IEEE International Conference on Control Applications, October 4-6, 2006, Munich, Germany

Modeling of Reheating-Furnace Dynamics Using Neural Network Based on Improved Sequential-Learning Algorithm

Yingxin Liao, Min Wu and Jin-Hua She

***Abstract*— In order to model the dynamics of a billet reheating furnace, a multi-input multi-output radial-basis-function neural network is constructed based on an improved sequential-learning algorithm. The algorithm employs an improved growing-and-pruning algorithm based on the concept of the significance of hidden neurons, and an extended Kalman filter improves the learning accuracy. Verification results show that the model thus obtained accurately predicts the temperatures of the various zones of the furnace.**

I. Introduction

A billet reheating furnace is a key apparatus in the steel rolling process, and is also the apparatus that consumes the most energy. The main features of the furnace dynamics are that they are nonlinear and time-varying and have distributed parameters. The uniformity of the temperature of billets directly affects the quality and quantity of steel products, the amount of energy consumed, and the lifetime of the rolling mill. Due to the soaring cost of raw materials and energy, and increasingly stringent requirements for product quality, steel companies are carrying out a great deal of research on the modeling and model-based control of reheating furnaces [1]–[3]. A common way of modeling reheating-furnace dynamics is to divide the furnace into zones and then to construct a mathematical model for each one. Since control is usually implemented independently in each zone, without taking the coupling between zones into account, the control performance is not satisfactory [1].

Since radial-basis-function (RBF) neural networks (NNs) have a topologically simple structure and can reveal the learning process in an explicit manner [4], they are widely used in the control of complex industrial processes [5]–[8]. Back-propagation (BP) learning is a common technique for estimating weights for connecting hidden and output neurons, and is based on the linear least-mean-square (LMS) method. It is a batch-learning method and has the drawbacks not only of being time-consuming, but also of producing an excessive number of hidden neurons [9]. In order to reduce the computational load, Platt presented a sequential-learning scheme, which adds hidden neurons to the network based on the novelty of new data [10]. Later, Kadirkamanathan and Niranjan presented an extended Kalman filter (EKF) to replace the LMS method of adjusting network parameters [11]. Their method provides better learning precision and produces a compact network. However, there is a serious drawback in Platt's and Kadirkamanathan and Niranjan's methods in that, once a hidden neuron is created, it cannot later be removed. Yingwei *et al.* presented a node-removing method to solve the problem [12], but the selection of the neurons to be pruned requires a great deal of computation. In [13], Huang *et al.* introduced a concept called *significance* for hidden neurons based on their statistical characteristics, and used it in a learning algorithm to build a parsimonious network. Their scheme has the characteristics of rapid learning, producing a compact network, and good overall performance.

This study employed an RBF NN to construct a dynamic model of a reheating furnace. A sequential-learning algorithm that makes use of the concept of significance is employed in the modeling process. Unlike existing models, the new one is a multi-input multi-output (MIMO) model, and takes the coupling of zones into account. The model yields accurate temperature estimates and thus is suitable for temperature control. It is a promising way of improving the temperature control performance of a reheating furnace.

In this paper, first, a model for predicting the temperature distribution of the furnace is constructed using an RBF NN. Then, a growing-and-pruning (GAP) algorithm is introduced to simplify the network based on the concept of the significance of hidden neurons, and an EKF is used to adjust the parameters of the neurons to improve the learning precision. A strategy of nearest-neuron regulation is employed to identify the hidden neurons to be pruned. The GAP algorithm in this paper not only takes less time to calculate parameters, but also provides more rapid learning, than the GAP algorithms in common use these days. Verification results demonstrate the validity of the model.

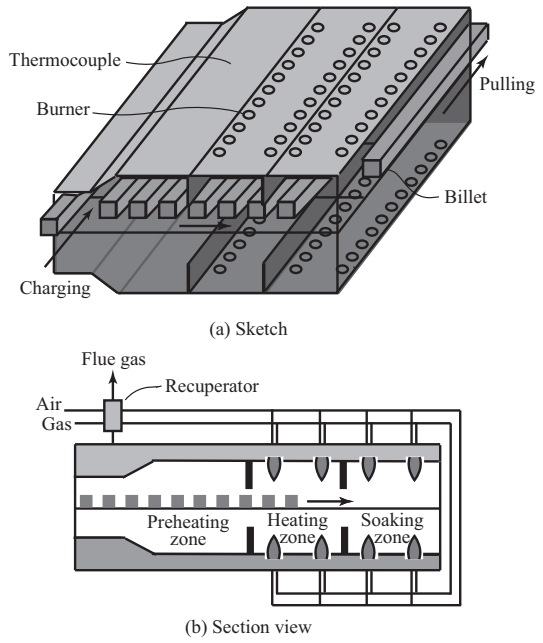
II. Model of Furnace Temperature

The object to be modeled is a walking-beam reheating furnace, which is used to reheat billets in a steelworks. The furnace is divided into three zones (Fig. 1): preheating, heating, and soaking. The preheating zone is relatively long, enabling the billets to be thoroughly preheated by

