Structure and Stability Analysis of General Mamdani Fuzzy Dynamic Models

Hao Ying*
Department of Electrical and Computer Engineering,
Wayne State University, Detroit, MI 48202, USA

Mamdani fuzzy models have always been used as black-box models. Their structures in relation to the conventional model structures are unknown. Moreover, there exist no theoretical methods for rigorously judging model stability and validity. I attempt to provide solutions to these issues for a general class of fuzzy models. They use arbitrary continuous input fuzzy sets, arbitrary fuzzy rules, arbitrary inference methods, Zadeh or product fuzzy logic AND operator, singleton output fuzzy sets, and the centroid defuzzifier. I first show that the fuzzy models belong to the NARX (nonlinear autoregressive with the extra input) model structure, which is one of the most important and widely used structures in classical modeling. I then divide the NARX model structure into three nonlinear types and investigate how the settings of the fuzzy model components, especially input fuzzy sets, dictate the relations between the fuzzy models and these types. I have found that the fuzzy models become type-2 models if and only if the input fuzzy sets are linear or piecewise linear (e.g., trapezoidal or triangular), becoming type 3 if and only if at least one input fuzzy set is nonlinear. I have also developed an algorithm to transfer type-2 fuzzy models into type-1 models as far as their input–output relationships are concerned, which have some important properties not shared by the type-2 models. Furthermore, a necessary and sufficient condition has been derived for a part of the general fuzzy models to be linear ARX models. I have established a necessary and sufficient condition for judging local stability of type-1 and type-2 fuzzy models. It can be used for model validation and control system design. Three numeric examples are provided. Our new findings provide a theoretical foundation for Mamdani fuzzy modeling and make it more consistent with the conventional modeling theory.

1. INTRODUCTION

Applications have shown the utility of fuzzy modeling techniques, which are discrete-time black-box modeling approaches developed for modeling complex dynamic systems (e.g., Refs. 1–4). Compared with the conventional black-box modeling techniques that can only utilize numerical data, fuzzy modeling approaches allow one to take advantage of both qualitative and quantitative information (e.g., Refs. 5–7). This advantage is practically important and even crucial

*e-mail: hao.ying@wayne.edu.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 20, 103–125 (2005)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.20056

in many circumstances. Qualitative information, such as expert/operator knowledge and experience about a physical system to be modeled, can readily be incorporated into the fuzzy models in the form of fuzzy sets, fuzzy logic, and fuzzy rules. Various learning schemes involving neural networks or genetic algorithms have also been attempted to automatically configure one or more of these components so that a fuzzy model can be established when qualitative/quantitative information is available (e.g., Refs. 8 and 9).

There exist two major types of fuzzy models, namely, Mamdani fuzzy models and TS fuzzy models (e.g., Refs. 10 and 11). I focus on the former in this article. Despite some practical applications, analytical study of Mamdani fuzzy modeling is still virtually nonexistent, which is in sharp contrast to the relatively rich analytical results on Mamdani fuzzy control and systems (e.g., Ref. 12). The fuzzy models have always been treated and used as black-box models. Their structures are currently unknown, let alone possible connections with their conventional counterparts, which include ARX (AutoRegressive with the eXtra input) models, AR models, and moving-average models.[13–15] I point out that the explicit structures of the conventional models are actually known; they are still called black-box models because the structures are artificially assigned, as opposed to being established according to the laws of physics. The models thus do not necessarily reflect the underlying physics of the dynamic systems. Comparatively, the fuzzy models also do not necessarily describe the underlying system physics; worse, though, their structures are even unknown. Hence, fuzzy models are much darker black-box models than the classical ones.

A related problem is that the vast majority of the fuzzy models in the literature were used merely for empirically mimicking measured input–output data of the physical systems. However, a fuzzy model seemingly mimicking input–output data does not necessarily mean the model is valid. Rigorous analysis and validation are required to ensure the model's validity and quality. Yet, at present there exists no theoretical means for checking the quality of a Mamdani fuzzy model via some measures, such as stability, and invalidate it if the result warrants.

In this article, I address these problems for a general class of Mamdani fuzzy models. They are actually contained in a more general class of fuzzy models, which, as I proved,[16] are universal approximators in that they can approximate any continuous functions arbitrarily well.[12] I also established sufficient approximation conditions by means of formulas capable of computing the number of fuzzy sets and rules needed for any given function and approximation accuracy.[16]

Knowing the fuzzy models to be universal approximators is important but not enough. Determining their model structures and connections with the traditional models was the first objective of the current research. Structure of a fuzzy model is determined by all the model components such as fuzzy sets, fuzzy rules, fuzzy logic AND operator, fuzzy inference method, and the defuzzifier. Different structures will result if different types of the components are used, and I explore their relationships.

The second goal was to use the newly obtained model structure information in developing a model quality measure. I derived a necessary and sufficient local stability condition for the fuzzy models. In addition to the stability determination, it can also be used to theoretically check the model quality. Moreover, with the criterion, (locally) stable control systems can be mathematically designed that involve the fuzzy models. I provide three numerical examples to demonstrate the utility of my new findings.

2. CONFIGURATION OF GENERAL MAMDANI FUZZY DYNAMIC MODELS

Designate $y(n-i)$ and $u(n-j)$ as the output and input of the physical system being modeled at times $n-i$ and $n-j$, respectively. The general Mamdani fuzzy models use $y(n-i)$ and $u(n-j)$, where $i = 0, \ldots, m$ and $j = 0, \ldots, p$, as input variables, resulting in total $m+p+2$ variables. And, $y(n-i)$ is defined on $[a_1, b_1]$ and $u(n-j)$ on $[a_2, b_2]$. For $y(n-i)$, $[a_1, b_1]$ is divided into $M_i - 1$ subintervals, and M_i continuous input fuzzy sets of arbitrary shape are defined. Like most fuzzy systems, one fuzzy set is defined over the first subinterval and one over the last. Each of the remaining $M_i - 2$ fuzzy sets is defined over every two consecutive subintervals. There are $M_i - 1$ pairs of such consecutive subintervals and each pair has a fuzzy set defined over it. In other words, each subinterval is covered by two fuzzy sets. The ℓth fuzzy set for fuzzifying $y(n-i)$ is designated as $\tilde{A}_{i,\ell}$, whose membership function is denoted as $\mu_{\tilde{A}_{i,\ell}}(y(n-i))$. The fuzzy sets for $u(n-j)$ are defined on $[a_2, b_2]$ in the same fashion. There are N_j continuous input fuzzy sets of arbitrary shape and the ℓth fuzzy set for $u(n-j)$ is denoted as $\tilde{B}_{j,\ell}$.

The input space is configured by $m+1$ identical intervals $[a_1, b_1]$ and $p+1$ identical intervals $[a_2, b_2]$ and it is $m+p+2$-dimensional. The subintervals of these intervals produce total

$$\Phi = \prod_{i=0}^{m}(M_i - 1)\prod_{j=0}^{p}(N_j - 1)$$

divisions of the input space, each of which is $m+p+2$-dimensional and is designated as Θ_i, $i = 1, \ldots, \Phi$.

There are a total of $\Omega = \prod_{i=0}^{m}\prod_{j=0}^{p} M_i N_j$ fuzzy modeling rules to cover all possible combinations of the input fuzzy sets. The rules are in the following format:

$$\text{IF } y(n) \text{ is } \tilde{A}_{0,I_0} \text{ AND } \ldots \text{ AND } y(n-m) \text{ is } \tilde{A}_{m,I_m} \text{ AND } u(n) \text{ is } \tilde{B}_{0,J_0}$$
$$\text{AND } \ldots \text{ AND } u(n-p) \text{ is } \tilde{B}_{p,J_p} \text{ THEN } y(n+1) \text{ is } \widetilde{V}_k$$

where $\widetilde{V}_k$ is a singleton output fuzzy set for $y(n+1)$ defined on $[a_1, b_1]$. That is, $\widetilde{V}_k$ is nonzero only at one location in $[a_1, b_1]$ and that location is designated as V_k. The AND operator can be either Zadeh type (i.e., the minimum operation) or the product type (i.e., the product operation). As for reasoning, any fuzzy inference method may be used, including the Mamdani inference method. They will produce the same inference outcome because the output fuzzy sets are of the singleton type.[12] The popular centroid defuzzifier is employed to combine the inference outcomes of the individual rules:

$$y(n+1) = \frac{\sum_{h=1}^{\Omega} \mu_h(\mathbf{y}(n), \mathbf{u}(n)) \cdot \overline{V}_h}{\sum_{h=1}^{\Omega} \mu_h(\mathbf{y}(n), \mathbf{u}(n))} \tag{1}$$

where $\mu_h(\mathbf{y}(n), \mathbf{u}(n))$ is the result of executing all the fuzzy logic AND operations in the hth rule, whereas $\overline{V}_h$ signifies the nonzero value of the singleton output fuzzy set in the rule.

