Proceedings of the 1998 IEEE
International Conference on Control Applications
Trieste, Italy 1-4 September 1998

Identification of Fuzzy Models for A Glass Furnace Process

M. Hadjili, A. Lendasse, V. Wertz and S. Yurkovich

ABSTRACT

In this paper a study is described for several approaches to the identification of models for the temperature within the melter portion of a glass furnace. The focus is on developing models from the gas input to the throat (melter outlet) temperature. Conventional linear techniques for system identification proved to be inadequate for this problem, but proved useful as baseline comparisons for further studies involving nonlinear techniques from intelligent control for model building. Various combinations of input and output variables in a variety of model structures using fuzzy and neuro-fuzzy system modeling approaches are developed, and comparisons are drawn. Approaches reported on here investigate nonlinear Takagi-Sugeno (TS) fuzzy model formulations, where a linear-in-the-parameter identification problem is formulated for various combinations of measured variables and system delays. A fuzzy-neuro formulation is then discussed for parameter selection in the TS model structure while simultaneously optimizing the membership functions associated with the inputs of the TS fuzzy system. Simulation results for data collected from an operating glass furnace process are presented.

INTRODUCTION

While linear identification is by now a well established field, thoroughly described in several textbooks and the open literature, identification of nonlinear models is still a subject of intense research. A large variety of methods exist, depending on whether one wishes (or is able) to use (i) so called white-box models, where the nonlinearities are perfectly described using prior knowledge; (ii) gray-box models, where nonlinearities can be partially suggested by physical insight (mainly nonlinear combinations of some measured signals) but still require some tuning; or, (iii) black-box models where no physical insight can be used *a priori* to guide the model builder. A comprehensive overview of nonlinear black-box modeling is given in [1]. A specific algorithm for grey-box modeling has been described in [2]. White-box models are, of course, problem dependent.

When there is no clear indication of which nonlinear combinations of measured signals to use in the regressors of the model, then neural networks and fuzzy models have been considered by several authors; see for example [3] for the neural network approach and [4], [5], [6], and [7] for work in fuzzy modeling. Following on this work, a promising structure for nonlinear identification, and subsequent control design based on the identified models, is the *functional fuzzy system*. One way to view the functional fuzzy system is as a nonlinear interpolator between the mappings that are defined by the functions in the consequents of the rules. This is especially appealing to the control engineer if one considers the topical area of gain scheduling, whereby a nonlinear controller is constructed by scheduling linear controllers designed for linear models along an operating line. That is, in the special case where the consequent takes the form of an affine mapping, we refer to the representation as a Takagi-Sugeno (TS) fuzzy system. Furthermore, if the mappings are linear, the fuzzy system (when inference is performed on the rules) essentially performs a nonlinear interpolation between the linear mappings (in the spirit of the gain scheduling controller).

This paper describes several approaches to the identification of models for the temperature within the melter portion of a glass furnace, using data from an actual process with gas flow (to the burners) as process control input and temperature at the throat (melter outlet) as process output. Although conventional linear techniques for system identification proved to be inadequate for this problem, they were useful in understanding the process for subsequent nonlinear identification studies.

GLASS FURNACE PROCESSES

The glass melting process in industrial furnaces involves the fusion of raw materials, followed by homogenization and conditioning. Thereafter, the melt is pulled out of the furnace (at the *throat*) to be processed further, e.g. in blowing of containers, or flat glass production in a float process. The energy required for the melting is generally provided by fuel or gas burners, with sometimes an additional source coming from electrodes sunk in the melt. In recent years, efforts have been spent for a better control of the melting process, with the aim of increasing the homogeneity of the product, the energy efficiency of the process and the lifetime of the furnace (see [8]).

