Intelligent Nonlinear Predictive Control

Ebrahim A. Al-Gallaf

Department of Electrical and Electronics Engineering
College of Engineering, University of Bahrain, P. O. Box 13184, Kingdom of Bahrain.

***Abstract* — This research article presents a Fuzzy structure for a Model Predictive Control (MPC) system. MPC theorem has recently been incorporated with fuzzy models. Such an integration provides controller design methods for an MPC control system. The paper concentrates on aspects of fuzzy based MPC for multivariable systems. Mathematical formulation of linearized MPC is utilized to introduce the concept of fuzzy based MPC scheme, then fuzzy MPC is constructed based on a modeled pH reactor.**

***Index Terms* — Fuzzy Control, MPC, Nonlinear System.**

I. INTRODUCTION

Traditionally, fuzzy controllers have been designed without an explicit model of the process being controlled. However, in fuzzy systems, mathematical models are explicitly used [1]. It has been found, that predictive control principle has recently been incorporated with fuzzy models. This provides design methods for fuzzy model based controllers, since predictive methods have several advantages that make them good candidates for industrial applications.

An important requirement for any system identification technique is the ability to exploit available priori knowledge. Many conventional approaches rely on depth physical knowledge describing the system. However, for complex ill-defined systems such knowledge is unavailable or limited. In these instances, an expert can often describe the behavior of the system using natural language.

Since Zadeh's first paper [2], fuzzy algorithms behavior has been used to build models based on such humanistic descriptions. Despite the apparent success of fuzzy systems, there are many aspects of their behavior which are unsuited to system identification and modeling. The major criticism is that these models are mathematically opaque, and there is no formal mathematical representation of the system's behavior. In addition, due to the vagueness and subjectivity of natural language statements, fuzzy systems based on qualitative knowledge alone are unlikely to adequately model simple system. To circumvent these inadequacies, as in conventional empirical modeling, available data should be used to adjust and validate the model's behavior. Efforts to combine both empirical and qualitative modeling have lead to the development of *fuzzy modeling* techniques. Such techniques allow both linguistic system description and empirical data to be fully utilized during system identification cycle.

II FUZZY SYSTEM IDENTIFICATION

Fuzzy models are useful for describing processes where the underlying physical mechanisms are not completely known and where a process behavior is understood in qualitative terms. An important property of fuzzy models is their capability to represent nonlinear dynamic systems. The global operation of a nonlinear process is divided into several local operating regions. Within each region R_i, a reduced order linear model in ARMAX form is used to represent the process behavior. This is not restrictive, and any appropriate model forms can be used. Fuzzy sets are used to define the process operating conditions such that fuzzy dynamic model of a nonlinear process can be described in the following way :

$$\begin{aligned} R_i: &\quad \textit{If operating condition} \\ \textit{Then} &\quad \hat{y}_i(k) = \sum_j^o a_{ij}\, y(k-j) + \sum_j^i b_{ij}\, u(k-j) (i = 1,2,\ldots,r) \end{aligned} \tag{1}$$

Finally the associated model output is via the center of gravity as given in (2), [3]:

$$\hat{y}(k) = \frac{\sum_{i=1}^{r} \mu_i \hat{y}_i(k)}{\sum_{i=1}^{r} \mu_i} \tag{2}$$

III Fuzzy Models via Multi-layer Neural Networks

Fuzzy model described in section (2) can be represented by a special type of network topology which is termed here a neuro-fuzzy. Fuzzy reasoning is capable of handling uncertain and imprecise information while a neural network is capable of learning from examples. Neuro-fuzzy intend to combine the advantages of both fuzzy reasoning and neural networks.

A. *Neuro-Fuzzy Architecture*

For simplicity, we assume the fuzzy inference system under consideration has two inputs x(*k–1*) and y(*k–R*) and one output *y*(*t*). For instant, if the rule base contains two fuzzy if-then rules of Takagi and Sugeno's type [4], a rule can thus be expressed as:

Rule 1:
If x(k-1) is A_1 and y(k-2) is B_1,

then $f_1 = p_1 x(k-1) + q_1 y(k-2) + r_1$

Rule 2:
If x(k-1) is A_2 and y(k-2) is B_2,

then

$$f_2 = p_2 x(k-1) + q_2 y(k-2) + r_2 \qquad (3)$$

where p, q, and *r* are constants and called parameter set. That is, the if parts of the rules are same as in the ordinary fuzzy if-then rules, then parts are linear combinations of the input variables.