3. STRUCTURE ANALYSIS OF GENERAL MAMDANI FUZZY DYNAMIC MODELS

In conventional modeling, one of the most important and popular structures is the nonlinear ARX (NARX) model, which is expressed by

$$y(n+1) = F(\mathbf{y}(n), \mathbf{u}(n)) + e(n+1) \tag{2}$$

where $\mathbf{y}(n) = [y(n) \ldots y(n-m)]$, $\mathbf{u}(n) = [u(n) \ldots u(n-p)]$, and the term $e(n+1)$ represents either random or deterministic error.[17] Because F can be any nonlinear function, this model structure is very general. In fact, it is so general that without any further research we can immediately conclude that the fuzzy models described in Equation 1 are NARX models without the random or deterministic error term since Equation 1 satisfies Equation 2. Apparently, this kind of conclusion is of little use. To more specifically classify and characterize the fuzzy models in a meaningful way, I divide the NARX models described in Equation 2 into three categories.

3.1. Classification of NARX Model Structures

In classic modeling theory, a linear time-invariant ARX dynamic model is described by (e.g., Ref. 13)

$$y(n+1) + \sum_{i=0}^{m} \alpha_i y(n-i) = \sum_{j=0}^{p} \beta_j u(n-j) + e(n+1) \tag{3}$$

where α_i and β_j are constant coefficients. For the purpose of this research and better presentation, I use this model as a base to divide the NARX models into three nonlinear groups.

Definition 1. *A dynamic model is called a type-1 NARX model if its model structure satisfies*

$$y(n+1) + \sum_{i=0}^{m} \alpha_i(\mathbf{y}(n), \mathbf{u}(n)) y(n-i) = \sum_{j=0}^{p} \beta_j(\mathbf{y}(n), \mathbf{u}(n)) u(n-j) \tag{4}$$

It is required that the denominator of $\alpha_i(\mathbf{y}(n), \mathbf{u}(n))$ is not cancelable by $y(n-i)$ and the denominator of $\beta_j(\mathbf{y}(n), \mathbf{u}(n))$ is not cancelable by $u(n-j)$). These constraints still apply even if $\alpha_i(\mathbf{y}(n), \mathbf{u}(n))$ and $\beta_j(\mathbf{y}(n), \mathbf{u}(n))$ are not fractional expressions (just treat them as fractional expressions with denominators 1).

DEFINITION 2. *A dynamic model is called a type-2 NARX model if its structure satisfies*

$$y(n+1) + \sum_{i=0}^{m} \alpha_i(\mathbf{y}(n), \mathbf{u}(n)) y(n-i) = \sum_{j=0}^{p} \beta_j(\mathbf{y}(n), \mathbf{u}(n)) u(n-j) + \delta(\mathbf{y}(n), \mathbf{u}(n)). \tag{5}$$

The constraints on $\alpha_i(\mathbf{y}(n), \mathbf{u}(n))$ and $\beta_j(\mathbf{y}(n), \mathbf{u}(n))$ are the same as described in Definition 1. In addition, the numerator of $\delta(\mathbf{y}(n), \mathbf{u}(n))$ must be constant.

DEFINITION 3. *A dynamic model is called a type-3 NARX model if its structure satisfies Equation 2 but complies with neither Definition 1 nor Definition 2.*

The restrictions on $\alpha_i(\mathbf{y}(n), \mathbf{u}(n))$, $\beta_j(\mathbf{y}(n), \mathbf{u}(n))$, and $\delta(\mathbf{y}(n), \mathbf{u}(n))$ are necessary to prevent misclassifications as exemplified below. One example misclassification would be to rewrite the following model

$$y(n+1) + y(n) + a \sin(y(n)) = u(n)$$

which is a type-3 NARX model, as

$$y(n+1) + y(n) = u(n) + \delta(\mathbf{y}(n))$$

by letting $\delta(\mathbf{y}(n)) = -a \sin(y(n))$, and misclassify the new expression as a type-2 NARX model. The violation arises because the numerator of $\delta(\mathbf{y}(n), \mathbf{u}(n))$ is not constant.

Accordingly, I classify the general Mamdani fuzzy models into three types.

DEFINITION 4. *A fuzzy model satisfying Definition k, $k = 1,2,3$, is called a type-k fuzzy NARX model.*

A fuzzy model may have different structures in different parts (i.e., divisions) of the input space. It is considered as a type-1 (or type-2 or type-3) fuzzy NARX model only when its input–output relationships in the *entire* input space satisfy the type-1 (or type-2 or type-3) NARX relationship.

3.2. Structure Analysis of the General Mamdani Fuzzy Dynamic Models

The general fuzzy models configured in Section 2 may be type-1, type-2, or type-3 NARX models. We now study their structures.

THEOREM 1. *The general Mamdani fuzzy dynamic models become type-2 NARX models in the entire input space if and only if all the input fuzzy sets are linear or piecewise linear in every Θ_i, $i = 1, \ldots, \Phi$.*

The condition in the theorem is necessary and sufficient. Figure 1 provides a graphical definition of piecewise linear fuzzy sets by giving one example. Note that the popular triangular and trapezoidal fuzzy sets are special cases of this example fuzzy set. For better presentation, the proof of the theorem is given in the Appendix.

Based on Theorem 1 and Definitions 1 and 2, the following condition is obvious.

THEOREM 2. *The necessary condition for the general Mamdani fuzzy dynamic models to become type-1 NARX models in the entire input space is that all the input fuzzy sets are linear or piecewise linear in every* Θ_i, $i = 1, \ldots, \Phi$.

Proof. According to Theorem 1, using linear or piecewise linear input fuzzy sets leads to type-2 NARX models. If, at the same, the model components are chosen such that the $\delta(\mathbf{y}(n), \mathbf{u}(n))$ term does not exist in any Θ_i, $i = 1, \ldots, \Phi$, the resulting models are actually type-1 NARX models. This, however, is not guaranteed to happen whenever the input fuzzy sets are linear or piecewise linear. Hence, the condition is only necessary, not sufficient. ■

Based on Theorems 1 and 2, the result below immediately follows.

COROLLARY 1. *The general Mamdani fuzzy dynamic models become type-3 NARX models in the entire input space if and only if at least one input fuzzy set is nonlinear (i.e., neither linear nor piecewise linear).*

When most physical systems are in steady state, their output is zero when the system input is zero (i.e., $y(n) = 0$ when $u(n) = 0$, where n is sufficiently large).

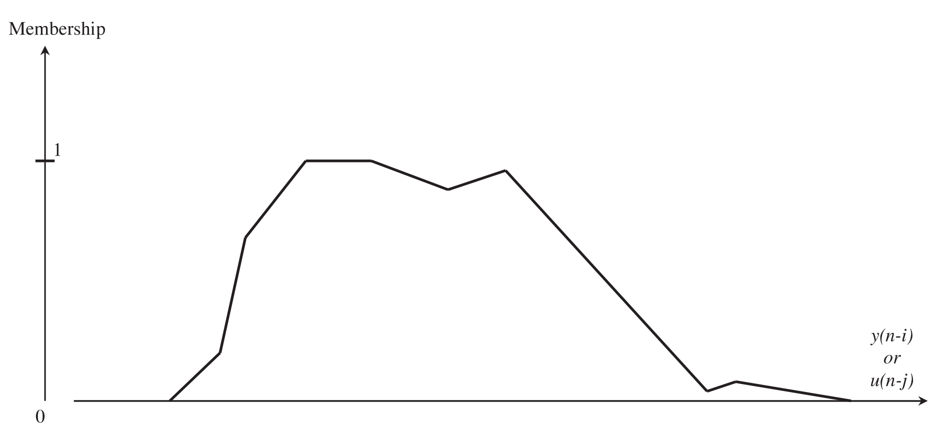

Figure 1. An example of piecewise linear fuzzy sets defined either for input variables (e.g., $y(n-i)$) or output variables (e.g., $u(n-j)$). Although this particular shape is hardly used in practice, the widely used triangular and trapezoidal fuzzy sets are the special cases of this piecewise linear fuzzy set.

This fundamental property, however, is generally not guaranteed by type-2 fuzzy NARX models because of the inherent existence of the $\delta(\mathbf{y}(n), \mathbf{u}(n))$ term. In practice, the fuzzy model developer needs to make an effort to properly choose those input fuzzy sets and fuzzy rules that cover the steady state so that the steady-state condition is satisfied. This may not be an easy task already. The task becomes much more difficult if one wants to manually prevent $\delta(\mathbf{y}(n), \mathbf{u}(n))$ from showing up in each and every Θ_i, $i = 1, \ldots, \Phi$, because there does not exist any theory that can guide the developer to make correct component choices. Hence, he is forced to use the trial-and-error method to blindly search for the correct choices. There are at least two obstacles that will prevent him from achieving the goal. First, the number of fuzzy model components and their possible selections are numerous. Second, the explicit expressions of $\alpha_i(\mathbf{y}(n), \mathbf{u}(n))$, $\beta_j(\mathbf{y}(n), \mathbf{u}(n))$, and $\delta(\mathbf{y}(n), \mathbf{u}(n))$ are difficult to derive and hence are unavailable in most cases.

This issue also occurs when the fuzzy models are constructed automatically using, for example, neural network learning methods or genetic algorithm optimization techniques.

