Adaptive Fuzzy Sliding Mode Control

적응 퍼지 슬라이딩 모드 제어

Byung Kook Yoo, Sa Cheul Jeoung, Woon Chul Ham
(유 병 국, 정 사 철, 함 운 철)

요 약 : 본 논문에서는 퍼지추정기와 슬라이딩모드제어이론이 고려되었다. 비선형시스템에 대한 슬라이딩모드제어기 설계 시에 그 시스템의 비선형함수를 추정하기 위하여 퍼지논리시스템이 사용되는 두 가지의 적응퍼지슬라이딩 모드제어방식을 제안한다. 첫번째 방식에서는 비선형시스템, $x^{(n)}=f(\underline{x},t)+b(\underline{x},t)u$ 의 알지 못하는 함수 f를 추정하기 위하여 하나의 퍼지논리시스템이 사용되어진다. 두번째 방식에서는 비선형시스템의 f와 b에 대한 추정기로서 두 개의 퍼지논리시스템이 각각 사용되어진다. 각각의 방식에 대하여 제어시스템의 안정도를 보장하도록 하는 적응법칙을 설계하며 퍼지추정기와 비선형함수와의 추정오차를 줄이기 위해 각각에 대한 강인한 제어법칙을 제안한다. 제안된 네 가지의 제어법칙에 대한 안정성을 증명하고 컴퓨터시뮬레이션에서 역진자시스템에 적용하여 그에 대한 타당성과 각각의 비교를 보인다.

Keywords: adaptive law, sliding mode control, fuzzy logic system, universal approximation theorem

Ⅰ. Introduction

In the conventional control theory, most of the control problems are usually solved by mathematical tools based on the system models. But in the real world, there are many complex industrial processes whose accurate mathematical models are not available or difficult to formulate. Because the fuzzy control can often provide a good solution for these problems by incorporating linguistic informations from human experts, fuzzy control, as an alternative to conventional control techniques, is gaining increased interests both in the academic world and in the industrial field[1,2]. Despite its practical successes in many areas, fuzzy control seems to be deficient in formal analysis and robustness aspects. This is also a great resource of criticism from some conventional control researchers. To overcome this drawback, great efforts have been done in the field of fuzzy control during the recent years. This paper is motivated by Wang's and Mendel's works that fuzzy logic systems(center average defuzzifier, product-inference rule, singleton fuzzifier, and gaussian membership function) are capable of uniformly approximating any nonlinear function over compact input space[3,4,5]. That is, any nonlinear system can be modeled by the fuzzy logic systems. Especially, Wang[3] utilized the general error dynamics of adaptive control to design the adaptive fuzzy controller. Many other researchers have attempted to apply the fuzzy approximator or fuzzy logic concepts to the conventional control techniques[6,7,8,9]. However, most of their works don't have formality.

In general, we need to know the system functions and have to find the inverse form of inertia term in system dynamics in designing the sliding mode controller. However it is not difficult to find the inertia term and its inverse form and to formulate the accurate mathematical model of the nonlinear system.system. To solve these problems, In this paper, a fuzzy approximator theory and a SMC scheme are considered. That is, fuzzy logic system theory is applied to designing the sliding mode controller for the nonlinear system, $x^{(n)}=f(\underline{x},t)+b(\underline{x},t)u$. In the first method, fuzzy logic system is utilized to approximate the unknown function f of the nonlinear system. In the second method, two fuzzy logic systems are utilized to approximate f and b, respectively. In these control schemes, simple adaptive laws are designed to approximate the nonlinear functions by fuzzy logic systems. In the first method, a robust adaptive law is also introduced to reduce the approximation errors, the differences between true system functions and fuzzy approximators. In the second method, the control law which is robust to approximation error is also designed. The stabilities of proposed control schemes are proved. These proposed control schemes are applied to an inverted pendulum system.

This paper is organized as follows. Section II presents the general fuzzy logic system and fuzzy approximator. In Section III, a SMC scheme is introduced and the general sliding mode controller is designed under assumption that f and b are known. In section IV, the first method of adaptive SMC scheme using a fuzzy logic system is proposed and adaptive laws are designed. In section V, the second method is proposed and robust control law is also designed. In both cases, adaptive fuzzy sliding mode control schemes(AFSMCs) are designed under assumption that the system functions, f and b, are unknown. The stabilities of the proposed control schemes are proved. In section VI, an inverted pendulum system is considered to verify the validities of the proposed control schemes and comparisons between the cases of simple adaptive law and robust adaptive law are noted, respectively. Conclusions are drawn in the final section.

접수일자 : 1996. 1. 30., 수정완료 : 1996. 9. 30.
정사철, 함운철 : 전북대학교 공과대학 전자공학과
유병국 : 한려산업대학교 의공학과
※ 본 논문은 한국과학재단에서 지원하여 연구하였습니다.

II. Fuzzy Logic System

Fig. 1 shows the basic configuration of a fuzzy logic system considered in this paper. In what follows a brief description of each component and the basic fuzzy operations that it performs is discussed.

1. Knowledge Base Constructed with Fuzzy Rules

The knowledge base for the fuzzy logic system comprises a collection of fuzzy IF-THEN rules. In this paper multiple-input single-out(MISO) rules will be used in the formulation of the control law. The MISO IF-THEN rule(s) are of the form

$$R(j): \textit{If } x_1 \textit{ is } A_1^j \text{ and} \cdots \text{and } x_n \textit{ is } A_n^j, \textit{ Then } y \textit{ is } C^j \quad (2.1)$$

where $\underline{x}=(x,\cdots,x_n)^T \in V \subset R^n$ and $y \in W \subset R$ denote the linguistic variables associated with the inputs and output of the FLS. A_i^j and C^l are labels of the fuzzy sets in V and W, respectively, and i denotes the number of inputs(states) of FLS, *i.e.* $i=1,2,\cdots,n$, and j denotes the number of rules of FLS, *i.e.* $j=1,2,\cdots,M$. The fuzzy rule (2.1) can be implemented using fuzzy implication, which gives

$$A_i^j \times \cdots \times A_n^j \rightarrow C^j \quad (2.2)$$

which is a fuzzy set defined in the product space $V \times W$. Based on generalizations of implications in multivalue logic, many fuzzy implication rules have been proposed in the fuzzy logic literature. In this paper, we define the implication rule using a t-norm operator, which gives

$$\mu_{A_i^j \times \cdots \times A_n^j \rightarrow C^j}(\underline{x}, y) = \mu_{A_i^j}(x_1) \bigstar \cdots \bigstar \mu_{A_n^j}(x_n) \bigstar \mu_{C^j}(y) \quad (2.3)$$

where $\bigstar$ denotes a t-norm, which corresponds to the conjunction "min" or "product" in general.