The work was supported in part by the Teaching and Research Award Program for Outstanding Young Teachers in Higher Education Institutions of the Ministry of Education, P. R. China; the Hunan provincial "Science and Technology Plan" project under Grant No. 04FJ3029; and the Grant-in-Aid for Scientific Research (JSPS Postdoctoral Fellowships for Overseas Researchers), Japan Society for the Promotion of Science, under Grant No. 17-05780.

Y. Liao and M. Wu are with the School of Information Science and Engineering, Central South University, Changsha, 410083 China. Y. Liao is also with the School of Electron and Information Engineering, Central South Forestry University, Changsha, 410004 China. yxliao@edu.21cn.com, min@mail.csu.edu.cn

J.-H. She is with the School of Bionics, Tokyo University of Technology, 1404-1 Katakura, Hachioji, Tokyo 192-0982, Japan. she@cc.teu.ac.jp



(a) Sketch

(b) Section view

Fig. 1. Reheating furnace.

the high-temperature flue gas. This gives the gas a high heating efficiency. So, only the heating and soaking zones are heated with a gas-air mixture. Nozzles at the top and bottom of the furnace send the gas mixture into these zones. The temperature of the preheating zone is monitored, but not controlled; while the temperatures of the heating and soaking zones are controlled. The heating zone is further divided into two subzones (upper, lower); and the soaking zone is divided into four (upper left, upper right, lower left, lower right). The subzones are strongly coupled in that the temperature of each subzone is greatly influenced by the temperatures of the others.

Due to the complexity of the heating process and the dynamic changes that occur in it, it is difficult to obtain a rigorous mathematical model for the process through an analysis of the mechanism(s) involved.

An RBF NN is a type of feedforward network, and can be used to obtain a nonlinear mapping [14]. So, we decided to construct an RBF NN as a model for predicting temperatures within the furnace. The structure is shown in Fig. 2.

The network contains three layers: input, hidden, and output. The input signal is fed into the input neurons and propagates to the hidden neurons. The hidden layer uses a Gaussian function to produce its output. It performs a nonlinear mapping from the vector $x := [x_1,\ x_2, \cdots, x_l]^T$ to the vector $\phi := [\phi_1,\ \phi_2, \cdots, \phi_m]^T$:

$$\phi_k = \exp\left(-\frac{\|x-\mu_k\|^2}{\sigma_k^2}\right), \quad k = 1, \cdots, m. \tag{1}$$

And the output layer performs a linear mapping from ϕ to

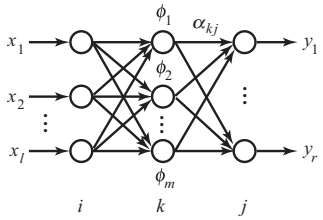


Fig. 2. Structure of RBF NN.

the output vector $y := [y_1,\ y_2, \cdots, y_r]^T$:

$$y_j = \sum_{k=1}^{m} \alpha_{kj}\phi_k, \quad j = 1, \cdots, r. \tag{2}$$

In (1) and (2), μ_k is the center vector; σ_k is the width of the hidden neuron k; and α_{kj} is the weight connecting k to the output neuron j for $k = 1, \cdots, m$ and $j = 1, \cdots, r$. $\|\cdot\|$ is the Euclidean norm.

The air/gas ratio usually remains within a narrow range, and is commonly taken to be constant. For simplicity, we do the same. Since the heating process has a time lag and is dynamic, the inputs of the network are chosen to be the delayed gas flux and temperature distributions in the temperature control zones; and the outputs are the estimated temperatures in those zones.

III. Sequential-Learning Algorithm

A practical model of furnace temperature requires on-line learning and real-time adjustment. On the other hand, since a BP learning algorithm produces a large number of hidden neurons, it is difficult (if not impossible) to use it to achieve real-time adjustment of the parameters. In this paper, a GAP algorithm and an EKF-based learning scheme solve the problem. The GAP algorithm utilizes the concept of the significance of hidden neurons. And an EKF is used to train the neural network because it provides faster convergence than gradient-based algorithms [15].

