

## *RNN Based Auto-tuning of PID Control Gains in Hot Strip Looper Controller*

Yoshihiro Abe \*

Dept. of Electrical and Electronic Engineering  
Okayama University  
3-1-1, Tsushima-Naka Okayama, 700-8530

Masami Konishi

Dept. of Electrical and Electronic Engineering  
Division of Industrial Innovation Science  
The Graduate School of Natural Science and  
Technology  
Okayama University  
3-1-1, Tsushima-Naka Okayama, 700-8530

Jun Imai

Dept. of Electrical and Electronic Engineering  
Division of Industrial Innovation Science  
The Graduate School of Natural Science and  
Technology  
Okayama University  
3-1-1, Tsushima-Naka Okayama, 700-8530

Tatsushi Nishi

Dept. of Electrical and Electronic Engineering  
Division of Industrial Innovation Science  
The Graduate School of Natural Science and  
Technology  
Okayama University  
3-1-1, Tsushima-Naka Okayama, 700-8530

(Received December 14, 2005)

In this study, auto tuning of PID control gains in hot strip looper controller is made based on RNN model. Neuro emulator is employed to model the characteristics of looper dynamics. Combining neuro emulator and RNN model, auto tuning system of PID control gains is constructed. As the inputs to RNN, plural evaluation functions which reflect individual preference of human experts. Further, Self learning mechanism is embedded to RNN model which enables adaptation to the change in rolling characteristics. Through numerical experiments, the effect of the proposed method is ascertained.

### 1 Introduction

In these years, plant control systems are being highly automated. But both the performances of plant and controller change with the passage of time, so it is necessary to tune them. This is why human experts intervene and tune the control system to improve the total plant performances. In this paper, it is targeted especially to tune looper control gains in the threading of hot strip mills. In hot strip mills, the looper height control system is set up between stands. By use of them, the rolling operation are kept to be stable. It is usual that the looper control system is governed by PID con-

troller. Usually PID gains are determined by gain table according to a variety of strip thickness and hardness, but the optimal gain also varies with time due to the deterioration of plant characteristics. Now, it is often the cases that the skillful operators tune these gains. It is necessary to tune the PID gains appropriately in order to stabilize the looper's behavior. In the past, the looper control in the steady state has been widely built up as the control technique [1]. Contrary to this, in threading condition of the looper's behavior tends to be unstable, and then human expert never fail to intervene and stabilize the looper's behavior by tuning PID gains and interposing manipulated variable of looper height control system. In general, it is difficult

\*y-abe@cntr.elec.okayama-u.ac.jp

to automatize human's action because of flexibility and diversity of his human operations not only in hot strip rolling but also in other objects. In this paper, we try to automatize tuning of PID gains by human. That is, the method of using recurrent neural networks (RNN) [2][3][4] is proposed. Because human can memorize the past tuning results whether old or new, human's tuning action depends on past tuning experiences. RNN is a simple mechanism generating its outputs depending on the past data. Firstly, neuro emulation is studied to emulate a dynamic behavior of hot strip rolling. The obtained neuro emulator is combined with PID controller to simulate the Looper height control dynamics human tuned PID gains of looper control system. Then, methods to update the RNN model by using learning algorithm will be proposed.

## 2 Looper Height Control in Hot Strip Rolling

The looper angle is determined by the difference of strip velocities between adjacent rolling stands. In the following, the mathematical model of looper control in hot strip rolling, strip thickness control and looper height control are described.

### 2.1 Mathematical Model for Hot Rolling

The fundamental equations for hot rolling process are described as follows.