B. *Training of the Neuro-Fuzzy System*

From the designed neuro-fuzzy architecture shown in Fig. 1., a given values of premise parameters, the entire output is expressed as a linear combinations of the consequent parameters. More precisely, output $\hat{\mathbf{y}}$ in Fig. 1. can be rewritten as:

$$\hat{y}_m = \frac{y_1}{y_1 + y_2}\hat{y}_1 + \frac{y_2}{y_1 + y_2}\hat{y}_2 + \cdots + \frac{y_{m-1}}{y_m + y_{m-1}}\hat{y}_{m-1}$$

$$\hat{y}_m = \bar{y}_1\hat{y}_1 + \bar{y}_2\hat{y}_2 + \ldots + \bar{y}_{m-1}\hat{y}_{m-1}$$

$$\hat{y}_m = (\bar{y}_1 x(k-1))\,p_1 + (\bar{y}_1 y(k-2))\,q_1 + (\bar{y}_1)r_1 + (\bar{y}_2 x(k-1))\,p_2 + (\bar{y}_2 y(k-2))\,q_2 + (\bar{y}_2)r_2 + \cdots$$

(4)

which is linear in the consequent parameters $(p_1, q_1, r_1, p_2, q_2 and r_2)$. As a result, we have:

$S =$ *set of total parameters*

$S_1 =$ *set of premise parameters*

$S_2 =$ *set of consequent parameters*

The consequent parameters thus identified are optimal (in the consequent parameter space) under the condition that the premise parameters are fixed.

C. *Parameter Identification*

A model's weights are conventionally identified by performing maximum likelihood estimation. Given a training data set $\mathbf{Z}^N = \{\mathbf{y}(k), \mathbf{x}(k)\}_{k=1}^N$ the task is to find a weight vector which minimizes the following cost function:

$$J_N(\mathbf{w}) = \frac{1}{N}\sum_{k=1}^{N}[\mathbf{y}(k) - \hat{\mathbf{y}}(\mathbf{x}(k), \mathbf{w})]^2 \qquad (5)$$

As the model, $\hat{\mathbf{y}}(\mathbf{x}(k), \mathbf{w})$, is nonlinear with respect to the weights, linear optimization techniques cannot be applied. Instead the popular Truncated Newton nonlinear optimization algorithm is employed. As an alternative a form of back fitting could be applied to models, but given the slow convergence times of back fitting the direct approach is preferred.

IV Fuzzy Model-based Predictive Control

In the last section we presented neuro-fuzzy systems. In this section we concentrate on employing these systems in model predictive control. Consider the following n inputs and m outputs model :

$$\hat{y}_1(k) = \sum_{j=1}^{n_{11}} c_{11}(j)u_1(k-j) + \sum_{j=1}^{n_{12}} c_{12}(j)u_2(k-j) + \cdots + \sum_{j=1}^{n_{1n}} c_{1n}(j)u_n(k-j)$$

$$\hat{y}_2(k) = \sum_{j=1}^{n_{21}} c_{21}(j)u_1(k-j) + \sum_{j=1}^{n_{22}} c_{22}(j)u_2(k-j) + \cdots + \sum_{j=1}^{n_{1n}} c_{2n}(j)u_n(k-j)$$

$$\hat{y}_m(k) = \sum_{j=1}^{n_{m1}} c_{m1}(j)u_1(k-j) + \sum_{j=1}^{n_{m2}} c_{m2}(j)u_2(k-j) + \cdots + \sum_{j=1}^{n_{mn}} c_{mn}(j)u_n(k-j)$$

(6)

in which $(\hat{y}_1, \ldots, \hat{y}_m)$ are process outputs, $(u_1, u_2, \ldots, u_n)$ are control variables, $c_{k1km}(j)'s$ and $n_{k1km}'s(k_1 = 1,2; k_2 = 1,2; j = 1, \ldots, n_{knkm})$ are the parameters and orders of the model. The model parameters $c_{knkm}(j)'s$ are constant for each given (k_1, k_m, j), since (6) is a linear time-invariant model.

