International Mathematical Forum, 1, 2006, no. 28, 1371 - 1382

INTELLIGENT CONTROL OF DYNAMIC SYSTEMS USING TYPE-2 FUZZY LOGIC AND STABILITY ISSUES

Oscar Castillo, Nohé Cázarez, and Dario Rico

Instituto Tecnológico de Tijuana
Calzada del Tecnológico S/N, Tijuana, B.C., México

Luis T. Aguilar

CITEDI-IPN
Ave. del Parque 1310, Mesa de Otay, Tijuana, B.C., México

Abstract

Stability is one of the more important aspects in the traditional knowledge of Automatic Control. Type-2 Fuzzy Logic is an emerging and promising area of application to Intelligent Control (in this case, Fuzzy Control). We present a design methodology based on the Margaliot work [10] for the design of Stable Mamdani Type-2 Fuzzy Logic Controllers.

1 Introduction

Fuzzy logic controllers (FLC's) are one of useful control schemes for plants having difficulties in deriving mathematical models or having performance limitations with conventional linear control schemes. Error e and change of error $\dot{e}$ are the most used fuzzy input variables in most fuzzy control works, regardless of the complexity of controlled plants. Also, either control input u (PD-type) or incremental control input Δu (PI-type) is typically used as a fuzzy output variable representing the rule consequent ("*then*" part of a rule) [5].

Stability has been one of the central issues concerning fuzzy control since Mandani's pioneer work [8], [9]. Most of the critical comments to fuzzy control are due to the lack of a general method for its stability analysis. But as Zadeh often points out, fuzzy control has been accepted by the fact that it is task-oriented control, while conventional control is characterized by setpoint-oriented control, and hence do not need a mathematical analysis of stability.

And as Sugeno says, in general, in most industrial applications, the stability of control is not fully guaranteed and the reliability of a control hardware system is considered to be more important than the stability [14].

The success of fuzzy control, however, does not imply that we do not need a stability theory for it. Perhaps the main drawback of the lack of stability analysis would be that we cannot take a model-based approach to fuzzy control design. In conventional control theory, a feedback controller can be primarily designed so that a close-loop system becomes stable [12], [13]. This approach of course restricts us to setpoint-oriented control, but stability theory will certainly give us a wider view on the future development of fuzzy control.

Therefore, many researchers have worked to improve the performance of the FLC's and ensure their stability. Li and Gatland [6] and [7] proposed a more systematic design method for PD and PI-type FLC's. Choi, Kwak and Kim [2] present a single-input FLC ensuring stability. Ying [17] present a practical design method for nonlinear fuzzy controllers, and many other researchers have results on the matter of the stability of FLC's.

This work is based on Margaliot [10] results, we use the Margaliot criterion to built an Stable Type-2 Fuzzy Logic Controller for a Nonlinear Plant, in this case, the Inverted Pendulum. The same criterion can be used either linear or nonlinear plants. This paper if organized as follows: In Section 2 we presents an introductory explanation of type-1 and type-2 FLC's. In Section 3 we extend the Margaliot result to build a general rule base for any type (1 or 2) of FLC's. Experimental results are presented in Section 4 and the concluding remarks are collected in Section 5.

2 Fuzzy Logic Controllers

We describe in this section both type-1 and type-2 fuzzy controllers, so that the differences between them are made more understandable.

2.1 Type-1 Fuzzy Control

Type-1 FLCs are both intuitive and numerical systems that map crisp inputs to a crisp output. Every FLC is associated with a set of rules with meaningful linguistic interpretations, such as

$$R^l: \text{ If } x_1 \text{ is } F_1^l \text{ and } x_2 \text{ is } F_2^l \text{ and } \ldots \text{ and } x_n \text{ is } F_n^l \text{ Then } w \text{ is } G^l$$

which can be obtained either from numerical data, or experts familiar with the problem at hand. Based on this kind of statement, actions are combined with rules in an antecedent/consequent format, and then aggregated according to approximate reasoning theory, to produce a nonlinear mapping from input space $U = U_1 \times U_2 \times \ldots \times U_n$ to the output space W, where $F_k^l \subset U_k$, $k = 1, 2, \ldots, n$, are the antecedent type-1 membership functions, and is the consequent type-1 membership function. The input linguistic variables are denoted by u_k, $k = 1, 2, \ldots, n$, and the output linguistic variable is denoted by w.

