In order to verify the performance of the proposed method, experiments with the NIST numeral database have been carried out and the performance of the proposed method has been compared with that of the previous ISR methods. The experimental results revealed that the proposed method had much better discrimination and generalization power than the previous ISR methods.

REFERENCES

[1] J. S. Denker *et al.*, "Neural network recognizer for handwritten ZIP code digits," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, vol. 1, pp. 323–331.

[2] Y. Le Cun *et al.*, "Constrained neural network for unconstrained handwritten digit recognition," in *Proc. 1st Int. Workshop Frontiers Handwriting Recogn.*, Montreal, P.Q., Canada, Apr. 1990, pp. 145–154.

[3] G. L. Martin and J. A. Pittman, "Recognizing hand-printed letters and digits," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, vol. 2, pp. 405–414.

[4] O. Matan *et al.*, "Reading handwritten digits: A ZIP code recognition system," *IEEE Comput. Mag.*, vol. 25, no. 7, pp. 59–63, July 1992.

[5] J. Keeler and D. E. Rumelhart, "A self-organizing integrated segmentation and recognition neural network," in *Advances in Neural Information Processing Systems*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, 1992, vol. 4, pp. 496–503.

[6] O. Matan, J. C. Burges, Y. Le Cun, and J. S. Denker, "Multi-digit recognition using a space displacement neural network," in *Advances in Neural Information Processing Systems*, J. E. Moody, S. J. Hanson, and R. P. Lippmann, Eds. San Mateo, CA: Morgan Kaufmann, 1992, vol. 4, pp. 488–495.

[7] G. L. Martin, M. Rashid, and J. A. Pittman, "Integrated segmentation and recognition through exhaustive scans or learned saccadic jumps," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 7, no. 4, pp. 831–847, 1993.

[8] K. Imai, K. Gouhara, and Y. Uchikawa, "Recognition of letters in lateral printed strings using a three-layered BP model with feedback connections," *Syst. Comput. Jpn.*, vol. 23, no. 12, pp. 83–92, 1992.

[9] H. Fujisawa and Y. Nakano, "Segmentation methods for character recognition: From segmentation to document structure analysis," *Proc. IEEE*, vol. 80, pp. 1079–1092, July 1992.

[10] R. M. Božinovič and S. N. Srihari, "Off-line cursive word recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, pp. 68–83, Jan. 1989.

[11] S.-W. Lee and J. S. Park, "Nonlinear shape normalization methods for the recognition of large-set handwritten characters," *Pattern Recognit.*, vol. 27, no. 7, pp. 895–902, 1994.

[12] K. Lang, A. Waibel, and G. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, vol. 3, pp. 23–43, 1990.

[13] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, D. Rumelhart and J. McClelland, Eds. Cambridge, MA: MIT Press, 1986.

[14] Y. Yamashita, K. Higuchi, Y. Yamada, and Y. Haga, "Classification of handprinted Kanji characters by the structured segment matching method, *Pattern Recognit. Lett.*, vol. 1, pp. 475–479, 1983.

[15] S.-W. Lee, "Off-line recognition of totally unconstrained handwritten numerals," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 648–652, June 1996.

Analytical Analysis and Feedback Linearization Tracking Control of the General Takagi-Sugeno Fuzzy Dynamic Systems

Hao Ying

***Abstract*—Takagi–Sugeno (TS) fuzzy modeling technique, a black-box discrete-time approach for system identification, has widely been used to model behaviors of complex dynamic systems. Analytical structure of TS fuzzy models, however, is presently unknown, nor is its possible connection with the traditional models, causing at least two major problems. First, the fuzzy models can hardly be utilized to design controllers for control of the physical systems modeled. Second, there lacks a systematic technique for designing a controller capable of controlling any given TS fuzzy model to achieve desired tracking or setpoint control performance. In this paper, we provide solutions to these problems. First of all, we have proved that a general class of TS fuzzy models is nonlinear time-varying Auto-Regressive with the eXtra input (ARX) model. The fuzzy models in this study are general because they use arbitrary continuous input fuzzy sets, any types of fuzzy logic AND operators, TS fuzzy rules with linear consequent and the generalized defuzzifier which contains the popular centroid defuzzifier as a special case. Furthermore, we have established a simple necessary and sufficient condition for analytically determining local stability of the general TS fuzzy dynamic models. The condition can also be used to analytically check quality of a TS fuzzy model and invalidate the model if the condition warrants. More importantly, we have developed a feedback linearization technique for systematically designing an output tracking controller so that output of a controlled TS fuzzy system, which may or may not be stable, of the general class achieves perfect tracking of any bounded time-varying trajectory. We have investigated stability of the tracking controller and established a necessary and sufficient condition, in relation to stability of nonminimum phase systems, for analytically deciding whether a stable tracking controller can be designed using our method for any given TS fuzzy system. Three numerical examples are provided to illustrate the effectiveness and utility of our results and techniques.**

***Index Terms*—AR models, feedback linearization, fuzzy control, fuzzy modeling, stability.**

I. INTRODUCTION

Numerous successful industrial applications have shown the power of Takagi–Sugeno (TS) fuzzy modeling approach [10], which is a black-box discrete-time modeling approach developed for modeling complex dynamic systems [1], [5], [16], [17], [19]. Compared with the conventional black-box modeling techniques [7] that can only utilize numerical data, TS modeling approach allows one to take advantage of both qualitative and quantitative information [8]. This advantage is practically important and even crucial in many circumstances. Qualitative information, such as expert/operator knowledge and experience about a physical system to be modeled, can readily be incorporated into TS fuzzy models in the form of fuzzy sets, fuzzy logic, or fuzzy rules. Virtually all the TS fuzzy models in the literature use linear functions of input variables as consequent of the fuzzy rules. Many learning schemes have been developed to automatically configure one or more components of TS fuzzy models so that a TS fuzzy model can quickly be established when qualitative/quantitative information is available [6], [15]. Despite of

Manuscript received November 10, 1996; revised September 17, 1998. This work was supported in part by Texas Higher Education Coordinating Board Grant 004952-054.

The author is with the Department of Physiology and Biophysics, Biomedical Engineering Center, The University of Texas Medical Branch, Galveston, TX 77555-0456 USA.

Publisher Item Identifier S 1094-6977(99)02771-6.

success of TS fuzzy systems in practice, analytical study has been scarce [24], [25], compared with the existing analytical results on Mamdani fuzzy systems (e.g., [18], [20]–[23]).

TS fuzzy modeling scheme will be more powerful if several major problems associated with it are overcome. First, almost all the TS fuzzy models developed so far were used merely for empirically mimicking measured input-output data sets of the physical systems modeled. However, a fuzzy model that can mimic a number of measured input-output data sets does not necessary mean the model is a valid one. More rigorous methods are needed to ensure model quality. Yet, there exists no analytical means for theoretically checking quality of a TS fuzzy model and possibly invalidate it. Second, TS fuzzy models were seldom utilized as control models (the only exceptions appear to be the studies [11], [12], [14] in which some TS fuzzy controllers were developed to control one type of TS fuzzy models. Stability and robustness of the closed-loop fuzzy control system were the main subjects in these research). This is in sharp contrast to the conventional black-box models, like Auto-Regressive (AR), Moving-Average (MA), and Auto-Regressive with the eXtra input (ARX) models [7], which were developed to facilitate design of a controller for controlling the physical system modeled. Third, analytical structure of TS fuzzy models is currently unknown, letting alone possible connection between the TS fuzzy models and the conventional models. The fuzzy models have always been treated and used as black boxes. Finally, there exist no systematic method that can be used to design a controller to control a given TS fuzzy model and achieve not only system stability but user-desired tracking or setpoint control performance.

