한국 퍼지 및 지능시스템 학회 2002, Vol. 12, No. 6, pp. 577-582

Neural-based Blind Modeling of Mini-mill ASC Crown

Gang-Hwa Lee*, Dong-Il Lee*, Seung-Joon Lee*, and Suk-Gyu Lee*,
Shin-Il Kim**, Hae-Doo Park**, and Seung-Gap Choi**

* School of EECS, Yeungnam University,
214-1 Dae-dong, GyongSan, GyongBuk.
** POSCO Technical Research Lab., Instrument and Control Research Group,
1, Goedong-dong, Nam-gu, Pohang, Gyeongbuk.

Abstract

Neural network can be trained to approximate an arbitrary nonlinear function of multivariate data like the mini-mill crown values in Automatic Shape Control. The trained weights of neural network can evaluate or generalize the process data outside the training vectors. Sometimes, the blind modeling of the process data is necessary to compare with the scattered analytical model of mini-mill process in isolated electro-mechanical forms. To come up with a viable model, we propose the 'blind neural-based range-division domain-clustering piecewise-linear' modeling scheme. The basic ideas are: 1) dividing the range of target data, 2) clustering the corresponding input space vectors, 3) training the neural network with clustered prototypes to smooth out the convergence and 4) solving the resulting matrix equations with a pseudo-inverse to alleviate the ill-conditioning problem. The simulation results support the effectiveness of the proposed scheme and it opens a new way to the data analysis technique. By the comparison with the statistical regression, it is evident that the proposed scheme obtains better modeling error uniformity and reduces the magnitudes of errors considerably. Approximatly 10-fold better performance results.

Key Words : neural network, blind modeling, data mining, clustering, condition number

1. Introduction

If the number of neurons in the hidden layer is sufficiently large, the two-layer neural network with tangent sigmoidal and pure linear activation function for the hidden- and output-layer respectively can approximate any nonlinear functions [1]. Although the convenient way of deciding the number of neurons in hidden layer is not established yet, there exists a heuristic way of judging the well-fitness of trained function by neural network [2].

A neural network can be trained to model an arbitrary function provided that we have enough input and output vectors for the supervised training. Besides, it has the generalization capability for the untrained input data set. In spite of these merits, the critical shortcoming of the neural network modeling is that the neural network training itself does not give any input-output mappings in elementary forms.

In many cases, we are interested in finding the input-output relations in closed-form equations. Unfortunately we often start with a very unformatted real world data like the mini-mill crown values in ASC (Automatic Shape Control) of steel rolling process. The ASC is typified by the control of strip crown which is defined by the difference between the center height and the average of edge heights in the cross-sectional shape of steel strip. In our case, we have twenty-one variables of input data space, and one output target vector of strip crown values. Some of the input variables are highly correlated and some are not. Since, in the rolling process, the steel strip is gradually rolled thinner through several rollers in tandem, any in/out strip thickness of rollers are tightly correlated. As a consequence, it usually requires a pre-processing of the data including the normalization, distribution and correlation analysis.

For the purpose of fully utilizing the generalization capability of neural network and comparing the neural-based model with the scattered analytical model of mini-mill process in isolated electro-mechanical forms [3], a problem at hand, in our case, is the 'blind modeling of input space versus output crown' without considering any prior data analysis. In the statistics field, aforementioned 'blind modeling' will be the 'exploratory data mining' [4].

The justification of this problem statement is that a given mini-mill rolling process should handle any different kinds of steel strips. The strip can have different starting thickness, different carbon concentration, and different cross-sectional width, etc.

접수일자 : 2002년 7월 30일
완료일자 : 2002년 11월 2일
This work has been supported by the grant of POSCO under the contract number 20026080.

In modeling a multivariate function (twenty-one variables in our case), the most difficult part is how to divide the original data set into the modeling and evaluating sets. Categorizing the original data into the one for training the neural network and the other for evaluating the network requires careful considerations. If we use well-behaved data only for modeling, the result may not reveal the details of data structures. On the other hand, if we use too much details of data variations, it may include abnormal behavior of data.

In this paper, we propose a new blind modeling (data mining) scheme of 'blind neural-based range -division domain-clustering piecewise-linear' modeling. The basic ideas are: 1) dividing the range of target data, 2) clustering the corresponding input space vector, 3) training the neural network with clustered prototypes to make the network convergence smooth and 4) solving the resulting matrix equations with pseudo-inverse to alleviate ill-conditioning problem. This proposed scheme considers the aspects of smoothing of data trends, early termination of neural network training, ill-conditioned matrix treatment, and the effective rolling process modeling.

