A revised adaptive fuzzy sliding mode controller for robotic manipulators

Xiaosong Lu*

Department of Systems and Computer Engineering,
Carleton University,
1125 Colonel By Drive,
Ottawa, Ontario, Canada
E-mail: luxiaos@sce.carleton.ca
*Corresponding author

Howard M. Schwartz

Department of Systems and Computer Engineering,
Carleton University,
1125 Colonel By Drive,
Ottawa, Ontario, Canada
E-mail: schwartz@sce.carleton.ca

Abstract: In this paper, we revised an adaptive fuzzy sliding mode control algorithm applied to a two-degree-of-freedom robotic manipulator. Stability of the overall system for the revised algorithm is proven in a Lyapunov sense. A discussion on the proof of Lyapunov stability for different algorithms is presented.

Keywords: Adaptive; sliding mode control; fuzzy systems.

Biographical notes: Xiaosong Lu was born in Shandong, China in 1976. He received his B.Eng. Degree and M.Eng. degree from Shandong University, Shandong, China in 1994 and 2001 respectively. He finished his MASc degree from Carleton University, Ottawa, Canada in 2007. He is currently working towards his Ph.D. in Systems and Computer Engineering at Carleton University. His research interests include adaptive control and fuzzy control.

Professor Howard M. Schwartz received his B.Eng. degree from McGill University, Montreal, Quebec in June 1981 and his M.S. degree and Ph.D. degree from M.I.T., Cambridge, Massachusetts in 1982 and 1987 respectively. He is currently a Professor in Systems and Computer Engineering at Carleton University, Ottawa, Canada. His research interests include adaptive and intelligent control systems, robotics and process control, system modelling, system identification, system simulation and computer vision systems.

1 Introduction

Classical sliding mode control is robust to model uncertainties and external disturbances. A sliding mode control method with a switching control law guarantees asymptotic stability of the system, but the addition of the switching control law introduces chattering into the system. One way of attenuating chattering is to insert a saturation function (Asada and Slotine, 1986) inside a boundary layer around the sliding surface. Unfortunately, this addition disrupts the Lyapunov stability of the closed loop system. Classical sliding mode control has difficulty in handling unstructured model uncertainties. One can overcome this problem by combining sliding mode control and fuzzy systems together. Fuzzy rules allow fuzzy systems to approximate arbitrary continuous functions (Wang, 1997). To approximate a time-varying nonlinear system, a fuzzy system requires a large amount of fuzzy rules. This large number of fuzzy rules will cause a high computational load. The addition of an adaptation law to a fuzzy sliding mode controller to online tune the parameters of the fuzzy rules in use will ensure a moderate computational load.

A number of adaptive fuzzy sliding mode control (AFSMC) algorithms have been proposed in the last decade (Yoo and Ham, 1998, 2000; Wang et al., 2001; Guo and Woo, 2003; Ho et al., 2004; Wai et al., 2004; Lin and Hsu, 2004; Akbarzadeh-T and Shahnazi, 2005; Shahnazi et al., 2006; Medhaffar et al., 2006). The adaptation laws in these algorithms are designed based on Lyapunov stability theory. Asymptotic stability of the closed loop system for these algorithms is also proved in the sense of Lyapunov. Adaptive fuzzy sliding mode controllers can be classified into two categories: indirect adaptive fuzzy sliding mode controllers and direct adaptive fuzzy sliding mode controllers (Wang, 1994).

In an indirect adaptive fuzzy sliding mode control method, the controller is used to estimate the parameters of the system's dynamics. Yoo and Ham (1998) proposed a single-input single-output (SISO) fuzzy system to approximate the unknown functions of a nonlinear system. Based on Yoo and Ham (1998)'s approach , Medhaffar et al. (2006) designed an indirect adaptive fuzzy sliding mode control algorithm applied to robotic manipulators. The multi-input multi-output (MIMO) fuzzy systems used in Medhaffar et al. (2006)'s approach are applied to estimate the dynamic equations of the robotic manipulator and the fuzzy rules are reduced by introducing sliding surfaces as the inputs. To avoid chattering in Yoo and Ham (1998)'s algorithm, a fuzzy system is used to substitute for the discontinuous control term in the algorithms proposed by Medhaffar et al. (2006) and Wang et al. (2001).