The objectives of this research were to solve these problems for a general class of TS fuzzy dynamic systems that use arbitrary continuous input fuzzy sets, any types of fuzzy logic AND operators, fuzzy rules with linear consequent and the generalized defuzzifier which contains the popular centroid defuzzifier as a special case.

II. A General Class of TS Fuzzy Dynamic System Models

A TS fuzzy model is composed of input fuzzy sets, fuzzy logic AND operators, fuzzy rules with linear functions of input variables, and a defuzzifier. For the general TS fuzzy models studied in this paper, we denote the jth TS fuzzy rule as R_j $(1 \le j \le \Omega, \Omega$ being the total number of the fuzzy rules)

$$\begin{aligned} R_j\colon\quad & \text{IF } y(n) \text{ is } A_{0j} \text{ AND } y(n-1) \text{ is } A_{1j} \text{ AND} \\ & \quad \cdots \text{AND } y(n-m) \text{ is } A_{mj} \\ & \text{THEN } y(n+1) = a_{0j}y(n) + a_{1j}y(n-1) + \cdots \\ & \qquad + a_{mj}y(n-m) + b_{0j}u(n) \\ & \qquad + b_{1j}u(n-1) + \cdots + b_{pj}u(n-p) \\ & \qquad = \sum_{i=0}^{m} a_{ij}y(n-i) + \sum_{k=0}^{p} b_{kj}u(n-k) \end{aligned} \tag{1}$$

where $y(n)$ and $u(n)$ are, respectively, model output and input at time n (n, a positive integer, represents sampling time nT where T is sampling period). Here, a_{ij} and b_{kj} are constant parameters. A_{ij} is a fuzzy set fuzzifying $y(n-i)$, and we denote its membership function as $\mu_{ij}(y(n-i))$ which may be any shape but required to be continuous. We suppose that there are P_i different fuzzy sets for fuzzification of $y(n-i)$ in all the fuzzy rules. Subsequently, there exist $\Omega = P_0 \times P_1 \cdots \times P_m$ different combinations of the fuzzy sets and that many fuzzy rules are needed to cover all the combinations. To combine the membership values of the fuzzy sets in the rule antecedent, any types of fuzzy logic AND operators may be used and different types of AND operators may be used in different rules. Using $\otimes$ as a symbol to represent an arbitrary type of fuzzy logic AND operator, the combined membership for $y(n+1)$ in the rule consequent in the jth rule is

$$\mu_j(\vec{y}(n)) = \mu_{0j}(y(n)) \otimes \mu_{1j}(y(n-1)) \otimes \cdots \otimes \mu_{mj}(y(n-m))$$

where

$$\vec{y}(n) = (y(n), y(n-1), \cdots, y(n-m)).$$

We represent $y(n) = y(n-1) = \cdots = y(n-m) = 0$ by $\vec{y}(n) = 0$ and $\mu_j(\vec{y}(n))$ in such a case is expressed as $\mu_j(0)$.

To produce crisp model output, the generalized defuzzifier [3] is used and the model output is

$$y(n+1) = \frac{\displaystyle\sum_{j=1}^{\Omega} \mu_j^{\alpha}(\vec{y}(n)) \left(\sum_{i=0}^{m} a_{ij}y(n-i) + \sum_{k=0}^{p} b_{kj}u(n-k) \right)}{\displaystyle\sum_{j=1}^{\Omega} \mu_j^{\alpha}(\vec{y}(j))} \tag{2}$$

where α $(0 \le \alpha < +\infty)$ is a design parameter. Different defuzzification strategies can be realized by using different values for α. The popular centroid defuzzifier and mean of maximum defuzzifier are just two special cases when $\alpha = 1$ and ∞, respectively [3].

III. Analytical Structure and Local Stability of the General TS Fuzzy Dynamic System Models

In this section, we first reveal the analytical structure of the above-defined general TS fuzzy dynamic system models and relate the resulting structure to ARX model. We then establish a necessary and sufficient condition for analytically judging local stability of the fuzzy dynamic system models at the equilibrium point (i.e., origin). Finally, we show how to use the condition to check and possibly invalidate a TS fuzzy model.

Theorem 1: The general TS fuzzy dynamic system models are nonlinear time-varying ARX dynamic models.

Proof: We rewrite (2) as

$$\begin{aligned} y(n+1) &= \frac{\displaystyle\sum_{j=1}^{\Omega} \mu_j^{\alpha}(\vec{y}(n)) \left(\sum_{i=0}^{m} a_{ij}y(n-i) + \sum_{k=0}^{p} b_{kj}u(n-k) \right)}{\displaystyle\sum_{j=1}^{\Omega} \mu_j^{\alpha}(\vec{y}(n))} \\ &= \sum_{i=0}^{m} \theta_i(\vec{y}(n))y(n-i) + \sum_{k=0}^{p} \varphi_k(\vec{y}(n))u(n-k) \end{aligned} \tag{3}$$

where

$$\theta_i(\vec{y}(n)) = \frac{\displaystyle\sum_{j=1}^{\Omega} \mu_j^{\alpha}(\vec{y}(n)) \cdot a_{ij}}{\displaystyle\sum_{j=1}^{\Omega} \mu_j^{\alpha}(\vec{y}(n))} \quad \text{and}$$

$$\varphi_k(\vec{y}(n)) = \frac{\displaystyle\sum_{j=1}^{\Omega} \mu_j^{\alpha}(\vec{y}(n)) \cdot b_{kj}}{\displaystyle\sum_{j=1}^{\Omega} \mu_j^{\alpha}(\vec{y}(n))}. \tag{4}$$

The result of (3) can be expressed as

$$y(n+1) - \sum_{i=0}^{m} \theta_i(\vec{y}(n))y(n-i) = \sum_{k=0}^{p} \varphi_k(\vec{y}(n))u(n-k). \tag{5}$$

Recall that the linear time-invariant ARX dynamic model [7] is

$$y(n+1) + \sum_{i=0}^{m} c_i y(n-i) = \sum_{k=0}^{p} d_k u(n-k) + e(n+1) \tag{6}$$

where c_i and d_k are constant parameters and $e(n+1)$ represents random error. Comparing (5) with (6), one sees that the general TS fuzzy dynamic system models are nonlinear time-varying ARX models without the $e(n+1)$ term. The nonlinearity and time-variation of the fuzzy models are due to $\theta_i(\vec{y}(n))$ and $\varphi_k(\vec{y}(n))$, whose values are determined by $\mu_j(\vec{y}(n))$'s that change with $\vec{y}(n)$ and hence with time. ■

This theorem shows that, in the context of traditional dynamic system modeling, TS fuzzy modeling is indeed a rational and viable way to construct nonlinear time-varying dynamic models. A significant and unique advantage of the fuzzy modeling approach is that both qualitative information (e.g., knowledge and experience of system expert/operator) and quantitative information (e.g., measured numerical data) can be utilized during modeling. Further, traditional modeling mainly focuses on linear time-invariant dynamic systems whereas, as we show here, TS fuzzy modeling scheme can handle nonlinear time-varying dynamic systems. Therefore, TS fuzzy modeling approach may be more desirable and effective when dealing with complex systems.