If a physical system being modeled is of type-1 NARX structure, using linear or piecewise linear input fuzzy sets for the fuzzy modeling is necessary, not sufficient, according to Theorem 2. I have developed a novel offset elimination algorithm to deal with the $\delta(\mathbf{y}(n), \mathbf{u}(n))$ term. The idea is simple. Although the term is intrinsic to the type-2 fuzzy models and cannot be removed, its *effect* on the model output can be eliminated. My algorithm will numerically eliminate the effect of the term in every sampling period so that the type-2 fuzzy NARX model becomes a type-1 NARX model as far as its input–output relationship is concerned. Because the aim is not to get rid of the term itself, the effect elimination can be achieved without knowing the analytical structure of the fuzzy model. Hence it is practically useful. I name this technique of transforming the input–output relationship of a type-2 NARX model to that of a type-1 NARX model the *Offset Elimination Algorithm*. I now first use an example to show how my approach works and then provide the general algorithm.

Suppose that the example fuzzy model uses Zadeh fuzzy AND operator, two input variables, $y(n)$ and $u(n)$, and more than two input fuzzy sets for each variable. At time n, $y(n) = y^*$ and $u(n) = u^*$. Due to the way the input fuzzy sets are defined in Section 2, each variable can be fuzzified by two fuzzy sets. Without loss of generality, assume that y^* is fuzzified by linear fuzzy sets

$$\mu_{\tilde{A}_{0,\ell}}(y^*) = c_{0\ell}\, y^* + d_{0\ell} = 2y^* + 0.5$$

$$\mu_{\tilde{A}_{0,\ell+1}}(y^*) = c_{0(\ell+1)}(y^*) + d_{0(\ell+1)} = -3y^* + 0.5$$

and u^* by linear fuzzy sets

$$\mu_{\tilde{B}_{0,r}}(u^*) = C_{0r}\, u^* + D_{0r} = 1.2u^* + 0.4$$

$$\mu_{\tilde{B}_{0,r+1}}(u^*) = C_{0(r+1)}\, u^* + D_{0(r+1)} = -2.5u^* + 0.7$$

The fuzzification results supposedly are $\mu_{\tilde{A}_{0,\ell}}(y^*) = 0.7$, $\mu_{\tilde{A}_{0,\ell+1}}(y^*) = 0.2$, $\mu_{\tilde{B}_{0,r}}(u^*) = 0.52$, and $\mu_{\tilde{B}_{0,r+1}}(u^*) = 0.45$. The fuzzification result for the remaining fuzzy sets is zero. Only the following four fuzzy rules are supposedly activated among many rules in the fuzzy model:

$$\text{IF } y(n) \text{ is } \tilde{A}_{0,\ell} \text{ AND } u(n) \text{ is } \tilde{B}_{0,r} \text{ THEN } y(n+1) \text{ is } \widetilde{V}_3 \tag{r1}$$

$$\text{IF } y(n) \text{ is } \tilde{A}_{0,\ell} \text{ AND } u(n) \text{ is } \tilde{B}_{0,r+1} \text{ THEN } y(n+1) \text{ is } \widetilde{V}_7 \tag{r2}$$

$$\text{IF } y(n) \text{ is } \tilde{A}_{0,\ell+1} \text{ AND } u(n) \text{ is } \tilde{B}_{0,r} \text{ THEN } y(n+1) \text{ is } \widetilde{V}_1 \tag{r3}$$

$$\text{IF } y(n) \text{ is } \tilde{A}_{0,\ell+1} \text{ AND } u(n) \text{ is } \tilde{B}_{0,r+1} \text{ THEN } y(n+1) \text{ is } \widetilde{V}_4 \tag{r4}$$

where V_3, V_7, V_1, and V_4 are assumed to be 6, 1, −5, and 8, respectively. Obviously, the result of the Zadeh AND operations can be computed as follows: 0.52 for r1, 0.45 for r2, 0.2 for r3, and 0.2 for r4. Thus, the model output at $(y^*(n), u^*(n))$ is

$$\begin{aligned} y(n+1) &= \frac{\mu_{\tilde{B}_{0,r}}(u^*)\cdot V_3 + \mu_{\tilde{B}_{0,r+1}}(u^*)\cdot V_7 + \mu_{\tilde{B}_{0,r}}(u^*)\cdot V_1 + \mu_{\tilde{A}_{0,\ell+1}}(y^*)\cdot V_4}{\mu_{\tilde{B}_{0,r}}(u^*) + \mu_{\tilde{B}_{0,r+1}}(u^*) + \mu_{\tilde{B}_{0,r}}(u^*) + \mu_{\tilde{A}_{0,\ell+1}}(y^*)} \\ &= 3.044 \end{aligned}$$

A part of this output is due to the $\delta(\mathbf{y}(n), \mathbf{u}(n))$ term, which must be deducted in order for the input–output relationship of the fuzzy model, which is of type-2 NARX, to be that of the type-1 NARX model. Because the mathematical expression for $y(n+1)$ around the $(y^*(n), u^*(n))$ region is not available, one cannot directly remove the $\delta(\mathbf{y}(n), \mathbf{u}(n))$ term for that region. I show how to eliminate the *effect* of this term numerically under the constraint.

Substitute the linear membership functions to the above expression and then separate and group all the terms in the numerator that are not associated with y^* or u^*. One gets

$$\begin{aligned} \delta(y^*(n), u^*(n)) &= \frac{D_{0r}\cdot V_3 + D_{0(r+1)}\cdot V_7 + D_{0r}\cdot V_1 + d_{0(\ell+1)}\cdot V_4}{\mu_{\tilde{B}_{0,r}}(u^*) + \mu_{\tilde{B}_{0,r+1}}(u^*) + \mu_{\tilde{B}_{0,r}}(u^*) + \mu_{\tilde{A}_{0,\ell+1}}(y^*)} \\ &= 3.723 \end{aligned} \tag{6}$$

Subtracting 3.044 from this number gives −0.679, which is the output of the desired type-1 NARX model. Carrying out this model output modification in every sampling period, the input–output relationship of the type-2 NARX model becomes that of the type-1 NARX model.

If the product AND operator is used instead,

$$\begin{aligned} &y(n+1) \\ &= \frac{\mu_{\tilde{A}_{0,\ell}}(y^*)\cdot\mu_{\tilde{B}_{0,r}}(u^*)\cdot V_3 + \mu_{\tilde{A}_{0,\ell}}(y^*)\cdot\mu_{\tilde{B}_{0,r+1}}(u^*)\cdot V_7 + \mu_{\tilde{A}_{0,\ell+1}}(y^*)\cdot\mu_{\tilde{B}_{0,r}}(u^*)\cdot V_1 + \mu_{\tilde{A}_{0,\ell+1}}(u^*)\cdot V_4}{\mu_{\tilde{A}_{0,\ell}}(y^*)\cdot\mu_{\tilde{B}_{0,r}}(u^*) + \mu_{\tilde{A}_{0,\ell}}(y^*)\cdot\mu_{\tilde{B}_{0,r+1}}(u^*) + \mu_{\tilde{A}_{0,\ell+1}}(y^*)\cdot\mu_{\tilde{B}_{0,r}}(u^*) + \mu_{\tilde{A}_{0,\ell+1}}(y^*)\cdot\mu_{\tilde{B}_{0,r+1}}(u^*)} \\ &= 3.092 \end{aligned}$$

and $\mu_1 = \mu_{\tilde{A}_{0,\ell}}(y^*)\cdot\mu_{\tilde{B}_{0,r}}(u^*)$, $\mu_2 = \mu_{\tilde{A}_{0,\ell}}(y^*)\cdot\mu_{\tilde{B}_{0,r+1}}(u^*)$, $\mu_3 = \mu_{\tilde{A}_{0,\ell+1}}(y^*)\cdot\mu_{\tilde{B}_{0,r}}(u^*)$, and $\mu_4 = \mu_{\tilde{A}_{0,\ell+1}}(y^*)\cdot\mu_{\tilde{B}_{0,r+1}}(u^*)$. Moreover, $\varphi_1 = d_{0\ell}\cdot D_{0r}$, $\varphi_2 = d_{0\ell}\cdot D_{0(r+1)}$, $\varphi_3 = d_{0(\ell+1)}\cdot D_{0r}$, and $\varphi_4 = d_{0(\ell+1)}\cdot D_{0(r+1)}$. Hence,

$$\delta(y^*(n), u^*(n)) = \frac{\sum_{h=1}^{4} \varphi_h \cdot \overline{V}_h}{\sum_{h=1}^{4} \mu_h} = 3.837$$

where $\overline{V}_1 = V_3$, $\overline{V}_2 = V_7$, $\overline{V}_3 = V_1$, and $\overline{V}_4 = V_4$. Subsequently,

$$\hat{y}(n+1) = 3.092 - 3.837 = -0.745$$

I now extend these specific cases to the general case.

Offset Elimination Algorithm

Step 1. At sampling time n, compute the type-2 fuzzy NARX model output $y(n+1)$ using Equation 1.