A Fuzzy Logic System (FLS) as the kernel of a FLC consist of four basic elements (Fig. 1): the *type-1 fuzzyfier*, the *fuzzy rule-base*, the *inference engine*, and the *type-1 defuzzifier*. The fuzzy rule-base is a collection of rules in the form of R^l, which are combined in the inference engine, to produce a fuzzy output. The type-1 fuzzyfier maps the crisp input into type-1 fuzzy sets, which are subsequently used as inputs to the inference engine, whereas the type-1 defuzzyfier maps the type-1 fuzzy sets produced by the inference engine into crisp numbers.

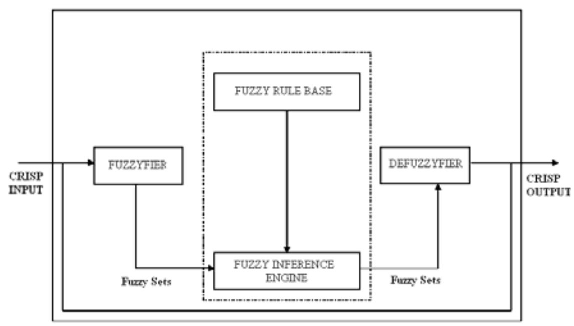

Figure 1: Structure of a type-1 fuzzy controller.

Fuzzy sets can be interpreted as membership functions u_X that associate with each element x of the universe of discourse, U, a number $u_X(x)$ in the interval [0,1]:

$$u_X : U \rightarrow [0, 1]. \tag{1}$$

For more detail of Type-1 FLS see [1], [3], [16].

2.2 Type-2 Fuzzy Control

As the type-1 fuzzy set, the concept of type-2 fuzzy set was introduced by Zadeh [18] as an extension of the concept of an ordinary fuzzy set.

A FLS described using at least one type-2 fuzzy set is called a type-2 FLS. Type-1 FLSs are unable to directly handle rule uncertainties, because they use type-1 fuzzy sets that are certain. On the other hand, type-2 FLSs, are very useful in circumstances where it is difficult to determine an exact, and measurement uncertainties [11].

It is known that type-2 fuzzy set let us to model and to minimize the effects of uncertainties in rule-based FLS. Unfortunately, type-2 fuzzy sets are more difficult to use and understand that type-1 fuzzy sets; hence, their use is not widespread yet.

Similar to a type-1 FLS, a type-2 FLS includes *type-2 fuzzyfier*, *rule-base*, *inference engine* and substitutes the defuzzifier by the output processor. The output processor includes a *type-reducer* and a *type-2 defuzzyfier*; it generates a type-1 fuzzy set output (from the type reducer) or a crisp number (from the defuzzyfier). A type-2 FLS is again characterized by IF-THEN rules, but its antecedent of consequent sets are now type-2. Type-2 FLSs, can be used when the circumstances are too uncertain to determine exact membership grades. A model of a type-2 FLS is shown in Fig. 2.

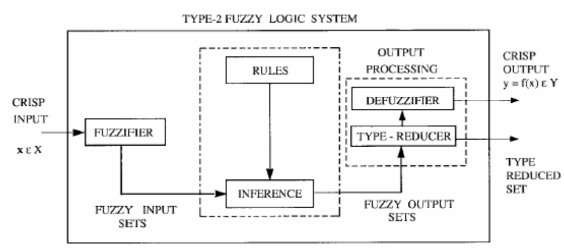

Figure 2: Structure of a type-2 fuzzy logic system.

In the case of the implementation of the type-2 FLCs, we have the same characteristics as in type-1 FLC, but we used type-2 fuzzy sets as member-ship functions for the inputs and for the outputs. Fig. 3 shows the structure of a control loop with a FLC.

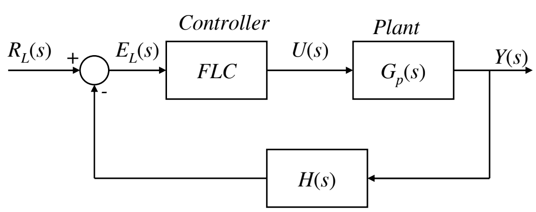

Figure 3: Fuzzy control loop.