Disclosure of the analytical structure of the general TS fuzzy dynamic system models makes it possible to theoretically and precisely investigate various aspects of the fuzzy system models. In present work, we focused on stability of the fuzzy models, which characterizes one of the most important aspects of physical systems. There exist two types of system stability: global stability and local stability, and one type cannot replace the other as each has its distinctive advantages and disadvantages. Generally speaking, global stability conditions for nonlinear systems are, in most cases, sufficient conditions, and necessary ones are uncommon. Except for linear systems, it is rare that a global stability condition is a necessary and sufficient condition. The most widely used methodology for global stability determination is the one developed by Lyapunov, which requires a Lyapunov function to be found for the system involved. Regardless of methodologies, their foremost assumption/requirement is that the complete and analytical expression of the system is explicitly available. This is impractical to the general TS fuzzy systems as their complete structures are usually not analytically derivable. A TS fuzzy system model is made up of several interrelated nonlinear components: input fuzzy sets, fuzzy rules, fuzzy logic AND operators and a defuzzifier. As such, the structures of most of the fuzzy systems are inherently complex and can be any nonlinear and time-varying forms, making analytical derivation of the complete structures for the whole input space virtually impossible.

Aside from the structure availability, even when the assumption/requirement is met, properly determining global stability is still very difficult or likely impossible. Constructing Lyapunov functions is more an art than science and heavily involves trial and error. Due to the structural and parametric complexity of the general fuzzy systems, finding a proper Lyapunov function for all the systems is practically impossible.

In view of these difficulties as well as nonlinear and time-varying nature of the TS fuzzy system models and generality of their components, we decided to concentrate on local stability. Determining local stability requires much less information. For the general TS fuzzy systems, we only need to know:

1) the structure of the fuzzy system around the equilibrium point (i.e., $\vec{y}(n) = 0$);
2) the linearizability of the fuzzy system at the equilibrium point.

Furthermore, the stability condition that we developed is a necessary and sufficient one, making it practically useful.

We now establish the local stability condition. Since stability is an inherent property of a system, it is unrelated to system input. In other words, stability of fuzzy dynamic system models (3) is determined by the following nonlinear time-varying difference equation:

$$y(n+1) - \sum_{i=0}^{m} \theta_i(\vec{y}(n))y(n-i) = 0. \tag{7}$$

If (7) is linearizable at $\vec{y}(n) = 0$, then Lyapunov's linearization method [9] can be utilized to judge local stability of the resulting linear difference equation, which will provide stability information about nonlinear systems (7) around the equilibrium point. Local stability can practically be determined by the following simple necessary and sufficient condition.

Theorem 2: If nonlinear difference equation (7) for a TS fuzzy dynamic system model of the general class (3) is linearizable at the equilibrium point, the fuzzy model is locally stable at the equilibrium point if and only if its corresponding linearized system

$$y(n+1) - \sum_{i=0}^{m} \theta_i(0)y(n-i) = 0 \tag{8}$$

is stable, where according to (4)

$$\theta_i(0) = \frac{\sum_{j=1}^{\Omega} \mu_j^{\alpha}(0) \cdot a_{ij}}{\sum_{j=1}^{\Omega} \mu_j^{\alpha}(0)}. \tag{9}$$

Proof: If nonlinear difference equation (7) of a TS fuzzy system model is linearizable at the equilibrium point, that is, if

$$\left.\frac{\partial \theta_i(\vec{y}(n-i))}{\partial y(n-i)}\right|_{\vec{y}(n)=0} = \text{unique constant}, \quad \text{for all } i \tag{10}$$

then the linearized system model is

$$y(n+1) - \sum_{i=0}^{m} \theta_i(0)y(n-i) = 0.$$

Using Lyapunov's linearization method, the establishment of Theorem 2 immediately follows. ■

The easiest way to determine stability of (8) is to use the z-transform. That is, (8) is stable if and only if all the roots of the corresponding z-transform equation

$$z - \sum_{i=0}^{m} \theta_i(0)z^{-i} = 0$$

are inside the unit circle. Later in this paper, we will use an unstable TS fuzzy system model as an example (see Example 1) to show how easily Theorem 2 can be employed for the determination of local stability.

In addition to the local stability determination, another use of Theorem 2 is to qualitatively check quality of a TS fuzzy system model. If the physical system modeled is known to be stable at the equilibrium point, the result of applying Theorem 2 to the fuzzy system model should confirm it. If the confirmation occurs, the model builder can be more confident about the quality of the fuzzy system model, at least about its behaviors around the equilibrium point. Otherwise, the fuzzy system model is incorrect and a new fuzzy system model needs to be established. This simple qualitative model verification can practically be important and useful for the TS fuzzy system modeling technique as there previously did not exist any analytical means for checking and invalidating a TS fuzzy dynamic system model. At present, the common practice on validation of a fuzzy system model is using computer simulation, which is not only time-consuming but, most importantly, can lead to erroneous validation because fuzzy systems are nonlinear and time-varying and no simulation can be comprehensive enough to cover all possible situations.

System identification and controller design are two closely related issues in theory and practice of conventional control and modeling [13]. Knowing the analytical structure of the general TS fuzzy dynamic system models enabled us to develop a design technique to systematically design an output tracking controller for them in achieving perfect tracking of any desired trajectory that is bounded and time-varying. Without the derivation of the above analytical structure of the fuzzy dynamic models, it is impossible to develop the design technique presented below.

IV. Systematic Design of Output Tracking Controllers Based on Feedback Linearization

In this section, we develop a systematic controller design technique for output tracking control of the general TS fuzzy dynamic systems (3). We assume that (1) the fuzzy system model is a true representation of the physical system to be controlled; and (2) the fuzzy system can either be stable or unstable. Our control objective is to make output of the general TS fuzzy dynamic systems achieve perfect tracking of any bounded time-varying trajectories. We denote such a trajectory as r(n). Our another requirement is that the output of the controller that we design must always be bounded, including when $n \to \infty$. The objective of our controller design is to produce such controller output that $y(n) = r(n)$ all the time (i.e., perfect tracking). The principle underlying our design method is feedback linearization, a well-established nonlinear controller design technique [4], [9]. The essence of this technique is using feedback to cancel internal nonlinearities of the system to be controlled and making the closed-loop control system linear so that linear controller design techniques can be used.

Note that at time n, we know the values of $r(n+1), u(n-1), \cdots, u(n-p), y(n), \cdots, y(n-m)$, and we can calculate the values of $\theta_0(\vec{y}(n)), \cdots, \theta_m(\vec{y}(n)), \varphi_0(\vec{y}(n)), \cdots, \varphi_p(\vec{y}(n))$. To be general, supposedly we do not know explicit expressions of $\theta_0(\vec{y}(n)), \cdots, \theta_m(\vec{y}(n)), \varphi_0(\vec{y}(n)), \cdots, \varphi_p(\vec{y}(n))$ (indeed, one will not be able to obtain them in many cases). We assume that $\varphi_0(\vec{y}(n)) \neq 0$ for any n, meaning output of the fuzzy systems always depends upon input of the systems. Using feedback linearization, we choose tracking controller for the general TS fuzzy dynamic systems (3) as follows:

$$u(n) = \frac{1}{\varphi_0(\vec{y}(n))}\left(-\sum_{i=0}^{m} \theta_i(\vec{y}(n))y(n-i) - \sum_{k=1}^{p} \varphi_k(\vec{y}(n))u(n-k) + r(n+1)\right). \quad (11)$$

Substituting (11) into (5), we obtain the output of the closed-loop fuzzy control systems

$$y(n+1) = r(n+1), \quad \text{for any } n$$

which means that we have achieved perfect tracking. The perfect tracking always starts from the beginning of the control, i.e., from time $n = 0$.