Step 2. Designate the number of activated fuzzy rules at sampling time n as $\omega(n)$. Number the rules in an order from 1 to $\omega(n)$. Designate the consequent of the hth rule as $\overline{V}_h$ and the result of fuzzy AND operations for the rule as μ_h. For the hth rule, if Zadeh AND operator is used, denote the y-intercept of the linear fuzzy set generating the membership of μ_h as φ_h (in the case of the product AND operator, the product of the y-intercepts of all the linear fuzzy sets in the rule is designated as φ_h). Then, calculate

$$\delta(\mathbf{y}(n), \mathbf{u}(n)) = \frac{\sum_{h=1}^{\omega(n)} \varphi_h \cdot \overline{V}_h}{\sum_{h=1}^{\omega(n)} \mu_h} \tag{7}$$

Step 3. Type-1 fuzzy NARX model output is $\hat{y}(n+1) = y(n+1) - \delta(\mathbf{y}(n), \mathbf{u}(n))$. Go to Step 1 at next sampling time.

Obviously, the values of $\omega(n)$, $\overline{V}_h$, μ_h, and φ_h may change with time due to the change of $\mathbf{y}(n)$ and $\mathbf{u}(n)$. I stress that my algorithm can be implemented in any part of the input space (i.e., Θ_i) without any knowledge of the analytical structure of the fuzzy models, even though the value of $\delta(\mathbf{y}(n), \mathbf{u}(n))$ constantly changes with $\mathbf{y}(n)$ and $\mathbf{u}(n)$.

Linking this algorithm to the example above, $\omega(n) = 4$ and the rules are numbered from 1 to 4 with $\overline{V}_1$ to $\overline{V}_4$ being V_3, V_7, V_1, and V_4. μ_1 to μ_4 are

$\mu_{\tilde{B}_{0,r}}(u^*)$, $\mu_{\tilde{B}_{0,r+1}}(u^*)$, $\mu_{\tilde{B}_{0,r}}(u^*)$, and $\mu_{\tilde{A}_{0,\ell+1}}(y^*)$, respectively, and φ_1 to φ_4 are D_{0r}, $D_{0(r+1)}$, D_{0r}, and $d_{0(\ell+1)}$, respectively. The link when product AND operator is used is already provided in the example.

THEOREM 3. *As far as the input–output relationship is concerned, a sufficient condition for a type-2 fuzzy NARX model to become a type-1 fuzzy NARX model in the entire input space is that the Offset Elimination Algorithm is used.*

Proof. In light of the Offset Elimination Algorithm and Theorem 2, this theorem immediately follows. ■

3.3. Conditions for a Class of General Mamdani Fuzzy Dynamic Models to be Linear ARX Models

One may wonder what necessary and sufficient conditions there will be for the type-1 or type-2 fuzzy NARX models to become linear ARX models 3. The answer depends on several factors, including the intervals on which $y(n-i)$ and $u(n-j)$ are defined, the input and output fuzzy sets, and fuzzy rules. I now show this point by establishing necessary and sufficient conditions for a class of the general fuzzy models.

These fuzzy models use $y(n)$, $y(n-1)$, and $u(n)$ as input variables, where $y(n)$ and $y(n-1)$ are defined on $[-L_1, L_1]$ and $u(n)$ on $[-L_2, L_2]$, where L_1 and L_2 are design parameters. All the input fuzzy sets are linear (Figure 2). There are eight fuzzy rules ($\Omega = 2^3 = 8$):

IF $y(n)$ is Positive AND $y(n-1)$ is Positive AND $u(n)$ is Positive
THEN $y(n+1)$ is $\widetilde{V}_1$

IF $y(n)$ is Positive AND $y(n-1)$ is Positive AND $u(n)$ is Negative
THEN $y(n+1)$ is $\widetilde{V}_2$

IF $y(n)$ is Positive AND $y(n-1)$ is Negative AND $u(n)$ is Positive
THEN $y(n+1)$ is $\widetilde{V}_3$

IF $y(n)$ is Positive AND $y(n-1)$ is Negative AND $u(n)$ is Negative
THEN $y(n+1)$ is $\widetilde{V}_4$

IF $y(n)$ is Negative AND $y(n-1)$ is Positive AND $u(n)$ is Positive
THEN $y(n+1)$ is $\widetilde{V}_5$

IF $y(n)$ is Negative AND $y(n-1)$ is Positive AND $u(n)$ is Negative
THEN $y(n+1)$ is $\widetilde{V}_6$

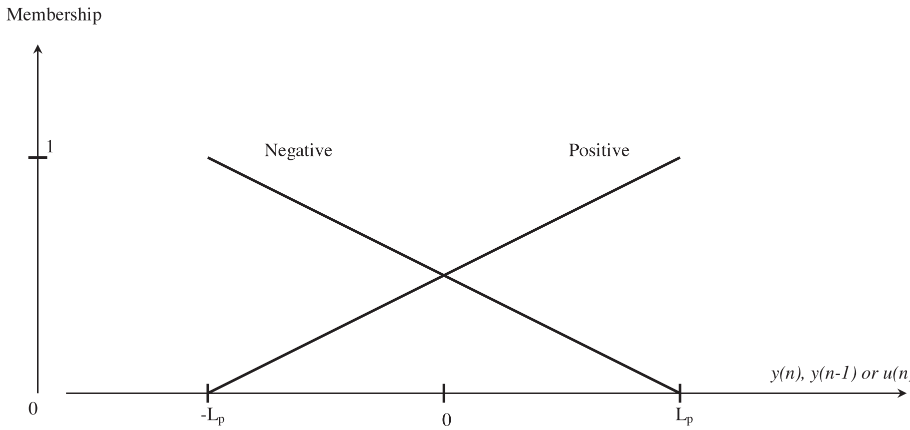

Figure 2. Definitions of three pairs of linear fuzzy sets for $y(n)$, $y(n-1)$, and $u(n)$. Here, $p = 1$ (i.e., L_1) is for $y(n)$ and $y(n-1)$ whereas $p = 2$ (i.e., L_2) is for $u(n)$. L_1 and L_2 are design parameters and their values can be different.

IF $y(n)$ is Negative AND $y(n-1)$ is Negative AND $u(n)$ is Positive

THEN $y(n+1)$ is $\widetilde{V}_7$

IF $y(n)$ is Negative AND $y(n-1)$ is Negative AND $u(n)$ is Negative

THEN $y(n+1)$ is $\widetilde{V}_8$

The rules use the product fuzzy AND operator.

Because of these linear fuzzy sets and the use of the product AND operator, it can be easily proved that the denominator of the defuzzifier 1 is always 1.[12] The numerator contains the summation of eight terms, one for a rule. Each term is the product of three linear membership functions in the rule. After simplifying and grouping the numerator, one gets

$$y(n+1) = c_1 \cdot y(n) + c_2 \cdot y(n-1) + c_3 \cdot u(n) + c_4 \cdot y(n)y(n-1) + c_5 \cdot y(n)u(n) + c_6 \cdot y(n-1)u(n) + c_7 \cdot y(n)y(n-1)u(n) + c_0 \tag{8}$$

with c_i being constants, $i = 0, \ldots, 7$. These constant coefficients are related to the parameters of the fuzzy models and can be described in a matrix form:

$$\mathbf{c} = \mathbf{MV}$$

where $\mathbf{c}^T = [c_0\, c_1\, c_2\, c_3\, c_4\, c_5\, c_6\, c_7]$, $\mathbf{V}^T = [V_1\, V_2\, V_3\, V_4\, V_5\, V_6\, V_7\, V_8]$,

$$\lambda_1 = \frac{1}{8L_1}, \qquad \lambda_2 = \frac{1}{8L_2}, \qquad \lambda_3 = \frac{1}{8L_1^2}, \qquad \lambda_4 = \frac{1}{8L_1 L_2}, \qquad \lambda_5 = \frac{1}{8L_1^2 L_2}$$

and

$$\mathbf{M} = \begin{bmatrix} \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} & \frac{1}{8} \\ \lambda_1 & \lambda_1 & \lambda_1 & \lambda_1 & -\lambda_1 & -\lambda_1 & -\lambda_1 & -\lambda_1 \\ \lambda_1 & \lambda_1 & -\lambda_1 & -\lambda_1 & \lambda_1 & \lambda_1 & -\lambda_1 & -\lambda_1 \\ \lambda_2 & -\lambda_2 & \lambda_2 & -\lambda_2 & \lambda_2 & -\lambda_2 & \lambda_2 & -\lambda_2 \\ \lambda_3 & \lambda_3 & -\lambda_3 & -\lambda_3 & -\lambda_3 & -\lambda_3 & \lambda_3 & \lambda_3 \\ \lambda_4 & -\lambda_4 & \lambda_4 & -\lambda_4 & -\lambda_4 & \lambda_4 & -\lambda_4 & \lambda_4 \\ \lambda_4 & -\lambda_4 & -\lambda_4 & \lambda_4 & \lambda_4 & -\lambda_4 & -\lambda_4 & \lambda_4 \\ \lambda_5 & -\lambda_5 & -\lambda_5 & \lambda_5 & -\lambda_5 & \lambda_5 & \lambda_5 & -\lambda_5 \end{bmatrix}$$