In a direct adaptive fuzzy sliding mode control method, the controller is used to directly adjust the parameters of the control law without estimating the system's dynamics. Yoo and Ham (2000) proposed a MIMO fuzzy system to compensate for model uncertainties of a robotic manipulator. Unfortunately, using a MIMO fuzzy system requires an inordinate number of fuzzy rules which leads to a high computational load. Guo and Woo (2003) applied a SISO fuzzy system to adjust the control gain in the control law for a robotic manipulator which both decreased the number of fuzzy rules and attenuated chattering. Different from Guo and Woo (2003)'s method, Ho et al. (2004) applied a PI controller inside a boundary layer to attenuate chattering and the parameters of this PI controller are online adjusted by adaptation laws. Wai et al. (2004) designed a AFSMC method to estimate the bound of the approximation error for electrical servo drives. However, algorithms proposed by Yoo and Ham (2000), Guo and Woo (2003), Ho et al. (2004) and Wai et al. (2004) can only tune the consequence part of the fuzzy rules, which places the onus of designing the premise part upon the designer. Therefore, Lin and Hsu (2004) presented a direct AFSMC method to online tune both the premise and consequence parts of fuzzy rules. In Lin and Hsu (2004)'s algorithm, a fuzzy controller and a compensation controller are proposed to construct a control law and the bound of the compensation controller is adjusted by adaptation laws. Since Lin and Hsu (2004)'s algorithm is only designed for induction servo motor systems, it is not applicable to robotic manipulators.

In this paper, we will revise the adaptation laws of the algorithm proposed by Lin and Hsu (2004) so that it is suitable for the robotic application. We then prove Lyapunov stability of the closed loop system for the revised algorithm. We will also discuss stability issues for the algorithms proposed by Yoo and Ham (2000), Guo and Woo (2003), Wang et al. (2001) and Medhaffar et al. (2006): Lyapunov stability in a practical sense for the algorithms proposed by Yoo and Ham (2000) and Guo and Woo (2003) and problems on the proof of Lyapunov stability for the algorithms proposed by Wang et al. (2001) and Medhaffar et al. (2006).

This paper is organized as follows: an overview of fuzzy systems and fuzzy rules; the design of the revised algorithms for robotic manipulators; the stability analysis of four algorithms in the sense of Lyapunov; and, the conclusion.

2 Fuzzy Systems

The fuzzy rule base of a multi-input multi-output (MIMO) fuzzy system is comprised of the following fuzzy if-then rules

$$\begin{aligned} R^{(l)}: \text{ If } x_1 \text{ is } A_1^l \text{ and } \cdots \text{ and } x_n \text{ is } A_n^l, \\ \text{then } y_1 \text{ is } B_1^l \text{ and } \cdots \text{ and } y_m \text{ is } B_m^l \end{aligned} \quad (1)$$

where $l = 1, 2, \cdots, M$ denotes the number of fuzzy if-then rules; $x_1, \cdots, x_n$ and $y_1, \cdots, y_m$ are the input and output variables of the fuzzy system. A MIMO fuzzy system can be decomposed into a collection of multi-input single-output(MISO) fuzzy systems (Wang, 1997). A MISO fuzzy rule base consists of a collection of fuzzy IF-THEN rules in the following form (Wang, 1994):

$$R^{(l)}: \text{ If } x_1 \text{ is } A_1^l \text{ and } \cdots \text{ and } x_n \text{ is } A_n^l, \text{ then } y \text{ is } B^l$$

where $x_1, \cdots, x_n$ are input variables and y is the output variable. The membership functions of the fuzzy sets A_i^l and B^l are defined as $\mu_{A_i^l}(x)$, $\mu_{B^l}(y)$. The MISO fuzzy systems with center average defuzzifier, product-inference rule and singleton fuzzifier are of the following form (Wang, 1994):

$$y(\boldsymbol{x}) = \frac{\sum_{l=1}^{M} y^l(\prod_{i=1}^{n} \mu_{A_i^l}(x_i))}{\sum_{l=1}^{M}(\prod_{i=1}^{n} \mu_{A_i^l}(x_i))} \tag{2}$$