$$P_i = \sigma_i \sqrt{R'_i(H_i - h_i)} Q_i \quad (1)$$

$$F_i = \frac{1}{2} \arctan \sqrt{\frac{H_i - h_i}{h_i}} - \frac{\pi}{8} \sqrt{\frac{h_i}{R'_i}} \ln \frac{h_i}{R'_i} \quad (2)$$

$$Q_i = \sqrt{\frac{h_i}{H_i - h_i}} \left( \frac{\pi}{2} \arctan \sqrt{\frac{H_i - h_i}{h_i}} - \sqrt{\frac{R'_i}{h_i}} \ln \frac{h_i}{R'_i} + \frac{1}{2} \sqrt{\frac{R'_i}{h_i}} \ln \frac{H_i}{h_i} \right) - \frac{\pi}{4} \quad (3)$$

$$R'_i = R \left( 1 + \frac{c P_i}{H_i - h_i} \right) \quad (c = 0.214 \times 10^{-3}) \quad (4)$$

$$h_i = s_i + \frac{P_i}{M} \quad (5)$$

$$\phi_i = \sqrt{\frac{h_i}{R'_i}} \tan F_i \quad (6)$$

$$f_i = \frac{1}{2} \phi_i^2 \left( \frac{2R'_i}{h_i} - 1 \right) \quad (7)$$

$$b_i = \frac{h_i}{R'_i} (1 + f_i) - 1 \quad (8)$$

where,  $P_i$  is rolling force,  $\sigma$  is flow stress,  $H_i$  is entry strip thickness,  $h_i$  is exit strip thickness,  $R_i$  is roll radius,  $s_i$  is roll gap,  $f_i$  is forward slip ratio,  $b_i$  is backward slip ratio,  $M_i$  is mill modulus, and subscript  $i$  denotes  $i^{th}$  stand. Using  $f_i$  and  $b_i$  led by solving above equations, exit strip speed  $v_{fi}$  and entry strip speed  $v_{bi}$  can be obtained as follows

$$v_{fi} = V_{Ri}(1 + f_i) \quad (9)$$

$$v_{bi} = V_{Ri}(1 + b_i) \quad (10)$$

where  $V_{Ri}$  is the rolling velocity of  $i^{th}$  mill.

### 2.2 Mathematical Model for Looper System

As shown in Fig. 1, the distance between mills is  $L = L_1 + L_2$ , exit strip speed of  $i^{th}$  mill is  $v_{fi}$ , and entry strip speed of  $(i + 1)^{th}$  mill is  $v_{bi+1}$ . Assuming  $L_1 = L_2$  and  $\theta = \theta'$ , the strip loop angle  $\theta$  can be obtained as follows

$$\sec \theta = 1 + \frac{1}{L} \int_0^t (v_{fi} - v_{bi+1}) dt \quad (11)$$

where the integral is started when the strip gets threads  $(i + 1)^{th}$  mill. The strip loop figure can be approximated as isosceles triangle. Thus, and looper angle  $\alpha$  is given by

$$\sin \alpha = \frac{1}{l} \left( \frac{L}{2} \tan \theta + \delta \right) \quad (12)$$

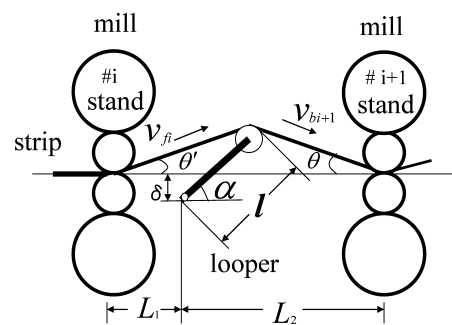


Fig. 1: Looper and interstands schematics

Variables  $v_{fi}$ ,  $v_{bi}$ ,  $\theta'$ ,  $\theta$ ,  $\alpha$ ,  $\delta$ ,  $l$ ,  $L$ ,  $L_1$ ,  $L_2$  are explained in Fig. 1

## 2.3 Mass Flow Model for Rolling velocity

Mass flow rule is used in order to determine set point of rolling velocity. Mass flow rule can be described as

$$h_1^o V_{R1}(1 + f_1) = h_2^o V_{R2}(1 + f_2) \quad (13)$$

where  $h_i^o$  is the desired value of strip thickness at  $i_{th}$  stand.

