A Systematic Method for Fuzzy Modeling from Numerical Data

Min-You Chen and D. A. Linkens

Dept. of Automatic Control & Systems Engineering
The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK

Abstract:

A systematic fuzzy modeling method that includes the initial fuzzy model self-generation, significant input selection, partition validation, parameter optimisation and rule-base simplification is proposed in this paper. In this framework, the whole procedure of structure identification and parameter optimisation is carried out automatically and efficiently by the combined use of a self-organisation network, fuzzy clustering, adaptive back-propagation learning and similarity analysis. The proposed fuzzy modeling approach has been used to non-linear system identification and mechanical property prediction in hot rolled steels. Experimental studies demonstrate that the proposed fuzzy models have a good balance between model accuracy and interpretability.

Keywords

Fuzzy Modelling, Rule base self-generation, Fuzzy Clustering, Neurofuzzy systems.

1. Introduction

Fuzzy modelling is a very important and active research field in fuzzy logic systems. Compared to traditional mathematical modeling and pure neural network modeling, fuzzy modeling possesses some distinctive advantages, such as the mechanism of reasoning in human understandable terms, the capacity of taking linguistic information from human experts and combining it with numerical data, and the ability of approximating complicated non-linear functions with simpler models. In recent years, a variety of different fuzzy modelling approaches have been developed and applied in engineering practice [1]-[7]. These approaches provided powerful tools to solve complex non-linear system modeling and control problems. However, most existing fuzzy modeling approaches concentrate on model accuracy that simply fit the data with the highest possible accuracy, paying little attention to simplicity and interpretability of the obtained models, which is considered as a primary merit of fuzzy rule-based systems. In many cases, users require the model to not only predict the system's output accurately but also to provide useful description of the system that generated the data. Such a description can be elicited and possibly combined with the knowledge of experts, helping to understand the system and validate the model acquired from data. Thus, it is desired to establish a fuzzy model with satisfactory accuracy and good interpretation capability.

Our aim is to develop a systematic fuzzy modeling mechanism, without any assumption about the structure of the data, which is capable of 1) generating a rule base automatically from numeric data, 2) finding the optimal number of the rules and fuzzy sets, 3) optimising the parameters of fuzzy membership functions, and 4) providing a readily interpretable model. To achieve these objectives, this paper proposes a systematic neuro-fuzzy modeling paradigm by incorporating a modified self-organisation network, fuzzy c-means (FCM) clustering associated with a proposed cluster validity measure, similarity analysis and back-propagation learning. The methodology and implementation of the neuro-fuzzy modeling will be described in the following section.

2. Methodology of Fuzzy Modeling

Consider a collection of N data points $\{\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_N\}$ in a m+1 dimensional space that combines both input and output dimensions. A generic fuzzy model is presented as a collection of fuzzy rules in the following form

R_i: IF x_1 is A_{i2} and x_2 is A_{i2} ... and x_m is A_{im}
THEN $y=z_i(\boldsymbol{x})$

where $\boldsymbol{x}=(x_1, x_2, \ldots, x_m)\in U$ and $y\in V$ are linguistic variables, A_{ij} are fuzzy sets of the universes of discourse $U_i\in R$, and $z_i(\boldsymbol{x})$ is a function of the input variables. Typically, z can take one of the following three forms: singleton, fuzzy set or linear function. Fuzzy logic systems with center of average defuzzification, product-inference-rule and singleton fuzzification are of the following form:

$$y=\sum_{i=1}^{p} z_i[\prod_{j=1}^{m}\mu_{ij}(x_j)] \Big/ \sum_{i=1}^{p}\prod_{j=1}^{m}\mu_{ij}(x_j) \qquad (1)$$

where $\mu_{ij}(x_j)$ denotes the membership function of x_j belonging to the ith rule. Very commonly, Gaussian function is chosen as the membership function. Thus, equation (1) can be rewritten as

$$y=\sum_{i=1}^{p} z_i m_i(x) \Big/ \sum_{i=1}^{p} m_i(x) \qquad (2)$$

where $m_i(x)=\exp(-\|x-c_i\|^2/\sigma_i^2)$ represents the matching degree of the current input x to the ith fuzzy rule.