3 Systematic Design of the Stable Fuzzy Controller

In this section we provide some explanation and details concerning to the well-known problem of designing a fuzzy stabilizing controller for the inverted

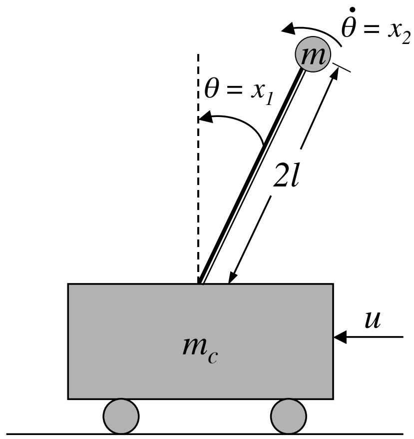

Figure 4: Inverted pendulum system.

pendulum system (Fig. 4) governed by the following nonlinear equations ([13]):

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x_1, x_2) + g(x_1, x_2)u \end{aligned} \tag{2}$$

with

$$f(x_1, x_2) = \frac{9.8(m_c + m)\sin x_1 - ml x_2^2 \sin(x_1)\cos(x_1)}{\frac{4}{3}l(m_c + m) - ml\cos^2(x_1)} \tag{3}$$

$$g(x_1, x_2) = \frac{\cos x_1}{\frac{4}{3}l(m_c + m) - ml\cos^2(x_1)} \tag{4}$$

where $x_1 = \theta \in \mathbb{R}$ is the angle of the pendulum, $x_2 = \dot{\theta} \in \mathbb{R}$ is the angular velocity, m_c is the mass of the cart, m is the mass of the pole, $2l$ is the pole length, and u is the applied force (control). The physical parameters $m_c = 0.5$ [Kg], $m = 0.2$ [Kg], and $l = 0.3$ [m] were taken from [13]. To apply the fuzzy Lyapunov synthesis method we assume the following:

A1) The states x_1 and x_2 are measurable variables.

A2) The exact equations (2)-(4) are unknown.

A3) The angular acceleration $\dot{x}_2$ is proportional to u, that is, when u increases (decreases) $\dot{x}_2$ increases (decreases).

A4) The initial conditions $x(0) = (x_1(0), x_2(0))^T$ belong to the set $\mathcal{N} = \{x \in \mathbb{R}^2 : \|x - x^\star\| \leq \epsilon\}$ where $x^\star$ is the equilibrium point.

The *control objective* is to design the rule-base of a fuzzy controller $u = u(x_1, x_2)$ to stabilize the inverted pendulum around its upright equilibrium

$$x_1^\star = 0 \quad x_2^\star = 0.$$

Theorem 3.1 that follows establishes conditions that help in the design of the fuzzy controller to ensure asymptotic stability. The proof can be found in [4].

Theorem 3.1 (Asymptotic stability) *Consider the nonlinear system (2) with an equilibrium point at the origin, i.e. $f(0) = 0$, and let $x \in \mathcal{N}$, then the origin is asymptotically stable if there exist a scalar Lyapunov function $V(x)$ with continuous partial derivatives such that*

(a) $V(x)$ is positive definite

(b) $\dot{V}(x)$ is negative definite.

If conditions of theorem 3.1 hold implies that the control input $u = u(x)$ enforces the trajectories $x(t)$ to the equilibrium point as t tends to ∞ for any initial condition $x(0) \in \mathcal{N}$, thus achieving the control objective.

The fuzzy controller design proceeds as follows. Let us introduce the Lyapunov function candidate

$$V(x_1, x_2) = \frac{1}{2}(x_1^2 + x_2^2) \tag{5}$$

that is positive-definite and radially unbounded function. The time derivative of $V = V(x_1, x_2)$ results in:

$$\dot{V} = x_1\dot{x}_1 + x_2\dot{x}_2 = x_1x_2 + x_2\dot{x}_2. \tag{6}$$

To guarantee stability of the equilibrium point $(x_1^\star, x_2^\star)^T = (0, 0)^T$ we wish to have:

$$x_1x_2 + x_2\dot{x}_2 < 0. \tag{7}$$

We can now derive sufficient conditions so that (7) will hold: If x_1 and x_2 have opposite signs, then $x_1x_2 < 0$ and (7) will hold if $\dot{x}_2 = 0$; if x_1 and x_2 are both positive, then (7) will hold if $\dot{x}_2 < -x_1$; and if x_1 and x_2 are both negative, then (7) will hold if $\dot{x}_2 > -x_1$.