Practically, whether a controller so designed can achieve perfect tracking for the actual physical system represented by the fuzzy model used in the design depends on how accurate the model is. The perfect tracking can be achieved if the model accurately describes the physical system. Otherwise, the perfect tracking control performance will not be guaranteed, owing to incomplete cancellation of the system nonlinearities. The extent of the performance degradation relates to the degree of mismatch between the fuzzy model and the real system. The issue here is about the robustness of the resulting physical control system. Obviously, this issue is not peculiar to the control of the fuzzy systems; rather it is a general, difficult and still open issue to nonlinear system control as whole. It has hardly been addressed [2].

A controller designed by our feedback linearization technique can always achieve perfect tracking, starting at time $n = 0$, for any given desired trajectory. However, the controller output may or may not be bounded, i.e., the controller is not guaranteed to be stable. A controller is practically meaningless if its output is not bounded, because such a controller cannot physically be realized. We now study what determines stability of the controller and under what conditions the controller is stable or unstable. For better presentation, we divide general fuzzy systems (3) into two groups: the general TS fuzzy systems with $p = 0$ and the general TS fuzzy systems with $p \geq 1$, and study the controller stability accordingly.

A. Controller Stability for the General TS Fuzzy Dynamic Systems with $p = 0$

According to (3), the general TS fuzzy dynamic systems with $p = 0$ are described by

$$y(n+1) - \sum_{i=0}^{m} \theta_i(\vec{y}(n))y(n-i) = \varphi_0(\vec{y}(n))u(n). \quad (12)$$

This class of fuzzy dynamic systems is widely used in theory and practice of fuzzy control and modeling. According to (11), controllers designed using our method for these fuzzy systems are

$$u(n) = \frac{1}{\varphi_0(\vec{y}(n))}\left(-\sum_{i=0}^{m} \theta_i(\vec{y}(n))y(n-i) + r(n+1)\right). \quad (13)$$

Because a desired trajectory $r(n)$ is always bounded and $y(n-i) = r(n-i)$ for $i = 0, 1, \cdots, m, y(n-i)$ are bounded, too. Thus, $u(n)$ is always bounded and the controllers are always stable.

B. Controller Stability for the General TS Fuzzy Dynamic Systems with $p \geq 1$

We now study the controller stability for the general TS fuzzy dynamic systems with $p \geq 1$ in (3). For this group of fuzzy systems, if desired trajectory constantly varies, the controller output will, too. Because of the time-varying and nonlinear nature of the fuzzy systems, it is difficult to analyze the controller stability if desired trajectory endlessly changes. A related important question is: if a desired trajectory does not change forever, say it is only a step function, will the controller designed be guaranteed always stable? The answer, as we will show now, is no.

Assume a desired trajectory has a final and fixed position. Our tracking control task is to make output of the general TS fuzzy dynamic systems with $p \geq 1$ follow a desired trajectory to reach a final and fixed position within a finite period of time. One example of such tracking control is to park a car while another one is to reach a still object by a robot arm. Without loss of generality, we assume that the desired trajectory varies with time before time N and becomes unchanged thereafter. Mathematically, a desired trajectory is described by a time series: $r(0), r(1), \cdots, r(N), r_f, r_f, \cdots,$ where r_f is the final fixed position and $r(n) = r_f$ when $n > N$.

Since a controller designed using our method always achieves perfect tracking, output of the fuzzy systems is always r_f after time N. This means that $y(n) = \cdots = y(n-m) = r_f$ when $n > N+m$. Additionally, when $n > N+m, \varphi_k(\vec{y}(n))$ and $\theta_i(\vec{y}(n))$ in (11) become constants because $\vec{y}(n)$ becomes constant $\vec{r}_f$ (i.e., $y(n) = \cdots = y(n-m) = r_f$). We denote $\varphi_k(\vec{r}_f)$ and $\theta_i(\vec{r}_f)$ as respective values of $\varphi_k(\vec{y}(n))$ and $\theta_i(\vec{y}(n))$ when $n > N+m$. Using all these facts, nonlinear time-varying controller (11) becomes a linear time-invariant controller when $n > N+m$

$$\sum_{k=1}^{p} \varphi_k(\vec{r}_f) u\ (n-k) = \left(1 - \sum_{i=0}^{m} \theta_i(\vec{r}_f)\right) r_f, \qquad k \geq 1. \quad (14)$$

In order for the controller to be stable (i.e., $u(n)$ is bounded), all the roots of the z-transform equation of (14)

$$\sum_{k=1}^{p} \varphi_k(\vec{r}_f) z^{-k} = 0 \quad (15)$$

must be inside the unit circle. One sees that whether a controller is stable depends on $\varphi_k(\vec{r}_f)$, which are the parameter values of the fuzzy dynamic system to be controlled when $\vec{y}(n) = \vec{r}_f$. The controller stability depends not only on the parameters of the fuzzy system but also on the final fixed position of the desired trajectory, r_f. For the same fuzzy system, it is possible that the controller is stable for one final position but unstable for another one. We will show this in Examples 2 and 3 in next section.

If a controller is stable, the controller output corresponding to r_f, designated as u_f, can be computed by letting $u(n-1) = \cdots = u(n-p) = u_f$ in (14), which yields

$$u_f = \frac{\left(1 - \sum_{i=0}^{m} \theta_i(\vec{r}_f)\right) r_f}{\sum_{k=1}^{p} \varphi_k(\vec{r}_f)}. \quad (16)$$

It can easily be proved that if the denominator of (16) is replaced by $\varphi_k(\vec{r}_f), u_f$ computed will be for the general TS fuzzy systems with $p = 0$, meaning (16) contains the steady-state controller output for those fuzzy systems as a special case. Controller output will reach and stay at u_f after time $N+m$ for the fuzzy systems with $p = 0$. For the fuzzy systems with $p \geq 1$, controller output will reach and stay at u_f after time $N+\tau$, where $\tau > m$. According to (14), how large τ is depends on how stable the controller is, which is determined by $\varphi_k(\vec{r}_f)$. The more stable the controller, the smaller the τ.

Requiring all the roots of (15) be inside the unit circle is equivalent to requiring the general TS fuzzy dynamic systems be minimum phase systems when $\vec{y}(n) = \vec{r}_f$ (note that the fuzzy systems become linear time-invariant systems when $\vec{y}(n) = \vec{r}_f$). A discrete-time system that has open-loop zeros outside the unit circle is a nonminimum phase system [9]. All the fuzzy systems with $p = 0$ are always minimum phase systems, regardless of the desired trajectory. As such, controllers designed using our feedback linearization method are always stable. For any fuzzy system with $p \geq 1$, it belongs to one of the following three situations:

1) it is a minimum phase system for any value of r_f;
2) it is a nonminimum phase system for any value of r_f;
3) it is a minimum phase system for some values of r_f and is a nonminimum phase system for the remaining values.