A. Definition and Estimation of Significance

The *significance*, $E_{sig}(k)$, of the hidden neuron k is defined to be its statistical contribution to the overall output of the RBF NN:

$$\begin{aligned} E_{sig}(k) &= \|\alpha_k\|_1 \int_x \phi_k(x)p(x)dx \\ &= \|\alpha_k\|_1 \int_x \exp\left(-\frac{\|x-\mu_k\|^2}{\sigma_k^2}\right) p(x)dx, \\ & \qquad k = 1, \cdots, m, \end{aligned} \tag{3}$$

where α_k is the weighting vector connecting the k-th hidden neuron to the overall output; $\phi_k(x)$ is the response of the k-th hidden neuron to the input vector x for $k = 1, \cdots, m$; $p(x)$ is the probability density function for x; and $\|\cdot\|_1$ is the 1-norm.

(3) involves an integration of the probability density function, $p(x)$, over the sampling range. This can be done analytically for some simple but commonly used $p(x)$, e.g.,

uniform, normal, exponential, and Rayleigh functions [13]. After analyzing sample data from the furnace, we selected the probability density function to be the normal sampling distribution, and $E_{sig}(k)$ to be the sum of the statistical contributions of neuron k (1-norm) to the temperatures of all the zones for $k = 1, \cdots, m$.

B. Growing-and-Pruning Criteria

The following growing-and-pruning criteria are used in this study.

1) Growing criterion: The learning process involves the allocation of new hidden neurons as well as the adjustment of network parameters. The RBF NN begins with no hidden neurons. As inputs are received sequentially during training, some of them may initiate new hidden neurons based on the following growing criterion.

Assume that n observations have been made, the input and output for the latest observation are $x(n)$ and $y(n)$, respectively, and $\mu_m(n)$ is the center of the hidden neuron nearest $x(n)$ in the sense of the Euclidean distance. Also, let $e_{\min}$ be the expected approximation accuracy of the output and ϵ be a threshold suitable for the Euclidean distance between $x(n)$ and $\mu_m(n)$. For the latest observation,

$$e(n) = y(n) - f(x(n)). \tag{4}$$

Two conditions must be satisfied before a new neuron (the $(k+1)$-th) is added. First,

$$\|x(n) - \mu_m(n)\| > \epsilon, \tag{5}$$

where ϵ is a positive number. Second, the significance of the new hidden neuron must satisfy

$$E_{sig}(k+1) = \|\alpha_{k+1}\|_1 \int_x \exp\left(-\frac{\|x - \mu_{k+1}\|^2}{\sigma_{k+1}^2}\right) p(x)dx > e_{\min}, \tag{6}$$

where

$$\begin{aligned} \alpha_{k+1} &:= e(n), \\ \mu_{k+1} &:= x(n), \\ \sigma_{k+1} &:= \rho\|x(n) - \mu_m(n)\|, \end{aligned}$$

and ρ (> 0) is an overlap factor that determines the overlap of the responses of the hidden neurons in the input space. If these conditions are met, then the $(k+1)$-th neuron is added to the network.

(5) ensures that a new neuron is added only under the condition that the input data is sufficiently far from existing neurons. And (6) ensures that the newly added neuron significantly improves the learning accuracy (its significance is larger than $e_{\min}$).

2) Pruning criterion: If a neuron has a significance of less than $e_{\min}$, then its contribution to the network is too small and it should be removed. Otherwise, it should be retained.

Considering the above conditions, one might think that, after learning from each observation, it would be necessary to check the significance of all the neurons for possible pruning, which is a computationally intensive task. However, a close examination of the network reveals that only the nearest neuron in the last learning step might have become insignificant and needs to be checked for pruning. So, there is no need to compute the significance of all the hidden neurons in the network. This keeps the computational load small, as explained below.

C. Speeding Up of Parameter Adjustment and Neuron Pruning

The parameters of the neurons need to be adjusted and the neurons need to be checked for possible pruning to improve the learning accuracy and to keep the network compact. Note that only the parameters of the neuron nearest the latest input might need to be adjusted when no new neuron is added, and also that only the nearest neuron in the last learning step needs to be checked for pruning. So, for each sampling, we check only the nearest neuron for adjustment and pruning. Since only a *single* neuron is checked and/or adjusted, learning is very rapid. This ensures that the network can be used for real-time temperature estimation.