In section 2, background information for the neural network modeling is introduced. We propose the range-division modeling scheme for the real steel rolling process in section 3. The modeling example of POSCO process data is presented in section 4. The concluding remarks recap the salient ideas.

2. Function Modeling with Neural Network

As the FFN (Feed Forward Network) structure with two layers may have tangent sigmoid and purely linear activation functions for hidden and output layer respectively, FFN can be used to model arbitrary functions. It is quite analogous to the Fourier Series function approximation scheme. Unfortunately, in the neural network training, there is no such well established procedures except the back propagation. As a result, it is quite uncommon to get the closed-form models out of neural network training.

In the function approximation paradigm, many researches have been devoted to the proof of ridge (sigmoidal) function approximation capability for the neural network applications [1, 2, 5]. In spite of those efforts, there are not many constructive proofs of deciding the weights of network for an arbitrary approximation precision.

Three important aspects of neural-based modeling relevant to our purpose are: 1) the adequate partition of the real data for the best modeling, 2) the decision criteria for the network training strategy, and 3) the selection of an appropriate modeling structure.

The most natural way of training-data partition policy is the random test vector selection [6]. As a simple example, if there are 500 data, we can choose 250 data for neural network training and the other 250 data for evaluation of network convergence. Moreover, the influence of outliers contained in the real data might be critical to the modeling behavior of neural network. Some techniques for identifying the outliers and reducing the effect of these outliers are proposed in literature [7].

Open problem of using neural network is the decision of the number of neurons for a specific function modeling. In addition to this matter, a closely related issue is the over- and under-fitting of network by a given number of neurons in a hidden layer. One good strategy avoiding (especially) an overfitting of the network is the so-called 'early-stopping' [2, 6]. By early stopping the network training before it enters into an overfitting status, we can reduce the possibility of outlier dominance and the abnormal modeling behavior.

3. Range-division Modeling Scheme

3.1 Basic Idea

In real data application, the modeling structure can never be determined by the correlative properties of predictor variables (input domain space) since we do not know the governing function between the input and output relations.

The ultimate input-output mapping should be guided by the target values of the responsive (range) variable. In the network learning paradigm, it is the 'supervised learning.' As an unavoidable consequence, the first step to the real data modeling is the division of the target ranges. Especially in the multivariate case when the cross-correlations are not clear yet, the target range is the only guidance to begin with.

Once we divide the range, to say, in equal intervals, all the corresponding input space data can be partitioned effectively. Binary-tree like partition strategy is reported in [8]. Once again, since we do not have the model function, a simple but an effective way of manipulating the input space is the unsupervised clustering of that space. The most popular categorization can be done by the FCM algorithm. We train the neural network with clustered prototypes (cluster means) and the common target values. To alleviate the problem of unequally populated range-division categories, we may use the best partition strategies such as the binary-tree partition or the majority-voting FCM [9].

After the partition, the center value of a given interval is used for the representative value for the corresponding clusters. Since the domain data are clustered into more than one, there exist several clusters to share the common representative target value and ultimately this common target value may cause the ill-conditioning problem of resulting matrix equations.

To cope with the over-fitting, we stop the network before it enters into a fully trained status. The resulting under-fitted target values show the discrepancies comparing with the representative target values. As the

neural network tries to reduce the output errors in such a way to adjust the weights in the opposite direction to the gradient error derivatives, we substitute the original target values with the values that are subtracted by the scaled discrepancies. By this arrangement, the ill-conditioning tendency might be mitigated. Besides, the slow convergence of network training is not a critical problem, in this case, due to the early-stopping.

The final step for the neural-based modeling is the fixations of the resulting matrix equations. If we try to solve the equations with only one target sub-range, we will encounter the under-determined case. To combine several target sub-ranges together into one matrix equation gives the over-determined case. Then it can be easily solved by the pseudo-inverse [4]. Moreover, in the piecewise linear model, combining many intervals into a larger one reduces the total number of piecewise linear model segments.

3.2 Domain-clustering Piecewise-linear Model

We start with the three days' amount of POSCO (GyangYang Works) real data which are specially arranged for the purpose of our approach. The mini-mill process data (ASC function turned off) can be expressed as the tuple (X, C), where X is the input vector (independent variables) and C is the output vector (dependent variable; crown). The tuple (X, C) consists of 885×22 data matrix where X (885×21) has 21 variables of sample size 885 and the last column (885×1) is the target crown vector.