We summarize these controller stability results in the form of theorem as follows.

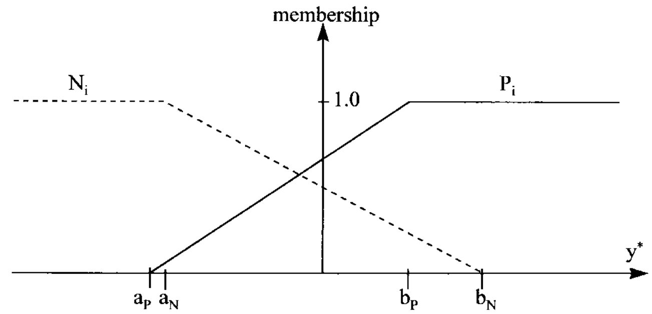

Fig. 1. Illustrative membership function definition of the six fuzzy sets used in Examples 1, 2, and 3. The mathematical definitions are given in (17) and (18) and the values of the parameters are listed in Table I.

TABLE I
VALUES OF THE PARAMETERS IN THE SIX MEMBERSHIP FUNCTIONS USED IN EXAMPLES 1, 2, AND 3. THE MATHEMATICAL DEFINITIONS ARE GIVEN IN (17) AND (18)

	a_P	b_P	k_P	d_P	a_N	b_N	k_N	d_N
P_0	-1	0.8	0.5556	0.5556				
P_1	-1.1	1.7	0.3571	0.3929				
P_2	-1.1	1.3	0.4167	0.4583				
N_0					-0.7	0.9	-0.625	0.5625
N_1					-0.6	1.4	-0.5	0.7
N_2					-0.6	1	-0.625	0.625

Theorem 3: Controller (11) designed for general TS fuzzy dynamic systems (3) with $p = 0$ is always stable for any bounded time-varying trajectory. Controller (11) designed for a fuzzy system with $p \geq 1$ is stable at a given value of r_f if and only if the fuzzy system is a minimum phase system at that value.

Since the designed fuzzy control system always achieves perfect tracking, the controller can be regarded stable between time 0 and $N+\tau$. If the system satisfies Theorem 3, it is stable for the rest of the time. Therefore, the fuzzy control system is stable in a global sense, not in a local sense (i.e., around the origin only).

For any given fuzzy system with $p \geq 1$, before utilizing our design method, one should use (15) to check stability of the controller to be designed. If the controller is determined to be stable, then design it. Otherwise, the desired perfect tracking is not achievable for the given fuzzy system model because the stability condition stated by Theorem 3 is a necessary and sufficient one.

V. NUMERICAL EXAMPLES

We now demonstrate three examples that are related to each other to show how to use our new results and methods. The first example displays how to use Theorem 2 to analytically determine local stability of a TS fuzzy dynamic system model. We purposely use an unstable fuzzy system.

Example 1: Suppose that we have identified a physical system using the TS fuzzy modeling technique. Assume the resulting TS fuzzy system model has the following eight TS fuzzy rules:

R_1: IF $y(n)$ is P_0 AND $y(n-1)$ is P_1 AND $y(n-2)$ is P_2

THEN $y(n+1) = -2y(n) - y(n-1) - 3.1y(n-2) + u(n) + 0.5u(n-1)$

R_2: IF $y(n)$ is P_0 AND $y(n-1)$ is P_1 AND $y(n-2)$ is N_2

THEN $y(n+1) = -4y(n) - 18y(n-1) - 1.8y(n-2) + 9u(n) + 0.4u(n-1)$

R_3: IF $y(n)$ is P_0 AND $y(n-1)$ is N_1 AND $y(n-2)$ is P_2

THEN $y(n+1) = -7y(n) - 2.3y(n-1) - 1.9y(n-2) - 2u(n) + 4.6u(n-1)$

R_4: IF $y(n)$ is P_0 AND $y(n-1)$ is N_1 AND $y(n-2)$ is N_2

THEN $y(n+1) = -3y(n) - 8.5y(n-1) - 1.1y(n-2) + u(n) + 0.5u(n-1)$

R_5: IF $y(n)$ is N_0 AND $y(n-1)$ is P_1 AND $y(n-2)$ is P_2

THEN $y(n+1) = -7.5y(n) - 2.6y(n-1) - 2y(n-2) + 0.5u(n) + 0.7u(n-1)$

R_6: IF $y(n)$ is N_0 AND $y(n-1)$ is P_1 AND $y(n-2)$ is N_2

THEN $y(n+1) = -3y(n) - 5.5y(n-1) - 1.2y(n-2) + 1.2u(n) - u(n-1)$

R_7: IF $y(n)$ is N_0 AND $y(n-1)$ is N_1 AND $y(n-2)$ is P_2

THEN $y(n+1) = -1.7y(n) - 4.2y(n-1) - 1.8y(n-2) + 4.2u(n) + 0.2u(n-1)$

R_8: IF $y(n)$ is N_0 AND $y(n-1)$ is N_1 AND $y(n-2)$ is N_2

THEN $y(n+1) = -1.3y(n) - 5.8y(n-1) - 2.9y(n-2) - 3.6u(n) + 0.2u(n-1)$.

Here, P_i and N_i ($i = 0$, 1, 2 and P stands for "Positive" whereas N stands for "Negative") are six fuzzy sets (Fig. 1 illustrates the definitions). The membership functions of P_i are described by

$$\mu_P(y^*) = \begin{cases} 0, & y^* \le a_P \\ k_P y^* + d_P, & a_P < y^* \le b_P \\ 1, & y^* > b_P \end{cases} \tag{17}$$

whereas the membership functions of N_i are defined by

$$\mu_N(y^*) = \begin{cases} 1, & y^* \le a_N \\ k_N y^* + d_N, & a_N < y^* \le b_N \\ 0, & y^* > b_N \end{cases} \tag{18}$$

where y^* is $y(n), y(n-1)$ or $y(n-2)$. The other parameters define the shape of the membership functions and their values are listed in Table I. Product AND fuzzy logic is used for all the AND's in the rules. Also, the popular centroid defuzzifier is used (i.e., $\alpha = 1$).

The question is: is this TS fuzzy dynamic system model stable around $\vec{y}(n) = 0$?

Solution: According to Theorem 1, this TS fuzzy dynamic system model is a nonlinear time-varying ARX system

$$y(n+1) - \sum_{i=0}^{2} \theta_i(\vec{y}(n))y(n-i) = \sum_{k=0}^{1} \varphi_k(\vec{y}(n))u(n-k).$$

In order to obtain the corresponding linearized system

$$y(n+1) - \theta_0(0)y(n) - \theta_1(0)y(n-1) - \theta_2(0)y(n-2) = \varphi_0(0)u(n) + \varphi_1(0)u(n-1)$$

we first need to determine whether the nonlinear system is linearizable at $\vec{y}(n) = 0$. For this specific system, we can find it out without the explicit expressions of $\theta_i(\vec{y}(n))$ and $\varphi_k(\vec{y}(n))$. The nonlinear system is linearizable because the conditions expressed in (10) hold. This is due to:

1) all the membership functions for $y(n), y(n-1)$, and $y(n-2)$ are differentiable at $\vec{y}(n)) = 0$;
2) $\mu_j(\vec{y}(n))$ yielded by product AND fuzzy logic are differentiable at $\vec{y}(n) = 0$.