For the sampling time n, the parameter vector of the nearest neuron is

$$w_m(n) = [\,\alpha_m^T(n) \quad \mu_m^T(n) \quad \sigma_m(n)\,]^T, \tag{7}$$

where $\alpha_m(n)$ is the coefficient vector of the linear mapping from the nearest neuron to the output, defined as in (2), and $\sigma_m(n)$ is the width of the nearest neuron. If the latest observation, $(x(n), y(n))$, does not meet the criterion for adding a new hidden neuron, the parameters of the nearest neuron are adjusted using the EKF method as follows:

$$\begin{cases} w_m(n) = w_m(n-1) + K(n)e(n), \\ K(n) = P(n-1)A(n)\left[A^T(n)P(n-1)A(n) + R(n)\right]^{-1}, \\ P(n) = \left[I - K(n)A^T(n)\right]P(n-1) + QI, \\ A^T(n) = [I \quad \phi_m(x(n))I \\ \qquad\qquad \phi_m(x(n))\dfrac{2\alpha_m(n)}{\sigma_m^2(n)}(x(n) - \mu_m(n))^T \\ \qquad\qquad \phi_m(x(n))\dfrac{2\alpha_m(n)}{\sigma_m^3(n)}\|x(n) - \mu_m(n)\|^2\Big], \end{cases} \tag{8}$$

where $e(n)$ is defined in (4); $K(n)$ is the gain matrix of the EKF; $P(n)$ is the error covariance matrix; $A(n)$ is the gradient matrix; $R(n)$ is the variance of the measurement noise; and $\phi_m(x(n))$ is the nonlinear mapping from $x(n)$ to the nearest neuron, for which the entries are defined in (1). All these values are for the latest sampling time, n. Q is a scalar that determines the number of allowable random steps in the direction of the gradient vector.

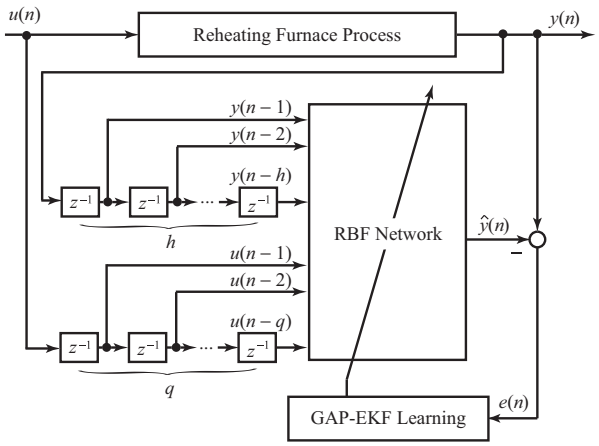


Fig. 3. Configuration of dynamic RBF model.

IV. Dynamic RBF-Based Model

Models for predicting the various temperatures of a reheating furnace are usually based on the principle of the conservation of energy [1]. But the precision of the model is low, and its real-time performance is also unsatisfactory due to the large computational load. In this paper, we employ an RBF NN to model the heating process. However, a standard RBF NN permits only static mapping; but since the process is dynamic, it is necessary to construct a dynamic NN to describe it. This can be accomplished by feeding the delayed output of the network back into the input. That is, the output of the network is produced as follows:

$$\begin{aligned} &\hat{y}(n) = G(x(n)), \\ &x(n) = [y(n-1) \ \cdots \ y(n-h) \ u(n-1) \ \cdots \ u(n-q)]^T, \\ &\qquad\qquad\qquad\qquad\qquad\qquad n = 0, 1, \cdots, \end{aligned} \tag{9}$$

where $G(\cdot)$ is the mapping generated by the RBF NN; $x(n)$ ($n = 0, 1, \cdots$) is the input of the network; and q and h are the numbers of delay steps of the input and output, respectively. This dynamic RBF NN is used to simulate the nonlinear dynamic heating process of the furnace. Note that $u(n) = 0$ and $y(n) = 0$ for $n < 0$.

(9) is a forward predictor. It is important for modeling the continuous rise and fall in the temperature of the furnace, and is used to predict the temperature at the next sampling time. Fig. 3 shows the configuration of the dynamic RBF model. In the figure, the control vector $u(n)$ ($n = 0, 1, \cdots$) is the gas flux and the output vector $y(n)$ ($n = 0, 1, \cdots$) gives the temperatures of the different zones of the furnace.

V. Verification and Discussion

The prediction of the temperatures of the zones of the furnace is the foundation of temperature optimization and control, and the prediction accuracy determines whether or not the control results are satisfactory. This section presents the results of verification tests based on actual runs that show the modeling method described above to be very effective. The method was used to build an RBF NN to predict the temperatures of the heating and soaking zones of a walking-beam reheating furnace. The results of actual runs (sampling period: 5 s) were used to construct the training and testing data sets. As for the number of delay steps for the output and control input, since fewer steps yield a model with a lower order, it is desirable to use as few steps as possible. Adjustment through trial and error showed that a one-step delay for both the output and control input produced a model with sufficient accuracy for practical use. So,

$$h = 1, \ q = 1 \tag{10}$$

was used in the simulation. The model initially creates one hidden node based on the first observation. Then, learning based on the improved sequential-learning algorithm explained in the previous section is used to make adjustments. Regarding the choice of $e_{\min}$ and ϵ, smaller values yield a larger number of hidden nodes. However, since only the parameters of the nearest node are adjusted, if the number of hidden nodes becomes large, the precision of the model may be degraded. So, it is desirable to use large values for $e_{\min}$ and ϵ. Thus, the parameters of the GAP algorithm were chosen to be:

$$e_{\min} = 80, \ \epsilon = 40, \ \rho = 0.4. \tag{11}$$

The verification results in Figs. 4-6 show that the error in the estimated temperature of each subzone is always less than 5.0°C, except for two sharp peaks at 345 and 1265 seconds. The unbiased deviation in the error of the estimated temperature for each zone is shown in Table I.

