T. Kondo, "Development of mold Level Control in Continuous Casting by H Control Theory," *2nd IEEE Conference on Control Applications*. September 13-16, Vancouver, B.C., 1993.

[11] M. Vidyasagar, , *Nonlinear System Analysis*, Prentice Hall, Englewood Cliffs, New Jersey, 1978.

[12] R. Li and M. Mukaidono, "A New Approach to Rule Learning Based on Fusion of Fuzzy Logic and Neural Networks," *IEICE Trans. Inf. And Syst.*, Vol. E78-D, No. 11, pp. 1509-1514, 1995.

[13] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981

Chang-Gyoon Lim

Chang-Gyoon Lim received his B.S. and M. S. degrees in Computer Science from Chosun University in 1984 and 1990, respectively, and Ph.D. degree in Computer Engineering from Wayne State University, Detroit, Michigan, in 1997. He is currently an Assistant Professor of Computer Engineering at Yosu National University. His research interests are in fuzzy logic, reinforcement learning, autonomous robot navigation, and soft computing.

Yeong-Seob Kueon

Yeong-Seob Kueon received the B.S. degree from Chosun University, the M.S. degree from the University of Oklahoma, and the Ph.D. Degree in Mechanical Engineering from Wayne State University, in 1983, 1991, and 1994, respectively. He is currently a senior researcher in Instrumentation & Control Research Group, POSLAB, POSCO, since 1994. His research interests include intelligent control, fuzzy logic control, and nonlinear control systems.

Yigon Kim

Yigon Kim received the MS degree in avionic electric engineering from Hankuk aviation university, Seoul Korea, in 1986 and 1988 respectively. He received Ph.D. degree in electrical engineering from Chonnam National university in 1993. He performed research at Tokyo Institute of Technology by research member in 1991 and at Iowa State University by visiting professor in 2000-2001. He is an associate professor in the School of ECC at Yosu National University. His interests fuzzy-Neuro Modeling and its application to diagnosis and control industrial systems.