where $l = 1, \cdots, M$ denotes the number of fuzzy if-then rules, $i = 1, \cdots, n$ denotes the number of input variables, and y^l is the point at which $\mu_{B^l}(y)$ achieves its maximum value (we assume $\mu_{B^l}(y^l) = 1$). Rewrite (2) as follows:

$$f(\boldsymbol{x}) = \sum_{l=1}^{M} \theta^l \xi^l(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{\xi}(\boldsymbol{x}) \tag{3}$$

where $\boldsymbol{\theta} = [\theta^1, \cdots, \theta^M]^T$, $\boldsymbol{\xi}(\boldsymbol{x}) = [\xi^1(\boldsymbol{x}), \cdots, \xi^M(\boldsymbol{x})]^T$, and

$$\xi^l(\boldsymbol{x}) = \prod_{i=1}^{n} \mu_{A_i^l}(x_i) / \sum_{l=1}^{M}(\prod_{i=1}^{n} \mu_{A_i^l}(x_i)). \tag{4}$$

For a single-input single-output(SISO) fuzzy system, each fuzzy if-then rule is represented as the form of

$$R^{(l)} : \text{ If } x \text{ is } A^l, \text{ then } y \text{ is } B^l \tag{5}$$

The SISO fuzzy system is described as

$$f(x) = \sum_{l=1}^{M} \theta^l \xi^l(x) = \boldsymbol{\theta}^T \boldsymbol{\xi}(x) \tag{6}$$

where $\boldsymbol{\theta} = [\theta^1, \cdots, \theta^M]^T$, $\boldsymbol{\xi}(x) = [\xi^1(x), \cdots, \xi^M(x)]^T$, and $\xi^l(x) = \mu_{A^l}(x) / \sum_{l=1}^{M} \mu_{A^l}(x)$. In order to use fuzzy systems to estimate nonlinear functions, we introduce the following universal approximation theorem (Wang, 1997).

Theorem 2.1. *For any given real continuous function $g(\boldsymbol{x})$ on a compact set $U \subset R^n$ and arbitrary $\epsilon > 0$, there exists a fuzzy logic system $f(\boldsymbol{x})$ in the form of (3) such that*

$$\sup_{\boldsymbol{x} \in U} |f(\boldsymbol{x}) - g(\boldsymbol{x})| < \epsilon . \tag{7}$$

3 Design of adaptive fuzzy sliding mode controller for robotic manipulators

The dynamic equation of an m-link robotic manipulator is

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{8}$$

where $\boldsymbol{q} = [q_1, \cdots, q_m]^T$ is an $m \times 1$ vector of joint position, $\boldsymbol{M}(\boldsymbol{q})$ is an $m \times m$ inertial matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is an $m \times m$ matrix of Coriolis and centrifugal forces, $\boldsymbol{G}(\boldsymbol{q})$ is an $m \times 1$ gravity vector and $\boldsymbol{\tau} = [\tau_1, \cdots, \tau_m]^T$ is an $m \times 1$ vector of joint torques.

3.1 Revised Lin and Hsu's algorithm

The adaptive fuzzy sliding mode controller can only tune the consequence part of their respective fuzzy rules in the algorithms proposed by Yoo and Ham (1998), Yoo and Ham (2000), Wang et al. (2001), Guo and Woo (2003), Ho et al. (2004), Wai et al. (2004), Akbarzadeh-T and Shahnazi (2005), and Medhaffar et al. (2006). Lin and Hsu (2004) surpassed this limitation by presenting a number of adaptation laws that are capable of online tuning of both the premise and consequence parts of the fuzzy rules in question. Since Lin and Hsu (2004)'s algorithm is specifically applied to an induction servomotor drive, we need to revise this algorithm for the control of robotic manipulators which we refer to as the revised Lin and Hsu's algorithm.