TABLE I
Unbiased deviation in error of estimated temperature for each zone.

Heating zone	Upper	1.5824
	Lower	1.4583
Soaking zone/Upper	Left	1.6510
	Right	1.6044
Soaking zone/Lower	Left	1.5079
	Right	1.5161

Sharp spikes appear in the curves for the error in the predicted temperature. They are caused by an abrupt change in the air/gas ratio of the lower-left soaking zone. It should be recalled that the dynamic RBF model was built on the assumption that the air/gas ratio was constant. However, since it was not possible to manually change the gas and air fluxes simultaneously, the change in the air/gas ratio was sometimes relatively large in the left soaking zone (Fig. 7), although it remained within a narrow range in the other zones. This is the reason for the spikes. It is desirable to keep the air/gas ratio constant in the reheating process; and this can be accomplished by using some method to automatically adjust the ratio. On the other hand, the spikes are not really a serious problem: If the predicted temperature is treated as a perturbed signal, the spikes can easily be

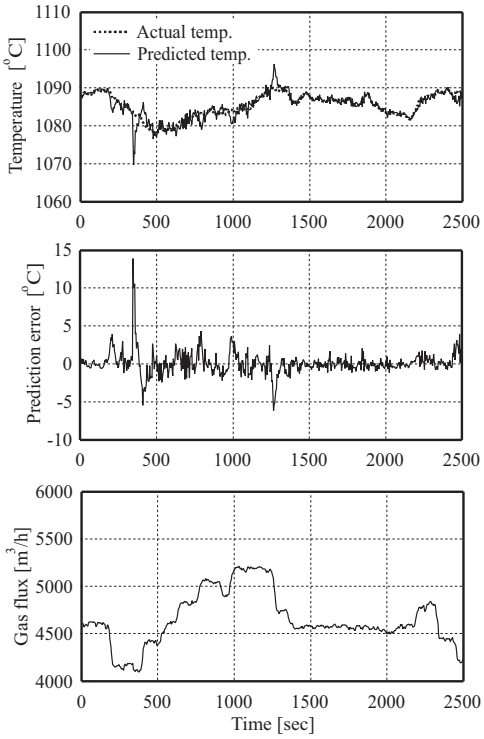
filtered out by existing signal processing techniques. As an example, the error in the predicted temperature of the upper heating zone obtained by the five-point moving-average method is shown in Fig. 8. Since our model takes the coupling of zones into account, changes in the air/gas ratio of the lower soaking zone affect not only the predicted temperature of the lower soaking zone, but also the prediction results for other zones. Thus, this method produces more realistic values than other methods. The engineering specification for the tolerance in the estimated temperature is usually less than $30°\mathrm{C}$. The filtered estimates clearly satisfy this requirement. A larger ρ produces smaller spikes in the outputs. e.g., Fig. 9 shows simulation results for the upper-left soaking zone with $\rho = 0.8$. Clearly, the spikes are much smaller than those in Fig. 6 (a). For comparison, verification results for the upper-left soaking zone based on BP learning are shown in Fig. 10. A bias appears in the predicted temperature, and the unbiased deviation is $7.9412°\mathrm{C}$. Clearly, the GAP-EKF learning algorithm is superior to the BP learning algorithm.

The GAP-EKF learning algorithm converged within seven steps. Since the NN is trained at every sampling time using the latest observation data, and since only the parameters of the neuron nearest the latest input are adjusted, the learning process is very rapid, and the network makes highly accurate temperature predictions. In the verification test, it took only 0.01 s to train the network, while the sampling period was 5 s. So, the speed of sequential learning based on a GAP algorithm and an EKF easily meets the requirements for real-time online learning.