As a result, $\theta_i(\vec{y}(n))$ and $\varphi_k(\vec{y}(n))$ are differentiable at $\vec{y}(n) = 0$. The values of $\theta_i(0)$ and $\varphi_k(0)$ can easily be computed using (4), and the resulting linearized system at $\vec{y}(n) = 0$ is

$$y(n+1) + 3.4555y(n) + 4.4621y(n-1) + 1.9681y(n-2) = 0.669u(n) + 0.0459y(n-1). \tag{19}$$

The corresponding z-transform equation is

$$z^3 + 3.4555z^2 + 4.4621z + 1.9681 = 0.$$

The three roots are $z = -0.9342$ and $z = -1.2606 \pm 0.7194i$. The last two roots are outside the unit circle. Hence, the given TS fuzzy dynamic system is unstable at $\vec{y}(n) = 0$. Fig. 2 shows the system output when a very small initial value is given ($y(0) = 0.0001$). The system output diverges with time, clearly demonstrating instability of the system and confirming our analytical result.

In the second example below, we exhibit how to use our feedback linearization design method to design a stable controller for the unstable TS fuzzy system given in Example 1 and achieve perfect output tracking performance.

Example 2: Using the feedback linearization method presented in this paper, design a tracking controller for the TS fuzzy dynamic system in Example 1 so that the output of the fuzzy system perfectly follows the following trajectory (Fig. 3):

$$r(n) = \begin{cases} 0.8\sin(3\pi n/100), & 0 \le n \le 50 \\ 0.4, & 51 \le n \le 100. \end{cases}$$

Is the controller so designed stable?

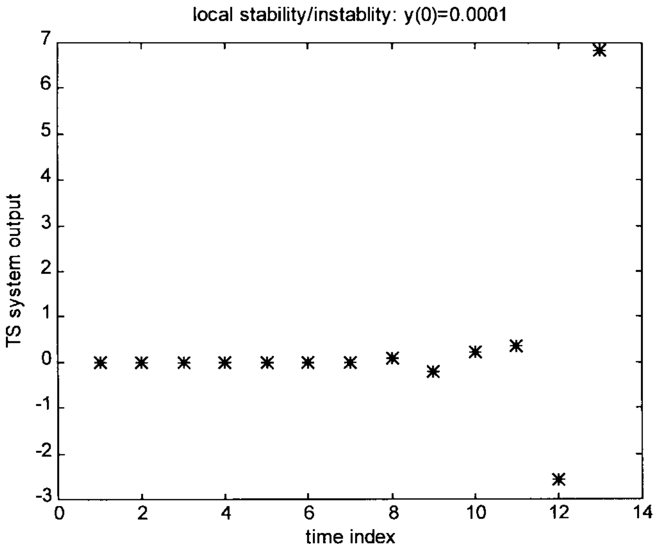

Fig. 2. Simulated output of the TS fuzzy dynamic system given in Example 1, confirming local instability of the fuzzy system, which is determined analytically by our necessary and sufficient stability condition. The initial system output is set 0.0001.

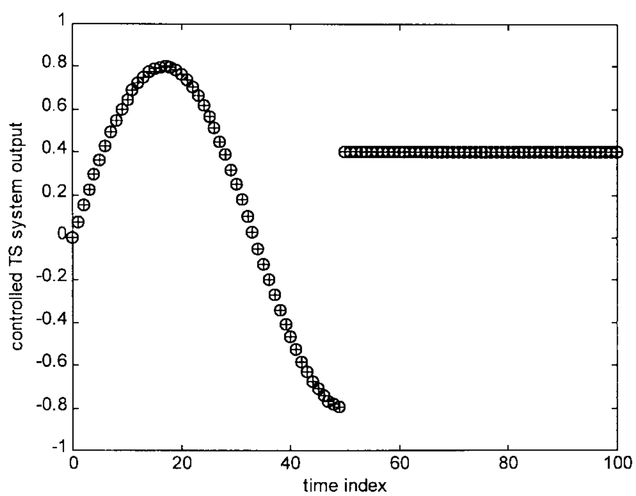

Fig. 3. Output of the unstable TS fuzzy dynamic system controlled by an output tracking controller in Example 2, which is designed using our feedback linearization technique. Sign ∘ represents the desired output trajectory whereas sign + represents the fuzzy system output. The figure shows that perfect tracking is achieved. Note that the final fixed position of the desired trajectory, r_f, is 0.4.

Solution: Before designing the controller, we should use Theorem 3 to determine whether the controller to be designed will be stable. According to the given desired trajectory, the final fixed position is: $r_f = 0.4$. According to (15), the z-transform equation for the controller stability determination is

$$\phi_0(\vec{r}_f) + \varphi_1(\vec{r}_f)z^{-1} = 0$$

whose root is

$$z = -\frac{\varphi_1(\vec{r}_f)}{\varphi_0(\vec{r}_f)}.$$

It can be calculated easily from the given TS fuzzy system that $\varphi_0(\vec{r}_f) = 1.4641$ and $\varphi_1(\vec{r}_f) = 1.2623$, and hence the root is $z = -0.8622$, indicating that the fuzzy system is a minimum phase system when $\vec{y}(n) = r_f = 0.4$, for $n > 50$. Thus, the tracking controller to be designed will be stable. The tracking controller is

$$u(n) = \frac{1}{\varphi_0(\vec{y}(n))}\left(-\sum_{i=0}^{2} \theta_i(\vec{y}(n))y(n-i) - \varphi_1(\vec{y}(n))u(n-1) + r(n+1)\right).$$

According to (16), the steady-state output of the designed controller at r_f is

$$u_f = \frac{\left(1 - \sum_{i=0}^{2} \theta_i(\vec{r}_f)\right)r_f}{\sum_{k=0}^{1} \varphi_k(\vec{r}_f)}.$$

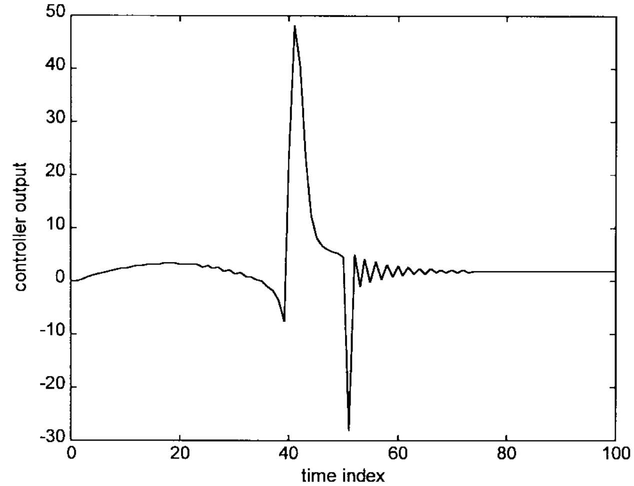

Fig. 4. Output of the output tracking controller designed using our feedback linearization technique in Example 2. The controller is stable, confirming the result of the analytical determination. The steady-state output of the controller is 1.8461, the same as the value computed using (16).