We can translate these conditions into the following fuzzy rules:

- If x_1 is *positive* and x_2 is *positive* Then $\dot{x}_2$ must be *negative big*
- If x_1 is *negative* and x_2 is *negative* Then $\dot{x}_2$ must be *positive big*
- If x_1 is *positive* and x_2 is *negative* Then $\dot{x}_2$ must be *zero*

- If x_1 is *negative* and x_2 is *positive* Then $\dot{x}_2$ must be *zero*.

However, using our knowledge that $\dot{x}_2$ is proportional to u, we can replace each $\dot{x}_2$ with u to obtain the fuzzy rule-base for the stabilizing controller:

- If x_1 is *positive* and x_2 is *positive* Then u must be *negative big*
- If x_1 is *negative* and x_2 is *negative* Then u must be *positive big*
- If x_1 is *positive* and x_2 is *negative* Then u must be *zero*
- If x_1 is *negative* and x_2 is *positive* Then u must be *zero*.

It is interesting to note that the fuzzy partitions for x_1, x_2, and u follow elegantly from expression (6). Because $\dot{V} = x_2(x_1 + \dot{x}_2)$, and since we require that $\dot{V}$ be negative, it is natural to examine the signs of x_1 and x_2; hence, the obvious fuzzy partition is *positive*, *negative*. The partition for $\dot{x}_2$, namely *negative big*, *zero*, *positive big* is obtained similarly when we plug the linguistic values *positive*, *negative* for x_1 and x_2 in (7). To ensure that $\dot{x}_2 < -x_1$ ($\dot{x}_2 > -x_1$) is satisfied even though we do not know the exact magnitude of x_1, only that it is *positive* (*negative*), we must set $\dot{x}_2$ to *negative big* (*positive big*). Obviously, it is also possible to start with a given, pre-defined, partition for the variables and then plug each value in the expression for $\dot{V}$ to find the rules. Nevertheless, regardless of what comes first, we see that fuzzy Lyapunov synthesis transforms classical Lyapunov synthesis from the world of exact mathematical quantities to the world of computing with words [19].

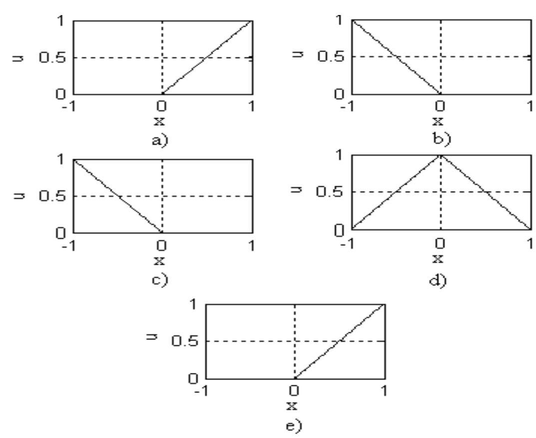

Figure 5: Kind of type-1 membership functions: a) positive, b) negative, c) negative big, d) zero and e) positive big.

To complete the controllers design, we must model the linguistic terms in the rule-base using fuzzy membership functions and determine an inference method. Following [15], we characterize the linguistic terms *positive*, *negative*, *negative big*, *zero* and *positive big* by the type-1 membership functions shows in Fig.5 for a Type-1 Fuzzy Logic Controller, and by the type-2 membership functions shows in Fig.6 for a Type-2 Fuzzy Logic Controller. Note that the type-2 membership functions are extended type-1 membership functions.

To this end, we had systematically developed a FLC rule-base that follows the Lyapunov Stability criterion. At Section 4 we present some experimental using our fuzzy rule-base to build a Type-2 Fuzzy Logic Controller.

4 Experimental Results

In Section 3 we had systematically develop a stable FLC rule-base, now we are going to show some experimental results using our stable rule-base to built Type-2 FLC. The plant parameters used in the experiments are the same shows in Section 3.

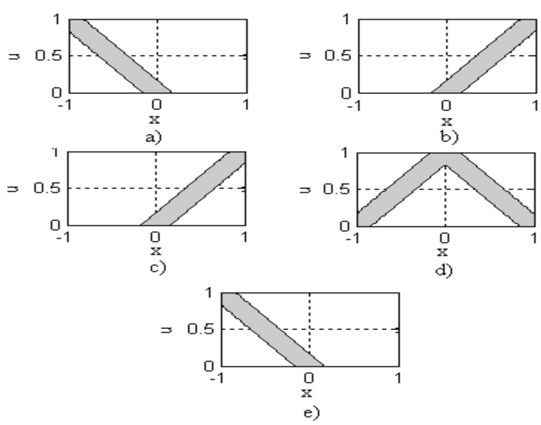

Figure 6: Kind of type-2 membership functions: a) negative, b) positive, c) positive big, d) zero and e) negative big.