According to the fuzzy modelling paradigm proposed in [8], a fuzzy modelling problem is equivalent to solving the following problems: 1) generating an initial fuzzy rule-base from data; 2) selecting the important input variables; 3) determining the optimal number of fuzzy rules; 4) optimising the parameters both in the antecedent part and consequent part of the rules; and 5) optimise the acquired fuzzy model by removing the redundant membership functions. The methodology of fuzzy model construction and implementation will be discussed in the following sub-sections.

A. Fuzzy Model Initialisation

Creating the initial fuzzy model can be considered as a clustering process which groups the data scattered in the input-output space into a collection of clusters. A fuzzy competitive neural network is exploited as a pre-processor to extract a number of clusters which can be viewed as an initial fuzzy model from raw data [8]. This step is used to perform fuzzy classification with the two objectives of providing the fuzzy model for subsequent input selection and reducing the total number of training instances for model optimisation.

Consider a set of N data points $\{P_1, P_2, ..., P_N\}$, the input-output data pair can be represented as $P_k=(x_{k1}, x_{k2}, ..., x_{km}, y_k)$, $k=1, 2, ..., m$. The self-organising network is introduced to produce the sub-clusters based on the given data set. In contrast to the Kohonen network [9], the proposed network has a variable structure in which the number of nodes in the competitive layer can be changed dynamically in response to the incoming data. Starting with one node at the competitive layer, as the learning process proceeds with increasing iteration, the number of the nodes grows accordingly. The number of clusters can be controlled by the cluster radius and the activation threshold. The procedure for the *modified competitive learning* algorithm is described as follows.

Step 1. Initialisation.

Given unlabelled data set $\{\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_N\}$, where; input the first input pattern P_1; set the iteration number $l=1$. Let the first weight vector be $\mathbf{w}_1(0)=P_1$, i.e. $w_{1i}(0)=x_{1i}$, $i=1, 2, ..., m+1$. Specify the valid radius δ for all nodes. Set the number of nodes $N_t=1$, and the activation number of node 1 $N_{s1}=1$;

Step 2. Dissimilarity calculation

For the lth input pattern (at the lth sampling instance): find the node J which has the minimum distance to the current input pattern P_l by

$\|P_l-\mathbf{w}_J(l)\|=\min_i \|P_l-\mathbf{w}_i(l)\|$, $i=1, 2, ..., N_t$.

The distance is defined as $\|P_l-\mathbf{w}_i\|=(P_l-\mathbf{w}_i)^T(P_l-\mathbf{w}_i)$

Step 3. Determine the winning node.

Use the following rule:

If $\|P_l-\mathbf{w}_J(l)\|\leq\delta$ Then node J is the winner

If $\|P_l-\mathbf{w}_J(l)\|>\delta$ Then create a new node

If J is the winner, modify the weight vector of node J to $\mathbf{w}_J(l)=\mathbf{w}_J(l-1)+\alpha(P_l-\mathbf{w}_J(l))$

where α is the learning rate which is determined by $\alpha=\alpha_0/(N_{sJ}+1)$, where $\alpha_0\in[0,1]$ is the initial rate; and set $N_{sJ}=N_{sJ}+1$; $l=l+1$.

If creating a new node, then the weight vector of the new node is given by

$\mathbf{w}_{n_t}(l)=x_l$, $N_t=N_t+1$;

If $l<N$, go to Step 2, otherwise set $p=N_t$ and stop.

The activation value of an output node is defined as: $o_j=\mathbf{w}_j$; $j=1, 2, ..., p$, where $\mathbf{w}_j=(w_{j1}, w_{j2}, ..., w_{jn})^T$ represents the prototype of the jth fuzzy cluster in input-output space.