## 2.4 Strip Thickness Control Model

Gauge meter equation is written by

$$h_i = s_i + \frac{P_i}{M} \quad (14)$$

where  $h_i$ ,  $s_i$ ,  $P_i$  and  $M$  are exit strip thickness, roll gap, rolling force and mill modulus. The control scheme of Gauge Meter AGC(Automatic Gauge Control) is given by

$$\Delta h_i = h_i - h_i^o \quad (15)$$

$$\Delta s_i = -k_i \Delta h_i \quad (16)$$

where  $k_i$  is AGC gain. AGC is started to operate soon after the head end of strip arrives at the next stand.

## 3 Problem description of Looper Control System

### 3.1 PID Control of Looper Angle

PID control is described as

$$m(t) = K_p \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (17)$$

where  $m(t)$  is manipulated variable and in this case,  $e(t)$  is error and means looper deviation,  $K_p$  is proportional gain,  $T_i$  is integral time, and  $T_d$  is derivative time of PID control. In simulator, PID control is used with velocity form algorithm and lagged derivative.

$$u_n = u_{n-1} + K_p \left[ (e_n - e_{n-1}) + \frac{T_s}{T_i} e_n + \Delta \mu_n \right] \quad (18)$$

$$\eta_n = \mu_{n-1} + \Delta \mu_n \quad (19)$$

$$\Delta \mu_n = (\gamma \beta - 1) \mu_{n-1} + \beta (e_n - e_{n-1}) \quad (20)$$

$$\beta = \frac{T_d}{T_s + \gamma T_d} \quad (21)$$

where  $T_s$  is sampling time instance,  $e_n$  is error at sampling instance  $nT_s$ ,  $e_{n-1}$  is error at sampling instance  $(n-1)T_s$ . and  $\frac{1}{\gamma}$  is derivative gain.

## 3.2 Inter Stands Looper Dynamics

The problem treated in this paper consists of two stands and one looper facility between them. The looper angle is controlled by changing rolling speed of the upstream stand. In the following, construction of the problem for hot strip looper height control system will be stated.

Fig. 2 shows two stands hot strip mill system with controllers. The block diagram showing dynamic characteristics of inter stands looper are given in the lower part of Fig. 3.

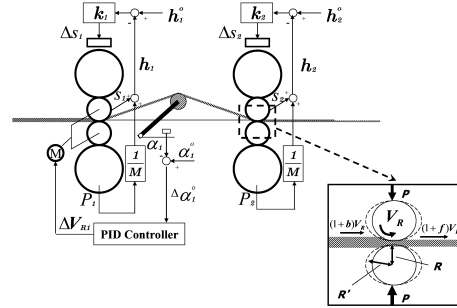


Fig. 2: Hot Strip Mill

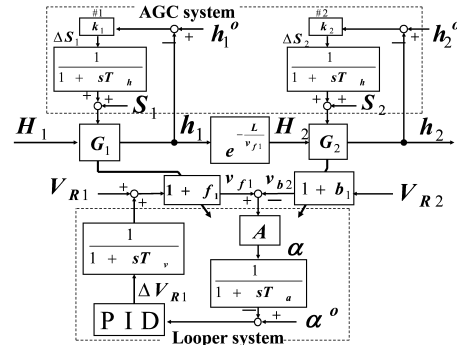


Fig. 3: Block Diagram of Hot Strip Mill with Controllers

In this paper, based on the model in Fig. 3, numerical simulations are carried out to generate looper dynamics for the construction of gain tuning system.

## 4 Auto Tuning of PID Gains by RNN

The Construction of Gain Tuning System is shown in Fig. 4. the upper part of Fig. 4 shows real plant involving hot strip looper and PID controllers. The lower part of Fig. 4 shows the proposed gain tuning mecha-