At time index (k), the controller needs to determine the control action $(u_1(k),\ldots,u_n(k))$ based on feedback $(y_1(k),\ldots,y_m(k))$ to drive a process to reach the desired outputs $(r_1,\ldots,r_m)$. In predictive control, prediction equations should be developed to predict the outputs. (7) can directly yield the following recursive prediction equations, for a two inputs with two outputs system.

$$
\begin{aligned}
\hat{y}_1(k+1) = \hat{y}_1(k) &+ \sum_{j=1}^{n_{11}} c_{11}(j,u_2(k-j))u_1(k-j) \\
&- \sum_{j=1}^{n_{11}} c_{11}(j,u_2(k-1-j))u_1(k-1-j) \\
&+ \sum_{j=1}^{n_{12}} c_{12}(j,u_1(k-j))u_2(k-j) \\
&- \sum_{j=1}^{n_{12}} c_{12}(j,u_1(k-1-j))u_2(k-1-j) \\
\hat{y}_2(k+1) = \hat{y}_2(k) &+ \sum_{j=1}^{n_{21}} c_{21}(j,u_2(k-j))u_1(k-j) \\
&- \sum_{j=1}^{n_{21}} c_{21}(j,u_2(k-1-j))u_1(k-1-j) \\
&+ \sum_{j=1}^{n_{22}} c_{22}(j,u_1(k-j))u_2(k-j) \\
&- \sum_{j=1}^{n_{22}} c_{22}(j,u_1(k-1-j))u_2(k-1-j)
\end{aligned} \tag{7}
$$

where $(k \geq 1)$ and with initials

$$
\begin{aligned}
\hat{y}_1(k) &= y_1(k) \\
\hat{y}_2(k) &= y_2(k)
\end{aligned} \tag{8}
$$

where notations $c_{11}(j,u_2(k-j)),\ldots$, emphasize that $c_{11}(j),\ldots$, are dependent on $u_2(k-j))$. Since we are dealing with constrained linear MPC formulation, to achieve a good control, it is required the following cost function to be minimized:

$$
\begin{aligned}
&\min_{\mathbf{u}(k)} J = \frac{1}{2}\mathbf{u}(k)^T \underset{u}{\mathbf{H}}\,\mathbf{u}(k) - \mathbf{G}(k+1)^T\mathbf{u}(k) \\
&\textit{subject to} \\
&\qquad \underset{u}{\mathbf{C}}\,\mathbf{u}(k) \geq \underset{u}{\mathbf{c}}(k+1) \\
&\qquad \mathbf{u}_{\min} \leq \mathbf{u}(k) \leq \mathbf{u}_{\max}
\end{aligned} \tag{9}
$$

where

$$\underset{u}{\mathbf{H}} = \xi_p^m \Gamma^T \Gamma \xi_p^m + \Lambda^T \Lambda \qquad (\textit{the QP Hessian matrix})$$

and

$$\mathbf{G}(k+1) = \xi_p^m \Gamma^T \Gamma \mathbf{e}(k+1) \quad (\textit{the QP gradient vector})$$

Solution of (9) by a **QP** algorithm at each sampling interval (k) produces an optimal set of moves u(k) which satisfies the constraints. Since $\underset{u}{\mathbf{H}}$ is likely to be fixed at all sampling intervals, a parametric QP algorithm is used to reduce on-line computation time.

V MODEL PREDICTIVE CONTROL FOR NONLINEAR PROCESSES

The principal components of a linearized MPC system are a reference neuro-fuzzy model, process output predictor, optimization routines or an adaptation mechanism and MPC controller. In this sense, the MPC embodies the linearized model parameters of the overall system. In practice, many nonlinear processes are approximated by reduced order models, possibly linear, which are clearly related to the underlying process characteristics. However, these models may only be valid within certain specific operating ranges. When operating conditions change, different model may be required to be employed or the model parameters may need to be adapted. Our approach to the modeling of nonlinear processes (via neuro-fuzzy system) is to divide the whole envelope of process operation into several operating regions, hence to use a local reduced order model to approximate the process in each region. The role of the MPC mechanism is to select the model and the tuning parameters of the controller in response to the error between the outputs of neuro-fuzzy reference model and plant [5].