The algorithm for the proposed model is shown by the flow chart diagram in Figure 1. The target crown is divided into 100 sub-ranges. Therefore, the sub-interval has the width of approximately 1μm of crown values. We use the center value of each interval for the representative target value for neural network training. The resulting 100 sub-ranges' input spaces are clustered by FCM algorithm.

The maximum allowable number of prototypes is decided by the off-line simulation using s-validity measure [10]. Since the use of optimum number of prototypes is not critically required in our case, we use a fixed number of clusters for every sub-interval. With five most-populated sub-interval simulations, the maximum allowable number of prototypes is chosen to eight by average. The sub-intervals that contain less data samples than maximum allowable prototypes keep the original data without FCM clustering. The total number of prototypes (including the samples without FCM clustering) obtained is 402.

Then, these prototypes and the representative target crowns are inputted to the neural network which is structured with one hidden layer of 100 neurons and one output node equipped with tangent sigmoidal and purely linear activation functions respectively.

The number of the hidden layer neurons has been guided by the heuristic that the maximum number of neurons is calculated by $n(2n+1)$ where n is the number of input variables adapted by [2]. Since the number of hidden layers obtained above is the upper bound for the function approximator, we simulate the neural network with starting hidden layer neurons of 800. Then, we down-size the number of hidden layer neurons by 25 neurons and re-simulate. The MSE (mean square error) values of network training at 300 epoch are recorded in each simulation and we choose the smallest possible number of hidden layer neurons that does not increase the MSE considerably. We found that 40 to 50 neurons increase the MSE considerably. For safety margin, 100 neurons in the hidden layer is selected for the subsequent procedures.

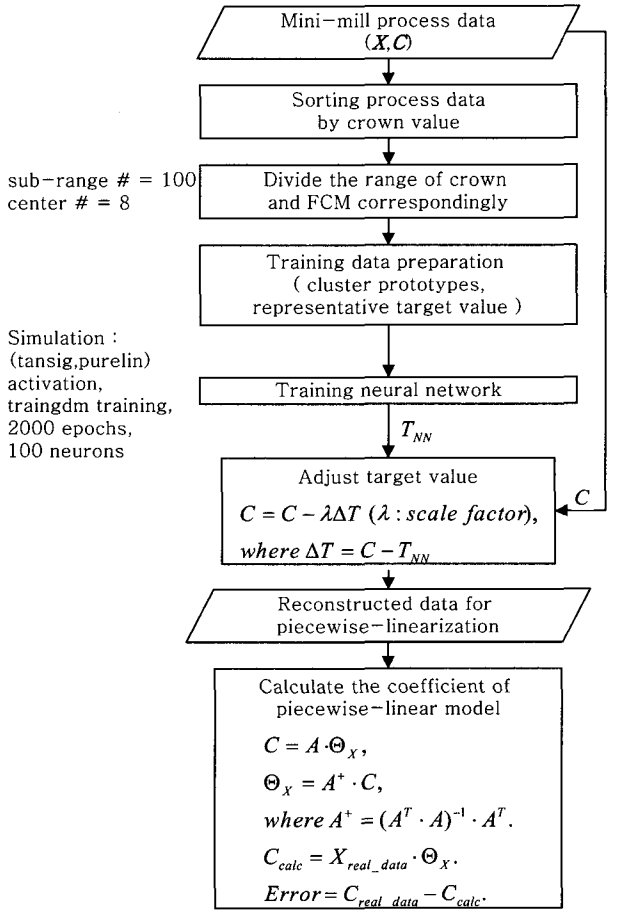


Figure 1. The flowchart of the piecewise-linear model proposed

The decision of the early-stop epochs is made after the 'random equal partition' strategy proposed by [6]. We randomly select half of the data samples for the neural network training and use the other half for generalization. We record the generalization error for every particular partition. We chose 2000 epochs, the appropriate number of epochs, by averaging the minimum error epochs after five such random equal partition simulations (see Figure 4, in section 4.2).

Since each sub-range has multiple prototypes and use

the same representative crown value for the matrix equation, it is desirable to perturb the crown value a little bit. For this purpose, the result of neural network training TNN is absorbed into the matrix equation as a scaled negative perturbation. The representative crown value C is substituted by

$$C - \lambda \Delta T, \tag{1}$$

where ΔT is $C - T_{NN}$.

The final matrix equation for the piecewise linear model is arranged as follows. The parameterized model has the general structure of

$$C = a_1 X_1 + a_2 X_2 + \cdots + a_{21} X_{21} + a_0, \tag{2}$$

where a_0 is the bias.