This control requires better models describing the dynamics of the heat transfer in the melt. Besides simulation models based on physical properties, which are usually extremely expensive in terms of computing time and not appropriate for controller synthesis, identification of black-box models has been used in several applications [9], [10]. One of the main challenges is to identify an accurate, yet simple model using available measurements for a process where the operating conditions can vary significantly over time. One of the main changes of operating conditions is produced by pull changes in the furnace which are dictated by customer demand. However, a precise and reliable measurement of the pull is not always available. There are, of course, other unmeasured perturbations affecting the process (e.g., change of glass color, changes in electrical heating, gas bubbling, and so on), so that identification of linear models is not always realistic.

The focus of these studies is a collection of data supplied by Air Products, Inc., of Allentown, PA, USA,

V. Wertz and M. Hadjili are with CESAME, Catholic University of Louvain, 1348 Louvain-la-Neuve, Belgium, A. Lendasse is with DICE, Catholic University of Louvain.

S. Yurkovich is with the Department of Electrical Engineering, The Ohio State University, 2015 Neil Avenue, Columbus OH 43210-1272 USA, *yurkovich@ee.eng.ohio-state.edu*

obtained in November of 1996 from a glass furnace in operation in the Czech Republic. The data, consisting of 10,000 samples of several variables measured on the furnace, have been re-sampled using one minute to one hour sampling times for use in identification studies. Because every 20 minutes (i.e., every 20 data points) the gas flow is cut during inversion (that is, burners are "cycled" on-and-off one side, then the other), some peaks in the measurements of the process variable can result; thus, care was taken in formulation of the various data sets to remove any undesirable spikes. Sampling at one-hour intervals leaves 167 samples for identification. The identification results reported on herein are for the full scale signals (i.e., without removal of the mean).

Linear model identification for various combinations of inputs and outputs has been carried out on this data. These studies formed a baseline for comparison for the current studies which involve nonlinear modeling and custom code. That study produced four types of linear throat temperature models of the furnace:

- gas setpoint (controlled) as input and the throat temperature as output;
- actual gas as input and the throat temperature as output;
- crown temperature (inside top of melter) as input and the throat temperature as output;
- actual gas and crown temperature as inputs and throat temperature as output.

Based on those linear modeling studies, the following conclusions were possible: (i) Looking at the model step responses, as well as the dominant time constants, a one-hour sampling period was deemed appropriate. One could even consider longer sampling periods, e.g. two hours, which would have the advantage of covering an equal number of left and right inversions of the control inputs. But the available data is too short for testing such sampling periods. (ii) The best linear model simulation results exhibit variations of the throat temperature which are not accounted for by the model. These are primarily due to various nonlinearities, including the effect of load variations in the furnace, which are not measured and hence cannot be directly introduced as input in the model.

It is important to note that the results presented here, for models developed from actual furnace data, are *simulations* using the model output, as opposed to *predictive* estimates which use the real data in an "on-line" fashion for estimation. That is, past values of the output are not considered in order to compute the future values. However, some control strategies, e.g., predictive control methods, rely on predictions of the output that can be obtained using *all* information available until the present time, including the actual outputs. Prediction results are generally much better than the simulation results, and so are not presented herein.

TAKAGI-SUGENO FUZZY SYSTEM FORMULATION

In this section we will describe the Takagi-Sugeno fuzzy system formulation, and discuss how it may be tuned via recursive least squares (RLS) techniques to produce a nonlinear model of the plant between the gas input and the throat temperature output. A more complete description of the theory, with extensions, may be found in [4] and [5].

We denote the process inputs by u_i, and the process output by y. For the Takagi-Sugeno fuzzy system we use singleton fuzzification and the i^{th} rule has the form

$$\textbf{If } \tilde{u}_1 \text{ is } \tilde{A}_1^j \textbf{ and } \tilde{u}_2 \text{ is } \tilde{A}_2^k \textbf{ and}, \ldots, \textbf{and } \tilde{u}_n \text{ is } \tilde{A}_n^l$$

$$\textbf{Then } c_i = f_i(\cdot)$$

(where "$\cdot$" simply represents the argument of the function f_i). The premise of this rule is defined the same as for the rules for the standard fuzzy system (e.g., $\tilde{A}_i^j$ j^{th} linguistic value on the i^{th} universe of discourse u_i, and the notation "$\tilde{u}_i$" refers to the linguistic characterization of the input). The consequents are, of course, different from the standard fuzzy system in that instead of a linguistic term with an associated membership function, in the consequent we use a crisp function $c_i = f_i(\cdot)$ that has no associated membership function.