2. Fuzzy Inference Engine

The fuzzy inference engine performs a mapping from fuzzy sets in V to fuzzy sets in R, based upon the fuzzy IF-THEN rules in fuzzy rule base and the compositional rule of inference. Let B be a fuzzy set in V, then the fuzzy relational equation $B \circ R^j$, where " $\circ$ " is the sup-star composition, results in M fuzzy sets. Using the t-norm operator yields

$$\mu_{B \cdot R^j}(y) = sup_{\underline{x}} [\mu_B(\underline{x}) \bigstar \mu_{A_1^j \times \cdots \times A_n^j \rightarrow C^j}(\underline{x}, y)] \quad (2.4)$$

In order to combine the M fuzzy sets into one fuzzy set t-conorm can be employed, which results in

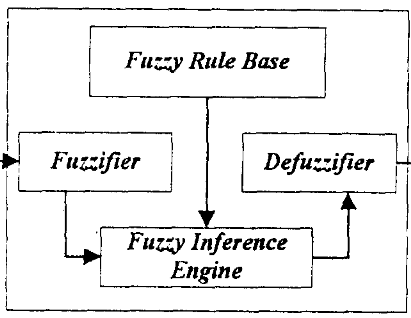


Fig. 1. A block diagram of basic fuzzy logic system.

$$\mu_{B \cdot (R^1,\cdots,R^{(M)})}(y) = \mu_{B \cdot R^{(1)}}(y) \dot{+} \cdots \dot{+} \mu_{B \cdot R^{(M)}}(y) \quad (2.5)$$

where $\dot{+}$ denotes the t-conorm(s-norm), the most commonly used operation for $\dot{+}$ is "max". If we use the product operation and choose $\bigstar$ in (2.3) and (2.4) to be an algebraic product, then the inference is called product inference. Using product inference, (2.4) becomes

$$\mu_{B \cdot R^j}(y) = sup_{\underline{x} \in V} [\mu_B(\underline{x}) \mu_{A_1^j}(x_1) \cdots \mu_{A_n^j}(x_n) \mu_{C^j}(y)]. \quad (2.6)$$

3. Fuzzifier

The fuzzifier maps a crisp point $\underline{x}$ into a fuzzy set B in V. In general, there are two possible choices of this mapping namely, singleton or nonsingleton. In this paper, we use the singleton fuzzifier mapping, *i.e.*,

$$\mu_B(\underline{x}') = \begin{cases} 1 & \text{for } (\underline{x}') = \underline{x} \\ 0 & \text{for } \textit{otherwise} \end{cases}, \text{ for } \underline{x}' \in V. \quad (2.7)$$

4. Defuzzifier

The defuzzifier maps fuzzy sets in R to a crisp point in R. In general, there are three possible choices of this mapping namely, maximum, center-average, and modified center-average defuzzifiers. In this paper, we use the center-average defuzzifier mapping, *i.e.*

$$y = \frac{\sum_{j=1}^{M} \bar{y}^j (\mu_{B \cdot R^{(j)}}(\bar{y}^j))}{\sum_{j=1}^{M} (\mu_{B \cdot R^{(j)}}(\bar{y}^j))} \quad (2.8)$$

where $\bar{y}^j$ is the point in R at which μ_{C^j} achieves its maximum value (assume that $\mu_{C^j}(\bar{y}^j)=1$).

5. Fuzzy Bases Function

The fuzzy logic system with the *center-average defuzzifier* (2.8), *product inference* (2.6), and the *singleton fuzzifier* (2.7) is of the following form:

$$y(\underline{x}) = \frac{\sum_{j=1}^{M} \bar{y}^j (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))}{\sum_{j=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))} \quad (2.9)$$

If we fix the $\mu_{A_i^j}(x_i)$'s and view the $\bar{y}^j$'s as adjustable parameters, then (2.9) can be written as

$$y(\underline{x}) = \theta^T \xi(\underline{x}), \quad (2.10)$$

where $\theta = (\bar{y}^1, \cdots, \bar{y}^M)^T$ is a parameter vector, and $\xi(\underline{x}) = (\xi^1(\underline{x}), \cdots, \xi^M(\underline{x}))^T$ is a regressive vector with the regressor $\xi^j(\underline{x})$ defined as

$$\xi^j(\underline{x}) = \frac{\prod_{i=1}^{n} \mu_{A_i^j}(x_i)}{\sum_{j=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^j}(x_i))}, \quad (2.11)$$

which are called *FBF*s(fuzzy bases functions). It is proved that a fuzzy logic system is universal approximator in [3]. We can fix all the parameters in $\xi^j(\underline{x})$ at the beginning of the FBF expansion design procedure, so that the only free design parameters are $\bar{y}^j$. In this paper, we use these fuzzy logic systems constructed FBFs with adaptive parameter vectors θ, θ_f and θ_b as alternatives of unknown functions $f(\underline{x}, t)$ and

$b(\underline{x}, t)$, respectively. In computer simulations, the adaptive parameter vector θ_f consists of random values (in first and second method) and θ_b consists of appropriate positive values (related with the boundary of b : the second method) at beginning. The reason that the values of θ_b have to be chosen the positive values is drawn at the Remark 1 in section V. Therefore, we need an assumption that the boundary of $f(\underline{x}, t)$ and $b(\underline{x}, t)$ are known. These boundaries are used in defining the universe of discourse of W for each fuzzy logic system.

III. Sliding Mode Control

Consider the nth-order nonlinear systems of the form

$$x^{(n)} = f(x, \dot{x}, \cdots, x^{(n-1)}) + b(x, \dot{x}, \cdots, x^{(n-1)})u + d(t) \quad y = x \tag{3.1}$$

where f and b are *unknown* continuous functions, $d(t)$ is the unknown external disturbance, $u \in R$ and $y \in R$ are the input and output of the system, and $\underline{x} = (x_1, x_2, \cdots, x_n)^T = (x, \dot{x}, \cdots, x^{(n-1)})^T \in R^n$ is the state vector of the system which is assumed to be available for measurement. In order for (3.1) to be controllable, we require that $b>0$ for $\underline{x}$ in certain controllability region $U_c \subset R^n$. And we have to make an assumption that $d(t)$ have upper bound D, that is, $|d(t)| \le D$. The control problem is to force the state $\underline{x}$ to track the desired state $\underline{x}_d$. With the tracking error

$$\underline{e} = \underline{x}(t) - \underline{x}_d(t) = (e, \dot{e}, \cdots, e^{(n-1)})^T \tag{3.2}$$

In general, *a sliding surface* is defined by

$$s(\underline{e}) = \underline{c}\,\underline{e} = 0, \tag{3.3}$$

where $\underline{c} = (c_1, c_2, \cdots, c_{n-1}, 1)$ in which the c_i's are all real and all roots of the polynomial $p^{n-1} + c_{n-1}p^{n-2} + \cdots + c_1 = 0$ are in the open left half-plane(p:Laplace operator).

Starting from the initial conditions $\underline{e}(0) = \underline{0}$, the tracking problem $\underline{x} = \underline{x}_d$ can be considered as the state error vector $\underline{e}$ remaining on the sliding surface $s(\underline{e}, t) = 0$ for all $t \ge 0$. A sufficient condition for this behavior is to choose the control input so that

$$\frac{1}{2} \cdot \frac{d}{dt}(s^2(\underline{e})) \le -\eta \cdot |s|, \qquad \eta \ge 0. \tag{3.4}$$

Considering $s^2(\underline{e})$ a Lyapunov function, it follows from (3.4) that the system controlled is stable. Looking as the phase plane, we obtain : the system is controlled in such a way that the state always moves towards the sliding surface and hits it. The sign of the control value must change at the intersection of state trajectory $\underline{e}(t)$ and the sliding surface. In this way, the trajectory is forced to move always towards the sliding surface. A sliding mode along the sliding surface is thus obtained.