If we use the variable name A (since the data are changed after neural training) instead of X, and rewrite (2) in matrix form, then

$$C = A\ \Theta_X, \tag{3}$$

where the estimation parameter vector

$$\Theta_X = [a_1\ a_2 \cdots a_{21}\ a_0]^T. \tag{4}$$

The parameter vector Θ_X can be conveniently solved by pseudo-inverse format.

4. Modeling Example

4.1 Steel Rolling System Configuration

The steel rolling process of POSCO plant can be figured briefly as in Figure 2. From the furnace, the thin slab of below 100 mm thickness is fed into the reduction unit. After the re-coiling, the coil is fed into the rollers F of five in tandem. To control the shape of steel, the profile meters to measure the thickness are distributed along the cross-sectional width. To maintain the quality of production steel shape, the rolling force, the bending force of working rolls, pair crossing angle of working roll, etc., are controlled by appropriate measures.

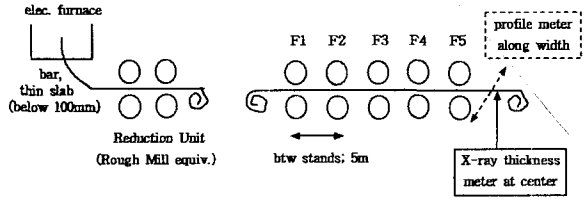


Figure 2. Steel rolling configuration of mini-mill process.

4.2 Linear Model Results

We use the POSCO (GyangYang works) real data, that are acquired from three days' amount of logged process data, as the test bed. For the enhanced complexity of the modeling, the ASC function has been shut off.

In figure 3, we present a simple neural network training result trained up to 2,000 epochs. The 'o' marks indicate the original sorted samples and the '+' marks are the trained crown values. The neural network used in the simulation has 100 neurons in one hidden-layer. The training epochs are allowed to be only 2,000 since, at that training epoch, the network begins to overfit. The training result is used for the target performance criterion of our proposed scheme.

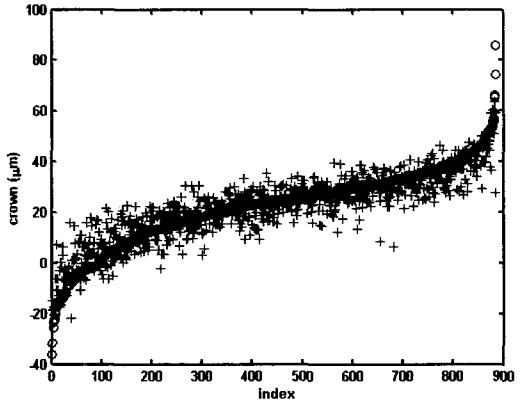


Figure 3. Crown data trained 2,000 epochs with 100 neurons: 'o'; original sample, '+'; nnet crown.

In Figure 4, the early-stopping epoch simulation results are represented. The beginning epoch of overfitting is chosen from the minimum point of the generalization error. In this particular simulation, the generalization MSE (solid-line) has the minimum at 1744 epoch. After 5 such simulations, we choose 2000 epoch by average.

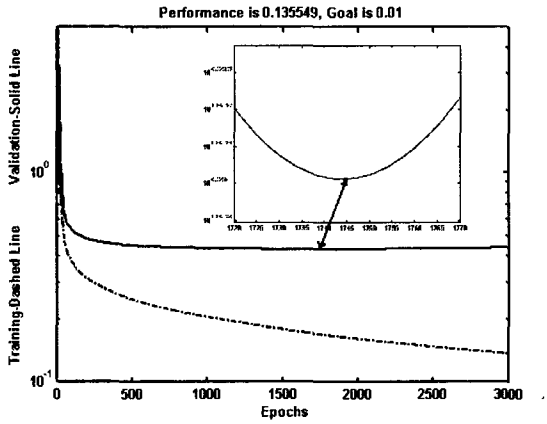


Figure 4. MSE trends as simulation step evolves: 'solid'; generalization MSE, 'dash'; training MSE. (100 neurons, 'traingdm' learning)

As mentioned before, it is not our objective to train the neural network with given data, and use those resulting weights to generalize. Therefore, we do not need to train the network as much period of epochs. The

training comes to a halt at 2000 epochs and the result has been fed to the piecewise modeling routine to perturb the representative crown values as indicated by Equation (1).