Rewriting Equation 8 yields the following type-2 NARX model:

$$\begin{aligned} y(n+1) = {} & \alpha_0(\mathbf{y}(n), \mathbf{u}(n)) \cdot y(n) + \alpha_1(\mathbf{y}(n), \mathbf{u}(n)) \cdot y(n-1) \\ & + \beta_0(\mathbf{y}(n), \mathbf{u}(n)) + \delta(\mathbf{y}(n), \mathbf{u}(n)) \end{aligned} \tag{9}$$

where

$$\begin{aligned} \alpha_0(\mathbf{y}(n), \mathbf{u}(n)) &= c_1 + c_5 u(n) + c_7 y(n-1) u(n) \\ \alpha_1(\mathbf{y}(n), \mathbf{u}(n)) &= c_2 + c_4 y(n) \\ \beta_0(\mathbf{y}(n), \mathbf{u}(n)) &= c_3 + c_6 y(n-1) \\ \delta(\mathbf{y}(n), \mathbf{u}(n)) &= c_0 \end{aligned}$$

To turn this model into a linear ARX model, c_i must satisfy $\mathbf{c}^* = [0 \quad \alpha_0 \quad \alpha_1 \quad \beta_0 \quad 0 \quad 0 \quad 0 \quad 0]$, where α_0, α_1, and β_0 are constants, so that all the cross-product terms in Equation 8 will disappear. Thus, as long as the rank of $\mathbf{M}$ is full (this should be the case at least most of the time), the necessary and sufficient conditions being sought can be calculated from

$$\mathbf{V} = \mathbf{M}^{-1}\mathbf{c}^*$$

with the result being

$$\begin{aligned} \mathbf{V}^T = [\, & \rho_1 + \rho_2 + \rho_3,\ \rho_1 + \rho_2 - \rho_3,\ \rho_1 - \rho_2 + \rho_3,\ \rho_1 - \rho_2 - \rho_3, \\ & -\rho_1 + \rho_2 + \rho_3,\ -\rho_1 + \rho_2 - \rho_3,\ -\rho_1 - \rho_2 + \rho_3,\ -\rho_1 - \rho_2 - \rho_3] \end{aligned}$$

where $\rho_1 = \alpha_0 L_1$, $\rho_2 = \alpha_1 L_2$, and $\rho_3 = \beta_0 L_3$. That is, if V_k are chosen according to these relations, the fuzzy models will become

$$y(n+1) = \alpha_0 \cdot y(n) + \alpha_1 \cdot y(n-1) + \beta_0 \cdot u(n)$$

which is a second-order linear ARX model. Clearly, the necessary and sufficient conditions are determined by the relationships among V_k, L_1, L_2, α_0, α_1, and β_0, as I have pointed out at the beginning of this section.

I should emphasize that my purpose here is to establish necessary and sufficient conditions for a class of the general fuzzy models. I am not interested in realizing linear ARX models via fuzzy modeling, because it offers no practical or theoretical advantages. These conditions should be viewed from a different angle: If a fuzzy model is not properly configured, one can unconsciously end up with a linear model. Indeed, more work is needed to study how to correctly construct fuzzy models so that they will not be linear models.

4. STABILITY, MODEL VALIDATION, AND CONTROL SYSTEM DESIGN FOR TYPE-1 AND TYPE-2 FUZZY NARX MODELS

Stability characterizes one of the most important behaviors of physical systems. The above structure analysis results enable us to determine stability of the type-1 and type-2 fuzzy NARX models. I will also use the stability criterion to check the quality of the fuzzy models and to design fuzzy control systems that are at least locally stable.

4.1. Local Stability Condition for Type-1 and Type-2 Fuzzy NARX Models

I chose to focus on local stability of the fuzzy models rather than on their global stability for two reasons. First, meaningful global stability analysis requires the explicit expression of the entire model, which is rather unrealistic and impractical to the fuzzy modeling paradigm as a whole. Explicit structures of many fuzzy models are not analytically derivable. Second, most global stability conditions are merely sufficient conditions; necessary ones are uncommon. Except for linear models, necessary and sufficient conditions are rare. Thus, global stability conditions are (very) conservative, often too conservative to be useful.

In contrast, local stability can be determined with much less information and assumption on the fuzzy models. Only two pieces of information are needed: (1) the fuzzy model structure around the equilibrium point and (2) the linearizability of the model at the equilibrium point. Both are obtainable in many cases, including the type-1 and type-2 fuzzy NARX models (see Section 5 for detailed examples). Furthermore, a local stability criterion is inherently a necessary and sufficient condition, the tightest possible condition (i.e., not conservative). These advantages make local stability results practically more useful. I should remind the reader that local stability does not mean stability at an equilibrium point only. Rather, it means stability in a region around the point.

THEOREM 4. *If the nonlinear difference equation*

$$y(n+1) + \sum_{i=0}^{m} \alpha_i(\mathbf{y}(n), \mathbf{u}(n)) y(n-i) = 0 \tag{10}$$

of a type-1 or type-2 fuzzy NARX model is linearizable at $\mathbf{y}(n) = \mathbf{0}$ *and* $\mathbf{u}(n) = \mathbf{0}$ *(meaning all the elements of the two vectors are 0), the fuzzy model is locally stable at that point if and only if the linearized model corresponding to Equation 10*

$$y(n+1) + \sum_{i=0}^{m} \alpha_i(\mathbf{0},\mathbf{0}) y(n-i) = 0 \tag{11}$$

is stable, where

$$\alpha_i(\mathbf{0},\mathbf{0}) = \frac{\sum_{h=1}^{\Omega} \mu_h(\mathbf{0},\mathbf{0}) \cdot \overline{V}_h}{\sum_{h=1}^{\Omega} \mu_h(\mathbf{0},\mathbf{0})}$$

Proof. Stability is an inherent model characteristic and thus is unrelated to input signals. Consequently, (local) stability of the type-1 or type-2 fuzzy NARX models is determined by the nonlinear difference equation 10. If the equation is linearizable at $\mathbf{y}(n) = \mathbf{0}$ and $\mathbf{u}(n) = \mathbf{0}$, then Lyapunov's linearization method[18] can be utilized to judge local stability of the resulting linear difference equation 11. This leads to the necessary and sufficient stability condition of the present theorem. ■

For the purpose of determining the stability, it is proper to treat the point $\mathbf{y}(n) = \mathbf{0}$ and $\mathbf{u}(n) = \mathbf{0}$ as an equilibrium point for either a type-1 or a type-2 fuzzy NARX model. By definition, the point is always an equilibrium point for a type-1 fuzzy model. But depending on the fuzzy model configuration, the point is not an equilibrium point for a type-2 fuzzy model if $\delta(\mathbf{0},\mathbf{0}) \neq 0$. Whenever this is the case, the actual equilibrium point for $\mathbf{y}(n)$ shifts in an amount related to $\delta(\mathbf{y},\mathbf{0})$, which can be calculated for any given fuzzy model (see Example 2 in Section 5).

The easiest way to determine stability of model 11 is to use the z-transform. That is, the model is stable if and only if all the roots of the corresponding z-transform equation

$$z - \sum_{i=0}^{m} \alpha_i(\mathbf{0},\mathbf{0}) z^{-i} = 0$$

are inside the unit circle.

4.2. Model Validation and Control System Design Involving Type-1 and Type-2 Fuzzy NARX Models

In addition to stability determination, an important use of Theorem 4 is for model validation. If a physical system being modeled by a type-1 or type-2 fuzzy NARX model is known to be stable around an equilibrium point, applying Theorem 4 to the fuzzy model should confirm it. If the confirmation occurs, the model developer can be more confident about the quality of the fuzzy model. Otherwise, the fuzzy model is incorrect and a new model needs to be established. This simple qualitative model verification can be practically important and useful. Without it, there would have existed no analytical means for checking or invalidating any Mamdani fuzzy model. The present common practice on fuzzy model validation is using computer simulation, which is not only time-consuming but, more importantly, can lead to erroneous validation results. No simulation can be comprehensive enough to cover all possible situations for nonlinear models.

Theorem 4 also makes it possible to use the type-1 and type-2 fuzzy models as control models in the design of control systems. One can use the linearized model of a fuzzy model to design a controller, be it a conventional controller or a fuzzy controller, linear or nonlinear. The design goal is to make the linearized control system at least locally stable. Fine-tuning may be needed to bring the control system to global stability.

5. NUMERIC EXAMPLES

Three numeric examples are provided, all of which involve local stability determination. In addition, the first example demonstrates how to use the stability criterion for model validation, and the third example shows how to design a control system that is at least locally stable. Explicit structures of the fuzzy models around the equilibrium point are derived.