For this formulation we represent the premise as we would with a standard fuzzy system (e.g., minimum or product), whereas defuzzification is carried out using a center-of-gravity formulation in the manner

$$y = \frac{\sum_{i=1}^{R} c_i \mu_i}{\sum_{i=1}^{R} \mu_i} \tag{1}$$

where μ_i is the certainty of the premise of the rule (e.g., the minimum of the certainties of each of the premise terms). Note that an underlying assumption in the definition of the fuzzy system is that, regardless of inputs, $\sum_{i=1}^{R} \mu_i \neq 0$.

Expressing Equation (1) as

$$y = c^\top \xi \tag{2}$$

where "$\top$" denotes transpose, $c^\top = [c_1 \ \cdots \ c_R]$ and

$$\xi^\top = [\xi_1, \xi_2, ..., \xi_R] = \left[\frac{1}{\sum_{i=1}^{R} \mu_i}\right] [\mu_1, \ \cdots \ , \mu_R] \quad , \tag{3}$$

allows us to formulate the usual TS structure of a bias term with terms linear in the inputs and measured variables. That is, we will narrow our consequent form to the selection

$$c_i = f_i(\cdot) = a_{i,0} + a_{i,1} u_1 + \cdots + a_{i,n} u_n \tag{4}$$

(where the $a_{i,j}$ are real numbers). This fuzzy system essentially performs an interpolation between affine mappings (or, linear mappings in the case where $a_{i,0} = 0$ for all i).

Training of the system essentially consists of specifying (fixing) the membership functions associated with the "inputs" to the consequent models, and then subsequent adjustment of the $a_{i,j}$ terms of the consequents of the rules. It is possible to show that the nonlinear TS model is linear in these parameters, which allows us to use RLS techniques for identification.

To show this, note that when

$$c_i = a_{i,0} + a_{i,1} u_1 + \cdots + a_{i,n} u_n \tag{5}$$

is substituted into

$$y = \frac{\sum_{i=1}^{R} c_i \mu_i}{\sum_{i=1}^{R} \mu_i} \qquad (6)$$

we obtain the relationship

$$y = \frac{\sum_{i=1}^{R} a_{i,0}\mu_i}{\sum_{i=1}^{R} \mu_i} + \frac{\sum_{i=1}^{R} a_{i,1}u_1\mu_i}{\sum_{i=1}^{R} \mu_i} + \cdots + \frac{\sum_{i=1}^{R} a_{i,n}u_n\mu_i}{\sum_{i=1}^{R} \mu_i}. \qquad (7)$$

This is key to the development, since now we can employ the definition of ξ above by letting

$$\phi = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \vdots \\ \xi_R \\ u_1\xi_1 \\ u_1\xi_2 \\ \vdots \\ u_1\xi_R \\ \vdots \\ u_n\xi_1 \\ u_n\xi_2 \\ \vdots \\ u_n\xi_R \end{bmatrix}, \quad \theta = \begin{bmatrix} a_{1,0} \\ a_{2,0} \\ \vdots \\ a_{R,0} \\ a_{1,1} \\ a_{2,1} \\ \vdots \\ a_{R,1} \\ \vdots \\ a_{1,n} \\ a_{2,n} \\ \vdots \\ a_{R,n} \end{bmatrix} \qquad (8)$$

where θ is the vector of unknown parameters, so that it is straightforward to write the summary relationship

$$y = \theta^{\top}\phi \quad , \qquad (9)$$

which is a linear-in-the-parameter model. As is the usual case in such formulations, specification of the "regression vector" (in our case, completely specified once the inputs to the fuzzy system are known via the ξ) leads to the use of standard least squares methods to find the value of the parameter vector θ.