We define the tracking error as

$$\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q}_d \tag{9}$$

where $\boldsymbol{q}_d = [q_{1d}, \cdots, q_{md}]^T$ is the desired trajectories. The sliding surface is given as

$$\boldsymbol{s} = \dot{\boldsymbol{e}} + \boldsymbol{\lambda}\boldsymbol{e} \tag{10}$$

The reference state is given as

$$\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}} - \boldsymbol{s} = \dot{\boldsymbol{q}}_d - \boldsymbol{\lambda}\, \boldsymbol{e} \tag{11}$$

$$\ddot{\boldsymbol{q}}_r = \ddot{\boldsymbol{q}} - \dot{\boldsymbol{s}} = \ddot{\boldsymbol{q}}_d - \boldsymbol{\lambda}\, \dot{\boldsymbol{e}} . \tag{12}$$

The control input is given by

$$\boldsymbol{\tau} = \hat{\boldsymbol{M}}\ddot{\boldsymbol{q}}_r + \hat{\boldsymbol{C}}\dot{\boldsymbol{q}}_r + \hat{\boldsymbol{G}} - \hat{\boldsymbol{F}}(\boldsymbol{s}) - \boldsymbol{F}_{cp}(\boldsymbol{s}) \tag{13}$$

where $\hat{\boldsymbol{F}}(\boldsymbol{s}) = [\hat{f}_1(s_1), \cdots, \hat{f}_m(s_m)]^T$ and $\boldsymbol{F}_{cp}(\boldsymbol{s}) = [f_{cp_1}(s_1), \cdots, f_{cp_m}(s_m)]^T$. The fuzzy system $\hat{f}_j(s_j)(j = 1, \cdots, m)$ is defined as

$$\hat{f}_j(s_j) = \boldsymbol{\theta}_j^T \boldsymbol{\Phi}_j(s_j) \tag{14}$$

where $\boldsymbol{\theta}_j = [\theta_j^1, \theta_j^2, \cdots, \theta_j^M]^T$ and $\boldsymbol{\Phi}_j(s_j) = [\Phi^1(s_j), \Phi^2(s_j), \cdots, \Phi^M(s_j)]^T$. Assume $\sum_{l=1}^{M} \mu_{A_j^l}(s_j) = 1$ and (14) becomes

$$\hat{f}_j(s_j) = \boldsymbol{\theta}_j^T \boldsymbol{\Phi}_j(s_j) = \frac{\sum_{l=1}^{M} \theta_j^l \mu_{A_j^l}(s_j)}{\sum_{l=1}^{M} \mu_{A_j^l}(s_j)} = \sum_{l=1}^{M} \theta_j^l \mu_{A_j^l}(s_j) \tag{15}$$

where $\mu_{A_j^l}(s_j) = \Phi^l(s_j) = \exp\left[-(\sigma_j^l(s_j - \alpha_j^l))^2\right]$ $(l = 1, \cdots, M)$. Define $\tilde{f}_j$ such that

$$\begin{aligned} \tilde{f}_j &= f_j - \hat{f}_j(s_j) \\ &= \hat{f}_j^*(s_j) - \hat{f}_j(s_j) + \Delta_j \\ &= \boldsymbol{\theta}_j^{*T} \boldsymbol{\Phi}_j^* - \boldsymbol{\theta}_j^T \boldsymbol{\Phi}_j + \Delta_j \end{aligned} \tag{16}$$

where $\boldsymbol{\theta}_j^*$ and $\boldsymbol{\Phi}_j^*$ are the optimal values based on the universal approximation theorem in (7). We define $\tilde{\boldsymbol{\theta}}_j = \boldsymbol{\theta}_j^* - \boldsymbol{\theta}_j$, $\tilde{\boldsymbol{\Phi}}_j = \boldsymbol{\Phi}_j^* - \boldsymbol{\Phi}_j$ and (16) is rewritten as

$$\begin{aligned} \tilde{f}_j &= (\boldsymbol{\theta}_j + \tilde{\boldsymbol{\theta}}_j)^T(\boldsymbol{\Phi}_j + \tilde{\boldsymbol{\Phi}}_j) - \boldsymbol{\theta}_j^T \boldsymbol{\Phi}_j + \Delta_j \\ &= \boldsymbol{\theta}_j^T \tilde{\boldsymbol{\Phi}}_j + \tilde{\boldsymbol{\theta}}_j^T \boldsymbol{\Phi}_j + \tilde{\boldsymbol{\theta}}_j^T \tilde{\boldsymbol{\Phi}}_j + \Delta_j \end{aligned} \tag{17}$$