A. Linear Fuzzy Models

A Takagi-Sugeno-Kang fuzzy model, is constructed typically from the following rules [6] :

$$
\begin{aligned}
&IF\, u_1(k-1)\, is\, C_1^l\, and \cdots and\, u_n(k-n)\, is\, C_n^l, \\
&THEN\, y^l(k+1) = C_0^l + C_1^l u_1(k+1) + \cdots + C_n^l u_n(k+n)
\end{aligned} \tag{10}
$$

where C_i^l are fuzzy sets, C_i^l are constants, and l = 1,2, ... , m. That is, the IF parts of the rules are the same as in the ordinary fuzzy IF-THEN rules, but the THEN parts are linear combinations of the input variables. Given an input $\mathbf{u} = (u_1,\ldots,u_n)^T$, the output $\hat{\mathbf{y}}(k)$ of a fuzzy system is computed as the weighted average of the $\mathbf{y}^l$'s, that is,

$$\hat{\mathbf{y}}(k) = \frac{\sum_{l=1}^{M} \mathbf{y}^l \mathbf{w}^l}{\sum_{l=1}^{M} \mathbf{w}^l} \qquad (11)$$

In which the weights $\mathbf{w}^l$ are computed as

$$\mathbf{w}^l = \prod_{i=1}^{n} \mu c_i^l(\mathbf{x}_i) \qquad (12)$$

If an output of a fuzzy model appears as one of its input, we obtain a dynamic TSK fuzzy system. A dynamic TSK fuzzy system is constructed from the following rules :

$$IF\ y(k) is A_1^p\ and \cdots and\ y(k-n+1) is A_n^p\ and\ \mathbf{u}(k) is B^p$$
$$THEN\ y(k+1) = a_1^p y(k) + \cdots + a_n^p y(k-n+1) + b^p \mathbf{u}(k)$$
(21)

where $A_i^p and B^p$ are fuzzy sets, $a_i^p and b^p$ are constants, p=1,2,...,N, u(k) is the input to the system, and $\mathbf{x}(k) = (\mathbf{x}(k), \mathbf{x}(k-1), \ldots, \mathbf{x}(k-n+1))^T$ is the state vector of the system. Hence output of the dynamic fuzzy system is computed as :

$$\mathbf{x}(k+1) = \frac{\sum_{p=1}^{N} \mathbf{x}^p(k+1) \mathbf{v}^p}{\sum_{p=1}^{N} \mathbf{v}^p} \qquad (13)$$

where $\mathbf{x}^p(k+1)$ is given in (13) and

$$\mathbf{v}^p = \prod_{i=1}^{n} \mu A_i^p[\mathbf{x}(k-i+1)] \mu B^p[\mathbf{u}(k)] \qquad (14)$$

Hence, one can express the Least-Squares solution of **MPC** equations as the following quadratic minimization problem, with linear plant model obtained from the neuro-fuzzy system :

$$\min_{\mathbf{u}(k)} J = \frac{1}{2}\left[\xi_p^n \mathbf{u}(k) - \mathbf{e}(k+1)\right]^T \Gamma^T \Gamma \left[\xi_p^n \mathbf{u}(k) - \mathbf{e}(k+1)\right] + \frac{1}{2}\mathbf{u}(k)^T \Lambda^T \Lambda \mathbf{u}(k)$$
$$\Delta\mathbf{u}(k|k) = [(\xi_p^n)^T \Gamma^T \Gamma \xi_p^n + \Lambda^T \Lambda]^{-1} (\xi_p^n)^T \Gamma^T \Gamma \times [\mathbf{r}(k+1|k) - \hat{\mathbf{y}}(k+1|k)]$$
(15)

V SIMULATION RESULTS

The proposed technique has been successfully applied to a multivariable nonlinear system, a *pH* neutralization. Results of simulated identification of the nonlinear dynamic plant using neuro-fuzzy are presented. Hence, results of employing *neuro-fuzzy* model of Fig. 1. in predictive control systems are discussed at this stage.

