'95 KACC (1995. 10. 23 ~ 25)

ADAPTIVE SLIDING MODE CONTROL USING FUZZY LOGIC SYSTEM

°Byungkook Yoo, Sacheul Jeoung, Woonchul Ham

Department of Electronic Engineer., Chonbuk National Univ., Chonju, Chonbuk, Korea 560-756
phone: +82-652-70-2405 ; fax: +82-652-70-2263 ; e-mail: woonchul@mips.chonbuk.ac.kr

Abstracts In this study, the fuzzy approximator and sliding mode control(SMC) scheme are considered. An adaptive sliding mode control is proposed based on the SMC theory. This proposed control scheme is that a adaptive law is utilized to approximate the unknown function f by fuzzy logic system in designing the sliding mode controller for the nonlinear system. In order to reduce the approximation errors, the differences of nonlinear function and fuzzy approximator, an robust adaptive law is also intoduced and the stability of proposed control scheme are proven with simple adaptive law and roburst adaptive law. This proposed control scheme is applied to a single link robot arm.

Keywords Fuzzy Logic System, Sliding Mode Control, Adaptive Law, Robust Adaptive Law

1. INTRODUCTION

As an alternative to conventional control techniques, fuzzy control is gaining increased interests, both in the academic world and in the industrial field. For those systems whose accurate mathematical models are not avaiable or difficult to formulate, fuzzy control can often provide a good solution by incorporating linguistic informations from human experts. 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 the researchers in Wang'works that fuzzy logic systems (cneter average defuzzifier, product-inference rule, singletone fuzzifier, and gaussian membership function) are capable of uniformly approximating any nonlinear function over compact input space[3]. That is, any nonlinear system can be modeled by the fuzzy logic systems. Especially, Wang utilized the general error dynamics of adaptive control to design the adaptive fuzzy controller. Many other researchers have atempted to apply the fuzzy approximator or fuzzy logic concepts to the conventional control technics[4, 5, 6]. However, most of their works don't have formality.

In this paper, the fuzzy approximator and SMC scheme are considered. That is, fuzzy logic system theory is applied to design the sliding mode controller. In this procedure, a simple adaptive law is utilized to approximate the unknown function f by fuzzy logic system, an robust adaptive law is also intoduced to reduce the approximation errors, the differences of nonlinear function and fuzzy approximator, and the stability of proposed control scheme are proven. This proposed control scheme is applied to a single link robot arm.

This paper is organized as follows. Section II presents the general fuzzy logic system and fuzzy approximator. In Section III, sliding mode control thory is introduced and optimal sliding mode controller(system is known) is designed. In section IV, adaptive sliding mode control scheme using fuzzy logic system is proposed, adaptive law are designed, and stability of the proposed control scheme is proven. In section V, a single link robot arm is considered to verify the validities of proposed control scheme and comparison between the cases of simple adaptive law and robust adaptive law is noted. Conclusions are drawn in the final section.

2. FUZZY LOGIC SYSTEM

2.1 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

$$R^{(j)} \ : \ IF\ x_1\ is\ A_1^j\ and\ \cdots\ and\ x_n\ is\ A_n^j,\ THEN\ y\ is\ C^j \qquad (2.1)$$

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 input(state) of FLS, *i.e.* $i = 1, 2, \cdots, n$, and j denotes the number of rules of FLS, *i.e.* $j = 1, 2, \cdots, M$. Fuzzy rule (2.1) can be implemented using fuzzy implication, which gives

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

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 used t-norm operator, which gives

$$\mu_{A_1^j\times\cdots\times A_n^j\to C^j}(\underline{x},y) = \mu_{A_i^j}(x_1)\star\cdots\star\mu_{A_n^j}(x_n)\star\mu_{C^j}(y) \quad (2.3)$$

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

2.2 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 "o" is the sup-star composition, results in M fuzzy stes. Using the t-norm operator yields

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

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

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

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 $\star$ in (2.3) and (2.4) to be an algebraic product, then the inference is called product inference. Using product inference, (2.4) becomes

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

2.3 Fuzzifier

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

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

2.4 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 defuzzifier. In this paper, we use the center-average defuzzifier mapping, *i.e.*