The sliding condition of (3.4) can be rewritten as

$$s \cdot \dot{s} \le -\eta \cdot |s| \quad \text{or} \quad \dot{s} \cdot sgn(s) \le -\eta. \tag{3.5}$$

where η is a positive constant. By taking the time derivative of both sides of (3.4), we obtain

$$\begin{aligned} \dot{s} &= c_1 \dot{e} + c_2 \ddot{e} + \cdots + c_{n-1} e^{(n-1)} + x^{(n)} - x_d^{(n)} \\ &= \sum_{i=1}^{n-1} c_i e^{(i)} + x^{(n)} - x_d^{(n)} \\ &= \sum_{i=1}^{n-1} c_i e^{(i)} + f(\underline{x}, t) + b(\underline{x}, t)u + d(t) - x_d^{(n)}. \end{aligned} \tag{3.6}$$

Therefore the control problem is to obtain the proper SMC input u^* which guarantees the sliding condition,

$$\begin{aligned} s \cdot \dot{s} &= s\Big(\sum_{i=1}^{n-1} c_i e^{(i)} + f(\underline{x}, t) + b(\underline{x}, t)u^* + d(t) - x_d^{(n)}\Big) \\ &\le -\eta|s|, \end{aligned} \tag{3.7}$$

or

$$\begin{aligned} &sgn(s)\ \Big(\sum_{i=1}^{n-1} c_i e^{(i)} + f(\underline{x}, t) + b(\underline{x}, t)u^* + d(t) - x_d^{(n)}\Big) \\ &\le -\eta. \end{aligned} \tag{3.8}$$

If $f(\underline{x}, t)$ is known, we can design the proper sliding mode control law easily as follows.

i) *If* $s>0$,
$$u^* \le b(\underline{x}, t)^{-1}\Big(-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) - d(t) + x_d^{(n)} - \eta_\Delta\Big)$$
ii) *If* $s<0$,
$$u^* \ge b(\underline{x}, t)^{-1}\Big(-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) - d(t) + x_d^{(n)} + \eta_\Delta\Big)$$
iii) *If* $s=0$,
$$u^* = b(\underline{x}, t)^{-1}\Big(-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) - d(t) + x_d^{(n)}\Big)$$

where $\eta_\Delta \ge \eta > 0$, and iii) is the case of the instant that the state trajectory hit the sliding surface. Therefore proper SMC input u^* is

$$\begin{aligned} u^* &= b(\underline{x}, t)^{-1} \cdot \\ &\Big(-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) - d(t) + x_d^{(n)} - h \cdot sgn(s) \cdot \eta_\Delta\Big), \end{aligned} \tag{3.9}$$

where

$$h = \begin{cases} 1 & , \text{if } s \ne 0 \\ 0 & , \text{if } s = 0 \end{cases} \tag{3.10}$$

This SMC input guarantees the sliding condition of (3.4). However, $f(\underline{x}, t)$, $b(\underline{x}, t)$ and $d(t)$ are unknown. To solve this problem, we propose the adaptive scheme using the fuzzy logic system in the next section.

IV. Adaptive Fuzzy Sliding Mode Control : the First Method

If $f(\underline{x}, t)$ and $b(\underline{x}, t)$ are known, we can easily construct the SMC input u^* as in the previous section. However, f and b are unknown. To find the solution of this problem, we replace the $f(\underline{x}, t)$ by the fuzzy logic system $\hat{f}(\underline{x}|\theta)$ which is in the form of (2.9) or (2.10) and consider the term $\underline{b}^{-1} sgn(s)|F_1|$ in order to reduce the disturbance due to the uncertainty of the unknown control gain.

Where we assume that $0 < \underline{b} \le b(\underline{x}, t)$ and $b(\underline{x}, t) = \underline{b} + \Delta b(\underline{x}, t)$, and $\underline{b}$ is a known positive constant and $\Delta b(\underline{x}, t)$ is an unknown positive function. The resulting control input is as follows.

$$u_1 = \underline{b}^{-1}(-\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(\underline{x}|\theta) + x_d^{(n)} - h \cdot sgn(s) \cdot (D+\eta_\Delta)) - \underline{b}^{-1} sgn(s)|F_1|, \tag{4.1}$$

where $F_1 = -\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(\underline{x}|\theta) + x_d^{(n)} - h \cdot sgn(s)(D+\eta_\Delta)$.

Then

$$\dot{s} = f(\underline{x}, t) - \hat{f}(\underline{x}|\theta) - h\, sgn(s)\,(D+\eta_\Delta) + d(t) + \Delta b(\underline{x}, t)\, \underline{b}^{-1} F_1 - b(\underline{x}, t)\, \underline{b}^{-1} sgn(s)|F_1|. \tag{4.2}$$

1. Adaptive Law Synthesis

Letting the optimal parameter vector of fuzzy logic system be θ^*, we can define the minimum approximation error,

$$\omega = f(\underline{x}, t) - \hat{f}(\underline{x}|\theta^*). \tag{4.3}$$

So, (4.2) can be rewritten as

$$\dot{s} = \hat{f}(\underline{x}|\theta^*) - \hat{f}(\underline{x}|\theta) + \omega - h\, sgn(s)\,(D+\eta_\Delta) + d(t) + \Delta b(\underline{x}, t)\, \underline{b}^{-1} F_1 - b(\underline{x}, t)\, \underline{b}^{-1} sgn(s)|F_1| \tag{4.4}$$

If we choose $\hat{f}$ to be the fuzzy logic system in the form of (2.10), then (4.4) can be rewritten as

$$\dot{s} = \phi^T \xi(\underline{x}) + \omega - h\, sgn(s)\,(D+\eta_\Delta) + d(t) + \Delta b(\underline{x}, t)\, \underline{b}^{-1} F_1 - b(\underline{x}, t)\, \underline{b}^{-1} sgn(s)|F_1| \tag{4.5}$$

where $\phi = \theta^* - \theta$, and $\xi(\underline{x})$ is the fuzzy basis function (2.11).

Now consider the Lyapunov candidate

$$V_1 = \frac{1}{2}(s^2 + \frac{1}{\gamma_1}\phi^T \phi) \tag{4.6}$$

where γ_1 is a positive constant. The time derivative of V_1 is

$$\begin{aligned}
\dot{V}_1 &= s\dot{s} + \frac{1}{\gamma_1}\phi^T \dot{\phi} \\
&= s\phi^T \xi(\underline{x}) + s\omega - s\, h\, sgn(s) \cdot (D+\eta_\Delta) \\
&\quad + s \cdot d(t) + \frac{1}{\gamma_1}\phi^T \dot{\phi} \\
&\quad + s\Delta b(\underline{x}, t)\, \underline{b}^{-1} F_1 - |s| b(\underline{x}, t)\, \underline{b}^{-1}|F_1| \qquad (4.7)\\
&\le \frac{1}{\gamma_1}\phi^T(\gamma_1 s \xi(\underline{x}) - \dot{\theta}) + s\omega + s\Delta b(\underline{x}, t)\, \underline{b}^{-1} F_1 \\
&\quad - |s| b(\underline{x}, t)\, \underline{b}^{-1}|F_1| - |s|\, h\, \eta_\Delta \\
&< \frac{1}{\gamma_1}\phi^T(\gamma_1 \cdot s \cdot \xi(\underline{x}) - \dot{\theta}) + s\omega - |s|\, h\, \eta_\Delta,
\end{aligned}$$

where $\dot{\phi} = -\dot{\theta}$. Because the term $s \cdot \omega$ is of the order of the minimum approximation error and from the Universal Approximation Theorem, Wang expected that the ω should be very small, *i,e,* $\omega \le \varepsilon$, if not equal to zero in the adaptive fuzzy system[3]. Therefore, we can choose the adaptive law

$$\dot{\theta} = \gamma_1 \cdot s \cdot \xi(\underline{x}). \tag{4.8}$$

However, this approach is not complete, thus we consider the robust control techniques under the assumption that the upper bound of ω is known in the next subsection.