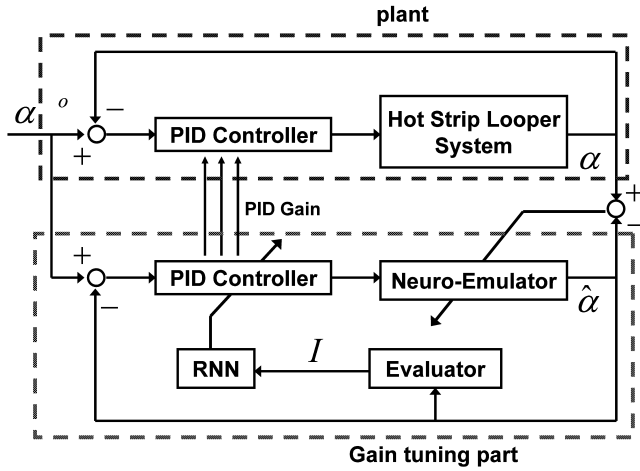


Fig. 4: Construction of Gain Tuning System

nisms. As shown here, neuro emulator is employed to generate emulated looper angle dynamics. The results of emulation are estimated in estimator whose outputs are used for the inputs to RNN model. RNN generates gains for PID controller. The parameters in neuro emulator are updated such that whose output becomes similar to the looper angle of hot rolling mills.

#### 4.1 Neuro Emulator model

The structure of neuro emulator model is shown in Fig. 5.

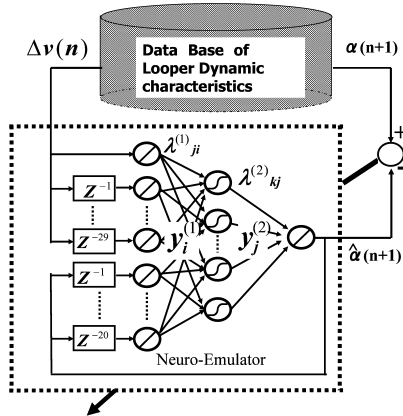


Fig. 5: Structure of Neuro Emulation model

The inputs to a neuro emulator are time series data of looper inputs, deviation of rolling speed, and past Looper angles. These data are stored in data base of looper dynamic characteristics and used for identification and traing of connecting weights in neuro emulator.

The methods of traing connecting weights in neuro emulator,  $\lambda_{kj}^{(2)}$  and  $\lambda_{ji}^{(1)}$  and descibed in the following equations.

$$\lambda_{ji}^{(1)} = \lambda_{ji}^{(1)} + \Delta\lambda_{ji}^{(1)} \quad (22)$$

$$\lambda_{kj}^{(2)} = \lambda_{kj}^{(2)} + \Delta\lambda_{kj}^{(2)} \quad (23)$$

$$\Delta\lambda_{kj}^{(2)} = -\eta_1 \frac{\partial E_1}{\partial \lambda_{kj}^{(2)}} \quad (24)$$

$$\Delta\lambda_{ji}^{(1)} = -\eta_1 \frac{\partial E_1}{\partial \lambda_{ji}^{(1)}} \quad (25)$$

$$\frac{\partial E_1}{\partial \lambda_{kj}^{(2)}} = (\hat{\alpha}[N] - \alpha[N])y_j^{(2)}[N] \quad (26)$$

$$\frac{\partial E_1}{\partial \lambda_{ji}^{(1)}} = (\hat{\alpha}[N] - \alpha[N])\lambda_{kj}^{(2)}[N](1 - y_j^{(2)}[N])y_j^{(2)}[N]y_i^{(1)}[N] \quad (27)$$

$$E_1 = \frac{1}{2}(\hat{\alpha}[n] - \alpha[n])^2 \quad (28)$$

when,  $E_1$  is difference between  $\alpha$  and  $\hat{\alpha}$ ,  $\eta_1$  is gain of traing,  $y_i^{(1)}$  is the output of  $i_{th}$  neuron in the input layer,  $y_j^{(2)}$  is the hidden of  $j_{th}$  neuron in the output layer. Thus, the looper dynamics can be represented by neuro emulator. As show in Fig. 6, looper dynamics can be replaced by neuro emulator. In the simulation for gaintuning, neuro emulator is connected to looper controller as show in Fig. 7.