From the given fuzzy system, we compute the values of $\theta_i(\vec{r}_f)$ and $\varphi_k(\vec{r}_f)$ as: $\theta_0(\vec{r}_f) = -3.9851, \theta_1(\vec{r}_f) = -5.5249, \theta_2(\vec{r}_f) = -2.0732, \varphi_0(\vec{r}_f) = 1.4641$, and $\varphi_1(\vec{r}_f) = 1.2623$. Consequently, $u_f = 1.8461$. Fig. 3 displays the system output along with the desired trajectory. The trajectory is always perfectly tracked. The corresponding controller output is exhibited in Fig. 4. The controller is stable and indeed the steady-state output is 1.8461, as expected.

In the last example, we show that the controller designed in Example 2 becomes unstable for the same fuzzy system at a different value of r_f.

Example 3: In Example 2, if the final fixed position of the desired trajectory is 0.7 instead of 0.4, for $51 \le n \le 100$, will the designed controller still be stable?

Solution: Now $r_f = 0.7$. One can calculate that $\varphi_0(\vec{r}_f) = 1.2081$ and $\varphi_1(\vec{r}_f) = 1.4867$, and hence the root is $z = -1.2307$ (outside the unit circle). This means that the fuzzy system becomes a nonminimum phase system when $\vec{y}(n) = \vec{r}_f = 0.7$ and consequently the designed controller becomes unstable for the new final position of the trajectory. Although the perfect tracking is still achieved, as shown in Fig. 5, the controller output grows without bound and the controller is unusable (Fig. 6), as predicted.

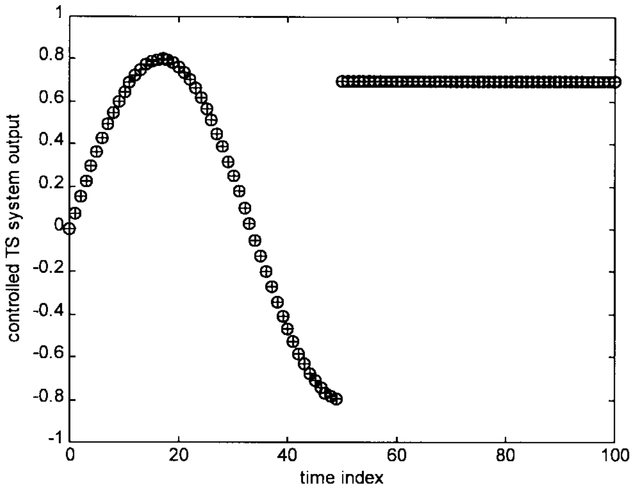

Fig. 5. Output of the unstable TS fuzzy dynamic system controlled by an output tracking controller in Example 3, which is designed in Example 2 using our feedback linearization technique. Sign ∘ represents the desired output trajectory whereas sign + represents the system output. The figure shows that perfect tracking is achieved. Note that the final fixed position of the desired trajectory, r_f, is 0.7, instead of 0.4 shown in Fig. 3 for Example 2.

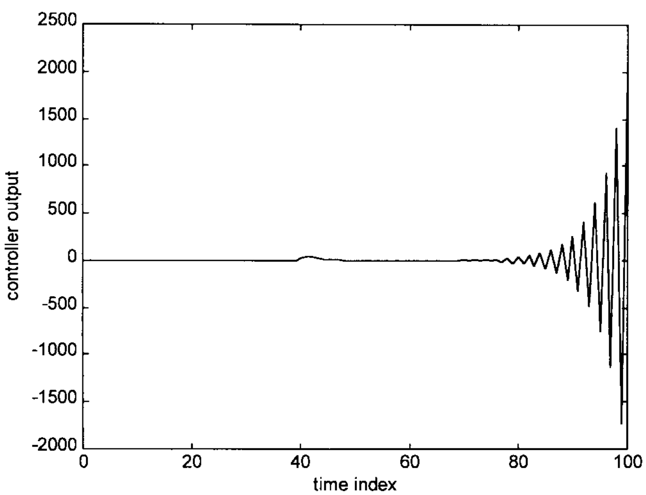

Fig. 6. Output of the output tracking controller in Example 3. Because of the change of the final position of the desired trajectory from 0.4 in Example 2 to 0.7 in Example 3, the controller becomes unstable, as predicted by using (15).

VI. Conclusion

We have proved that a general class of TS fuzzy dynamic systems is nonlinear time-varying ARX systems and have established a simple necessary and sufficient condition for analytically determining local stability of the fuzzy systems. The condition can also be used to check quality of a TS fuzzy model against the physical system modeled and invalidate the model if the conditions warrant. Based on the revealed structure of the fuzzy systems, we have developed a feedback linearization method for systematically designing a controller to control any given TS fuzzy system of the general class, stable or not, so that perfect output tracking is obtained. Our design method always produces stable controllers for a large portion of the general TS fuzzy systems that are commonly encountered [those with $p = 0$ in (3)], regardless of desired trajectories as long as they are bounded. For the remaining TS fuzzy systems [those with $p \geq 1$ in (3)], we have proved that whether the designed controller is stable depends on the fuzzy system to be controlled as well as the desired trajectory. We have derived a simple and practical necessary and sufficient condition, in relation to stability of nonminimum phase systems, for analytically determining the controller stability. We have given three concrete numerical examples to demonstrate practicality and utility of our new results.

The results obtained in this paper cover a very general class of TS fuzzy dynamic systems, which are actually, as we have shown, nonlinear and time-varying ARX systems. Our results are not only unique and hence valuable to fuzzy systems but also useful to the conventional studies of output tracking control of nonlinear time-varying systems.

References

[1] G. Chen, T. T. Pham, and J. J. Weiss, "Fuzzy modeling of control systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 31, pp. 414–429, 1995.

[2] B. Friedland, *Advanced Control System Design*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[3] D. P. Filev and R. R. Yager, "A generalized defuzzification method via BAD distributions," *Int. J. Intell. Syst.*, vol. 6, pp. 687–697, 1991.

[4] J. W. Grizzle and P. V. Kokotovic, "Feedback linearization of sampled-data systems," *IEEE Trans. Automat. Contr.*, vol. 33, pp. 857–859, 1988.

[5] D. A. Linkens and H. O. Nyongesa, "Learning systems in intelligent control: An appraisal of fuzzy, neural and genetic algorithm control applications," *Proc. Inst. Elect. Eng.*, vol. 143, pp. 367–386, 1996.

[6] K. Liu and F. L. Lewis, "Adaptive tuning of fuzzy logic identifier for unknown nonlinear systems," *Int. J. Adapt. Contr. Signal Process.*, vol. 8, pp. 573–586, 1994.

[7] L. Ljung, *System identification: Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[8] J. Sjoberg, Q. Zhang, and L. Ljung, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, vol. 31, pp. 1691–1724, 1995.

[9] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[10] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man Cybern.*, vol. SMC-15, pp. 116–132, 1985.

[11] K. Tanaka and M. Sano, "A robust stabilization problem of fuzzy controller systems and its applications to backing up control of a truck-trailer," *IEEE Trans. Fuzzy Syst.*, vol. 2, pp. 119–134, 1994.

[12] K. Tanaka, T. Ikeda, and H. O. Wang, "Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: Quadratic stabilizability, H^∞ control theory, and linear matrix inequalities," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 1–13, 1996.