We take Taylor series expansion of $\boldsymbol{\Phi}_j$ around two vectors $\boldsymbol{\alpha}_j$ and $\boldsymbol{\sigma}_j$ where $\boldsymbol{\alpha}_j = [\alpha_j^1, \cdots, \alpha_j^M]^T$ and $\boldsymbol{\sigma}_j = [\sigma_j^1, \cdots, \sigma_j^M]^T$(α_j^l and σ_j^l are defined in (15)):

$$\boldsymbol{\Phi}_j^* = \boldsymbol{\Phi}_j + \frac{\partial \boldsymbol{\Phi}_j}{\partial \boldsymbol{\alpha}_j}\tilde{\boldsymbol{\alpha}}_j + \frac{\partial \boldsymbol{\Phi}_j}{\partial \boldsymbol{\sigma}_j}\tilde{\boldsymbol{\sigma}}_j + h.o.t. \tag{18}$$

where $\tilde{\boldsymbol{\alpha}}_j = \boldsymbol{\alpha}_j^* - \boldsymbol{\alpha}_j$, $\tilde{\boldsymbol{\sigma}}_j = \boldsymbol{\sigma}_j^* - \boldsymbol{\sigma}_j$ and $h.o.t.$ denotes the higher order terms. We rewrite (18) as

$$\begin{aligned}\tilde{\boldsymbol{\Phi}}_j &= \frac{\partial \boldsymbol{\Phi}_j}{\partial \boldsymbol{\alpha}_j}\tilde{\boldsymbol{\alpha}}_j + \frac{\partial \boldsymbol{\Phi}_j}{\partial \boldsymbol{\sigma}_j}\tilde{\boldsymbol{\sigma}}_j + h.o.t. \\ &= \boldsymbol{B}_j\tilde{\boldsymbol{\alpha}}_j + \boldsymbol{D}_j\tilde{\boldsymbol{\sigma}}_j + h.o.t. \end{aligned} \tag{19}$$

where

$$\boldsymbol{B}_j = \begin{bmatrix} \frac{\partial \Phi_j^1}{\partial \alpha_j^1} & \frac{\partial \Phi_j^2}{\partial \alpha_j^1} & \cdots & \frac{\partial \Phi_j^M}{\partial \alpha_j^1} \\ \frac{\partial \Phi_j^1}{\partial \alpha_j^2} & \frac{\partial \Phi_j^2}{\partial \alpha_j^2} & \cdots & \frac{\partial \Phi_j^M}{\partial \alpha_j^2} \\ \vdots & & & \\ \frac{\partial \Phi_j^1}{\partial \alpha_j^M} & \frac{\partial \Phi_j^2}{\partial \alpha_j^M} & \cdots & \frac{\partial \Phi_j^M}{\partial \alpha_j^M} \end{bmatrix} \tag{20}$$

$$\boldsymbol{D}_j = \begin{bmatrix} \frac{\partial \Phi_j^1}{\partial \sigma_j^1} & \frac{\partial \Phi_j^2}{\partial \sigma_j^1} & \cdots & \frac{\partial \Phi_j^M}{\partial \sigma_j^1} \\ \frac{\partial \Phi_j^1}{\partial \sigma_j^2} & \frac{\partial \Phi_j^2}{\partial \sigma_j^2} & \cdots & \frac{\partial \Phi_j^M}{\partial \sigma_j^2} \\ \vdots & & & \\ \frac{\partial \Phi_j^1}{\partial \sigma_j^M} & \frac{\partial \Phi_j^2}{\partial \sigma_j^M} & \cdots & \frac{\partial \Phi_j^M}{\partial \sigma_j^M} \end{bmatrix}. \tag{21}$$

We substitute (19) into (17):

$$\begin{aligned}\tilde{f}_j &= \boldsymbol{\theta}_j^T(\boldsymbol{B}_j\tilde{\boldsymbol{\alpha}}_j + \boldsymbol{D}_j\tilde{\boldsymbol{\sigma}}_j + h.o.t.) + \tilde{\boldsymbol{\theta}}_j^T\boldsymbol{\Phi}_j \\ &\quad + \tilde{\boldsymbol{\theta}}_j^T\tilde{\boldsymbol{\Phi}}_j + \Delta_j \\ &= \boldsymbol{\theta}_j^T\boldsymbol{B}_j\tilde{\boldsymbol{\alpha}}_j + \boldsymbol{\theta}_j^T\boldsymbol{D}_j\tilde{\boldsymbol{\sigma}}_j + \tilde{\boldsymbol{\theta}}_j^T\boldsymbol{\Phi}_j + \varepsilon_j \end{aligned} \tag{22}$$