2. Robust Adaptive Law Synthesis

Let the control input

$$u_{1r} = \underline{b}^{-1}(-\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(\underline{x}|\theta) + x_d^{(n)} - h\, sgn(s)(D+\eta_\Delta + \hat{\rho})) - \underline{b}^{-1} sgn(s)|F_{1r}|, \tag{4.9}$$

where

$$F_{1r} = -\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(\underline{x}|\theta) + x_d^{(n)} - h \cdot sgn(s) \cdot (D+\eta_\Delta + \hat{\rho})$$

and $\hat{\rho}$ (the estimation of ω) is $\hat{\rho} = \rho^* - \tilde{\rho}$ and $\rho^* = |\omega|_{\max}$ *i.e.*, ρ^* is the upper bound of minimum approximation error of the fuzzy approximator. Therefore we can obtain $\dot{s}$:

$$\dot{s} = f(\underline{x}, t) - \hat{f}(\underline{x}|\theta) - h \cdot sgn(s) \cdot (D+\eta_\Delta + \hat{\rho}) + d(t) + \Delta b(\underline{x}, t) \cdot \underline{b}^{-1} F_{1r} - b(\underline{x}, t) \cdot \underline{b}^{-1} sgn(s)|F_{1r}|. \tag{4.10}$$

Now consider the Lyapunov candidate

$$V_{1r} = \frac{1}{2}(s^2 + \frac{1}{\gamma_1}\phi^T \phi + \frac{1}{\gamma_2}\tilde{\rho}^2) \tag{4.11}$$

Applying (4.10) to (4.11) and after straightforward manipulation, we obtain the time derivative of V_{1r}

$$\begin{aligned}
\dot{V}_{1r} &= \frac{1}{\gamma_1}\phi^T(\gamma_1 s \xi(\underline{x}) - \dot{\theta}) - s\, h\, sgn(s)\,(D+\eta_\Delta) \\
&\quad + s \cdot d(t) + s\omega - h \cdot |s| \cdot \hat{\rho} + \frac{1}{\gamma_2}\tilde{\rho}\dot{\tilde{\rho}} + \\
&\quad s\,(\Delta b(\underline{x}, t)\, \underline{b}^{-1} F_{1r} - b(\underline{x}, t)\, \underline{b}^{-1} sgn(s)|F_{1r}|) \\
&\le \frac{1}{\gamma_1}\phi^T(\gamma_1 s \xi(\underline{x}) - \dot{\theta}) + s\omega - h|s|\rho^* + \\
&\quad h|s|(\rho^* - \hat{\rho}) + \frac{1}{\gamma_2}\tilde{\rho}\dot{\tilde{\rho}} + s\Delta b(\underline{x}, t)\, \underline{b}^{-1} F_{1r} \\
&\quad - |s| b(\underline{x}, t)\, \underline{b}^{-1}|F_{1r}|) - |s| \cdot h \cdot \eta_\Delta \\
&< \frac{1}{\gamma_1}\phi^T(\gamma_1 s \xi(\underline{x}) - \dot{\theta}) + s\omega \\
&\quad - h \cdot |s| \cdot \rho^* + h \cdot |s| \cdot (\rho^* - \hat{\rho}) - |s| \cdot h \cdot \eta_\Delta \qquad (4.12)\\
&\le \frac{1}{\gamma_1}\phi^T(\gamma_1 s \xi(\underline{x}) - \dot{\theta}) + \frac{1}{\gamma_2}\tilde{\rho}(\dot{\tilde{\rho}} + \gamma_2 \cdot h \cdot |s|) \\
&\quad - |s| \cdot h \cdot \eta_\Delta.
\end{aligned}$$

Therefore we can choose the adaptive laws

$$\dot{\theta} = \gamma_1 \cdot s \cdot \xi(\underline{x}), \quad \dot{\tilde{\rho}} = \gamma_2 \cdot h \cdot |s|. \tag{4.13}$$

V. Adaptive Fuzzy Sliding Mode Control : the Second Method

In the previous section, we had to know the accurate lower bound of the control gain function $b(\underline{x}, t)$. But it is difficult to know this value precisely. So we propose another method that $f(\underline{x}, t)$ and $b(\underline{x}, t)$ are replaced by the fuzzy logic systems $\hat{f}(\underline{x}|\theta_f)$ and $\hat{b}(\underline{x}|\theta_b)$ in this section. We replace f and b with two fuzzy logic systems, $\hat{f}(\underline{x}|\theta_f)$ and $\hat{b}(\underline{x}|\theta_b)$, and append another input u_b in order to reduce the disturbance due to the uncertainty of the unknown control gain. The resulting control input is as follows.

$$u_2 = u_f + u_b, \tag{5.1}$$

where

$$u_f = \hat{b}^{-1}(x|\theta_b)(-\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(x|\theta_f) + x_d^{(n)} - h\cdot sgn(s)\cdot(D+\eta_\Delta)),$$

$u_b = -\Gamma_1 \cdot sgn(s)|u_f|$ and $\Gamma_1 \geq \frac{|\omega_b|_{max}}{b(x,t)|_{min}}$ and $b(x,t)|_{min}$ is the known minimum value of control gain $b(x,t)$.

Then

$$\dot{s} = f(x,t) - \hat{f}(x|\theta_f) + (b(x,t) - \hat{b}(x|\theta_b))u_f + b(x,t)u_b - h\cdot sgn(s)\cdot(D+\eta_\Delta) + d(t). \quad (5.2)$$

1. Adaptive Law Synthesis

Letting the optimal parameter vectors of fuzzy logic systems be θ_f^*, θ_b^*, we can define the minimum approximation errors,

$$\omega_f = f(x,t) - \hat{f}(x|\theta_f^*),\ \omega_b = b(x,t) - \hat{b}(x|\theta_b^*). \quad (5.3)$$

So, (5.2) can be rewritten as

$$\dot{s} = \hat{f}(x|\theta_f^*) - \hat{f}(x|\theta_f) + \omega_f - h\,sgn(s)(D+\eta_\Delta) + d(t) + (\hat{b}(x|\theta_b^*) - \hat{b}(x|\theta_b) + \omega_b)u_f + b(x,t)u_b. \quad (5.4)$$

If we choose $\hat{f}$ and $\hat{b}$ to be the fuzzy logic systems in the form of (2.10), then (5.4) can be rewritten as

$$\dot{s} = \phi_f^T \xi_f(x) + \omega_f + (\phi_b^T \xi_b(x) + \omega_b)u_f + b(x,t)u_b - h\cdot sgn(s)\cdot(D+\eta_\Delta) + d(t), \quad (5.5)$$

where $\phi_f = \theta_f^* - \theta_f$, $\phi_b = \theta_b^* - \theta_b$, and $\xi_f(x)$ and $\xi_b(x)$ are the fuzzy basis functions (2.11).

Now consider the Lyapunov candidate

$$V_2 = \frac{1}{2}(s^2 + \frac{1}{\gamma_1}\phi_f^T\phi_f + \frac{1}{\gamma_2}\phi_b^T\phi_b), \quad (5.6)$$

where γ_1, γ_2 are positive constants. The time derivative of V_2 is

$$\begin{aligned}\dot{V}_2 &= s\dot{s} + \frac{1}{\gamma_1}\phi_f^T\dot{\phi}_f + \frac{1}{\gamma_2}\phi_b^T\dot{\phi}_b \\ &= \frac{1}{\gamma_1}\phi_f^T(s\gamma_1\xi_f(x) + \dot{\phi}_f) + s\omega_f \\ &\quad + \frac{1}{\gamma_2}\phi_b^T(s\gamma_2\xi_b(x)u_f + \dot{\phi}_b) \\ &\quad - s\,h\,sgn(s)(D+\eta_\Delta) + sd(t) + s\,\omega_b\,u_f \\ &\quad - s\,\Gamma_1 b(x,t)sgn(s)|u_f| \\ &\leq \frac{1}{\gamma_1}\phi_f^T(\gamma_1 s\xi_f(x) + \dot{\phi}_f) \\ &\quad + \frac{1}{\gamma_2}\phi_b^T(\gamma_2 s\xi_b(x)u_f + \dot{\phi}_b) \\ &\quad + s\omega_f - |s|\cdot h\cdot\eta_\Delta,\end{aligned} \quad (5.7)$$

where $\dot{\phi}_f = -\dot{\theta}_f$ and $\dot{\phi}_b = -\dot{\theta}_b$. Because the term $s\cdot\omega_f$ is of the order of the minimum approximation error, we can choose the adaptive laws as follows.

$$\dot{\theta}_f = \gamma_1\cdot s\cdot\xi_f(x),\ \dot{\theta}_b = \gamma_2\cdot s\cdot\xi_b(x)\cdot u_f. \quad (5.8)$$

However, this approach is not complete, thus we also consider the robust control techniques in the next subsection.