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

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

2.5 Fuzzy Bases Function

The fuzzy logic system with *center-average defuzzifier* (2.8), *product inference* (2.6), and *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 writen 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

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

which are called *FBFs(fuzzy bases functions)*. These FBFs have been proved in [3] that they are universal approximators. We can fix all the parameters in $\xi^j(\underline{x})$ at the very beginning of the FBF expansion design procedure, so that the only free design parameters are θ_i. In this paper, we use this fuzzy logic system constructed FBFs with adaptive paramater vector θ as an alternative of unknown function $f(\underline{x})$ in adaptive sliding mode control system. In computer simulation, really, the adaptive parameter vector θ is consisted of random value at beginning. Therefore, we need an assumption that the boundary of $f(\underline{x})$ is known. This boundary is used in universe of discourse of W of the fuzzy logic system.

3. SLIDING MODE CONTROL

Consider the nth-order nonlinear systems of the form

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\cdots \\ \dot{x}_n &= f(x_1,\cdots,x_n) + bu + d(t) \quad (3.1) \\ y &= x_1 \end{aligned}$$

or equivalently of the form

$$\begin{aligned} x^{(n)} &= f(x,\dot{x},\cdots,x^{(n-1)}) + bu + d(t) \\ y &= x \end{aligned} \quad (3.2)$$

where f is a *unknown* continuous functions, b is a control gain, $d(t)$ is the unknown external disturbance, $u \in R$ and $y \in R$ are the input and output of the system, respectively, 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 (3.2) 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, $|d(t)| < D$. The control problem is to obtain the state $\underline{x}$ for tracking a desired state $\underline{x}_d$. With the tracking error

$$\underline{e} = \underline{x}(t) - \underline{x}_d(t) = (e,\dot{e},\cdots,e^{(n-1)})^T \quad (3.3)$$

In general, *a sliding surface* is defined by

$$s(\underline{e}) = \underline{c}\underline{e} = 0, \quad (3.4)$$

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 $h(p) = p^n + c_{n-1}p^{n-1} + \cdots + c_1$ 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 $t \geq 0$. A sufficient condition for this behavior is to choose the control input so that

$$\frac{1}{2} \cdot \frac{d}{dt}(s^2(\underline{e})) \leq -\eta \cdot |s|, \qquad \eta \geq 0. \tag{3.5}$$

Considering $s^2(\underline{e})$ a Lyapunov function, it follows from (3.5) 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 sliding surface. In this way, the trajectory is forced to move always towards the sliding surface(see fig.1). A sliding mode along the sliding surface is thus obtained.

The sliding condition of (3.5) can be rewriten as

$$s \cdot \dot{s} \leq -\eta \cdot |s| \quad or \quad \dot{s} \cdot sgn(s) \leq -\eta. \tag{3.6}$$

By taking the time derivative of both sides of (3.4), we obtain

$$\begin{aligned} \dot{s} &= \underline{c}\dot{\underline{e}} \\ &= c_1\dot{e} + c_2\ddot{e} + \cdots + c_{n-1}e^{n-1} + x^{(n)} - x_d^{(n)} \\ &= \sum_{i=1}^{n-1} c_i e^{(i)} + x^{(n)} - x_d^{(n)} \\ &= \sum_{i=1}^{n-1} c_i e^{(i)} + f(\underline{x}, t) + bu + d(t) - x_d^{(n)}. \end{aligned} \tag{3.7}$$

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

$$s \cdot \dot{s} = s \cdot (\sum_{i=1}^{n-1} c_i e^{(i)} + f(\underline{x}, t) + bu^* + d(t) - x_d^{(n)}) \leq -\eta|s|, \tag{3.8}$$

or

$$sgn(s) \cdot (\sum_{i=1}^{n-1} c_i e^{(i)} + f(\underline{x}, t) + bu^* + d(t) - x_d^{(n)}) \leq -\eta \tag{3.9}$$

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

$$i) \; If \;\; s > 0, \tag{3.10}$$

$$u^* \leq \frac{1}{b}(-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) - d(t) + x_d^{(n)} - \eta_\Delta)$$

$$ii) \; If \;\; s < 0, \tag{3.11}$$

$$u^* \geq \frac{1}{b}(-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) - d(t) + x_d^{(n)} + \eta_\Delta)$$

$$iii) \; If \;\; s = 0, \tag{3.12}$$

$$u^* = \frac{1}{b}(-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) - d(t) + x_d^{(n)})$$