Our experiments were executed with Interval Type-2 Fuzzy Sets, this kind of Fuzzy Sets are ones in which the membership grade of every do-main point is a crisp set whose domain is some interval contained in [0,1] [11]. On Fig.6 are depicted some Interval Type-2 Fuzzy Sets, for each fuzzy set, the grey area is known as the *Footprint Of Uncertain* (FOU) [11], and this one is bounded by an upper and a lower membership function as shown in Fig.7.

In our experiments we increase and decrease the value of ε to determine how much can be extended or perturbed the FOU with out loss of stability in the FLC. The initial conditions considered in the experiments was with an angle $\theta(0) = 0.1$ [rad].

At Fig. 8 we can see a simulation of the plant made with a Type-1 FLC, as can be seen, the plant has been assented in around 0.5 sec, and Fig.9 shows the graph of (6) which is always negative definite and consequently is stable.

Fig. 10 shows the simulation results of the plant made with the Type-2 FLC increasing and decreasing ε in the range of [0,1], as can be seen the plant has been assented in less than 0.5 sec, and the graph of (6) depicted at Fig.11 is always negative definite and consequently is stable.

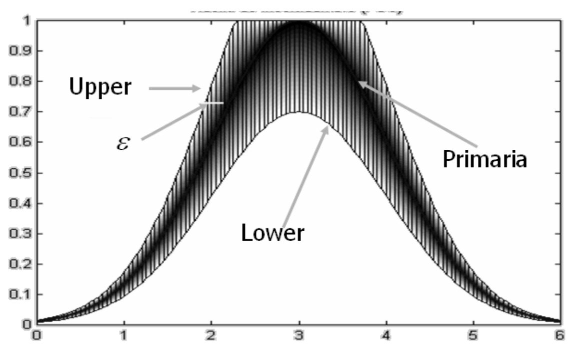

Figure 7: Type-2 fuzzy set.

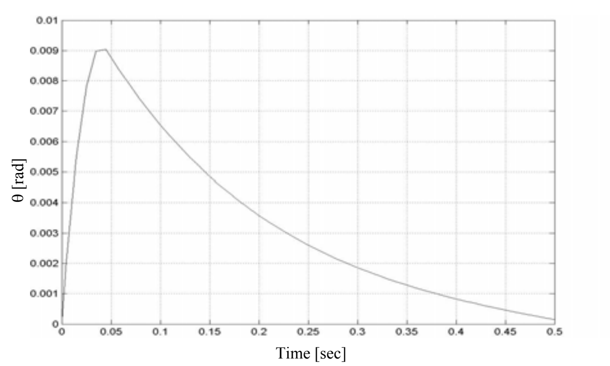

Figure 8: Response for the Type-1 FLC.

5 Conclusions

Margaliot's approach for the design of FLC is now proved to be valid for both, Type-1 and Type-2 Fuzzy Logic Controllers. On Type-2 FLC's membership functions we can perturb or change the definition domain of the FOU without loss of stability of the controller while we give an propor-tional gain to the control loop gains. In our example of the inverted pendu-lum, the stability

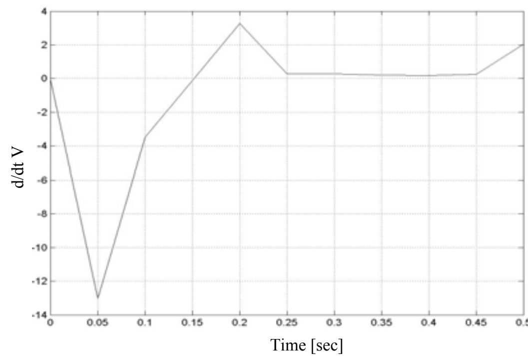

Figure 9: $\dot{V}$ for the Type-1 FLC

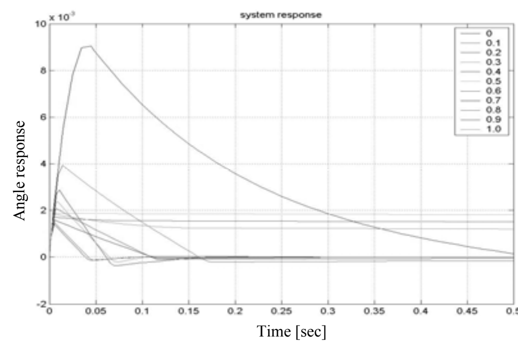

Figure 10: Angle response for the Type-2 FLC ($\varepsilon \rightarrow [0, 1]$).