Example 1. Suppose that we have established a Mamdani fuzzy model for a physical system known to be stable around the equilibrium point $\mathbf{y}(n) = \mathbf{0}$ and $\mathbf{u}(n) = \mathbf{0}$. Assume that there are many fuzzy sets for $y(n)$ and $u(n)$. To determine the local stability, however, all one needs is the input fuzzy sets covering the area around the equilibrium point. Figure 3 shows graphically such fuzzy sets for both input variables. The rest of the fuzzy sets are not shown, as they are not needed. The mathematical definitions for Figure 3 are as follows:

$$\mu_{\tilde{P}_0}(y(n)) = y(n) + 0.6, \qquad \mu_{\tilde{N}_0}(y(n)) = -0.3y(n) + 0.2,$$

$$\mu_{\tilde{P}_1}(u(n)) = 0.4u(n) + 0.3, \quad \text{and} \quad \mu_{\tilde{N}_1}(u(n)) = -1.3u(n) + 0.7$$

The complete fuzzy model must use many fuzzy rules to cover all the combinations of the input fuzzy sets. But for the stability judgment, only the following four rules covering $\mathbf{y}(n) = \mathbf{0}$ and $\mathbf{u}(n) = \mathbf{0}$ are relevant:

$$\text{IF } y(n) \text{ is } \tilde{P}_0 \text{ AND } u(n) \text{ is } \tilde{P}_1 \text{ THEN } y(n+1) \text{ is } \tilde{V}_1$$

$$\text{IF } y(n) \text{ is } \tilde{P}_0 \text{ AND } u(n) \text{ is } \tilde{N}_1 \text{ THEN } y(n+1) \text{ is } \tilde{V}_2$$

$$\text{IF } y(n) \text{ is } \tilde{N}_0 \text{ AND } u(n) \text{ is } \tilde{P}_1 \text{ THEN } y(n+1) \text{ is } \tilde{V}_3$$

$$\text{IF } y(n) \text{ is } \tilde{N}_0 \text{ AND } u(n) \text{ is } \tilde{N}_1 \text{ THEN } y(n+1) \text{ is } \tilde{V}_4$$

where $V_1 = 1$, $V_2 = 0.23$, $V_3 = -2.408$, and $V_4 = -1$. $\tilde{P}_0$, $\tilde{N}_0$, $\tilde{P}_1$, and $\tilde{N}_1$ are the two pairs of linear fuzzy sets shown in Figure 3. The product fuzzy AND operator is used.

The question is: Is this fuzzy model valid?

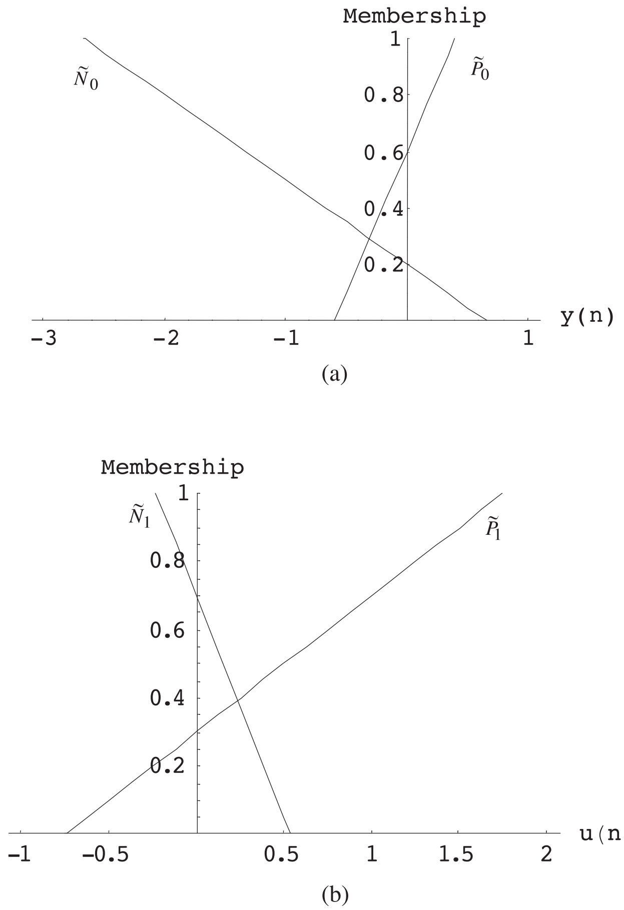

Figure 3. Two pairs of input fuzzy sets used in Examples 1 and 2: (a) the pair for $y(n)$, and (b) the pair for $u(n)$.

Solution. Around $\mathbf{y}(n) = \mathbf{0}$, the fuzzy model is a type-2 NARX model, according to Theorem 1. Using Equation 1, its structure around $\mathbf{y}(n) = \mathbf{0}$ is found to be as follows:

$$
\begin{aligned}
y(n+1) = {} & \frac{0.88775}{0.7y(n) - 0.63y(n)u(n) - 0.72u(n) + 0.8}\, y(n) \\
& + \frac{0.127933}{0.7y(n) - 0.63y(n)u(n) - 0.72u(n) + 0.8}\, u(n) \\
& - \frac{0.0079}{0.7y(n) - 0.63y(n)u(n)0.72u(n) + 0.8} \qquad (12)
\end{aligned}
$$

This model is obviously linearizable at $y(n) = 0$ and $u(n) = 0$. Let $y(n) = 0$ and $u(n) = 0$; we get the linearized model from the above model:

$$y(n+1) = 1.10969y(n) + 0.159917u(n) - 0.009875 \tag{13}$$

For stability determination, we only need to consider

$$y(n+1) = 1.10969y(n)$$

or equivalently

$$z - 1.10969 = 0$$

The linearized model is unstable because its pole is at 1.10969, which is outside the unit circle. Thus, the fuzzy model is unstable around $\mathbf{y}(n) = \mathbf{0}$. Figure 4 shows the model output when a very small initial value is given ($y(0) = 0.0001$). The output diverges very quickly with time, clearly demonstrating instability and confirming the analytical result. Model output at time index 14 calculated by the local model in Equation 12 is -4.952. This is an invalid result, as -4.952 is already outside the region covered by the fuzzy sets shown in Figure 3a.

Given that the actual physical system is stable, we can immediately conclude the fuzzy model to be wrong, because it is unstable. The model is invalidated.

Example 2. Suppose that only two conditions given in Example 1 are changed: specifically, V_2 is changed from 0.23 to 0.1 and V_3 from -2.408 to -1. Is the model locally stable now?

Solution. The calculation for Example 1 is repeated using the two new parameter values. The resulting fuzzy model around $\mathbf{y}(n) = \mathbf{0}$ is

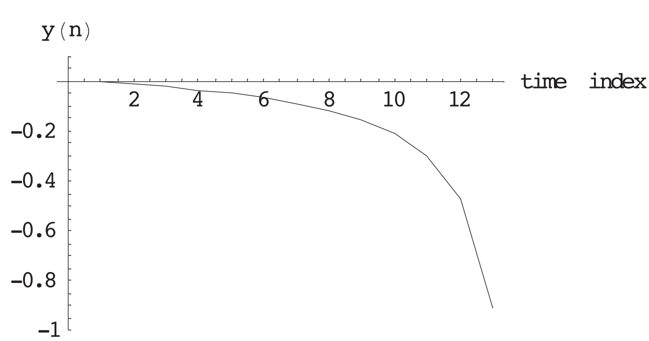

Figure 4. A graphic display for confirming the model instability around $\mathbf{y}(n) = \mathbf{0}$ that is theoretically determined in Example 1.

$$y(n+1) = \frac{0.67}{0.7y(n) - 0.63y(n)u(n) - 0.72u(n) + 0.8} y(n) + \frac{0.342}{0.7y(n) - 0.63y(n)u(n) - 0.72u(n) + 0.8} u(n) + \frac{0.22}{0.7y(n) - 0.63y(n)u(n)0.72u(n) + 0.8} \quad (14)$$

which is a type-2 NARX model and is linearizable. The linearized model is

$$y(n+1) = 0.8375y(n) + 0.4275u(n) + 0.0275$$

Thus, the modified fuzzy model becomes locally stable around $\mathbf{y}(n) = \mathbf{0}$ because the pole is at $z = -0.8375$ now. For a visual confirmation, Figure 5 provides the model output with the same initial condition $y(0) = 0.0001$. Clearly, the model is stable.

The final model output is not 0 but 0.1073 due to the effect of $\delta(\mathbf{y}(n), \mathbf{u}(n))$ of the local type-2 ARX model. This value can be computed using either Equation 7 or 14. In the latter equation, let $y(n+1) = y(n) = y(\infty)$ and $u(n) = 0$, and then solve the resulting equation. The result is $y(\infty) = 0.1073$. This part of the example illustrates that when $\mathbf{u}(n) = 0$, $\mathbf{y}(n)$ may not be $\mathbf{0}$. As such, that point is not an equilibrium point of the model. However, such a point can be treated as an equilibrium point for the purpose of the stability determination.

Example 3. Design a linear PD controller to control the unstable fuzzy model given in Example 1 so that the resulting closed-loop control system (Figure 6) is at least locally stable.