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

$$u^* = \frac{1}{b}(-\sum_{i=1}^{n-1} c_i e^{(i)} - f(\underline{x}, t) - d(t) + x_d^{(n)} - h \cdot sgn(s) \cdot \eta_\Delta), \tag{3.13}$$

where

$$h = \begin{cases} 1 & , if \;\; s \neq 0 \\ 0 & , if \;\; s = 0 \end{cases}. \tag{3.14}$$

This optimal control input guarantees the sliding condition of (3.5).

Howerever, $f(\underline{x}, t)$ and $d(t)$ are unknown. To solve this problem, we propose the adaptive scheme using the fuzzy logic system in next section.

4. ADAPTIVE SLIDING MODE CONTROL USING FUZZY LOGIC SYSTEM

If $f(\underline{x}, t)$ is known, we can easily construct the optimal control input u^* in the previous section. However, $f(\underline{x}, t)$ is 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). The resulting control input is as follows.

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

Then

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

4.1 Adaptive Law Synthesis

Let the optimal parameter vector of fuzzy logic system 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 rewriten as

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

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

$$\dot{s} = \phi^T \xi(\underline{x}) + \omega - h \cdot sgn(s) \cdot [D + \eta_\Delta] + d(t) \tag{4.5}$$

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

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

where γ_1 is positive constant. The time derivative of V is

$$\dot{V} = s\dot{s} + \frac{1}{\gamma_1}\phi^T\dot{\phi} \tag{4.7}$$

$$= s(\phi^T\xi(\underline{x}) + \omega - h \cdot sgn(s) \cdot [D + \eta_\Delta] + d(t)) + \frac{1}{\gamma_1}\phi^T\dot{\phi} \tag{4.8}$$

$$= s\phi^T\xi(\underline{x}) + s\omega - s \cdot h \cdot sgn(s) \cdot [D + \eta_\Delta] + s \cdot d(t) + \frac{1}{\gamma_1}\phi^T\dot{\phi} \tag{4.9}$$

$$= \frac{1}{\gamma_1}\phi^T[\gamma_1 \cdot s \cdot \xi(\underline{x}) - \dot{\theta}] - s \cdot h \cdot sgn(s) \cdot [D + \eta_\Delta] + s \cdot d(t) + s\omega \tag{4.10}$$

where $\dot{\phi} = -\dot{\theta}$, $s \cdot h \cdot sgn(s) \cdot [D + \eta_\Delta] + s \cdot d(t) \geq 0$. 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,* $\leq \epsilon$, if not equal to zero in the adaptive fuzzy system[3]. Therefore, we can choose the adaptive law

$$\dot{\theta} = \gamma_1 \cdot s \cdot \xi(\underline{x}). \tag{4.11}$$

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

4.2 Robust Adaptive Law Synthesis

Let the control input

$$\begin{aligned} u_{rob} &= \frac{1}{b}\Big(-\sum_{i=1}^{n-1} c_i e^{(i)} - \hat{f}(\underline{x}|\theta) + x_d^{(n)} \\ &\quad - h \cdot sgn(s) \cdot (D + \eta_\Delta + \hat{\rho})\Big), \end{aligned} \tag{4.12}$$

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

$$\dot{s} = f(\underline{x}, t) - \hat{f}(\underline{x}|\theta) - h \cdot sgn(s) \cdot (D + \eta_\Delta + \hat{\rho}) + d(t) \tag{4.13}$$

Now consider the Lyapunov candidate

$$V = \frac{1}{2}(s^2 + \frac{1}{\gamma_1}\phi^T\phi + \frac{1}{\gamma_2}\tilde{\rho}^2) \tag{4.14}$$