When unsupervised learning is completed, a collection of p fuzzy clusters $C=(c_1, c_2,...,c_p)$ is produced. Each cluster center $c_i=(x_i^*\ y_i^*)$ can be considered as a fuzzy rule that describes the system behavior. Intuitively, cluster centre c_i represents the rule "If input is around x_i^* Then output is around y_i^*". Given an input vector x^*, the degree to which rule i is fulfilled is determined by the membership function $\mu_i(x_i)$. This initial fuzzy model is used as the basis for subsequent model structure identification.

B. Input Selection

Selecting the important input variables from all possible input variables is important for system modeling. Obviously, incorporating only the important variables into a model provides a simpler, more useful and more reliable model. Based on the initial fuzzy model that incorporates all possible input variables, we can evaluate the importance of each input variable. The objective of this work is to reduce the input dimensionality of the model without significant loss in accuracy. It is known that the change of system output is contributed to by all input variables. The larger the output change caused by a specified input variable, the more important this input variable is. The fuzzy inference system provides an easy mechanism to test the importance of each input variable without having to generate new models. The basic idea is to let all antecedent clauses except one associated with a particular input variable (e.g., X_{2j} is A_{2j}) in the rules be assigned a truth value of 1 and then compute the fuzzy output with respect to this input variable. A neurofuzzy-network-based model, which can generate all fuzzy outputs in parallel with respect to every individual input variable and test the importance of all input variables simultaneously, was proposed in [10]. Based on the neurofuzzy model, the output of the fuzzy inference system using multiplication as the AND operator and defuzzification using the center of area algorithm can be computed by

$$z_{ij} = \sum_{j=1}^{p} \mu_{ij}(x_i^*) y_j^* \Big/ \sum_{j=1}^{p} \mu_{ij}(x_i^*) \quad (3)$$

The output vector $z_i=(z_{i1}, z_{i2}, ..., z_{ip})$ denotes the fuzzy inference output corresponding to the contribution of the ith input variable. On the basis of m fuzzy output vectors, the importance of the input variables can be determined by calculating the change range of the corresponding z_i, obtained by

$$Rz_i=\max(z_i)-\min(z_i).$$

Hence, the importance factor of the ith input can be defined by $F_i=Rz_i/R_m$;
where $R_m=\max\{Rz_1, Rz_2, ..., Rz_m\}$.

Clearly, $F_i=1$ corresponds to the most important input variable, the large range of the fuzzy output indicating the big influence of the corresponding input variable. A small value of F_i corresponds to a relatively unimportant input. When F_i is less than the chosen threshold, i.e. $F_i<\lambda$, $\lambda\in(0,1)$, the corresponding input variable is believed unimportant and can be removed.

Assume that there are r inputs with the values of $F_i \geq \lambda$, thus, a collection of r inputs is selected from m input variables. Further recognition of the closely related input variables (independent input variable testing) can be realized by calculating the correlation functions, $\rho(x_i, x_j)$, between the selected input variables:

$$\rho(x_i, x_j) = \frac{\frac{1}{N}\sum_{k=1}^{N}(x_i - \bar{x}_i)(x_j - \bar{x}_j)}{\sqrt{\phi_{x_i}\phi_{x_j}}} \quad (4)$$

where $\rho(x_i, x_j)\in[0,1]$, $\bar{x}_i$, $\bar{x}_j$, ϕ_{x_i}, ϕ_{x_j} are the means and variances of vector x_i and x_j respectively, $i, j=1,2, ..., r$. The independent variables among the r selected inputs can be recognized by the following rule: if $|\rho(x_i, x_j)|>\tau$, then x_i is closely related with x_j, thus, remove the one which has a smaller value of importance factor from the list of selected significant input variables. τ is the chosen threshold.

The task of input selection is completed and a collection of q ($q\leq r$) significant input variables is selected for the fuzzy model. It should be noted that the input selecting procedure is proceeded on the basis of central points $c_i=(x_i^* \; y_i^*)$ of the p clusters obtained through self-organising network instead of raw data. The computational cost in this stage is reduced drastically due to $p<<N$.