where $\varepsilon_j = \boldsymbol{\theta}_j^T(h.o.t.) + \tilde{\boldsymbol{\theta}}_j^T\tilde{\boldsymbol{\Phi}}_j + \Delta_j$ is assumed to be bounded by $|\varepsilon_j| \leq E_j$. E_j is a constant and the value of E_j is uncertain to the designer. We define E^* as the real value and the estimation error is given by

$$\tilde{E}_j = E_j^* - E_j \tag{23}$$

We produce an adaptation law to online tune the following parameters: $\boldsymbol{\theta}_j$ in (14), σ_j^l, α_j^l in (15) and the bound E_j in (23). The adaptation laws are given as

$$\dot{\boldsymbol{\theta}}_j = \eta_{j2}s_j\boldsymbol{\Phi}_j \tag{24}$$
$$\dot{\boldsymbol{\alpha}}_j = \eta_{j3}s_j\boldsymbol{B}_j^T\boldsymbol{\theta}_j \tag{25}$$
$$\dot{\boldsymbol{\sigma}}_j = \eta_{j4}s_j\boldsymbol{D}_j^T\boldsymbol{\theta}_j \tag{26}$$
$$f_{cp_j}(s_j) = E_j\text{sgn}(s_j) \tag{27}$$
$$\dot{E}_j = \eta_{j1}|s_j| \tag{28}$$

where $\eta_{j1}, \cdots, \eta_{j4}$ are positive constants, $\boldsymbol{\alpha}_j = [\alpha_j^1, \alpha_j^2, \cdots, \alpha_j^M]^T$, $\boldsymbol{\sigma}_j = [\sigma_j^1, \sigma_j^2, \cdots, \sigma_j^M]^T$ and $f_{cp_j}(s_j)$ is the compensation term defined in (13).

3.2 Stability proof of the revised Lin and Hsu's algorithm

We define the following Lyapunov function candidate:

$$V = \frac{1}{2}\boldsymbol{s}^T\boldsymbol{M}\boldsymbol{s} + \frac{1}{2}\sum_{j=1}^m\left(\frac{\tilde{E}_j^2}{\eta_{j1}} + \frac{\tilde{\boldsymbol{\theta}}_j^T\tilde{\boldsymbol{\theta}}_j}{\eta_{j2}} + \frac{\tilde{\boldsymbol{\alpha}}_j^T\tilde{\boldsymbol{\alpha}}_j}{\eta_{j3}} + \frac{\tilde{\boldsymbol{\sigma}}_j^T\tilde{\boldsymbol{\sigma}}_j}{\eta_{j4}}\right) \tag{29}$$

We take the time derivative of V:

$$\begin{aligned}\dot{V} &= \boldsymbol{s}^T\boldsymbol{M}\dot{\boldsymbol{s}} + \frac{1}{2}\boldsymbol{s}^T\dot{\boldsymbol{M}}\boldsymbol{s} + \sum_{j=1}^m\left(\frac{\tilde{E}_j\dot{\tilde{E}}_j}{\eta_{j1}} + \frac{\tilde{\boldsymbol{\theta}}_j^T\dot{\tilde{\boldsymbol{\theta}}}_j}{\eta_{j2}}\right. \\ &\quad \left. + \frac{\tilde{\boldsymbol{\alpha}}_j^T\dot{\tilde{\boldsymbol{\alpha}}}_j}{\eta_{j3}} + \frac{\tilde{\boldsymbol{\sigma}}_j^T\dot{\tilde{\boldsymbol{\sigma}}}_j}{\eta_{j4}}\right)\end{aligned} \tag{30}$$