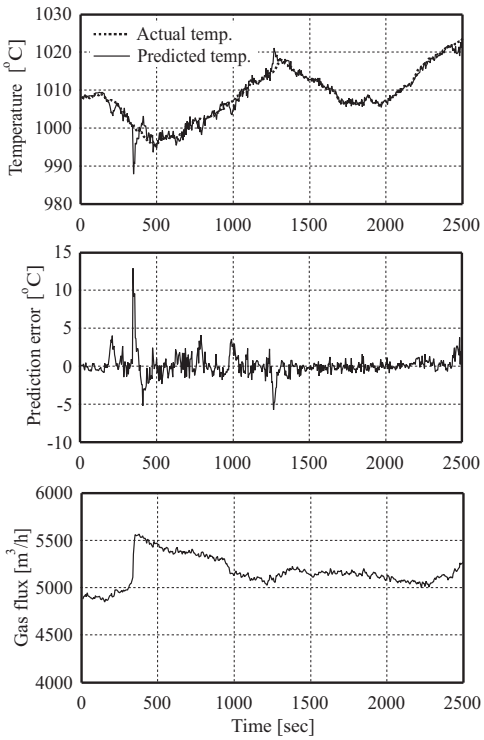
VI. Conclusion

This paper presented a dynamic model of the temperatures within a complex billet reheating furnace. The model is based on an RBF NN; and the combination of a GAP algorithm and an EKF provide the network with the ability to carry out random nonlinear mapping and quick sequential learning. These characteristics enable the model to handle nonlinearity, strong coupling, time variance, distributed parameters, etc., with sufficient speed and accuracy. Unlike conventional modeling methods, which, for example, only consider the heat exchange due to radiation [16] or ignore the generation of carbon monoxide [17], no such assumptions were made to predict the rise and fall in the temperatures of the furnace. Since our new model produces temperature estimates without the need to solve complex partial differential equations, it is fast enough for real-time learning. Even though the model is very simple, it provides the required estimation accuracy. Verification tests showed the model to be suitable for furnace process control, and the same method can also be used to model other complex industrial processes.

This NN reheating-furnace model was employed for the immune-based optimal control of furnace temperature, and the results were found to be superior to those obtained by

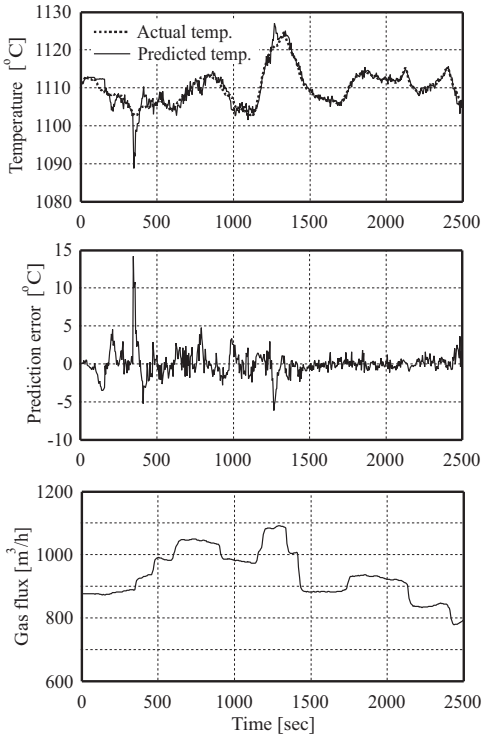


(a) Verification results in the upper heating zone.

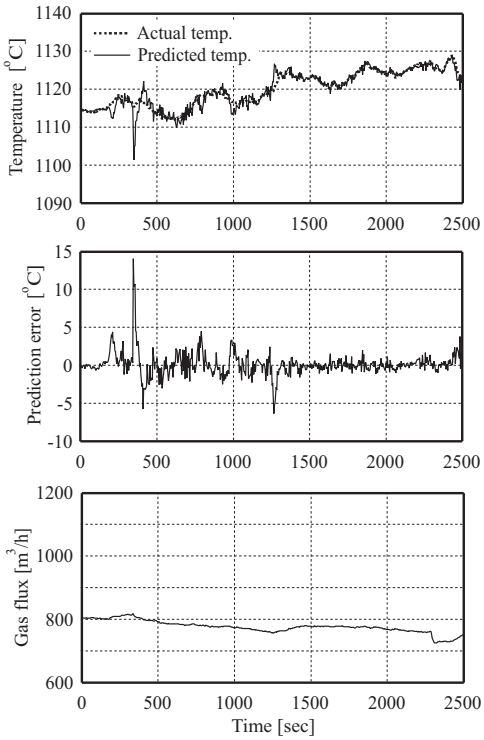


(b) Verification results in the bottom heating zone.

Fig. 4. Verification results for heating zone.