Figure 5 shows the corresponding condition numbers of the sectional 'A' matrixes. Each sectional tuple (A, C) that is used to piecewise regression consists of 40 equations. The purpose of this simulation is the identification of the ill-conditioned sectional matrixes by the condition numbers.

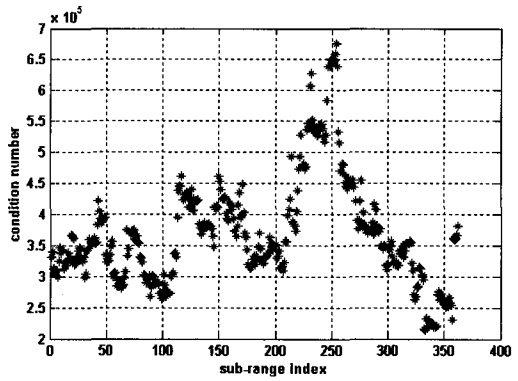


Figure 5. Sectional condition numbers of piecewise linear model

4.3 Comparison with Conventional Regression

The regression result by the conventional statistical regression method is used as the test vehicle to the evaluation of the proposed scheme. The procedures to obtain a regression model is quite well-known [4]. However, the modeling by the statistical regression also requires the modeling and generalization phases. Without these schemes, there are not many ways to accomodate the rejection of noise and/or outliers. Whereas, the proposed neural-based scheme aims to provide the generalization capability without separate modeling and generalization phases. The neural-based scheme implicitly includes the generalization phase.

Figures 6 and 7 summarizes the final neural-based modeling results. In each figure, the original crown samples are sorted and indicated by 'o' marks and the calculated crown values from the regression model by '+'. Figure 6 shows the result of the proposed, neural-based scheme. We can see the good generalization trends except in the middle section of the samples. We are now working on this problem to fix the causes of poorly-fitted middle part.

Figure 7 is generated by regressing the half of the sample data and generalizing with entire samples. Comparing with the proposed scheme, we can deduce that the absolute generalization error is uniformly scattered over all crown ranges.

Due to the different nature of the approaches, two methods in Figure 6 and Figure 7 can not be directly compared. However, when using the MSE as the performance measure, the neural-based method generates the MSE of 12.66 and the MSE of 113.93 by the conventional regression method. Hence, roughly 10-fold MSE reduction is accomplished.

During the simulation, we did not consider the issues of: 1) the complete outlier analysis, 2) any 'a priori' correlation analyses between the input variables. Albeit, the proposed scheme has been tested successfully and the results are very convincing.

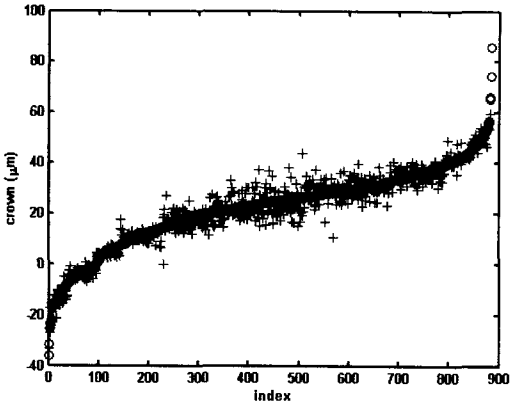


Figure 6. The generalization by the proposed neural-based regression scheme

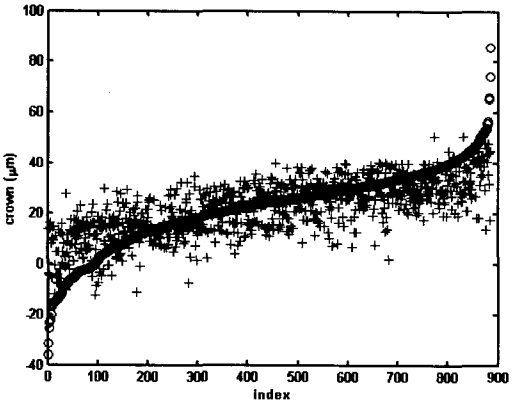


Figure 7. The generalization by the conventional statistical regression scheme

5. Concluding Remarks

We have proposed the neural-based 'blind piecewise -linear, range-division, domain-clustering' modeling scheme.

The proposed scheme considers the issues of: 1) over-fitting of neural network training, 2) partition of data validation, 3) potentially ill-conditioned matrix problems. Also, various ways of effectively clustering the input space by, to say, the majority-voting FCM are under consideration.