2. Robust Control Law

In order to reduce the disturbance due to ω_f, in this section, we consider the term $\Gamma_2\cdot sgn(s)$ in the control law. The resulting control input

$$u_{2r} = u_f + u_{rob} \quad (5.9)$$

where

$$u_{rob} = -\Gamma_1\,sgn(s)\,|u_f| - \Gamma_2\,sgn(s),\ \Gamma_2 \geq \frac{|\omega_f|_{max}}{b(x,t)|_{min}}$$

Therefore we can obtain $\dot{s}$:

$$\dot{s} = \phi_f^T\xi_f(x) + \omega_f + (\phi_b^T\xi_b(x) + \omega_b)u_f + b(x,t)u_{rob} - h\cdot sgn(s)\cdot(D+\eta_\Delta) + d(t). \quad (5.10)$$

Now considering the Lyapunov candidate V_2, and applying (5.10) to (5.6) and after straightforward manipulation, we obtain the time derivative of V_2

$$\dot{V}_2 \leq \frac{1}{\gamma_1}\phi_f^T(\gamma_1 s\xi_f(x) + \dot{\phi}_f) + \frac{1}{\gamma_2}\phi_b^T(\gamma_2 s\xi_b(x)u_f + \dot{\phi}_b) - |s|\cdot h\cdot\eta_\Delta. \quad (5.11)$$

Therefore we can choose the adaptive laws as (5.8).

Remark 1 : In the real implementation of the proposed controller, the condition that $\hat{b}(x|\theta_b)$ is not zero should be guaranteed, because the proposed controllers (5.1), (5.9) have the inverse term of $\hat{b}$. Therefore, in this paper, we choose the adaptation step size of θ_b, γ_2 is properly small and initial parameter values of θ_b is a positive constant in simulations (the output of the fuzzy logic system is positive if all elements of the parameter vector θ_b are positive : non-vanishes property ; appendix of [3]).

Remark 2 : Since the sliding mode control laws involve the sgn-functions, these control laws are discontinuous across the surface s, thus they lead to the control chattering. Chattering is, in general, highly undesirable in practice, since it involves extremely high control activity, and may excite high-frequency dynamics neglected in the course of modelling. To overcome this problem, in this paper, we use the sliding surface with the boundary layer proposed by Slotine and Sastry[10,11,12]. Therefore smoothing out the control discontinuity can be achieved by replacing the function $sgn(.)$ as $sat(.)$ in control laws (4.1), (4.9), (5.1), (5.9). Detailed expressions are omitted(refer [11,12]) and we investigate the continuous approximation of (4.1), (4.9) only through the simulations.

VI. Design Example

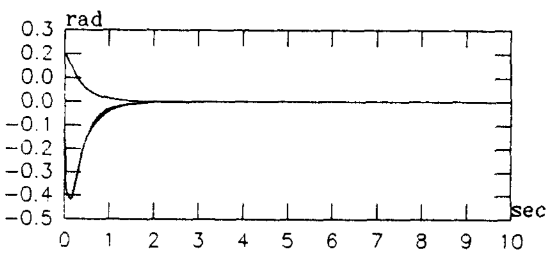
To illustrate the above design approaches, an inverted pendulum system is considered. The dynamics of the system can be derived using the Euler-Lagrange method,

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{g\sin x_1 - \cos x_1(\frac{ml}{m_c+m}x_2^2\sin x_1 - \frac{1}{m_c+m}u(t))}{\frac{4}{3}l - \frac{ml}{m_c+m}\cos^2 x_1},\end{aligned} \quad (6.1)$$

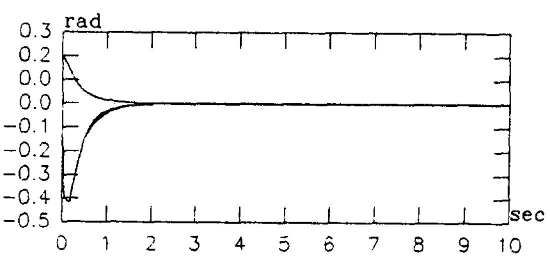
where x_1 and x_2 are the angular position and velocity of the pole, $u(t)$ is the control input force applied to v cart, g is the gravitational constant : $9.8(m/\sec^2)$, m_c is the mass of the cart, and m and l denote the mass and length of the pole, respectively. The control objective is to maintain $x_1(t)=x_{1d}(t)$ and $x_2(t)=x_{2d}(t)$, that is, tracking error $\underline{e}$ converges to $\underline{0}$. In this design example we choose the switching surface given by

$$s(\underline{e})=\underline{c}\,\underline{e}=0, \qquad (6.2)$$

where $\underline{e}=[e,\ \dot{e}\ ]^T$, $\underline{c}=[c_1, 1\ ]$. In (6.2), $m=0.1Kg$, $m_c=1Kg$, $l=0.5m$ and we choose the $\eta_d=0.03$ and $c_1=3.0$ in simulations of the first and second methods. In simulation of the first method, the inputs are designed by (4.1) and (4.9) and the boundary of the approximator is determined as $|\hat{f}(\underline{x},\theta)|\le 15$. This value is used in determining the initial random values of θ in u_1, u_{1r}. In the second method, the inputs are designed by (5.1) and (5.9). We have $|\hat{f}(\underline{x},\theta_f)|=15$ and $|\hat{b}(\underline{x},\theta_b)|=1.9$, so we choose 1.9s as the initial values of θ_b at beginning. In simulations(of all cases), we choose 5 fuzzy levels, *i.e.*, NB, NS, ZO, PS, PB on each universe of discourse of x_1 and x_2, and we use the fuzzy logic system with the center-average defuzzifier, product inference, the singleton fuzzifier and gaussian membership functions. Because the inverted pendulum system is of 2nd order with respect to angular displacement of the pole, the fuzzy logic systems, all of $\hat{f}$s and $\hat{b}$ are constructed with 25 rules (5^2: 2 input variables, 5 fuzzy levels), respectively.



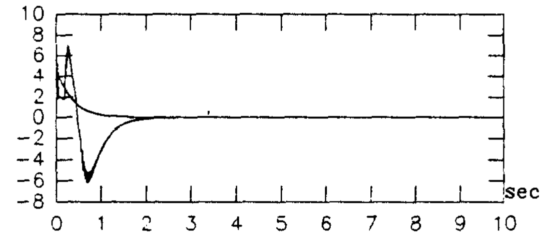
(a) input : (4.1)



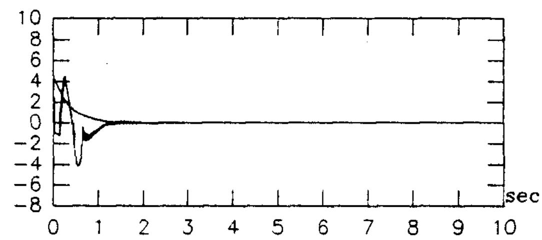
(b) input : (4.9)

Fig. 2. Angular displacement and velocity of the pole($x_1(0)=0.2$, $x_2(0)=0.0$, $\underline{x}_d=\underline{0}$).

Fig. 2 ~ Fig. 4 show the control results of the first method when the initial states and desired value are $x_1(0)=0.2$, $x_2(0)=0.0$, $\underline{x}_d=\underline{0}$. Fig. 5 ~ Fig. 8 show the control results of the second method when the initial states and the desired value are $x_1(0)=0.2$, $x_2(0)=0.0$, $\underline{x}_d=\underline{0}$. Fig. 9 and 10 show the control results of the first method and Fig. 11, 12 show the control results of the second method when the initial states and desired trajectory are $x_1(0)=0.2, x_2(0)=0.0$, $x_{1d}(t)=\frac{\pi}{10}(\sin(t)+0.3\sin(3t))$.