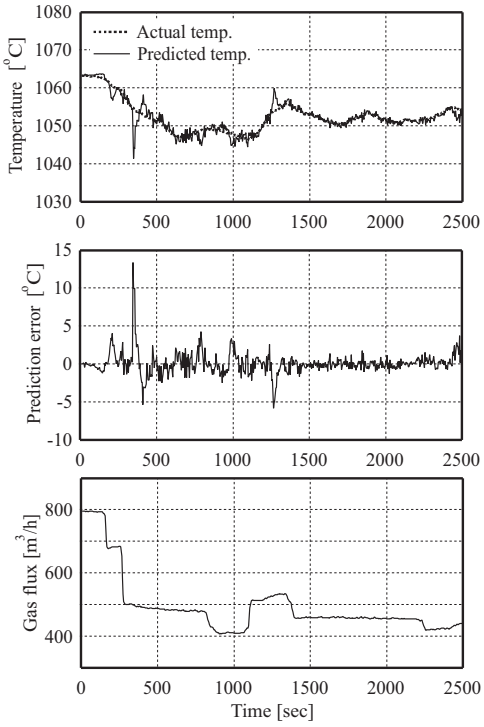


(a) Verification results in the upper left soaking zone.

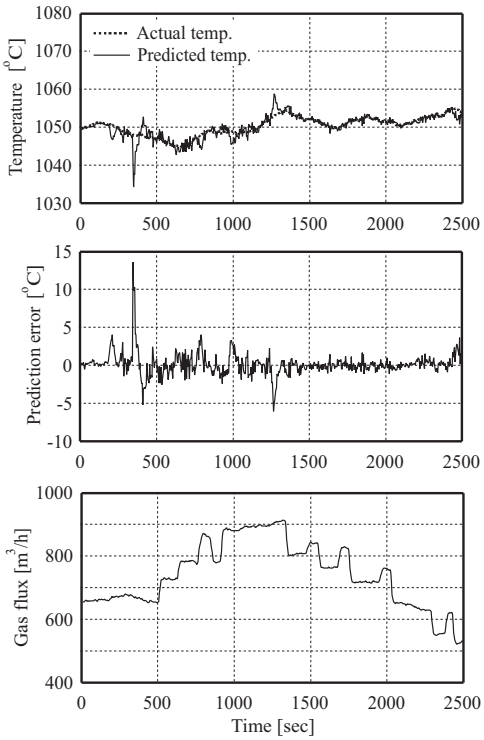


(b) Verification results in the upper right soaking zone.

Fig. 5. Verification results for upper soaking zone.



(a) Verification results in the bottom left soaking zone.



(b) Verification results in the bottom right soaking zone.

Fig. 6. Verification results for lower soaking zone.

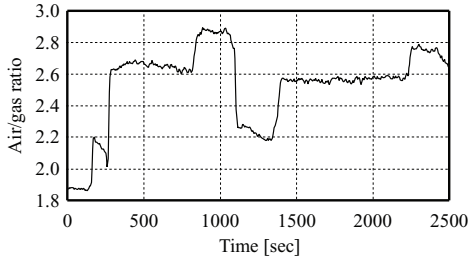


Fig. 7. Air/gas ratio of lower-left soaking zone.

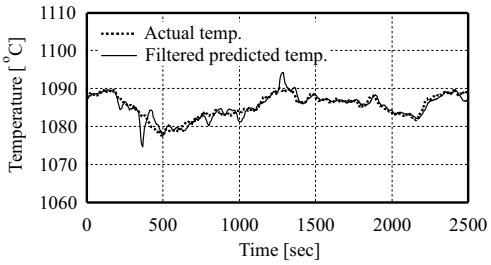


Fig. 8. Predicted temperature of upper heating zone filtered by the five-point moving-average method.

conventional models. A report on this will appear in the near future.

REFERENCES

[1] B. Bang, Z. G. Chen, L. Y. Xu, J. C. Wang, J. M. Zhang, and H. H. Shao, The Modeling and Control of a Reheating Furnace, *Proc. American Contr. Conf.*, 2002, pp. 3823-3828.

[2] B. Zhang, J. C. Wang, and J. M. Zhang, Dynamic Model of Reheating Furnace Based on Fuzzy System and Genetic Algorithm, *Control Theory & Application*, vol. 20, 2003, pp. 293-296.

[3] Z. J. Wang, T. Y. Chai, and C. Shao, Slab Temperature Prediction Model Based on RBF Neural Network, *Journal of System Simulation*, vol. 11, 1999, pp. 181-184 (in Chinese).

[4] K. Tao, A closer look at the radial basis function (RBF) networks, *Proc. 27th Asilomar Conf. Signals, Syst., Comput.*, 1993, pp. 401-405.

[5] C. Schiefer, F. X. Rubenzucker, H. P. Jorgl, and H. R. Aberl, A Neural Network Controls the Galvannealing Process, *IEEE Transactions on Industry Applications*, vol. 35, 1999, pp. 114-118.