[13] P. M. Van Den Hof and R. J. P. Schrama, "Identification and control—Closed-loop issues," *Automatica*, vol. 31, pp. 1751–1770, 1995.

[14] H. O. Wang, K. Tanaka, and M. F. Griffin, "An approach to fuzzy control of nonlinear systems: Stability and design issue," *IEEE Trans. Fuzzy Syst.*, vol. 4, pp. 14–23, 1996.

[15] L. Wang and R. Langari, "Building Sugeno-type models using fuzzy discretization and orthogonal parameter estimation techniques," *IEEE Trans. Fuzzy Syst.*, vol. 3, pp. 454–458, 1995.

[16] ——, "Identification of time-varying fuzzy systems," *Int. J. General Syst.*, vol. 25, pp. 203–218, 1996.

[17] ——, "Complex systems modeling via fuzzy logic," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, pp. 100–106, Feb. 1996.

[18] L.-X. Wang, *Adaptive Fuzzy Systems and Control—Design and Stability Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[19] R. R. Yager and D. P. Filev, *Essentials of Fuzzy Modeling and Control*. New York: Wiley, 1994.

[20] H. Ying, "The simplest fuzzy controllers using different inference methods are different nonlinear proportional-integral controllers with variable gains," *Automatica*, vol. 29, pp. 1579–1589, 1993.

[21] ——, "A fuzzy controller with linear control rules is the sum of a global two-dimensional multilevel relay and a local nonlinear proportional-integral controller," *Automatica*, vol. 29, pp. 499–505, 1993.

[22] ——, "General analytical structure of typical fuzzy controllers and their limiting structure theorems," *Automatica*, vol. 29, pp. 1139–1143, 1993.

[23] ——, "Sufficient conditions on general fuzzy systems as function approximators," *Automatica*, vol. 30, pp. 521–525, 1994.
[24] ——, "The Takagi–Sugeno fuzzy controllers using the simplified control rules are nonlinear variable gain controllers," *Automatica*, vol. 34, pp. 157–167, 1998.
[25] ——, "Constructing nonlinear variable gain controllers via the Takagi–Sugeno fuzzy control," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 226–234, 1998.

A Color Texture Based Visual Monitoring System For Automated Surveillance

George Paschos and Kimon P. Valavanis

***Abstract*—This paper describes a visual monitoring system that performs scene segmentation based on color and texture information. Color information is combined with texture and corresponding segmentation algorithms are developed to detect and measure changes (loss/gain) in a given scene or environment over a period of time. The xyY color space is used to represent the color information. The two chromaticity coordinates (x, y) are combined into one, thus, providing the chrominance (spectral) part of the image, while Y describes the luminance (intensity) information. The proposed color texture segmentation system processes luminance and chrominance separately. Luminance is processed in three stages: filtering, smoothing, and boundary detection. Chrominance is processed in two stages: histogram multi-thresholding, and region growing. Two or more images may be combined at the end in order to detect scene changes, using logical pixel operators. As a case study, the methodology is used to determine wetlands loss/gain. For comparison purposes, results in both the xyY and HIS color spaces are presented.**

I. Introduction

Texture has been widely accepted as a feature of primary importance in image processing and computer vision since it provides unique information about the physical characteristics of surfaces, objects, and scenes [1], [2]. An image may represent a specific textural pattern, while in other cases, an image may be composed of two or more textural patterns. In the first case, the problem encountered is that of classification, since a single texture has to be recognized. In the second case, one has to separate the different textures from each other within a single image, thus, performing an image segmentation task.

There has been considerable research in the area of texture analysis (i.e., description, segmentation, classification) [3]–[5]. However, most of the work has focused on methods using gray-level images, where only the luminance (intensity) component of the image signal is utilized. Only limited work has been reported in the literature related to the use of color in texture analysis [6], [7]. In order to incorporate the chromatic information into texture analysis, assuming that the RGB color space is used, the following choices exist.

1) Each color band (i.e., R, G, B) is processed separately.
2) Information across different bands (e.g., cross-correlations RG, RB, GB) is extracted.
3) Both individual color band and cross-band information is used.
4) A composite measure to describe the chromatic information is used.

Methods based on one of the first three choices have been recently reported [6], [7]. The fourth alternative is explored in this research using the xyY color space [8]. The proposed Color Texture Analysis System is shown in Fig. 1. The main goal of the system is to separate a given image into two parts, namely, a Region of Interest (ROI), and the rest of the image (i.e., the background). A ROI is typically an area of the image that represents something meaningful in the corresponding real-world scene. For example, an aerial image may capture a piece of land surrounded by water. The land, in this case, is the ROI, and the surrounding water is the background. The system performs analysis on luminance and chrominance in parallel, and, at the final stage, results are combined to detect changes (i.e., loss/gain) in a specific area of the image (ROI).

Processing starts by transforming a given image from RGB to xyY (Fig. 1). This produces the luminance component (Y) directly, whereas the two chromaticity values (x, y) are combined to provide for a single-valued chrominance. Textural information, such as sizes and orientations of basic image features (e.g., edges, blobs), is contained in the luminance component. Thus, a set of filters tuned to different sizes and orientations is applied on luminance and produces a corresponding set of filtered images. Smoothing of the filtered images follows, thus, eliminating spurious/negligible regions. The smoothed images are combined into a single image, based on a neighborhood pixel similarity measure, and boundaries of potential ROI's are extracted using a perceptron-type processing mechanism. The result of luminance processing is, thus, a Boundary Image.

Crominance processing proceeds in two stages. First, the chrominance histogram is computed and multiple thresholds are identified. Secondly, these thresholds are used to segment the chrominance image into a corresponding number of regions (i.e., potential ROI's). Thus, the result of chrominance processing is a Region Image. Using a region expansion algorithm, the Boundary and Region Images are combined to locate the desired Region of Interest (e.g., wetland area). The result is a ROI Image showing the identified ROI.

The final stage involves the comparison of two or more ROI images to locate possible scene changes. Typically, two or more images of the same real-world scene are taken at different times. Each of these images will result in a corresponding ROI Image, after going through the various segmentation stages (i.e., luminance and chrominance processing). Change detection and measurement is performed by comparing two such ROI Images using logical pixel operators.

The end result of this research is threefold:

1) incorporation of texture and color attributes for scene analysis;
2) development of computationally efficient and easily implementable algorithms for the analysis of color textures;
3) development of appropriate neural network architectures for image segmentation and classification.

One of the main applications of the proposed system is in the monitoring of wetlands. Such environments experience changes over time (i.e., partial loss/gain of wetland area). The development of autonomous surveillance systems capable of collecting data over a period of time and analyzing them using a variety of visual properties in order to identify such changes is, thus, important. The methodology presented in this paper provides the analysis component of such an autonomous system. It incorporates color and texture visual attributes into a unified framework and utilizes them to detect and measure loss/gain.

Manuscript received September 9, 1996; revised May 10, 1998.
G. Paschos is with the Computer Science Division, Florida Memorial College, Miami, FL 33054 USA.
K. P. Valavanis is with the Robotics and Automation Laboratory, A-CIM Center, University of Southwestern Louisiana, Lafayette, LA 70504 USA.
Publisher Item Identifier S 1094-6977(99)02770-4.