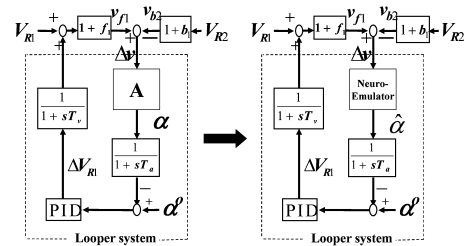


Fig. 6: Replacement of looper dynamics by Neuro Emulator

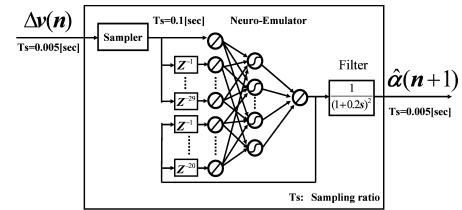


Fig. 7: Connecting method of neuro emulator with controller

## 4.2 RNN Model

The structure of RNN model is shown in Fig. 8. The network has three layers. The first layer is the input layer, which consist of two parts. The one is the plan layer, which represents input evaluation value  $I_i(i; 1 \sim 5)$  respectively.

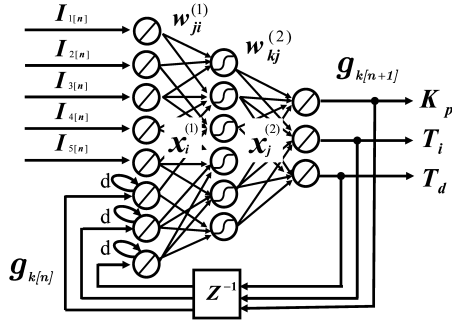


Fig. 8: Structure of RNN Model

The other is the context layer at number of rolling of  $N'$ , which is expressed by

$$c_k = \sum_{m=1}^{N'} d^{N'-m} g_k[m] \quad ; k = 1 \sim 3 \quad (29)$$

$$g_1 : K_p, g_2 : T_i, g_3 : T_d \quad (30)$$

where  $d$  is a positive constant and less than 1, which means forgetting effect and  $N'$  is the rolling number indicating  $N'_{th}$  preceding instant. As shown in Fig. 8, the context layer is the layer that the past output data of RNN is stored.

Then the output of input layer is

$$x_i^{(1)} = \begin{cases} I_i & \text{if } 1 \leq i \leq 5 \\ c_{i-5} & \text{if } 6 \leq i \leq 8 \end{cases} \quad (31)$$

The second layer is the hidden layer, which can be expressed as

$$s_j = \sum_{i=1}^6 w_{ji}^{(1)} x_i^{(1)} \quad ; j = 1 \sim 6 \quad (32)$$

$$x_0^{(1)} = -1 \quad (33)$$

$$x_j^{(2)} = \frac{1}{1 + e^{-s_j}} \quad ; j = 1 \sim 6 \quad (34)$$

where  $w_{ji}^{(1)}$  is the weight for the connection between  $i_{th}$  neuron in the input layer and  $j_{th}$  neuron in the hidden layer.

The third layer is the output layer, which is described as follows.

$$s_k = \sum_{j=0}^3 w_{kj}^{(2)} x_j^{(2)} \quad ; k = 1 \sim 3 \quad (35)$$

$$x_0^{(2)} = -1 \quad (36)$$

$$g_k = s_k \quad ; k = 1 \sim 3 \quad (37)$$

where  $w_{kj}^{(2)}$  is the weight for the connection between  $j_{th}$  neuron in the hidden layer and  $k_{th}$  neuron in the output layer.

Control performances are evaluated quantitatively by using evaluation equations (38) to (42).

$$I_1 = \int_{T_0}^{T_1} \left| \frac{\alpha}{\alpha^o} - 1 \right| dt \quad (38)$$

$$I_2 = \int_{T_1}^T \left| \frac{\alpha}{\alpha^o} - 1 \right| dt, \quad (39)$$

$$I_3 = \frac{1}{T - T_1} \int_{T_1}^T t \left| \frac{\alpha}{\alpha^o} - 1 \right| dt \quad (40)$$

$$I_4 = K \frac{\alpha_{over} - \alpha_{under}}{\alpha^o} \quad K = 10 \quad (41)$$

$$I_5 = \frac{t}{T - T_1} \quad t : \text{settling time} \quad (42)$$

Evaluation value  $I_i(i; 1 \sim 5)$  in eqs.(38) to (42) are shown in Fig. 9.