where $\tilde{E}_j = E_j^* - E_j$, $\tilde{\boldsymbol{\theta}}_j = \boldsymbol{\theta}_j^* - \boldsymbol{\theta}_j$, $\tilde{\boldsymbol{\alpha}}_j = \boldsymbol{\alpha}_j^* - \boldsymbol{\alpha}_j$, $\tilde{\boldsymbol{\sigma}}_j = \boldsymbol{\sigma}_j^* - \boldsymbol{\sigma}_j$. Since $\dot{\boldsymbol{M}} - 2\boldsymbol{C}$ is a skew-symmetric matrix, we can get $\boldsymbol{s}^T\boldsymbol{M}\dot{\boldsymbol{s}} + \frac{1}{2}\boldsymbol{s}^T\dot{\boldsymbol{M}}\boldsymbol{s} = \boldsymbol{s}^T(\boldsymbol{M}\dot{\boldsymbol{s}} + \boldsymbol{C}\boldsymbol{s})$. From (8) and (11)-(13) we get

$$\boldsymbol{M}\dot{\boldsymbol{s}} + \boldsymbol{C}\boldsymbol{s} = \boldsymbol{F} - \hat{\boldsymbol{F}}(\boldsymbol{s}) - \boldsymbol{F}_{cp}(\boldsymbol{s}) \tag{31}$$

where $\boldsymbol{F} = \Delta\boldsymbol{M}\ddot{\boldsymbol{q}}_r + \Delta\boldsymbol{C}\dot{\boldsymbol{q}}_r + \Delta\boldsymbol{G}$, $\Delta\boldsymbol{M} = \hat{\boldsymbol{M}} - \boldsymbol{M}$, $\Delta\boldsymbol{C} = \hat{\boldsymbol{C}} - \boldsymbol{C}$ and $\Delta\boldsymbol{G} = \hat{\boldsymbol{G}} - \boldsymbol{G}$. Then $\dot{V}$ becomes

$$\begin{aligned}\dot{V} &= \sum_{j=1}^m s_j(f_j - \hat{f}_j(s_j) - f_{cp_j}) - \sum_{j=1}^m\left(\frac{\tilde{E}_j\dot{E}_j}{\eta_{j1}} + \frac{\tilde{\boldsymbol{\theta}}_j^T\dot{\boldsymbol{\theta}}_j}{\eta_{j2}}\right. \\ &\quad \left. + \frac{\tilde{\boldsymbol{\alpha}}_j^T\dot{\boldsymbol{\alpha}}_j}{\eta_{j3}} + \frac{\tilde{\boldsymbol{\sigma}}_j^T\dot{\boldsymbol{\sigma}}_j}{\eta_{j4}}\right) \\ &= \sum_{j=1}^m s_j(\boldsymbol{\theta}_j^T\boldsymbol{B}_j\tilde{\boldsymbol{\alpha}}_j + \boldsymbol{\theta}_j^T\boldsymbol{D}_j\tilde{\boldsymbol{\sigma}}_j + \tilde{\boldsymbol{\theta}}_j^T\boldsymbol{\Phi}_j + \varepsilon_j - f_{cp_j}) \\ &\quad - \sum_{j=1}^m\left(\frac{\tilde{E}_j\dot{E}_j}{\eta_{j1}} + \frac{\tilde{\boldsymbol{\theta}}_j^T\dot{\boldsymbol{\theta}}_j}{\eta_{j2}} + \frac{\tilde{\boldsymbol{\alpha}}_j^T\dot{\boldsymbol{\alpha}}_j}{\eta_{j3}} + \frac{\tilde{\boldsymbol{\sigma}}_j^T\dot{\boldsymbol{\sigma}}_j}{\eta_{j4}}\right) \\ &= \sum_{j=1}^m\left[\tilde{\boldsymbol{\theta}}_j^T\left(s_j\boldsymbol{\Phi}_j - \frac{\dot{\boldsymbol{\theta}}_j}{\eta_{j2}}\right) + \tilde{\boldsymbol{\alpha}}_j^T\left(s_j\boldsymbol{B}_j^T\boldsymbol{\theta}_j - \frac{\dot{\boldsymbol{\alpha}}_j}{\eta_{j3}}\right)\right. \\ &\quad \left. + \tilde{\boldsymbol{\sigma}}_j^T\left(s_j\boldsymbol{D}_j^T\boldsymbol{\theta}_j - \frac{\dot{\boldsymbol{\sigma}}_j}{\eta_{j4}}\right)\right] + \sum_{j=1}^m\left(s_j\varepsilon_j - s_jf_{cp_j}\right. \\ &\quad \left. - \frac{\tilde{E}_j\dot{E}_j}{\eta_{j1}}\right)\end{aligned} \tag{32}$$