A. Neuro-Fuzzy System For a pH Reactor

A *pH* neutralization reactor is a multivariable nonlinear process. The process gains differ dramatically at different *pH* ranges. The steady state relationship between acid flow rate (f_1) and *pH* (*y*) in the reactor shows a nonlinear relation. It was found that the process gain is very high in the medium *pH* region, while it is quite low in both low and high *pH* regions. A neuro-fuzzy model is developed to model the nonlinear dynamic relationships between the acetic acid flow rate (f_1), the flow rate (f_2) and concentration (c_2) of sodium hydroxide and the *pH* in the reactor. Such nonlinear relationship suggests that the process operation can be partitioned into several regions based on the reactor *pH*. Hence, we first divided the process into three operating regions: *pH* low, *pH* medium, and *pH* high.

B. Neuro-Fuzzy Training

To generate training and testing data, random perturbations are added to $u_1, u_2 and u_3$. Five hundred data points are generated as shown in Fig. 2. After training the neuro-fuzzy system, a typical identified fuzzy model for three regions are :

R_1 : *If pH is low* :

$$y(t) = 0.1837\ y(t-1) + 0.0223\ u_1(t-1) + 0.0444\ u_2(t-1) + 0.0337\ u_3(t-1)$$

R_2 : *If pH is medium* :

$$y(t) = 0.201\ y(t-1) + 0.1912\ u_1(t-1) + 0.2777\ u_2(t-1) + 0.2712\ u_3(t-1) \qquad (16)$$

R_3 : *If pH is high* :

$$y(t) = 0.3028\ y(t-1) + 0.0097\ u_1(t-1) + 0.0098\ u_2(t-1) + 0.0243\ u_3(t-1)$$

C. Membership Functions Identification

The identified final membership functions are plotted in Fig. 3. to Fig. 4. It is interesting to observe that the sharp changes of the training data surface around the origin is accounted for by the moving of the membership functions toward the origin. Changes of membership shape from initial to final shows that the fuzzy system has in fact made a correct understanding of how the fuzzy system list the fuzzy rules.

D. Model Validation

To validate the constructed fuzzy model output, Fig. 5. shows a comparison between the actual process output and model output. From the figure it is clear how the process has been modeled with minimum error. The fuzzy model is tested by some testing data. It can be seen that the fuzzy model is very accurate, it has an RMS error of 0.0988 for the training data.

E. Surface Plot of Fuzzy Rules and MPC Response

In order to asset the behavior of the obtained fuzzy models, *three dimensional plots of the fuzzy rules* are viewed as shown in Fig. 6. In this sense, the predictor (model) output is obtained via the summation of identified fuzzy regions. Each fuzzy region over the entire 3D fuzzy surface represents a linear region. For instant, in Fig. 6, effect of pH (*k-1*) and sodium flow at (*k*) on the *pH* (*k*) is obtained via the designation of the fuzzy membership functions (i.e. small, medium and large). A typical rule can then be written as linear summation as follows :

If acid fl(k−1) is small AND sdm fl(k−1) is large THEN pH(k) is large.

The pH Neutralization Plant system being examined by the neuro-fuzzy is a multivariable pH neutralization process. In this respect, Fig. 7 shows the associated MPC simulation for the process, at two required set-points. As the set-point changes, the process model also changes via the neuro-fuzzy system. In this MPC simulation, (15) has been employed to compute the associated control signals via the utilization of the linearized process model. This depends entirely on the region within which the process is operating.

VI Conclusions

Within this article we have introduced the basic structure of a fuzzy system and how the inter-layers are configured to model a *pH* neutralization reactor plant. The modeling fuzzy network has shown great results in terms of reducing the prediction Root Mean Squared error. Once fuzzy models are obtained, we validated the fuzzy models that operate over linearized plant region of operation. In this respect, the fuzzy models have been employed to model linearized operating regions of the plant under linear model predictive control. The employed algorithm produced fast settling time and convergence, in addition to a good fuzzy structure for modeling the nonlinear behavior of the plant under control.

References

[1] Wu Z. and Harris C., "Modeling and Adaptive Filtering of Nonlinear System using Neural Network," *University of Southampton*, Technical Report, 2000.

[2] L. A. Zadeh, "Fuzzy sets. Information and Control," vol. 8, pp. 338-353, 1965.

[3] J. Zhang, and J. Morris, "Fuzzy Neural Networks for Nonlinear Systems Modeling," *IEE Proc.-Control Theory Application*, vol. 142, no. 6, 1995.

[4] R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Trans. on System Man and Cybernetics*, vol. 23, no. 3, pp. 665-685, 1995.