C. Partition Validation

Although we can generate the initial fuzzy model associated with only the significant inputs by using the modified competitive network, the number of rules of the initial rule-base is not optimal because the competitive learning process is not based on optimising any model of the data. Determination of the optimal number of the fuzzy rules is a very important issue, which is equivalent to finding out a suitable number of clusters for the given data set. It is noted that an effective partition in input-output space can lead to reducing the number of rules and thus improving the computational efficiency and interpretability of the fuzzy model. Numerous clustering algorithms have been developed. The most widely used algorithm is the fuzzy c-means (FCM) due to its efficacy and simplicity. However, the number c of clusters is required to be pre-determined. FCM algorithm partitions a collection of n data points ($X=\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$) into c fuzzy clusters such that the following objective function is minimised.

$$J_m=\sum_{k=1}^{n}\sum_{i=1}^{c}\mu_{ik}^{m}(x)\,\|\boldsymbol{x}_k-\boldsymbol{v}_i\|^2, \quad 1<m<\infty \quad (5)$$

where m is a fuzzy coefficient, $\boldsymbol{v}_i$ is the prototype of the ith cluster generated by fuzzy clustering, u_{ik} is the membership degree of the kth data belonging to the ith cluster represented by $\boldsymbol{v}_i$, $u_{ik}\in U$, U is a $c\times n$ fuzzy partition matrix which satisfies the constraints:

$$0 \leq u_{ik} \leq 1 \quad \forall i,k \;\; \& \; \sum_{i=1}^{c} u_{ik} = 1 \;\; \forall k$$

Partition validation is the problem of finding the best value for c subject to minimisation of J_m. Since J_m monotonically decreases with c, an efficient criterion for evaluating the performance is required. Various cluster validity criteria have been proposed, such as partition coefficient [11], partition entropy [12], hypervolume and partition density [13], and some other effective validity measures ([1],[14], [15]). These criteria provide useful tools for cluster validation, each of which has developed its own set of partially successful validation schemes. In this work, a new fuzzy partition measure is proposed as a cluster validity criterion associated with the FCM algorithm, which is defined as follows:

$$J(\mathrm{U},c)=\frac{1}{n}\sum_{k=1}^{n}\max_i(u_{ik}) - \frac{1}{K}\sum_{i=1}^{c-1}\sum_{j=i+1}^{c}[\frac{1}{n}\sum_{k=1}^{n}\min(u_{ik},u_{jk})] \quad (6)$$

where $K=\sum_{i=1}^{c-1} i$

It can be seen that the proposed criterion includes two items. The first item reflects the compactness within clusters. and the second item represents the separation between clusters. That is, the optimal clustering means minimizing J over the whole c space. It should be pointed out that the criterion J only involves the information of membership degree and concentrates on the maximum and minimum values of u_{ik}, thus it is computationally simple and fast. Simulation experiments demonstrated that the proposed validity measure works very well when $m\in[1.5, 3]$. Fortunately, $m=2$ is the most common choice in fuzzy clustering. Partition validation based on the above criterion is carried out by the fuzzy clustering algorithms through an iterative optimisation of J_m according to the following steps:

Step 1. Choose the maximum cluster number c_{max}, iteration limit T, weighting exponent m, and termination criterion $\varepsilon>0$.

Step 2. With c=2, 3, ..., c_{max}; initialise the position of cluster centres: $\boldsymbol{V}_0=(\boldsymbol{v}_{10}, \boldsymbol{v}_{20}, ..., \boldsymbol{v}_{c0})$;

Step 3. With the iteration number t=1,2,...,T;

calculate; $u_{ik,t}=1\Big/\sum_{j=1}^{c}(d_{ik}/d_{jk})^{2/(m-1)}$ (7)

where $d_{ik}=\|\boldsymbol{x}_k-\boldsymbol{v}_i\|$, i=1, 2,..., c; k=1, 2,..., p;

Calculate $\boldsymbol{v}_{i,t}=\sum_{k=1}^{n}(u_{ik,t})^m x_k\Big/\sum_{k=1}^{n}(u_{ik,t})^m$ (8)

If $\|V_t-V_{t-1}\|<\varepsilon$,go to next step, otherwise repeat step 3.