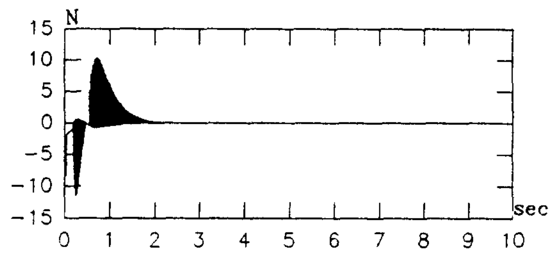


(a) input : (4.1)

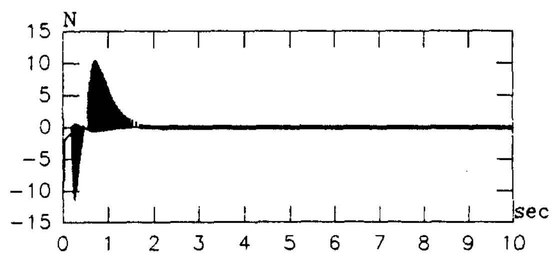


(b) input : (4.9)

Fig. 3. $f(\underline{x},t)$ and $\hat{f}(\underline{x}|\theta)$ ($x_1(0)=0.2$, $x_2(0)=0.0$ $\underline{x}_d=\underline{0}$).

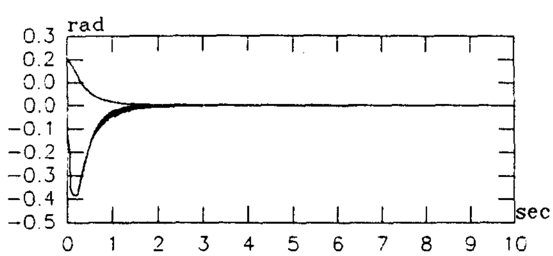


(a) input : (4.1)

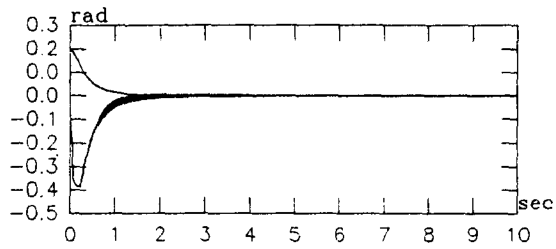


(b) input : (4.9)

Fig. 4. u_1 and u_{1r}($x_1(0)=0.2$, $x_2(0)=0.0$, $\underline{x}_d=\underline{0}$).

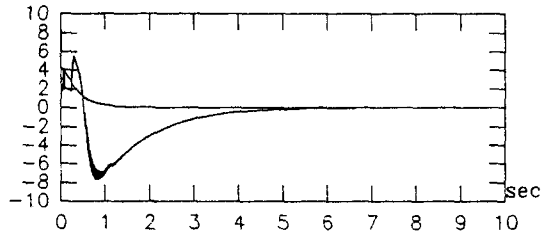


(a) input : (5.1)

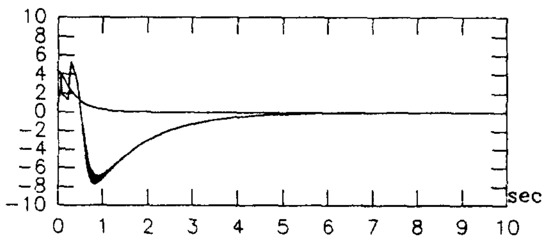


(b) input : (5.9)

Fig. 5. Angular displacement and velocity of the pole($x_1(0)=0.2$, $x_2(0)=0.0$, $\underline{x}_d=\underline{0}$).

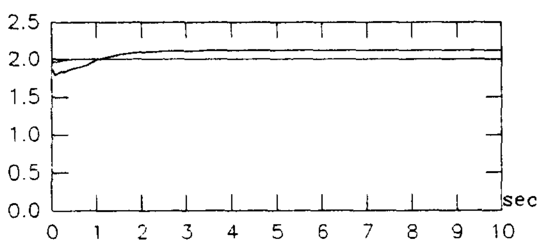


(a) input : (5.1)

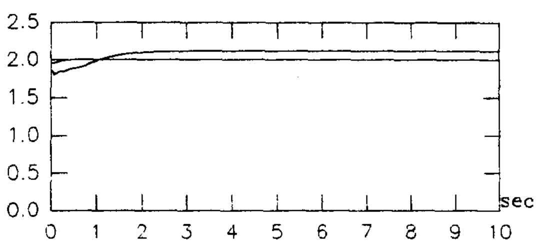


(b) input : (5.9)

Fig. 6. $f(\underline{x},t)$ and $\hat{f}(\underline{x}|\theta_f)$
($x_1(0)=0.2$, $x_2(0)=0.0$, $\underline{x}_d=\underline{0}$).



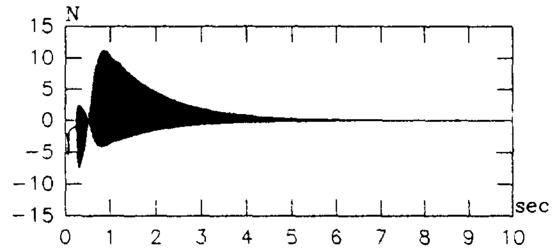
(a) input : (5.1)



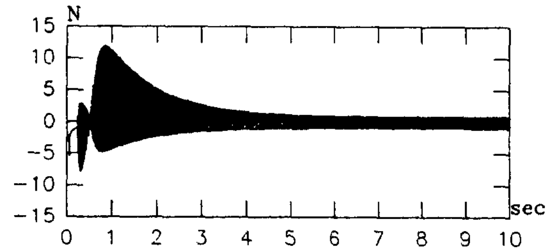
(b) input : (5.9)

Fig. 7. $b(\underline{x},t)$ and $\hat{b}(\underline{x}|\theta_b)$
($x_1(0)=0.2$, $x_2(0)=0.0$, $\underline{x}_d=\underline{0}$).

Fig. 13 ~ Fig. 16 show the control results of the continuous control law for the first method described in appendix. Fig. 2, 5, 9, 11, 13 and 15 show the angular displacement or velocity of the pole and Fig. 3, 6 and 7 show the comparisons between $\hat{f}(\underline{x}|\theta)$ and $\hat{f}(\underline{x}|\theta_f)$ or between $b(\underline{x},t)$ and $\hat{b}(\underline{x}|\theta_b)$, Fig. 4, 8, 10, 12, 14 and 16 show the comparisons between u_1 and u_{1r}, u_2 and u_{2r}, u_{1-cont} and $u_{1r-cont}$, respectevely.

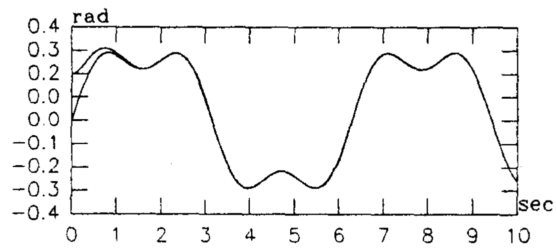


(a) input : (5.1)

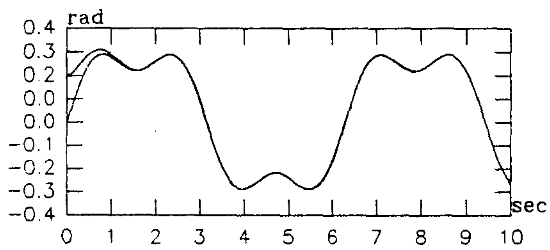


(b) input : (5.9)

Fig. 8. u_2 and u_{2r}
($x_1(0)=0.2$, $x_2(0)=0.0$, $\underline{x}_d=\underline{0}$).