Solution. According to Equation 13, the linearized model from the original type-2 fuzzy NARX model is

$$y(n+1) = 1.10969y(n) + 0.159917u(n) - 0.009875$$

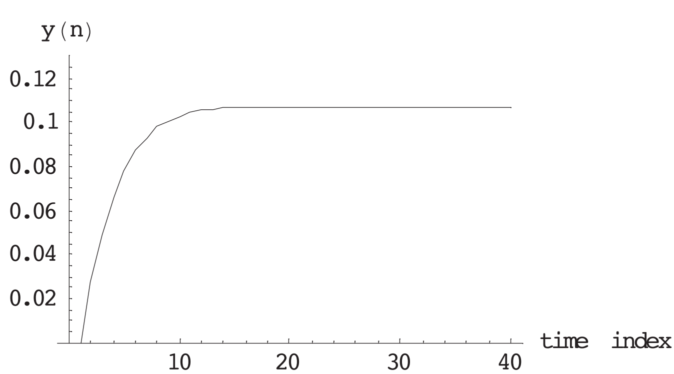

Figure 5. A graphic confirmation for the calculated model stability around $\mathbf{y}(n) = \mathbf{0}$ and the final model output value in Example 2.

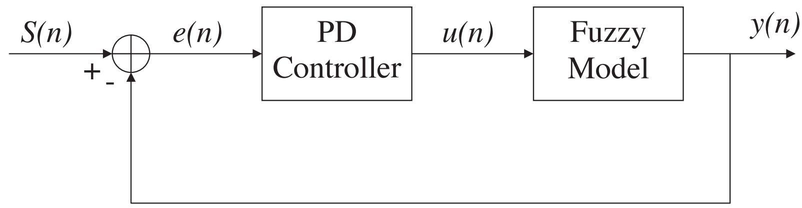

Figure 6. Structure of the closed-loop control system using a linear PD controller to control the unstable type-2 fuzzy NARX model given in Example 1.

Note that the steady state of $y(n)$, $y(\infty)$, is not 0. Rather, $y(\infty) = 0.009875/0.10969 = 0.09$. Let $x(n) = y(n) - y(\infty)$; the linearized model becomes

$$x(n+1) = 1.10969x(n) + 0.159917u(n)$$

and $x(n) = 0$ when $u(n) = 0$. A discrete-time linear PD controller can be described by

$$\begin{aligned} u(n) &= K_p e(n) + K_d[e(n) - e(n-1)] \\ &= K_p[S(n) - x(n)] + K_d[x(n-1) - x(n)] \end{aligned}$$

where $S(n)$ is output setpoint signal, and K_p and K_d are proportional-gain and derivative-gain, respectively. The closed-loop transfer function of the linearized control system is found to be

$$\frac{X(z)}{S(z)} = \frac{0.159917K_p}{z^2 + (0.159917K_p + 0.159917K_d - 1.10969)z - 0.159917K_d}$$

If we choose $K_p = 1$ and $K_d = 1$, the closed-loop poles are at $z = 0.957$ and $z = -0.1671$, both of which are inside the unit circle. Thus, the closed-loop fuzzy control system shown in Figure 6 is stable at least around the equilibrium point.

References

1. Cao SG, Rees NW, Feng G. Analysis and design for a class of complex control systems Part I: Fuzzy modelling and identification. Automatica 1997;33:1017–1028.
2. Sugeno M, Kang KT. Structure identification of fuzzy model. Fuzzy Sets Syst 1988; 28:15–33.
3. Wang L, Langari R. Identification of time-varying fuzzy systems. Int J Gen Syst 1996;25:203–218.
4. Wang L, Langari R. Complex systems modeling via fuzzy logic. IEEE Trans Syst Man Cybern B 1996;26:100–106.
5. Sugeno M, Yasukawa T. A fuzzy-logic-based approach to qualitative modeling. IEEE Trans Fuzzy Syst 1993;1:7–31.
6. Yager RR, Filev DP. Essentials of fuzzy modeling and control. New York: John Wiley & Sons; 1994.
7. Yen J, Wang L. Application of statistical information criteria for optimal fuzzy model construction. IEEE Trans Fuzzy Syst 1998;6:362–372.

8. Liu K, Lewis FL. Adaptive tuning of fuzzy logic identifier for unknown non-linear systems. Int J Adapt Control Signal Process 1994;8:573–586.
9. Wang LX. Adaptive fuzzy systems and control. Englewood Cliffs, NJ: PTR Prentice Hall; 1994.
10. Mouzouris GC, Mendel JM. Dynamic non-singleton fuzzy logic systems for nonlinear modeling. IEEE Trans Fuzzy Syst 1997;5:199—208.
11. Takagi T, Sugeno M. Fuzzy identification of systems and its applications to modeling and control. IEEE Trans Syst Man Cybern 1985;15:116–132.
12. Ying H. Fuzzy control and modeling: Analytical foundations and applications. New York: IEEE Press; 2000.
13. Ljung L. System identification: Theory for the user, 2nd ed. Englewood Cliffs, NJ: Prentice Hall; 1999.
14. Sjoberg J, Zhang Q, Ljung L, Benveniste A, Delyon B. Nonlinear black-box modeling in system identification: A unified overview. Automatica 1995;31:1691–1724.
15. Van Den Hof PM, Schrama RJP. Identification and control—Closed-loop issues. Automatica 1995;31:1751–1770.
16. Ying H. Sufficient conditions on general fuzzy systems as function approximators. Automatica 1994;30:521–525.
17. Milanes M, Belforte G. Estimation theory and uncertainty intervals evaluation in the presence of unknown but bounded errors: Linear families of models and estimators. IEEE Trans Auto Control 1982;27:408–414.
18. Slotine J-JE, Li W. Applied nonlinear control. Upper Saddle River, NJ: Prentice Hall; 1991.
19. Ying H, Siler W, Buckley JJ. Fuzzy control theory: A nonlinear case. Automatica 1990;26:513–520.

APPENDIX: PROOF OF THEOREM 1

The condition stated in the theorem is a necessary and sufficient one.

A1. When Zadeh Fuzzy Logic AND Operator Is Used

I first prove the necessity of the condition, that is, if at least one of the input fuzzy sets is not linear or piecewise linear in Θ_i, the fuzzy models cannot be type-2 ARX models in Θ_i. Figure 1 provides a graphical definition of piecewise linear fuzzy sets.

Without loss of generality, assume that $\mu_{\tilde{A}_{I,J}}(y(n-I))$ is the one and only one fuzzy set that is not linear or piecewise. Assume that $\omega(n)$ fuzzy rules are activated at sampling time n, where $\omega(n) \le \Omega$. We want to determine the analytical input–output relationship for the fuzzy models, not just the numeric relationship sought in most fuzzy modeling studies. Which rules are activated is known; so are their rules consequent (i.e., $\overline{V}_h$). Nevertheless, one must first determine $\mu_h(\mathbf{y}(n), \mathbf{u}(n))$, the *analytical* result of Zadeh fuzzy AND operations for each of the activated fuzzy rules before the defuzzifier in Equation 1 can produce the mathematical structure of the models. For each rule, $m + p + 2$ membership functions, one for each input or output variable, must be compared in the $m + p + 2$-dimensional input space to determine which membership function is the smallest. Such a membership function will then become the $\mu_h(\mathbf{y}(n), \mathbf{u}(n))$ for that rule. Because no single membership function can be smallest in the entire input space, the space must be so divided that in each subspace (still $m + p + 2$-dimensional), one of the $m + p + 2$ membership functions will always be the smallest.[19] Inevitably, $\mu_{\tilde{A}_{I,J}}(y(n-I))$ will be the smallest membership function in at least one subspace. Assume there is only one such subspace. The result of Zadeh AND operations for that subspace will be $\mu_{\tilde{A}_{I,J}}(y(n-I))$. Consequently, the numerator of the defuzzifier will contain the term $\mu_{\tilde{A}_{I,J}}(y(n-I))\cdot V_k$, where V_k represents the singleton output fuzzy set involved. The rest of the numerator is the summation of linear membership functions multiplied by constants $\overline{V}_h$. Because $\mu_{\tilde{A}_{I,J}}(y(n-I))$ is not a linear or piecewise linear function with respect to $y(n-I)$, after all the like terms in the numerator are combined, a nonlinear term of $y(n-I)$, designated as $f(y(n-I))$ where f is a nonlinear function, will *always* exist. Given that the defuzzifier denominator cannot be just $f(y(n-I))$, this nonlinear term in the numerator cannot be canceled by either the terms in the numerator or the denominator. Subsequently, the fuzzy models cannot be expressed in the form of Equation 5 because the term involving $y(n-I)$ cannot be expressed as $\alpha_I(\mathbf{y}(n),\mathbf{u}(n))y(n-I)$.