Identification Results with RLS Training

Membership functions for the model inputs are fixed at the beginning of the identification experiment. The centers and spread are chosen systematically; the centers are chosen such that the membership functions lie somewhere in the middle of the training data, and at regular spacing. For these studies, the shapes of the membership functions for the inputs of the model are

$$\mu(i) = \prod_{in=1}^{n} \mu_{in}(i) \quad , \qquad (10)$$

where the index $i = 1 \ldots R^n$, R is the number of membership functions per input, n is the number of inputs,

$$\mu_{in}(i) = e^{-\frac{1}{2}\left(\frac{x(in,k)-c(in,j_{in})}{\sigma_{in}}\right)^2} \quad , \qquad (11)$$

where $x(in,k)$ represents the input data, index k runs through the length of the data, index $in = 1, \ldots n$, and each index j_{in} runs from 1 to R. The membership function centers $(c(\cdot,\cdot))$ and their spread are computed according to the general formula

$$c(in, j_{in}) = MIN_{in} + (j_{in} - 1)ST(in) \qquad (12)$$

$$\sigma_{in} = 2ST(in) \quad , \qquad (13)$$

where MIN_{in} (MAX_{in}) is the minimum (maximum) value over all the data for input in, and

$$ST(in) = (scale)\frac{MAX_{in} - MIN_{in}}{R-1} \quad . \qquad (14)$$

The parameter *scale* is inserted to vary the spread/spacing of the membership functions.

Many model structures using this general TS formulation have been tested on the data supplied, for various numbers of data pairs, and for various combination of fuzzy system inputs. For purposes of brevity, we only present a summary of those results here. In all plots to follow, the output of the simulated model is represented by the dashed line, whereas the actual data is represented by the solid line.

Three Membership Functions per Input: We begin with a summary of results of models consisting of a reasonable but large number of parameters $a_{i,j}$. Consider first the model structure derived above, and represented in Equation 4:

$$c_i = a_{i,0} + a_{i,1}y(k-1) + a_{i,2}u(k) + a_{i,3}u(k-1) \quad , \qquad (15)$$

where data sampled every 60 minutes is used (that is, 167 data pairs), and where now we represent the model inputs in terms of the throat temperature measurement $y(k)$ and the actual process inputs $u(k)$ (sampled versions of the actual applied gas). With three membership functions ($R = 3$) for each of three consequent inputs ($n = 3$), this results in $R^n = 3^3 = 27$ rules, and 108 model parameters (four parameters per rule). A typical result of this identification is given in Figure 1, for the *scale* parameter set to 0.4. We note that little improvement results if twice the data is used (sampled every 30 minutes, for 334 data pairs).

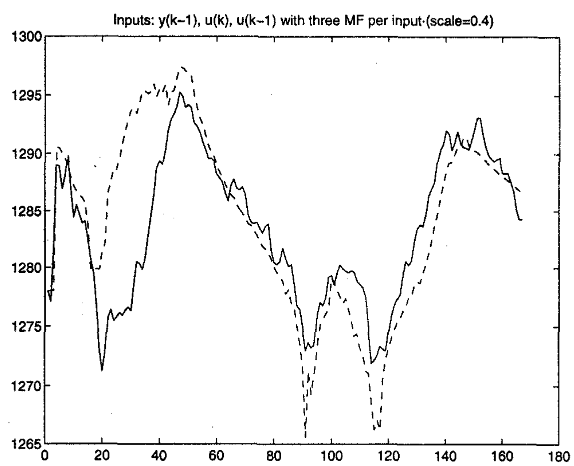

Fig. 1. TS model with throat temperature measurement $y(k-1)$, and gas inputs $u(k)$ and $u(k-1)$; in this case, $R = 3$, and data is sampled at 60-minute intervals ($scale = 0.4$).