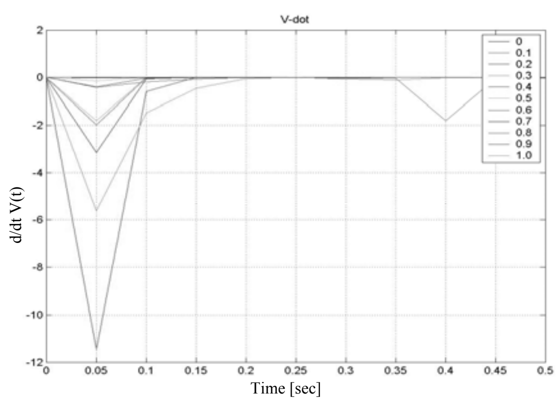

Figure 11: $\dot{V}$ response for the Type-2 FLC ($\varepsilon \rightarrow [0, 1]$).

holds extending the FOU on the domain [0,1], now we must to look for an explanation for this domain, a first look can be that we used interval type-2 sets, and these ones are defined in the interval [0,1].

References

[1] G. Chen and T.T. Pham, Introduction to fuzzy sets, fuzzy logic, and fuzzy control systems, CRC Press, USA, 2000.

[2] B.J. Choi, S.W. Kwak, and B. K. Kim, Design and Stability Analysis of Single-Input Fuzzy Logic Controller, IEEE Trans. Fuzzy Systems, 30 (2000), 303–309

[3] J.-S.R. Jang, C.-T. Sun and E. Mizutani, Neuro-Fuzzy and Soft Computing: a computational approach to learning and machine intelligence, Prentice Hall, USA, (1997).

[4] H. Khalil, Nonlinear systems: Third edition, Prentice Hall, USA, 2002.

[5] J. Lee, On methods for improving performance of PI-type fuzzy logic controllers, IEEE Trans. Fuzzy Systems, 1 (1993), 298–301.

[6] H.-X. Li and H.B. Gatland, A new methodology for designing a fuzzy logic controller, IEEE Trans. Systems Man and Cybernetics, 25 (1995), 505–512.

[7] H.-X. Li and H.B. Gatland, Conventional fuzzy control and its enhancement, IEEE Trans. Systems Man and Cybernetics, 25 (1996), 791–797.

[8] E. H. Mamdani and S. Assilian, An experiment in linguistic systhesis with a fuzzy logic controller, Int. J. Man-Machine Studies, 7 (1975), 1–13.

[9] E.H. Mamdani, Advances in the linguistic synthesis of fuzzy controllers, Int. J. Man-Machine Studies, 8 (1976), 669–679.

[10] M. Margaliot and G. Langholz, New Approaches to Fuzzy Modeling and Control: Design and Analysis, World Scientific, Singapore, (2000).

[11] J.M. Mendel, Uncertain Rule-Based Fuzzy Logic: Introduction and new directions, Prentice Hall, USA, (2000).

[12] K. Ogata, Ingenieria de Control Moderna, 3a Edicion, Prentice Hall, Spain, (1998).

[13] H. Paul and C. Yang, Sistemas de control en ingenieria, Prentice Hall, Spain, (1999).

[14] M. Sugeno, On Stability of Fuzzy Systems Expressed by Fuzzy Rules with Singleton Consequents, IEEE Trans. Fuzzy Systems, 7 (1999).

[15] L.X. Wang, A Course in Fuzzy Systems and Control, Prentice Hall, (1997).

[16] J. Yen and R. Langari, Fuzzy Logic: Intelligence, Control, and Information, Prentice Hall, New Jersey, (1998).

[17] H.Ying, Practical Design of Nonlinear Fuzzy Controllers with Stability Analysis for Regulating Processes with Unknown Mathematical Models", Automatica, 30 (1994), 1185–1195.

[18] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning, J. Information Sciences, 8 (1975), 43–80.

[19] L.A. Zadeh, Fuzzy Logic = Computing With Words, IEEE Trans. Fuzzy Systems, 4 (1996), 103–111.

Received: November 7, 2005