[5] S. Reay and W. Dunnigan, "Learning Issue in Model Reference Based Fuzzy Control," *IEE Proc.-Control Theory Application*, vol. 144, no. 6, 1997.

[6] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and its Application to Modeling and Control," *IEEE Trans. on System Man and Cybernetics*, vol. 15, pp. 116-132, 1999.

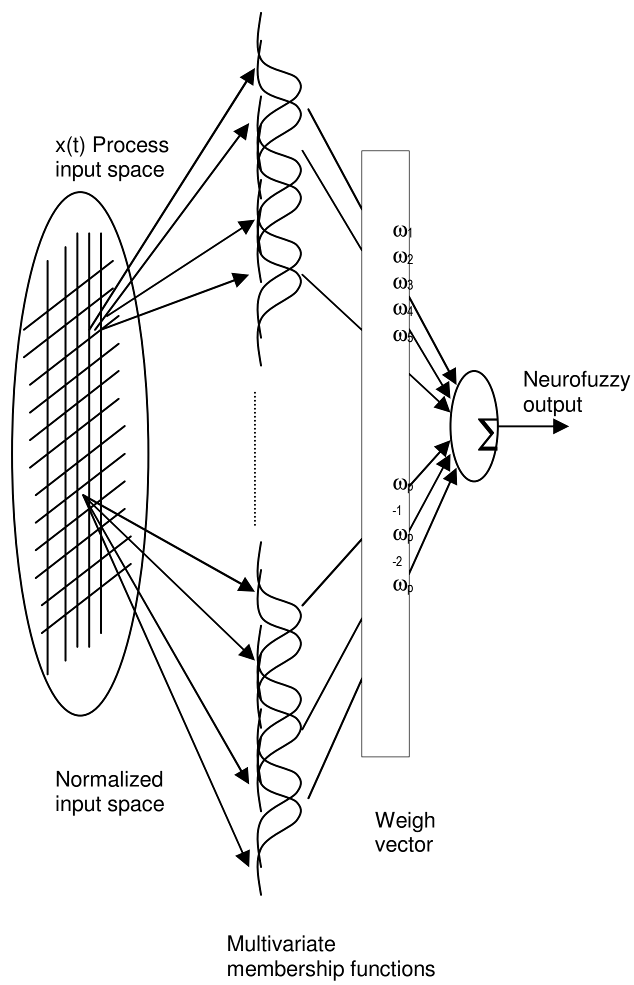

Fig. 1. Fuzzy system.

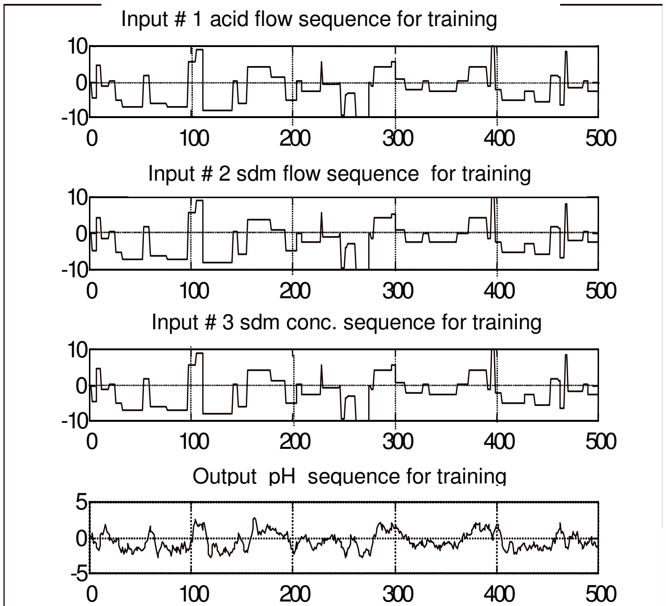

Fig. 2. Training sets for *pH* neutralization system.