[6] H. Peng, T. Ozaki, Y. Toyoda, H. Shioya, K. Nakano, V. Haggan-Ozaki, and M. Mori, RBF-ARX Model-based Nonlinear System Modeling and Predictive Control with Application to a NOx Decomposition Process, *Control Engineering Practice*, vol. 12, 2004, pp. 191-203.

[7] D. L. Yu and J. B. Gomm, Implementation of Neural Network Predictive Control to a Multivariable Chemical Reactor, *Control Engineering Practice*, vol. 11, 2003, pp. 1315-1323.

[8] W. Daosud, P. Thitiyasook, A. Arpornwichanop, P. Kittisupakorn, and M. A. Hussain, Neural Network Inverse Model-based Controller for the Control of a Steel Pickling Process, *Computers and Chemical Engineering*, vol. 29, 2005, pp. 2110-2119.

[9] A. G. Bors and M. Gabbouj, Minimal topology for a radial basis functions neural network for pattern classification, *Digital Signal Processing*, vol. 4, 1994, pp. 173-188.

[10] J. Platt, A Resource-allocating Network for Function Interpolation, *Neural Computat.*, vol. 3, 1991, pp. 213-225.

[11] V. Kadirkamanathan and M. Niranjan, A Function Estimation Approach to Sequential Learning with Neural Networks, *Neural Computat.*, vol. 5, 1993, pp. 954-975.

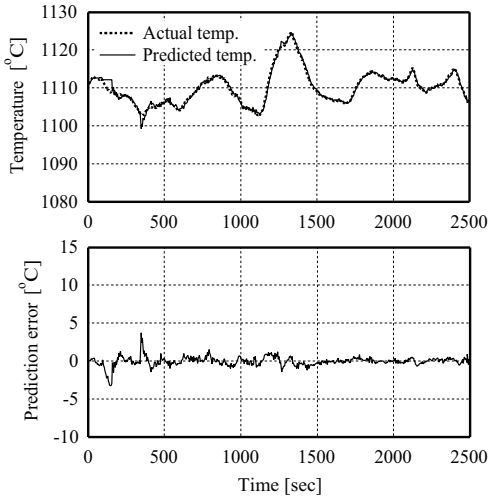


Fig. 9. Verification results for upper-left soaking zone with $\rho = 0.8$.

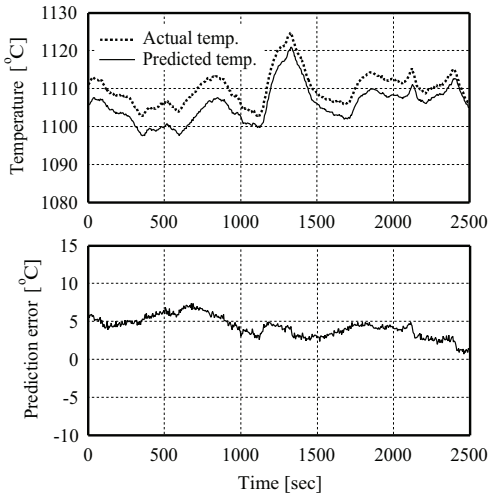


Fig. 10. Verification results for upper-left soaking zone obtained with BP learning algorithm.

[12] L. Yingwei, N. Sundararajan, and P. Saratchandran, A Sequential Learning Scheme for Function Approximation Using Minimal Radial Basis Function (RBF) Neural Networks, *Neural Computat.*, vol. 9, 1997, pp. 461-478.

[13] G. B. Huang, P. Saratchandran, and N. Sundararajan, A Generalized Growing and Pruning RBF (GGAP-RBF) Neural Network for Function Approximation, *IEEE Trans. Neural Networks*, vol. 16, 2005, pp. 57-67.

[14] W. Pedrycz, Conditional Fuzzy Clustering in the Design of Radial Basis Function Neural Networks, *IEEE Transactions on Neural Networks* vol. 9, 1998, pp. 601-612.

[15] J. S. Choi, C. C. L. Antonio, and H. Simon, Kalman Filter-Trained Recurrent Neural Equalizers for Time-Varying Channels, *IEEE Transactions on Communications*, vol. 53, 2005, pp. 472- 480.

[16] B. Zhang, Z. G. Chen, L. Y. Xu, J. C. Wang, J. M. Zhang, and H. H. Shao, The Modeling and Control of a Reheating Furnace, *Proceedings of the American Control Conference*, Anchorage, 2002, pp. 3823-3828.

[17] H. S. Ko, J. S. Kim, and T. W. Yoon, Modeling and Predictive Control of a Reheating Furnace, *Proceedings of American Control Conference*, 2002, pp. 2725-2729.