Applying (4.13) to (4.14) and after straightforward manipulation, we obtain the time derivative of V

$$\begin{aligned} \dot{V} &= \frac{1}{\gamma_1}\phi^T[\gamma_1 \cdot s \cdot \xi(\underline{x}) - \dot{\theta}] \\ &\quad - s \cdot h \cdot sgn(s) \cdot (D + \eta_\Delta) + s \cdot d(t) \\ &\quad + s\omega - h \cdot |s| \cdot \hat{\rho} + \frac{1}{\gamma_2}\tilde{\rho}\dot{\tilde{\rho}} \end{aligned} \tag{4.15}$$

$$\begin{aligned} &= \frac{1}{\gamma_1}\phi^T[\gamma_1 \cdot s \cdot \xi(\underline{x}) - \dot{\theta}] \\ &\quad - s \cdot h \cdot sgn(s) \cdot (D + \eta_\Delta) + s \cdot d(t) + s\omega \\ &\quad - h \cdot |s| \cdot \rho^* + h \cdot |s| \cdot (\rho^* - \hat{\rho}) + \frac{1}{\gamma_2}\tilde{\rho}\dot{\tilde{\rho}} \end{aligned} \tag{4.16}$$

where $s \cdot h \cdot sgn(s) \cdot [D + \eta_\Delta] + s \cdot d(t) \geq 0$, and $s\omega - h \cdot |s| \cdot \rho^* \leq 0$. Therefore we can choose the adaptive laws

$$\dot{\theta} = \gamma_1 \cdot s \cdot \xi(\underline{x}), \tag{4.17}$$

$$\dot{\hat{\rho}} = -\gamma_2 \cdot h \cdot |s|. \tag{4.18}$$

5. DESIGN EXAMPLE

To illustraite the above design approach, a single-link robot arm is considered. The dynamics of the robot arm can be derived using the Euler-Lagrange method,

$$ml^2\ddot{q}(t) + mglsin(q(t)) = \tau(t), \tag{5.1}$$

where $q(t)$ is the angular position of the robot arm, $\tau(t)$ is the control input torque applied to the robot arm, m is the mass of the robot arm, l is the length of the arm, g is the gravitational constant : $9.8(m/sec^2)$. Let $x_1(t) = q(t)$ and $x_2(t) = \dot{q}(t)$, the dynamic equation (5.1) of robot arm can be put in the following form:

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) \\ \dot{x}_2(t) &= -\frac{g}{l}sin(x_1(t)) + \frac{1}{ml^2}\tau(t). \end{aligned} \tag{5.2}$$

The control objective is to maintain $x_1(t) = x_d(t)$ and $x_2(t) = \dot{x}_{1d}(t)$, that is, $\underline{e} = \underline{0}$. In this design example we choose the switching surface given by

$$s(\underline{e}) = \underline{c}\underline{e} = 0, \tag{5.3}$$

where $\underline{e} = [e, \dot{e}]^T$ and $\underline{c} = [c_1, 1]$. In (5.2), $m = 1Kg$, $l = 1m$, $c_1 = 3$. The input of robot arm is designed by (4.1), we choose the $\eta_\Delta = 0.03$ and first need to determine the bound $\hat{f}(\underline{x}, \theta)$, we have $|f(\underline{x}, \theta)| = 9.8$. This value is used in determining the initial random values of θ at begining. In simulations, we chose 5 fuzzy level, *i.e.*, NB,NS,ZO,PS,PB on the each universe of discourse of x_1, x_2, and we use the fuzzy logic system with center-average defuzzifier, product inference, singleton fuzzifier and gaussian membership functions. Because the sigle-link robot system is 2-nd order system, the used fuzzy logic system in simulation was constructed with 25 rules (5^2 : 2 input variables, 5 fuzzy levels).

Fig. 1.,2. show the trajectories for angular displacement of the robot arm under using u_{adp} and desired value not changed/changed(0.0 → 0.6), respectively.

Fig. 3.,4. show the trajectories for angular displacement of the robot arm under using u_{rob} and desired value not changed/changed (0.0 → 0.6), respectively.

In these figures, angular displacement of the robot arm converge to desired value within about 2 seconds, and we can see that both proposed control schemes achieve the control objective well. In addition, the sliding mode control scheme based on the robust adaptive law is more efective than the simple adaptive law in transient time(without overshoot).