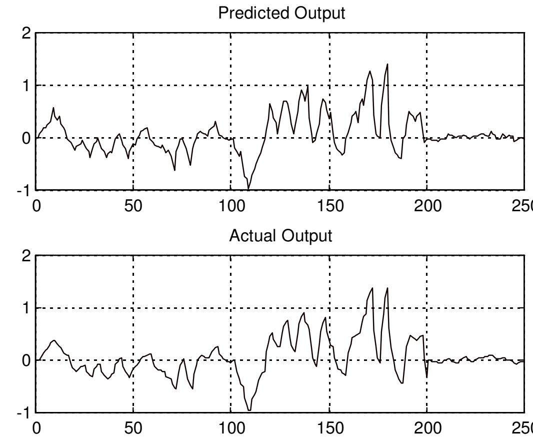

Fig. 5. Creating Fuzzy model and actual *pH* output.

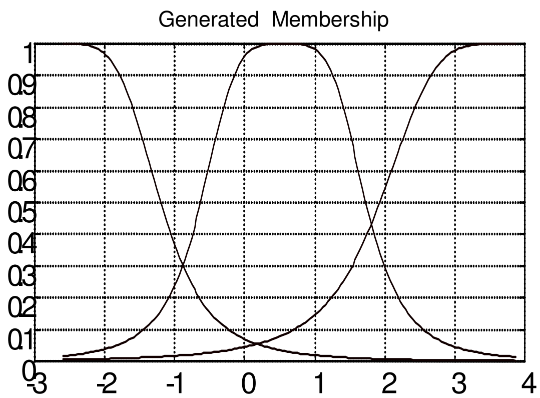

Fig. 3. Final membership functions of the acid f_1.

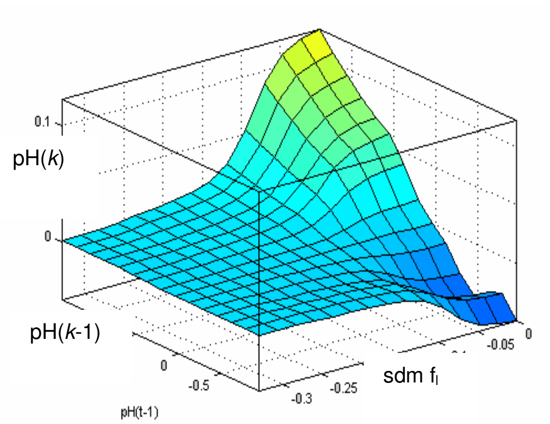

Fig .6. 3-D plot of fuzzy rules, and relation between pH(k–1), $sdmf_1$ and the process output pH(k).

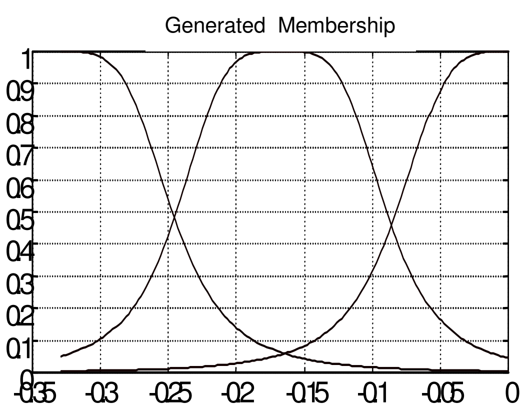

Fig. 4. Final membership functions of the *sdm. conc* in *pH* reactor plant.

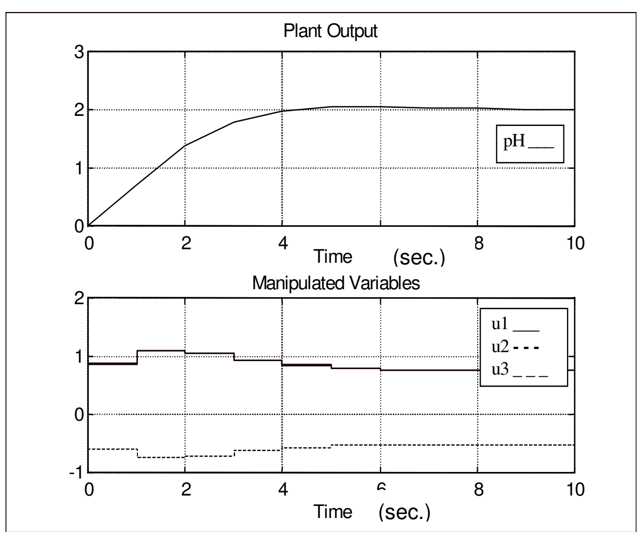

Fig. 7. Fuzzy model predictive control for the *pH* neutralization reactor plant.