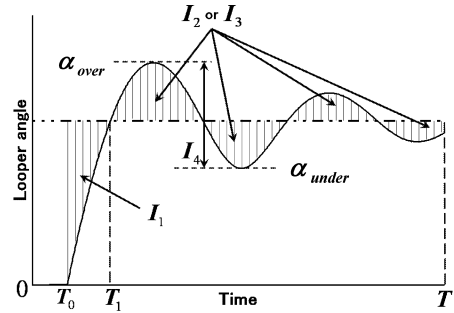


Fig. 9: Evaluation value

In the learning process of RNN model, weight parameters are optimised such that each evaluation value approaches to its expected value. In the learning, Sum of the differences of evaluation values  $I$ s from their expected values  $I^o$ s.

### 4.2.1 Method of PID Gains tuning by RNN

The structure of RNN model is updated referring differences of evaluation values with their aimed values.

Details are described in the following.

$$w_{ji}^{(1)} = w_{ji}^{(1)} + \Delta w_{ji}^{(1)} \quad (43)$$

$$w_{kj}^{(2)} = w_{kj}^{(2)} + \Delta w_{kj}^{(2)} \quad (44)$$

$$\Delta w_{kj}^{(2)} = -\eta_2 \frac{\partial E_2}{\partial w_{kj}^{(2)}} \quad (45)$$

$$\Delta w_{ji}^{(1)} = -\eta_2 \frac{\partial E_2}{\partial w_{ji}^{(1)}} \quad (46)$$

$$\frac{\partial E_2}{\partial w_{kj}^{(2)}} = \sum_{m=1}^5 (I_m - I_m^o) \frac{\partial I_m}{\partial g_k} x_j^{(2)} \quad (47)$$

$$\frac{\partial E_2}{\partial w_{ji}^{(1)}} = \sum_{k=1}^3 \left( \sum_{m=1}^5 (I_m - I_m^o) \frac{\partial I_m}{\partial g_k} \right) w_{kj}^{(2)} (1 - x_j^{(2)}) x_j^{(2)} x_i^{(1)} \quad (48)$$

where,  $E_2$  is defined by

$$E_2 = \frac{1}{2} \sum_{m=1}^5 (I_m - I_m^o)^2 \quad (49)$$

$\eta_2$  is gain of traing.

### 4.3 Training of RNN Model

The RNN model is trained by the back propagation algorithm (BP). BP algorithm is described as follows.

#### step1 ; initialization

The end condition of learning is determined, and initial value of weight parameters  $w_{ji}^{(1)}$ s and  $w_{kj}^{(2)}$ s are determined by generating random number between 0 and 1.

#### step2 ; input and preprocessing

The evaluation values,  $I_i[N'](i; 1 \sim 5)$  at  $N'_{th}$  rolling are input to the plan layer and PID gains at previous rolling number of rolling  $N'$  are input to the context layer. Where input to the context layer is 0 at  $N' = 1$ .

#### step3 ; forward propagation

By Eqs. (29) to (37), the output  $x_j^{(2)}$  of hidden layer and the output  $g_k^{(3)}[N' + 1]$  of output layer are calculated.

#### step4 ; back propagation

Eqs.(43) to (48) are used in back propagation to update the RNN model.

## 5 Numerical Experiments

According to iteration progress, PID gains simultaneously approach to certain convergence values. These convergent behaviors are shown in Fig. 10. As shown here, PID gains  $K_p$ ,  $T_i$  and  $T_d$  are tuned to their optimal values during iterations.

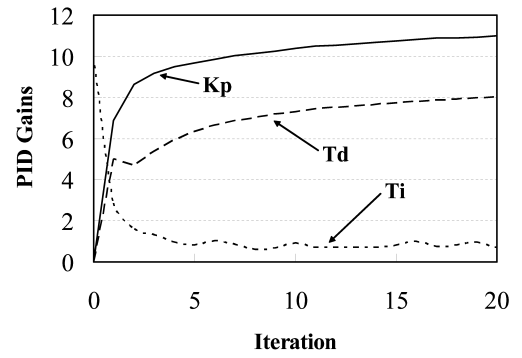


Fig. 10: Variation of PID gains during tuning

Fig. 10 shows offline gain tuning results. Variations of looper angle during tuning are shown in Fig. 11. The looper angles shown here are calculated by neuro emulator. As shown here, looper behavior becomes preferable during iterations.