Next we consider a structure where the process input variables (actual gas data) are delayed upon formulation in the model. That is, we consider

$$c_i = a_{i,0} + a_{i,1}y(k-1) + a_{i,2}u(k-5) + a_{i,3}u(k-7) \quad . \tag{16}$$

Again for $R = 3$ (three membership functions per model input), we simulate the system with the result shown in Figure 2, for a *scale* factor 1.5. The results are considerably better in this case, indicating that incorporation of delay into the consequent models can significantly improve the results.

Fig. 2. TS model with throat temperature measurement $y(k-1)$, and gas inputs $u(k-5)$ and $u(k-7)$; in this case, $R = 3$, and data is sampled at 60-minute intervals (*scale* = 1.5).

Two Membership Functions per Input: When the number of membership functions per input is reduced to $R = 2$, the total number of parameters to be identified is reduced significantly, and simulations/identifications may be carried out at a much more rapid pace[1]. Thus, we can include results for models of increased "dimension" in the sense of increasing the number of model inputs to four. Consider, for example, the structure

$$c_i = a_{i,0} + a_{i,1}y(k-1) + a_{i,2}u(k-5) + a_{i,3}u(k-7) \quad . \tag{17}$$

In this case, the total number of model parameters (the $a_{i,j}$) is only 32. The result for the best choice of scaling factor is shown in Figure 3. Space limitations do not permit illustrations of results for various other structures (for example, slightly better results can be obtained using an additional premise term of $y(k-2)$, resulting in a model with 40 parameters).

TS Modeling with Data Clustering

One of the drawbacks of the previous methods for fuzzy modeling is that the number of rules is determined *a priori*, even when the user has no clear indication about how many rules to use. Since the number of rules has a major influence on the number of parameters to be identified, this drawback can become cumbersome in some applications. Comparisons must be made to test several model structures, with different numbers of rules.

[1] On the 150MHz Pentium PC, a system with $R = 3$ and $n = 4$ took approximately 150 minutes to execute with the Matlab code.

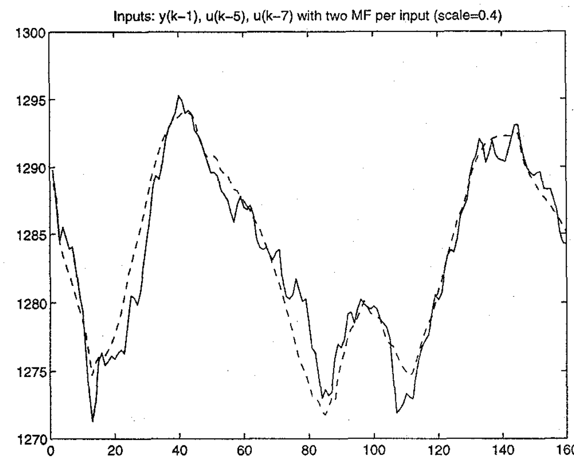

Fig. 3. TS model with throat temperature measurement $y(k-1)$, and gas inputs $u(k-5)$ and $u(k-7)$; in this case, $R = 2$, and data is sampled at 60-minute intervals (*scale* = 0.4).

As an alternative, data clustering algorithms can be used in conjunction with fuzzy modeling techniques. Basically, these data clustering algorithms are used to determine the number of data clusters in a given identification (or training) data set, each cluster yielding one rule of the fuzzy model. Then, projection of these data clusters on each of the components of the input space will determine the membership functions. In TS fuzzy modeling, the parameters of the rule consequents are subsequently identified using least squares techniques. Results for one of these clustering methods, namely mountain clustering, are given in this section. We refer to [11] for further details on this and other algorithms.

For the sake of comparison with previous results, several models using various combinations of premise structures have been estimated (we will only illustrate results for one here). Best results were obtained for a structure using $y(k-1)$, $u(k-5)$ and $u(k-7)$ in the premises of the rules, where, as before, $y(k)$ is the throat temperature and $u(k)$ is the gas flow. This means that the training data set is made of triples $\{y(k-1), u(k-5), u(k-7)\}$.