Step 4. Calculate $J(c)$ by (6); if $c<c_{max}$, repeat from Step 2. Otherwise, stop the program and set the optimal cluster number $c=c_m$, where c_m meets the following condition:

$J(c_m)=\min\{J(c)\}$, c=2, 3, ..., c_{max};

Based on cluster validation, both the number of rules and the prototypes of the clusters $\boldsymbol{v}_j=(v_{j1}, v_{j2}, ..., v_{jm}, v_{j,m+1})$ can be obtained, where j=1, 2, ..., c.

Let $\boldsymbol{a}_j=(a_{j1}, a_{j2}, ..., a_{jm})=(v_{j1}, v_{j2}, ..., v_{jm})$, $z_j=v_{j,m+1}$, then the vector $\boldsymbol{a}_j$ denotes the prototype of the jth fuzzy partition in the input space. It can also be viewed as the center values of the Gaussian membership functions in the antecedent of the jth rule, while z_j is the prototype of the jth fuzzy partition in the output space, and denotes the fuzzy output value in the consequent part of the jth rule.

Therefore, the rule-base which is composed of c fuzzy rules can be represented as

R_j: IF x_1 is A_{j1} and x_2 is A_{j2}...and x_q is A_{jq} THEN y is z_j

where R_j denotes the jth rule, A_{ji} is the fuzzy set defined by the Gaussian membership function; and $z_j=b_j$ or $z_j=\sum_{i=0}^{q}b_{ji}x_i$, is the jth rule output with respect to a Mamdani model or Takagi-Sugeno (TS) model respectively, with x_0=1.

D. Parameter Optimisation

After structure identification, we obtain not only the number of rules but also the initial parameters which can be used to build the fuzzy models. To improve the model performance and achieve higher modeling accuracy, the obtained parameters should be optimised under a certain performance index. There are several methods for training fuzzy models, that is, to learn the optimal membership function parameters a_{ji}, σ_{ji} and linear weights b_{ji}. The commonly used back-propagation learning algorithm is applied to optimise the parameters a_{ji}, σ_{ji} and b_{ji} under the performance index of *Mean Square Error* (MSE). Using the back-propagation technique, the parameter learning algorithms can be derived as

$\Delta b_{ji}(t)=\beta e_k x_i q_j(\boldsymbol{x})+\gamma\Delta b_{ji}(t-1)$ (9)

$\Delta a_{ji}(t)=\beta e_k\frac{(x_{ik}-a_{ji})}{\sigma_{ji}^2}(z_j-y_k)q_j(\boldsymbol{x})+\gamma\Delta a_{ji}(t-1)$ (10)

$\Delta\sigma_{ji}(t)=-\beta e_k\frac{(x_{ik}-a_{ji})^2}{\sigma_{ji}^3}(z_j-y_k)q_j(\boldsymbol{x})+\gamma\Delta\sigma_{ji}(t-1)$ (11)

where β is the learning rate, γ is momentum rate, t refers to the iteration number, x_0=1, $e_k=(d_k-y_k)$ and $q_j(\boldsymbol{x})=\mu_j(\boldsymbol{x})/\sum_{j=1}^{c}\mu_j(\boldsymbol{x})$.

To increase the convergence speed and improve the learning efficiency of the back-propagation algorithm, an adaptive learning rate tuning algorithm was introduced for self-adjusting both learning rate β and the momentum rate γ. The basic idea of this algorithm is to introduce the performance index $E=\frac{1}{n}\sum_{k=1}^{n}e_k^2$ values at the current and previous iteration as feedback information, and adjust the learning rate and momentum strength according to the change trend of E. The adaptive algorithm is represented as follows:

If $E(t)\geq E(t-1)$ then $\beta(t)=h_d\beta(t-1)$, γ=0

If $E(t)<E(t-1)$ & $|\Delta E/E(t)|<\delta$ then $\beta(t)=h_i\beta(t-1)$, $\gamma=\gamma_0$,