The results with POSCO real data support the effectiveness of the proposed modeling scheme. By the neural-based scheme, approximately 10-fold better MSE performance compared with the conventional regression

scheme has been demonstrated. Using the nonlinear spline model, and the principal component analysis are left for the further studies.

References

[1] F. Scarselli and A. C. Tsoi, "Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results," Neural Networks, vol.11, no.1, pp.15-37, 1998

[2] C. M. Bishop, Neural Networks for Pattern Recognition, Clarendon Press, Oxford, 1995.

[3] POSCO Technical Research Lab., Personal communications.

[4] W. L. Martinez and A. R. Martinez, Computational Statistics Handbook with MATLAB, Chapman & Hall/CRC, Boca Raton, FL, 2002.

[5] C. K. Chui and X. Li, "Approximation by Ridge Functions and Neural Networks with One Hidden Layer," Journal of Approx. Theory, vol. 70, pp. 131-141, 1992.

[6] I. V. Tetko and A. E. P. Villa, "Efficient Partition of Learning Data Sets for Neural Network Training," Neural Networks, vol. 10, no. 8, pp. 1361-1374, 1997.

[7] A. S. Kosinski, "A Procedure for the detection of multivariate outliers," Computational Statistics & Data Analysis, vol. 29, pp. 145-161, 1999.

[8] R. Siciliano and F. Mola, "Multivariate data analysis and modeling through classification and regression trees," Computational Statistics & Data Analysis, vol. 32, pp. 285-301, 2000.

[9] GangHwa Lee, YoonChul Lee, SoonHak Kwon and SukGyu Lee, "Majority-Voting FCM Algorithm in the Vague Fuzzy Classification," The 10th IEEE International Fuzzy Systems Conference, vol. 2, 2001.

[10] X. L. Xie and G. Beni, "A Validity Measure for Fuzzy Clustering," IEEE Trans. PAMI, vol. 13, no. 8, pp. 841-847, 1991.

저 자 소 개

이강화 (Gang-Hwa Lee)
1981년 서울대 전기공학과 졸업
1984년 서울대 대학원 전기공 석사
(84)-89 금성사 중앙연구소 선임
1994년 Oklahoma S. 전기공 석사
1997년 UT Arlington 전기공 박사수료
1997년 National Semiconductor co-op
2000년~현 영남대 대학원 전기공 박사수료. 대구대 정보통신공 겸임교수

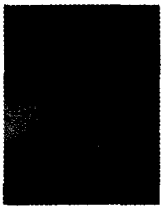
관심분야 : 시스템, 제어, VLSI, 통신

이동일 (Dong-Il Lee)
2001년 영남대 전기전자공학부 졸업
2001년~현 영남대 대학원
전기공학과 석사과정
관심분야: 제어 및 시스템, 이동 로봇, 정보 가전, 내장형 시스템

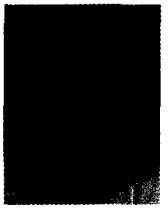
이승준 (Seung-Joon Lee)
2000년 영남대 전기공학과 졸업
2000년 삼성전자 LCD사업부
2001년~현 영남대 대학원
전기공학과 석사과정
관심분야: 제어 및 시스템, 이동 로봇, 정보 가전, 내장형 시스템

김신일 (Shin-Il Kim)
1991년 경북대 기계공학과 졸업
1993년 미국 U. of Bridgeport
기계공학 석사
1993년~현 포스코 기술연구소 연구원
관심분야: 철강공정 진단 및 제어

박해두 (Hae-Doo Park)
1981년 부산대 기계공학과 졸업
1983년 부산대 기계공학과 석사
1983년 포항제철소 입사
1995년 부산대 기계공학과 박사
1994년~현 포스코 기술연구소
광양계측제 어연구 팀장
관심분야 : 제어 및 시스템, 압연제어



최승갑 (Seung-Gap Choi)
1980년 서울대 전기공학과 졸업
1983년 서울대 제어계측공학과 석사
1997년 U. of Strathclyde(U.K),
Dept. EE Eng. 박사
1983년~현 포스코 기술연구소
관심분야: Process, Robust,
Auto-PID control



이석규 (Suk-Gyu Lee)
1979년 서울대 전기공학과 졸업
1981년 서울대 전기공학과 석사
1990년 UCLA 전기공학과 박사
1982년~현 영남대학교
전자정보공학부 교수

관심분야 : 퍼지 뉴럴 이론, 로봇 공학, 제어 시스템

Phone : 053-810-2487
Fax : 053-813-8230
E-mail : sglee@yu.ac.kr