The mountain clustering algorithm identifies five clusters, with consequent models of the form

$$c_i = A_{1,i}y(k-1) + B_{1,i}u(k-5) + B_{2,i}u(k-7) + \gamma_i \quad , \tag{18}$$

where now, to distinguish from previous results, we alter our notation slightly to indicate the (previously defined) $a_{i,j}$ parameters associated with the model inputs due to throat temperature measurements ($A_{1,i}$) and due to gas flow ($B_{1,i}$, $B_{2,i}$). The number of parameters in this fuzzy model is 16, and simulation results are shown in Figure 4.

NEURO-FUZZY TS MODELING WITH PREMISE OPTIMIZATION

As competing approaches for nonlinear modeling, neural networks and fuzzy systems each have strong

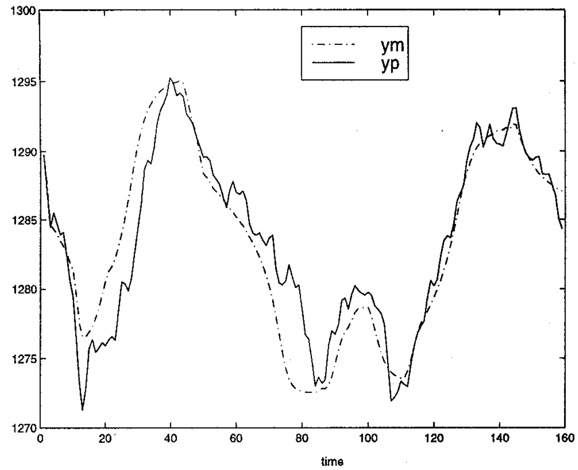

Fig. 4. Result of clustering, with delays in model.

and weak points. Neural networks can be considered as a powerful means of solving a nonlinear optimization problem, which is one way of recasting a nonlinear modeling task: one tries to minimize the modeling error, i.e., the difference between the real output of a system and the output computed via a nonlinear model which is basically a nonlinear function of past inputs and outputs. For a multilayered neural network, the output of the network can be a complicated nonlinear function of the network inputs so that even when the modeling error is small, little insight can be gained from the NN model.

Fuzzy models, on the other hand, often correspond to a description of the system that can be "linguistically" rephrased. Hence, the insight given by a fuzzy model is normally higher. However, even if a "gross" description of a fuzzy model can be easily obtained from intuitive knowledge, the fine tuning of the fuzzy model parameters, and in particular those parameters defining the fuzzy sets, is not always an easy task.

Hence, it may seem appealing to combine the advantages of both approaches: keep a linguistically meaningful interpretation of the model, but provide an efficient method for the optimization of the model parameters. This can be done using the ANFIS (adaptive neuro-fuzzy inference system) method [11], summarized next.

Problem Formulation

Figure 5 illustrates the ANFIS architecture; a multilayered neural network described in the following:

- Layer one contains the description of the membership functions corresponding to each input of the network, which are the inputs entering in the premises of a given TS fuzzy model. The number of nodes in this layer equals the number of inputs (n) in the premises, times the number of linguistic values ($m_i, i = 1, \ldots, n$) for each input. These nodes are parameterized and the parameters are referred to as premise parameters.
- Layer two contains as many nodes as there are rules in the TS model (typically $\prod_{i=1}^{n} m_i$) and simply performs the product of the membership values evaluated in layer one.
- Layer three contains as many nodes as there are rules and each node i calculates the ratio of the rules firing strength to the sum of all rules firing strengths.
- Layer four contains as many nodes as there are rules and each node is a parameterized function of past inputs, usually a linear or affine function. The parameters in these nodes are referred to as consequent parameters.
- Layer five consists in one single node which computes the summation of all incoming signals, which represents the weighted average of the expressions described in layer four.