If $E(t)<E(t-1)$ & $|\Delta E/E(t)|\geq\delta$ then $\beta(t)=\beta(t-1)$, $\gamma=\gamma(t-1)$

where $E(t)$ is the performance index value at the tth iteration, $\Delta E=E(t)-E(t-1)$, h_d and h_i are a decreasing factor and increasing factor respectively, $0<h_d<1$ and $h_i>1$, γ_0 is the initial momentum rate, and δ is the threshold of the relative index change rate. In this work, we set h_d=0.9, h_i=1.05 and δ=0.05.

It can be seen that the learning rate β and momentum rate γ are automatically adjusted according to the change trend of E. When the value of E is decreasing at a reasonable change rate, both β and γ remain unchanged. If the convergent speed is too slow, i.e. the decreasing rate is too small, the learning rate would be enhanced by increasing factor h_i, thus speeding up the learning process. On the other hand, once a divergent trend is observed, i.e. ΔE is non-negative, both β and γ will be reduced automatically, thus avoiding the occurrence of divergence. The learning efficiency is improved greatly due to introducing the adaptive learning algorithm. Through parameter learning, the optimal parameters of the fuzzy model are thus obtained.

E. Model Simplification

After parameter learning, the optimal fuzzy rule-base is not finally constructed. The obtained model may exhibit redundancy in terms of highly overlapping membership functions. To acquire an efficient and transparent fuzzy model, elimination of redundancy and making the fuzzy model as simple as possible is necessary. Based on the obtained fuzzy rule-based model, a rule base simplification approach to minimise the number of fuzzy sets in the universe of discourses of each input variable and remove possible redundant rules is performed as follows:

1. Removing redundant fuzzy rules
   If a fuzzy membership function is always near zero over its whole universe, i.e. $\mu_{ji}(x_j)\approx 0$, $\forall x_j\in U_j$, then remove the rule corresponding to this membership

function because this rule will almost never be fired, which means the output of this rule is always near zero.

2. Merging similar fuzzy sets

Calculate the similarity between fuzzy sets A_{ji} and A_{ki} by

$$S(A_{ji},A_{ki})=\frac{\sum_{i=1}^{n}\{\mu_{ji}(x_i)\wedge\mu_{ki}(x_i)\}}{\sum_{i=1}^{n}\{\mu_{ji}(x_i)\vee\mu_{ki}(x_i)\}} \quad (12)$$

where $i=1,\ldots,n$; $j,k=1,\ldots,c, j\neq k$.

If $S(A_{ji}, A_{ki})>\lambda_m$, i.e. fuzzy sets A_{ji} and A_{ki} are highly overlapping, then merge the two fuzzy sets A_{ji}, A_{ki} into one new fuzzy set A_j, where $\lambda_m\in(0,1)$ is the threshold for merging fuzzy sets that are similar to one another.

3. Removing redundant fuzzy sets

For each fuzzy set A_{ji}, calculate the similarity to the universal set U_i. If $S(A_{ji}, U_i)>\lambda_r$, (i.e. $\mu_{Ui}(x_i)$)$>\lambda_r$, $\forall x_i\in U_i$,), then remove A_{ji} from the antecedent of rule R_j; where $\lambda_r\in(0,1)$ is the threshold of similarity degree for removing fuzzy sets similar to the universal set.

After the completion of structure identification, parameter optimisation and rule-base simplification, the final fuzzy model is thus obtained.

3. Illustrative Examples

To verify the effectiveness of the proposed fuzzy modeling methodology and evaluate the performance of the obtained fuzzy models, different application examples including non-linear system identification and material property prediction for C-Mn steels are presented in this section.