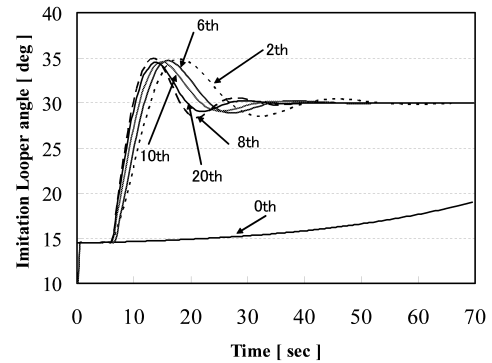


Fig. 11: Variation of Looper Angle( $\hat{\alpha}$ ) during tuning

At the same time, estimation values reach to their objective values as shown in Fig. 12. That is, by the effect of self learning of RNN model, optimizations of combining weights of RNN model are considered to be made.

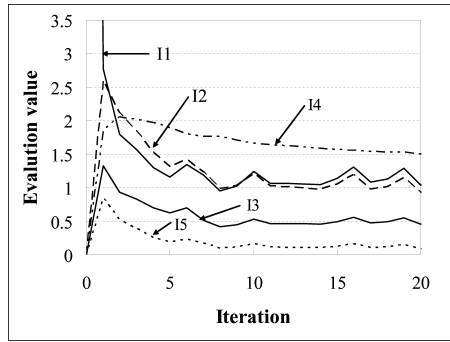


Fig. 12: Variation of Evaluation value during tuning

Fig. 13 shows simulated looper dynamics for each PID gains obtained by RNN model comparing emulated result in Fig. 11. As shown here, with excess of tuning, improvement of control performance is attained. Apparently, simulated results show the effectiveness of the proposed gain tuning method by RNN model.

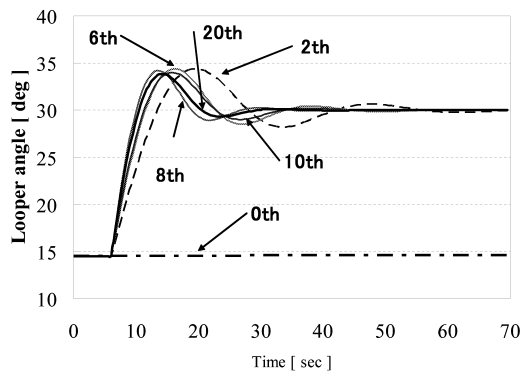


Fig. 13: Variation of Looper Angle by tuning( $\alpha^o = [30deg]$ )

ing for other control system.

**Bibliography**

- [1] S. H.Asada et al, "Adaptive and Robust Control-Method with Estimation of Rolling Characteristics for Looper Angle Control at Hot Strip Mill" , ISIJ International, Vol.43, No.3(2003), pp358-365
- [2] I.M.Mujtaba et al, , "Application of Neural Network to Generation of Robot Arm Trajectory", AROB9th, pp.380-383,2003
- [3] Lakhmi C.Jain et al, "INDUSTRIAL APPLICATIONS of NEURAL NETWORKS" , The CRC Press, 1999
- [4] Cihan H. Dagli, "ARTIFICIAL NEURAL NETWORKS for Intelligent Manufacturing" , Chapman Hall, 1994
- [5] S. Imajo et al. , "Human Model for Gain Tuning of Looper Control in Hot Strip Rolling", tetsu-to-Hagane,vol.90,No. 11,pp.933-940, 2004

**6 Conclusion**

In this paper, auto gain tunig method of PID controller for hot strip looper height control is proposed. Neuro emulator is newly employed to emulate the dynamics of looper height dynamics during rolling. RNN with self learning function is used to generate PID gains. As for the inputs to RNN model, evaluation functions of control performances are adopted which enables reflection of the human preferences. Throught numerical experiments, the applicability of the proposed method is confirmed. Combination of a neuro emulator and RNN is consider to be powerful tool of PID gain tun-