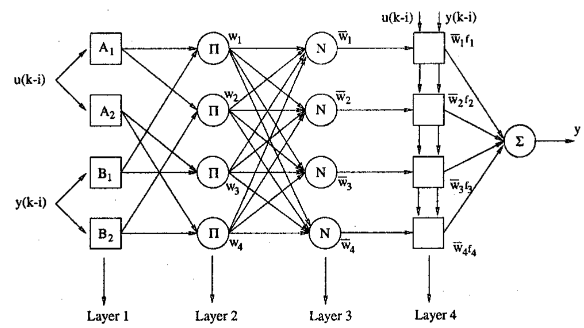

Fig. 5. ANFIS network structure.

From the description of this system, it is obvious that the overall output is linear in the consequent parameters and nonlinear in the premise parameters. Moreover, the results described above (where the input membership functions were scaled in various ways) indicates that some judicious tuning of the premise characterizations can improve results. This is the approach we adopt in the next section, using the ANFIS network structure to automate this tuning procedure.

Identification Results

Towards improved tuning of the TS model structure along the lines discussed above, [11] suggests application of a hybrid learning algorithm where the consequent parameters are estimated in a forward pass using the least squares method, while the premise parameters are estimated in the backward pass by a steepest descent method using back-propagated error signals; we refer to [11] for the details of the computations. This method has been applied to the glass furnace data with different choices for the inputs.

Inspired by the fuzzy modeling described above, and the best linear model (which is an ARX (2,3,1) structure), we have chosen to describe the functions in layer four of the network as

$$c_i = \alpha_1^i y(k-1) + \alpha_2^i y(k-2) + \alpha_3^i y(k-3) + \beta_1^i u(k-1) + \beta_2^i u(k-2) + \beta_3^i u(k-3) + \gamma^i \quad , \tag{19}$$

for appropriately defined model parameters α_j^i and β_j^i. The output of the model, denoted $y(k)$, is the throat temperature of the furnace, while the input $u(k)$ is the gas flow. However, using $y(k-i)$, $u(k-i)$, $i = 1, 2, 3$, as inputs for the network would have led to a very large number of rules. In addition, the dynamics of the process are slow, so that $y(k-i)$, $i = 1, 2, 3$, or $u(k-i)$, $i = 1, 2, 3$, are simultaneously big or small.

Hence, we will only take $y(k-1)$ and $u(k-1)$ as inputs and for each of these we take two different membership functions, parameterized as follows:

$$\mu_A(x) = \frac{1}{1 + |\frac{x-c_i}{a_i}|^{2b_i}} \quad . \tag{20}$$

We then have four rules (and four nodes in each of layers three and four) and the number of premises parameters is 12.

Evaluation of the membership functions for the inputs of the network (after training) indicates that only the a_i parameters of the membership functions in Equ. (20) change significantly from their initial values, which indicates that in this example one could have simplified the steepest descent optimization by tuning only four parameters, instead of 12. Figure 6 shows the result of the fuzzy model in simulation (layer four uses past values of the process input and past values of the *model* output).

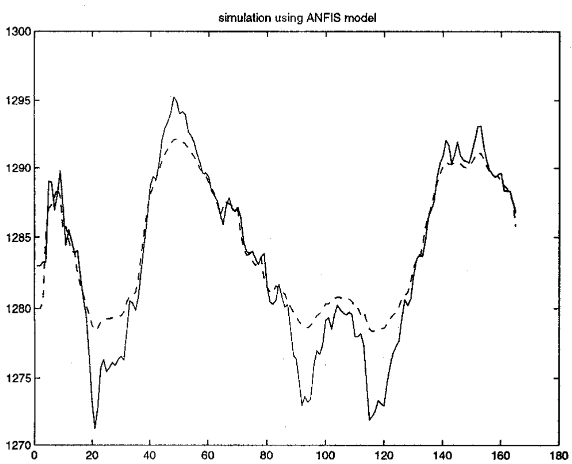

Fig. 6. Simulation using ANFIS network.