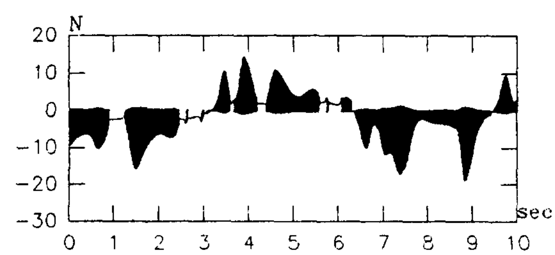


(a) input : (4.1)

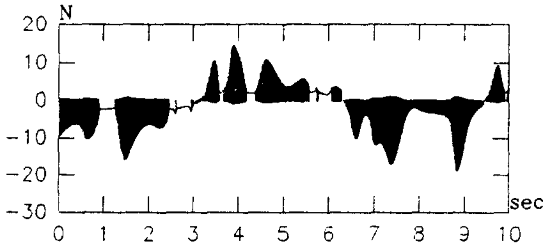


(b) input : (4.9)

Fig. 9. Angular displacement of the pole
($x_{1d}(t)=\frac{\pi}{10}(\sin(t)+0.3\sin(3t))$).

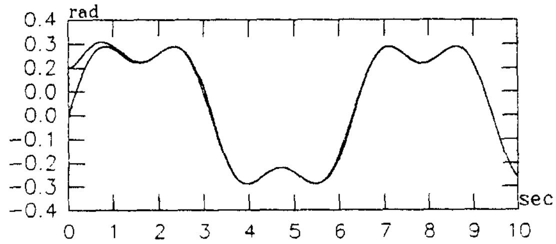


(a) input : (4.1)

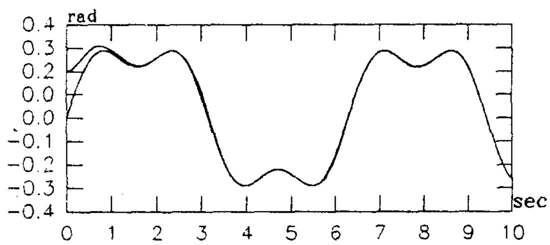


(b) input : (4.9)

Fig. 10. Comparisons between u_1 and u_{1r}
($x_{1d}(t) = \frac{\pi}{10}$ (sin (t) + 0.3 sin $(3t)$)).

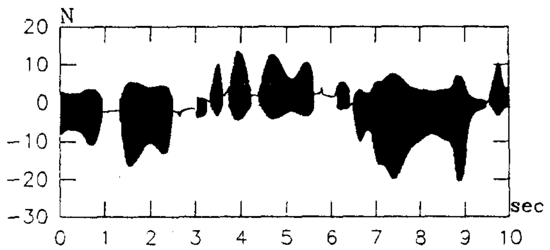


(a) input : (5.1)

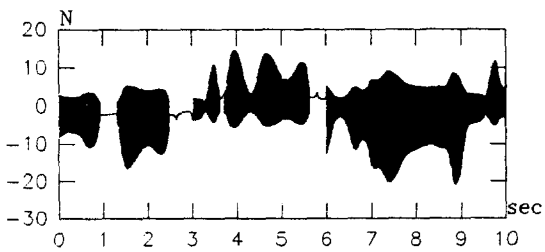


(b) input : (5.9)

Fig. 11. Angular displacement of the pole
($x_{1d}(t) = \frac{\pi}{10}(\sin(t) + 0.3\sin(3t))$).



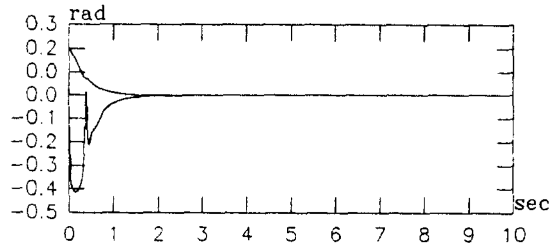
(a) input : (5.1)



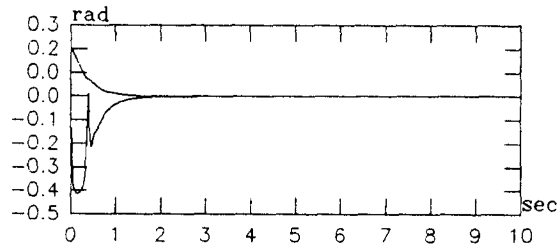
(b) input : (5.9)

Fig. 12. Comparisons between u_2 and u_{2r}
($x_{1d}(t) = \frac{\pi}{10}(\sin(t) + 0.3\sin(3t))$).

In these figures, angular displacements of the pole converge to desired value within about 2 seconds, and fuzzy logic systems, $\hat{f}(x|\theta)$, $\hat{f}(x|\theta_f)$, converge to real

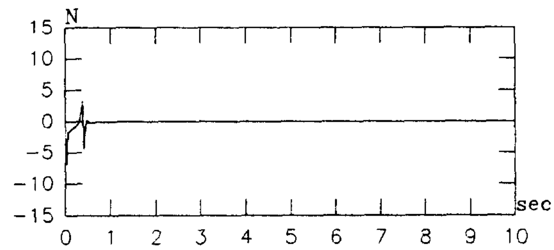


(a) continuous input of (4.1) : u_{1-cont}

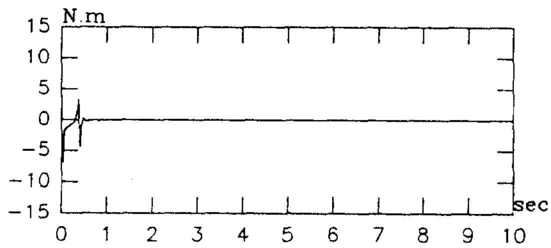


(b) continuous input of (4.9) : $u_{1r-cont}$

Fig. 13. Angular displacement and velocity of the pole($x_1(0) = 0.2$, $x_2(0) = 0.0$, $\underline{x}_d = \underline{0}$).



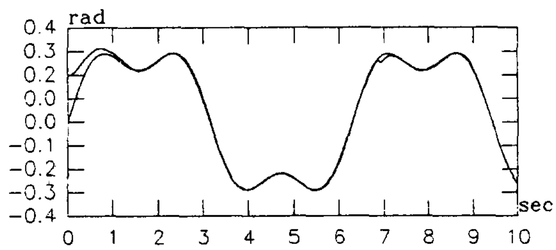
(a) continuous input of (4.1) : u_{1-cont}



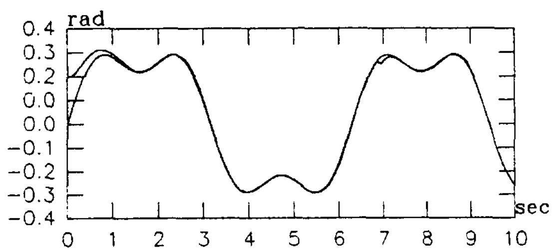
(b) continuous input of (4.9) : $u_{1r-cont}$

Fig. 14. u_{1-cont} and $u_{1r-cont}$
($x_1(0) = 0.2$, $x_2(0) = 0.0$, $\underline{x}_d = \underline{0}$).

system functions respectively. Especially, the cases of using inputs u_{1r} and u_{2r} are more effective than the cases of using u_1 and u_2 in Fig.s 3 and 6. From Fig. 8, we can see that $\hat{b}(x|\theta_b)$ does not converge to real b, but it converges to a positive constant. The differences between the results of u_1 and u_{1r}, u_2 and u_{2r} are not conspicuous in Fig. 2, 4, 5, 7, 9 and 11. So we can see that the minimum approximation errors are very small, if not equal to zero. In Fig. 3 and Fig. 6, especially, the case of using robust technique is more effective than simple adaptive technique. So we can see that the $\hat{\rho}$ terms in u_{1r} play the important role of reducing the approximation error in transient time. The chattering in the simulation results dues to the characteristics of sliding mode control. From Fig. 4, 14, 10 and 16, in addition, we can also see that continuous approximations are effectively achieved.