Now let us prove the condition to be sufficient. When all the input fuzzy sets are linear or piecewise linear in Θ_i, the smallest membership function in every subspace will be a linear function of either $y(n-i)$ or $u(n-j)$, but not both. (Note that a piecewise linear membership function can always be decomposed into a series of linear membership functions with the corresponding division of the subspaces into smaller subspaces.) As a result,

$$y(n+1) = \frac{\sum_{h=1}^{\Omega} \mu_h(\mathbf{y}(n),\mathbf{u}(n))\cdot\overline{V}_h}{\sum_{h=1}^{\Omega} \mu_h(\mathbf{y}(n),\mathbf{u}(n))} = \sum_{h=1}^{\Omega} \frac{\overline{V}_h}{\sum_{h=1}^{\Omega} \mu_h(\mathbf{y}(n),\mathbf{u}(n))} \mu_h(\mathbf{y}(n),\mathbf{u}(n))$$
$$= \sum_{h=1}^{\Omega} \varphi_h(\mathbf{y}(n),\mathbf{u}(n))\cdot\mu_h(\mathbf{y}(n),\mathbf{u}(n)) \tag{15}$$

where

$$\varphi_h(\mathbf{y}(n),\mathbf{u}(n)) = \frac{\overline{V}_h}{\sum_{h=1}^{\Omega} \mu_h(\mathbf{y}(n),\mathbf{u}(n))} \tag{16}$$

For any given h, $\mu_h(\mathbf{y}(n),\mathbf{u}(n))$ is the smallest linear membership function in the $m + p + 2$-dimensional subspace. Hence it is a linear function of either $y(n-i)$ or $u(n-j)$. Replacing all the $\mu_h(\mathbf{y}(n),\mathbf{u}(n))$ in Equation 15 by their corresponding linear membership functions, expanding the resulting expression, and then combining the like terms for $y(n-i)$ and $u(n-j)$, we get

$$\begin{aligned} y(n+1) = {} & \alpha_0(\mathbf{y}(n),\mathbf{u}(n))\cdot y(n) + \cdots + \alpha_m(\mathbf{y}(n),\mathbf{u}(n))\cdot y(n-m) \\ & + \beta_0(\mathbf{y}(n),\mathbf{u}(n))\cdot u(n) + \cdots + \beta_p(\mathbf{y}(n),\mathbf{u}(n))\cdot u_p(n-p) \\ & + \delta(\mathbf{y}(n),\mathbf{u}(n)) \end{aligned}$$

where $\alpha_i(\mathbf{y}(n), \mathbf{u}(n))$ is a linear combination of the $\varphi_h(\mathbf{y}(n), \mathbf{u}(n))$ terms from all the $\varphi_h(\mathbf{y}(n), \mathbf{u}(n)) \cdot y(n-i)$ terms whereas $\beta_j(\mathbf{y}(n), \mathbf{u}(n))$ is a linear combination of the $\varphi_h(\mathbf{y}(n), \mathbf{u}(n))$ terms from all the $\varphi_h(\mathbf{y}(n), \mathbf{u}(n)) \cdot u(n-j)$ terms. $\delta(\mathbf{y}(n), \mathbf{u}(n))$ is a linear combination of all the $\varphi_h(\mathbf{y}(n), \mathbf{u}(n))$ terms unassociated with either $y(n-i)$ or $u(n-j)$.

Note that $\alpha_i(\mathbf{y}(n), \mathbf{u}(n))$, $\beta_j(\mathbf{y}(n), \mathbf{u}(n))$, and $\delta(\mathbf{y}(n), \mathbf{u}(n))$ are fractional expressions. Their numerators are constants and denominators are the same, which is $\sum_{h=1}^{\Omega} \mu_h(\mathbf{y}(n), \mathbf{u}(n))$. The numerators of $\alpha_i(\mathbf{y}(n), \mathbf{u}(n))$ and $\beta_j(\mathbf{y}(n), \mathbf{u}(n))$ are linear combinations of some $\overline{V}_h$s, and the slopes of the linear input fuzzy sets involved whereas the numerator of $\delta(\mathbf{y}(n), \mathbf{u}(n))$ is a linear combination of some $\overline{V}_h$s and the y-intercepts of the input fuzzy sets. Unless the fuzzy sets are chosen such that the defuzzifier denominator $\sum_{h=1}^{\Omega} \mu_h(\mathbf{y}(n), \mathbf{u}(n))$ is a constant, the values of $\alpha_i(\mathbf{y}(n), \mathbf{u}(n))$ and $\beta_j(\mathbf{y}(n), \mathbf{u}(n))$ change with $\mathbf{y}(n)$ and $\mathbf{u}(n)$ and are hence variable coefficients. According to Definition 2, these fuzzy models are type-2 fuzzy NARX models.

A2. When the Product Fuzzy Logic AND Operator Is Used

In Θ_i, for the hth fuzzy rule, due to the product fuzzy AND operator,

$$\mu_h(\mathbf{y}(n), \mathbf{u}(n)) = \mu_{\tilde{A}_{0,I_0}}(y(n)) \times \cdots \times \mu_{\tilde{A}_{m,I_m}}(y(n-m)) \times \mu_{\tilde{B}_{0,J_0}}(u(n)) \times \cdots \times \mu_{\tilde{B}_{p,J_p}}(u(n-p))$$

and hence

$$y(n+1) = \sum_{h=1}^{\Omega} \varphi_h(\mathbf{y}(n), \mathbf{u}(n)) \times \mu_{\tilde{A}_{0,I_0}}(y(n)) \times \cdots \times \mu_{\tilde{A}_{m,I_m}}(y(n-m)) \times \mu_{\tilde{B}_{0,J_0}}(u(n)) \times \cdots \times \mu_{\tilde{B}_{p,J_p}}(u(n-p)). \tag{17}$$

$\varphi_h(\mathbf{y}(n), \mathbf{u}(n))$ is described in Equation 16. Suppose that one of the input fuzzy sets is not linear or piecewise linear. Without losing generality, assume that one fuzzy set for $y(n-I)$ is nonlinear. Then the above input–output relationship will become the sum of a type-2 NARX model and an extra term nonlinear with respect to $y(n-I)$. Therefore, it will be impossible to express the fuzzy models as type-2 NARX models. (See the necessity proof for Theorem 1.)

On the other hand, if all the input fuzzy sets are linear or piecewise linear in Θ_i, then in

$$y(n+1) = \sum_{h=1}^{\Omega} \varphi_h(\mathbf{y}(n), \mathbf{u}(n)) \cdot \mu_h(\mathbf{y}(n), \mathbf{u}(n))$$

$\mu_h(\mathbf{y}(n), \mathbf{u}(n))$ is the product of the linear membership functions of all the input variables. Carrying out all the multiplication operations and simplifying the resultant expression gives

$$
\begin{aligned}
y(n+1) = {} & \sum_{i=0}^{m} a_i(\mathbf{y}(n),\mathbf{u}(n))\cdot y(n-i) \\
& + \sum_{i=0}^{m}\sum_{j>i}^{m} a_{ij}(\mathbf{y}(n),\mathbf{u}(n))\cdot y(n-i)\cdot y(n-j) + \cdots \\
& + a_{0\ldots m}(\mathbf{y}(n),\mathbf{u}(n))y(n)\ldots y(n-m) + \sum_{i=0}^{p} b_0(\mathbf{y}(n),\mathbf{u}(n))\cdot u(n-i) \\
& + \sum_{i=0}^{p}\sum_{j>i}^{p} b_{ij}(\mathbf{y}(n),\mathbf{u}(n))\cdot u(n-i)\cdot u(n-j) \\
& + \cdots + b_{0\ldots p}(\mathbf{y}(n),\mathbf{u}(n))u(n)\ldots u(n-p) \\
& + \sum_{i=0}^{m}\sum_{j=0}^{p} c_{ij}(\mathbf{y}(n),\mathbf{u}(n))\cdot y(n-i)\cdot u(n-j) + \cdots \\
& + c_{0\ldots m0\ldots p}(\mathbf{y}(n),\mathbf{u}(n))y(n)\ldots y(n-m)u(n)\ldots u(n-p) \\
& + \delta\,(\mathbf{y}(n),\mathbf{u}(n))
\end{aligned}
$$

We want to change this expression so that it fits Definition 2. Note that the denominators of all the variable coefficients and $\delta(\mathbf{y}(n),\mathbf{u}(n))$ are the same, which are $\sum_{h=1}^{\Omega}\mu_h(\mathbf{y}(n),\mathbf{u}(n))$. Their numerators are determined by the slops and/or y-intercepts of the linear fuzzy sets. The numerator of $\delta(\mathbf{x})$ is a constant and is determined by the y-intercepts of the linear fuzzy sets only. All the cross-product terms (i.e., the terms other than those in $\sum_{i=0}^{m} a_i(\mathbf{y}(n),\mathbf{u}(n))\cdot y(n-i)$ and $\sum_{j=0}^{p} b_j(\mathbf{y}(n),\mathbf{u}(n))\cdot u(n-j)$ and the term $\delta(\mathbf{y}(n),\mathbf{u}(n))$) can be combined with either the $y(n-i)$ terms or the $u(n-j)$ terms to yield

$$
\begin{aligned}
y(n+1) = {} & \sum_{i=0}^{m}\alpha_i(\mathbf{y}(n),\mathbf{u}(n))y(n-i) \\
& + \sum_{j=0}^{p}\beta_j(\mathbf{y}(n),\mathbf{u}(n))u(n-j) + \delta(\mathbf{y}(n),\mathbf{u}(n))
\end{aligned}
$$

which describes the type-2 fuzzy NARX models. (As an illustration, see how the nonlinear model in Equation 8 is represented as a type-2 NARX model in Equation 9.)

This proves the sufficiency of the condition of the theorem. ■