SUMMARY

Different approaches to the nonlinear identification of models for control design have been described. While the basic structure of the model has been in the realm of the functional fuzzy system approach (specifically, the TS model structure), three different approaches have been described, revealing interesting results concerning the size of the model and the degree to which the TS model structure can approximate the given nonlinear process. The baseline TS modeling (with RLS tuning) has produced some of the best results, but required extensive (although not exhaustive) tuning (in terms of model input selection, membership function selection, and so on). The clustering technique eliminated some of the uncertainty associated with the baseline TS approach in terms of specifying the number of rules in the model; the results presented were comparable to the baseline TS results, leading one to believe that further investigation in that area may be warranted.

The ANFIS modeling approach has led to some possible conclusions regarding the number of membership functions necessary in the TS model structure. Although in that work the problem was formulated as an optimization problem, where parameters associated with bell-shaped input membership functions were optimized by a neural network, the result was useful in the TS modeling studies, and each technique generated ideas for the other. That is, each aimed at investigating the use of fewer functions for the TS model inputs, and in varying the shape and spread of the membership functions.

In the baseline TS modeling (with RLS tuning) and the ANFIS structure work, the resulting models are "complete" in the usual sense ([4]), but there may be very few data points contributing significantly to some rules. This implies that the corresponding parameters in the consequents are perhaps not identified very well. In contrast, in the algorithm using TS modeling with clustering, only rules corresponding to data "clusters" are identified. Hence, there are normally many points contributing to each rule, but the global model is not complete so that it may not work with other data corresponding to another region of the input space. This, of course, is important if one considers that the results reported on in this work are for modeling and simulation *on the same data.* Ordinarily one would like to model the process with one set of data, then test it with yet another. Because of the fact that we are operating with a limited amount of data, this was not possible in this study. A next step, therefore, would be to simulate the models obtained herein on another collection of data from the same process. This remains as a future direction.

REFERENCES

[1] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modelling in system identification : A unified overview," *Automatica*, vol. 31, no. 12, pp. 1691–1724, 1995.

[2] S. Graebe, *Theory and Implementation of Gray-Box Identification.* PhD thesis, Royal Institute of Technology, Sweden, 1990.

[3] S. Chen and S. Billings, "Neural networks for nonlinear dynamic system modelling and identification," *International Journal of Control*, vol. 56, pp. 319–346, 1992.

[4] K. M. Passino and S. Yurkovich, *Fuzzy Control.* Menlo Park, CA: Addison-Wesley Longman, Inc., 1998.

[5] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 15, pp. 116–132, Jan. 1985.

[6] J. C. Bezdek, C. Coray, R. Gunderson, and J. Watson, "Detection and characterization of cluster substructure: I. Linear case: Fuzzy c-lines, II. Fuzzy c-varieties and convex combinations thereof," *SIAM J. Appl. Math.*, vol. 40, pp. 339–372, 1981.

[7] Y. Nakamori and M. Ryoke, "Identification of fuzzy prediction models through hyperellipsoidal clustering," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 24, pp. 1153–1173, 1994.

[8] H. Muysenberg and R. Bauer (Editors), (TNO-TPD, Eindhoven), Maastricht, *Proceedings of the Workshop on "Advanced control for industrial glass melters"*, May 1997.

[9] V. Wertz, J. F. Simon, and M. Gevers, "Adaptive control of the temperature of a glass furnace," in *Proc. IFAC Symp. on Adaptive Control and Signal Processing*, (Grenoble, France), pp. 555–560, July 1992.

[10] R. Haber, J. Hetthessy, L. Keviczky, I. Vajk, A. Feher, N. Czeiner, Z. Csaszar, and A. Turi, "Identification and adaptive control of a glass furnace," *Automatica*, vol. 17, pp. 175–185, 1981.

[11] J.-S. R. Jang, C.-T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing.* Prentice-Hall, 1997.