A. Non-linear System Identification

The non-linear system is taken from [16].

$$y=x_2\sin(x_1)+x_1\cos(x_2) \quad (13)$$

100 training data were chosen randomly (instead of 441 evenly distributed data in [16]) from $0\leq x_1$, $x_2\leq\pi$, and the corresponding output data were obtained from equation (15). In order to illustrate input variable selection, two random variables x_3 and x_4 in the range of $[0,\pi]$ were added as the dummy inputs. Using the proposed input selection paradigm, the importance factors with respect to x_1, x_2, x_3, x_4 are obtained as: 0.702, 1.00, 0.295, 0.135. Clearly, x_1 and x_2 are important and thus selected as the model input variables. After structure identification and parameter learning, a 4-rule fuzzy model was produced as shown in Fig. 1. Compared to the 9-rule model with the mean-square error of 0.016 in [16], the mean-square error of this 4-rule model is 0.006. The system approximation result using the produced fuzzy model is shown in Fig.2. Using the proposed rule-base simplification approach, the four fuzzy items of x_2 were merged into two and it results in a simplified 4-rule with 6 membership functions model. The mean-square error of the simplified fuzzy model for this nonlinear system is 0.027.

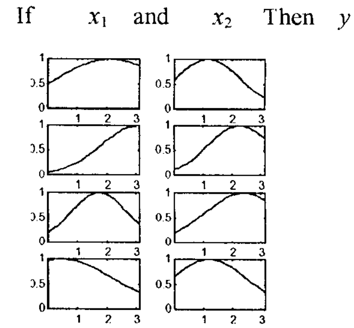

Fig.1. Fuzzy rule-based model

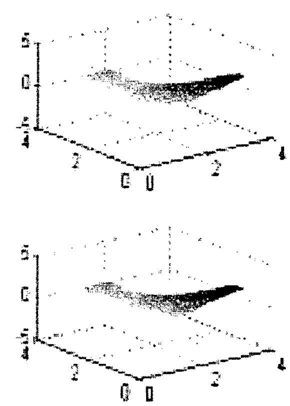

Fig.2. Simulation result.(a)Actual system output;(b)Model output

B. PROPERTY PREDICTION FOR C-Mn STEELS

The problem in modelling of hot-rolled metal materials can be broadly stated as: given a certain material which undergoes a specified set of manufacturing processes, what are the final properties of this material? By using, the proposed neural fuzzy modelling approach, we have developed composition-microstructure-property models for some classes of hot-rolled steels.

Applying the proposed input selection paradigm, 5 inputs (C, Si, Mn, Nb and Ferrite grain size $mm^{-1/2}$ ($D^{-1/2}$)), were selected from the 15 possible inputs variables. 358 industrial data were used, 50% for training and 50% for model testing. After partition validation and parameter learning, the final fuzzy models of the Mamdani type consisting of 6 rules were obtained. The rule-based fuzzy model is represented in Fig.3.

Experimental results of this model with RMSE= training and testing are shown in Fig. 4. According to the simulation result, the out-of-10% error band prediction patterns for the testing data is 2.2%. It can be seen that the fuzzy model gives good prediction and generalisation.