6. CONCLUSIONS

The proposed adaptive sliding mode control scheme in this study was motivated by the fuzzy approximator theory, research of [3], and we used the fuzzy approximator as the estimator to unknown function f.
We introduce the conventional sliding mode control theory and proposed the method that fuzzy approximator was applied to the variable structure control. We also proposed the robust controller using the robust adaptive law, *i.e.*, we used the adaptive law to reduce the error, difference of nonlinear function and fuzzy approximator. The stability of proposed control scheme was proven and we verified that the sliding mode control scheme based on the robust adaptive law is more effective than the simple adaptive law in transient time in simulation.

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] 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.

[5] 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.

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

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

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

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

[10] W.C.Ham, "Adaptive control based on explicit model of robot manipulators," *IEEE trans. Automat. Contr.*, vol. 38, Apr, pp.654-658, 1993.

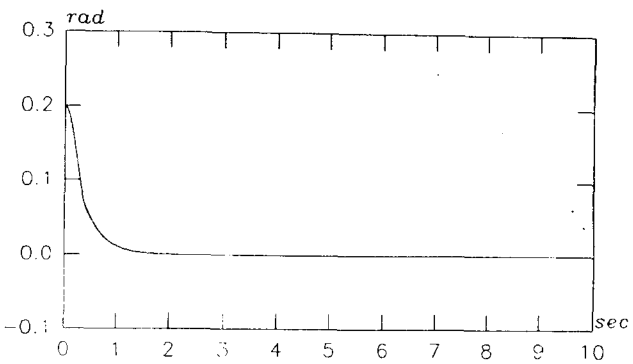

Fig. 1. Trajectory of angular displacement q (initial : 0.2, desired value : 0.0, input : u_{adp})

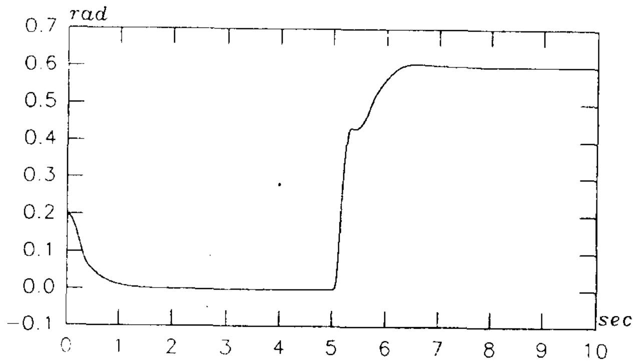

Fig. 2. Trajectory of angular displacement q (initial : 0.2, desired value : 0.0 → 0.6, input : u_{adp})

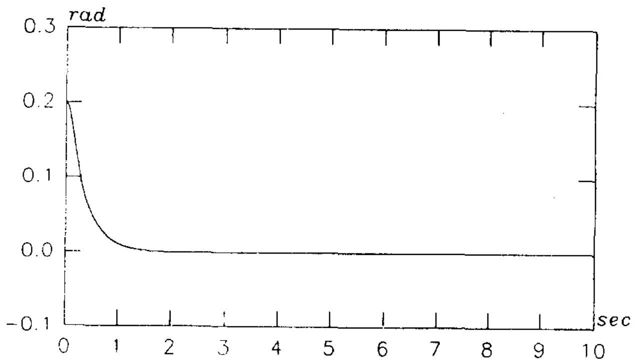

Fig. 3. Trajectory of angular displacement q (initial : 0.2, desired value : 0.0, input : u_{rob})

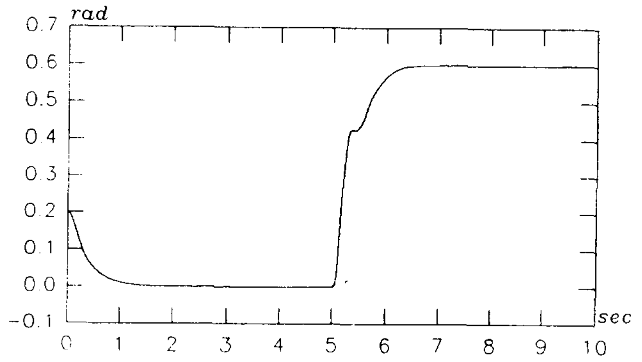

Fig. 4. Trajectory of angular displacement q (initial : 0.2, desired value : 0.0 → 0.6, input : u_{rob})