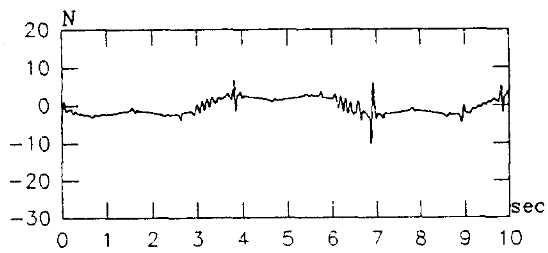


(a) continuous input of (4.1) : u_{1-cont}

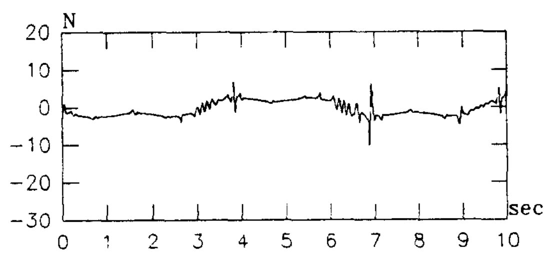


(b) continuous input of (4.9) : $u_{1r-cont}$

Fig. 15. Angular displacement and velocity of the pole($x_{1d}(t)=\frac{\pi}{10}(\sin(t)+0.3\sin(3t))$).



(a) continuous input of (4.1) : u_{1-cont}



(b) continuous input of (4.9) : $u_{1r-cont}$

Fig. 16. u_{1-cont} and $u_{1r-cont}$ ($x_{1d}(t)=\frac{\pi}{10}(\sin(t)+0.3\sin(3t))$).

Ⅶ. Conclusion

The proposed AFSMC schemes were motivated by the fuzzy approximator theory and results of [3,4,5], and we used the fuzzy approximators as the estimators to the unknown function f and b. We introduced the conventional sliding mode control theory and proposed two methods that fuzzy approximator could be applied to the siding mode control. We also proposed the robust controller using the robust adaptive law, that is, we used the adaptive law to reduce the error between the nonlinear function and the fuzzy approximator. The stabilities of proposed control schemes were proved and we verified that the minimum approximation error is very small and the fuzzy logic system can approximate the nonlinear function well, and the sliding mode control scheme based on the robust adaptive law is more effective than the simple adaptive law in simulations.

References

[1] C. C. Lee, "Fuzzy logic in control system : Fuzzy logic controller-Part I, II," *IEEE Trans. on Systems, Man, and Cybernetics,* vol. SMC-20, no. 2, pp. 404-435, 1990.

[2] H. J. Zimmermann, "*Fuzzy set theory and its applications*," Kluwer Academic Pulishers, 1991.

[3] L. X. Wang, "*Adaptive fuzzy systems and control*," Prentice-Hall International, Inc., 1994.

[4] L. X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least squares learning," *IEEE Trans. Neural Networks*, vol. 3, no. 5, pp. 807-814, Sept., 1992.

[5] H. M. Kim and J. M. Mendel, "Fuzzy basis functions: comparisons with other basis functions," *IEEE Trans. Fuzzy Systems,* vol. 3, no. 2, pp. 158-168, May, 1995.

[6] M. B. Ghalia and A. T. Alouani, "Sliding mode control synthesis using fuzzy logic," *Proceedings of the American Control Conference*, Seattle, Washington, pp. 1528-1532, June, 1995.

[7] G. C. Hwang and S. C. Lin, "A stability approach to fuzzy control design for nonlinear systems," *Fuzzy Sets and Systems*, vol. 48, pp. 279-287, 1992.

[8] R. Palm, "Sliding mode fuzzy control," *IEEE Int. Confer. on Fuzzy Systems*, San Diego, California, pp. 519-526, 1992.

[9] S. C. Lin and Y. Y. Chen, "Design of adaptive fuzzy sliding mode for nonlinear system control," *Third IEEE Int. Confer. on Fuzzy Systems*, Orlando, Florida, pp. 35-39, June, 1994.

[10] V. I. Utkin, "Variable structure systems with sliding modes," *IEEE Trans. Automat. Contr.*, vol. AC-22, no. 2, pp. 212-222, April, 1977.

[11] J. E. Slotine, "Sliding controller design for non-linear systems," *Int. J. Control*, vol. 40, no. 2, pp. 421-434, 1984.

[12] J. E. Slotine and S. S. Sastry, "Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators," *Int. J. Control*, vol. 38, no. 2, pp. 465-492, 1983.

[13] W. C. Ham and J. J. Lee, "Adaptive nonlinear control of one-link flexible arm," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Yokohama, Japan, vol. 1, pp. 3-7, 1993.

[14] W. C. Ham, "Adaptive control based on explicit

model of robot manipulators,"*IEEE Trans. Automat. Contr.*, vol. 38, pp. 654-658, Apr., 1993.

Appendix

The continuous control laws to eliminate chattering in the first method are as follows. These are the cases that $n=2$ and ε_w is the boundary layer width and tracking precision.

$$\begin{aligned} u_{1-cont} = \; & \underline{b}^{-1}(-c_1\dot{e}-\hat{f}(x|\theta)+\ddot{x}_d \\ & -h\,sat(s/(c_1\varepsilon_w))\,(D+\eta_\Delta)) \\ & -\underline{b}^{-1}sat(s/(c_1\varepsilon_w))|F_{1c}|, \end{aligned}$$

$$\begin{aligned} u_{1r-cont} = \; & \underline{b}^{-1}(-c_1\dot{e}-\hat{f}(x|\theta)+\ddot{x}_d \\ & -h\,sat(s/(c_1\varepsilon_w))\,(D+\eta_\Delta+\hat{\rho})) \\ & -\underline{b}^{-1}sat(s/(c_1\varepsilon_w))|F_{1rc}| \end{aligned}$$

where

$$\begin{aligned} F_{1c} = \; & -c_1\dot{e}-\hat{f}(x|\theta)+\ddot{x}_d \\ & -h\cdot sat(s/(c_1\varepsilon_w)\cdot(D+\eta_\Delta) \end{aligned}$$

$$\begin{aligned} F_{1r} = \; & -c_1\dot{e}-\hat{f}(x|\theta)+\ddot{x}_d \\ & -h\cdot sat(s/(c_1\varepsilon_w)\cdot(D+\eta_\Delta+\hat{\rho}). \end{aligned}$$

유 병 국

1992년 전북대학교 전자공학과 졸업(공학사). 1995년 전자공학과 석사과정 졸업(공학석사). 1995년 ~ 동대학원 박사과정 재학중. 1996년 ~ 현재 한려산업대학교 의공학과 전임강사. 주 관심분야는 퍼지제어, 가변구조 제어, 로봇제어등과 의료용 센서.

함 운 철

1979년 서울대학교 전자공학과 졸업(공학사). 1982년 서울대학교 전자공학과 석사과정 졸업(공학석사). 1988년 서울대학교 전자공학과 박사과정 졸업(공학박사). 1991년 미국 Perdue대 전기과 교환교수. 현재 전북대학교 전자공학과 정교수. 관심 분야는 적응제어, 가변구조제어, 비선형제어, 퍼지제어이론, 헬기제어등이다.

정 사 철

1988년 원광대 전자공학과 졸업(공학사). 1990년 전북대 전자공학과 석사 과정 졸업(공학석사). 1993년 ~ 현재 동 대학원 박사과정 재학중. 관심분야는 적응제어, 퍼지제어, 가변구조제어, 비선형제어.