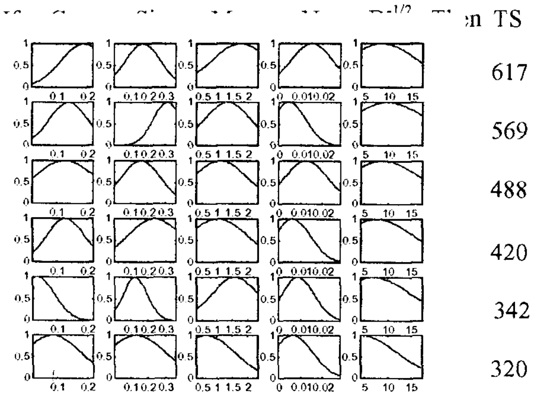

Fig.3. Fuzzy rule-based model for C-Mn steels

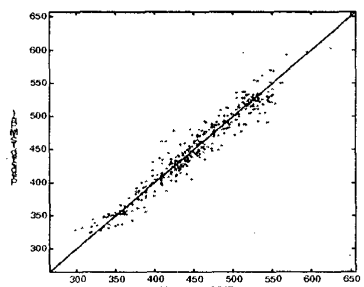

Fig.4. Measured versus predicted TS of C-Mn model

4. Conclusion Remarks

In this paper, a hybrid neuro-fuzzy modeling framework is proposed. It can be seen that a fuzzy rule-base can be generated and optimised automatically from the training data through the proposed hybrid neural fuzzy model. Due to its multi-paradigm nature, the proposed hybrid model not only takes advantage of the computational efficiency of the neural network approach but also maintains the clarity of the fuzzy rule-base paradigm, thus providing an explanation facility for the network. The proposed neural fuzzy model also provides a fast and effective mechanism for generating fuzzy models based on neural network and fuzzy clustering techniques.

Using this neurofuzzy modeling approach, the acquired fuzzy model has a simple structure and thus has low computation cost. Extensive experimental validation shows that the produced rule-based fuzzy models have satisfactory prediction accuracy and good interpretation features. Clearly, the proposed fuzzy modeling approach provides a simple and effective framework for system identification and prediction. Further improvement in the model optimisation and incorporation of linguistic information into the modeling procedure would be beneficial. Also, further work will be carried out for other industrial applications of microstructure modeling and property prediction.

References

[1] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling", *IEEE Trans. on Fuzzy Systems*, **1**(1), pp.7-31, 1993.

[2] J. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern.*, **23**(3), pp.665-685, 1993.

[3] Li-Xin Wang, "Adaptive Fuzzy Systems and Control, design and stability analysis", Prentice-Hall Inc. 1994.

[4] Chia-Feng Juang and Chin-Teng Lin, "An On-line Self-Constructing Neural Fuzzy Inference Network and Its Applications". *IEEE Trans. On Fuzzy Systems*, **6**(1), pp.12-32, 1998.

[5] J. Nie and D. A. Linkens, "Learning control using fuzzified self-organizing radial basis function network," *IEEE Trans. Fuzzy Syst.*, **1**(5), pp.280-287, 1993.

[6] Y.Yasugawa, W.Pedrycz and K.Hirota, "Construction of fuzzy models through clustering techniques", *Fuzzy Sets and Systems*, **54**, pp.157-165, 1993.

[7] Yinghur Lin and G.A.Cunningham, "A new approach to fuzzy-neural system modeling", *IEEE Trans. on Fuzzy Systems*, **3**(2), 190-197, 1995.

[8] Minyou Chen and D. A. Linkens, "A Fast Fuzzy Modelling Approach Using Clustering Neural Networks", *IEEE World Congress on Intelligent Computation* 1998, *Proceedings of Fuzzy-IEEE'98*, vol.2, pp. 1406-1411, 1998.

[9] T. Kohonen, "Self-Organizing Maps", *Springer Series in Information Science*,Vol.30, Springer-Verlag, 1995.

[10] D.A.Linkens and Minyou Chen, "Input Selection and Partition Validation for Fuzzy Modelling Using Neural Networks", *Fuzzy Sets and Systems*. **107**, pp.299-308, 1999.

[11] J. C. Bezdek, "Cluster validity with fuzzy sets," *J. Cybernetics*, vol.3, no.3, pp.58-72, 1974.

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

[13] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Trans. Patern Anal. Mach. Intell.*, vol.11, pp.773-781, 1989.

[14] M. P. Windham, "Cluster validity for the fuzzy *c*-means clustering algorithm," *IEEE Trans. Patern Anal. Mach. Intell.*, **4**(4), pp. 357-363, 1982.

[15] X. L. Xie and G. A. Beni, "Validity measure for fuzzy clustering," *IEEE Trans. Patern Anal. Mach. Intell.*, **13**(8), pp. 841-846, 1991.

[16] Yinghur Lin, G.A.Cunningham and S. V. Coggeshall, "Using fuzzy partitions to create fuzzy systems from input-output data and set the initial weights in a fuzzy meural network," *IEEE Trans. on Fuzzy Systems*, **5**(4), 614-621, 1997.