We substitute the adaptation law (24)-(28) into (32) and get

$$\begin{aligned}\dot{V} &= \sum_{j=1}^m\left[s_j\varepsilon_j - s_jE_j\text{sgn}(s_j) - \tilde{E}_j|s_j|\right] \\ &\leq \sum_{j=1}^m\left[|s_j||\varepsilon_j| - E_j^*|s_j|\right] \\ &= \sum_{j=1}^m\left[|s_j|(|\varepsilon_j| - E_j^*)\right] \leq 0\end{aligned} \tag{33}$$

where $\dot{V}$ is negative semidefinite. We define $\dot{V}_j = s_j\varepsilon_j - s_jE_j\text{sgn}(s_j) - \tilde{E}_j|s_j|$ and rewrite (33) as

$$\sum_{j=1}^m\dot{V}_j \leq \sum_{j=1}^m\left[|s_j(t)|(|\varepsilon_j| - E_j^*)\right] \leq 0 \tag{34}$$

From $\dot{V}_j \leq 0$, we can get $s_j(t)$ is bounded. We assume $|s_j(t)| \leq \eta_s$ and rewrite $\dot{V}_j \leq |s_j(t)|(|\varepsilon_j| - E_j^*)$ as

$$|s_j(t)| \leq \frac{1}{E_j^*}|s_j(t)||\varepsilon_j| - \frac{1}{E_j^*}\dot{V}_j \leq \frac{\eta_s}{E_j^*}|\varepsilon_j| - \frac{1}{E_j^*}\dot{V}_j \tag{35}$$

Then we take the integral on both sides of (35):

$$\begin{aligned}\int_0^t |s_j(\nu)|d\nu &\le \frac{\eta_s}{E_j^*}\int_0^t |\varepsilon_j|d\nu + \frac{1}{E_j^*}(V_j(0)-V_j(t)) \\ &\le \frac{\eta_s}{E_j^*}\int_0^t |\varepsilon_j|d\nu + \frac{1}{E_j^*}(|V_j(0)|+|V_j(t)|)\end{aligned} \tag{36}$$

If $\varepsilon_j \in L_1$, we can get $s_j \in L_1$ from (36). Since we can prove $\dot{s}_j$ is bounded (see proof in (Wang, 1994)), we have $\dot{s}_j \in L_\infty$ and s_j is uniformly continuous. Given that the right hand side of (36) is bounded, then by using Barbalat's lemma, we can get $\lim_{t\to\infty} s_j(t) = 0$ and $\lim_{t\to\infty} e_j(t) = 0$.

3.3 Simulation results

The dynamic equation of a two-link robotic manipulator is given as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{37}$$

where

$$\boldsymbol{M}(\boldsymbol{q}) = \begin{bmatrix} P_1 + 2P_2 + 2P_2\cos q_2 & P_2 + P_2\cos q_2 \\ P_2 + P_2\cos q_2 & P_2 \end{bmatrix},$$

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \begin{bmatrix} -2P_2\dot{q}_1\dot{q}_2\sin q_2 - P_2\dot{q}_2^2\sin q_2 \\ P_2\dot{q}_1^2\sin q_2 \end{bmatrix};$$

$P_1 = m_1 l^2 = 1.0$ and $P_2 = m_2 l^2 = 2.0$ (m_1, m_2 are mass; links 1 and 2 have the same length of l); q_1 and q_2 are joint positions of the links 1 and 2. Since robotic manipulators cannot follow a step sequence instantaneously, the desired trajectory will be the output of a filtered sequence of unit steps. We define the transfer function of the pre-filter for each joint of the robotic manipulator:

$$W_m(s) = \frac{4}{s^2+4s+4}. \tag{38}$$

The initial values of the robotic manipulators' joint positions are set to 0.5 radians. The estimated mass and Coriolis matrices are given by

$$\hat{\boldsymbol{M}} = \begin{bmatrix} \hat{P}_1 + 2\hat{P}_2 + 2\hat{P}_2\cos q_2 & \hat{P}_2 + \hat{P}_2\cos q_2 \\ \hat{P}_2 + \hat{P}_2\cos q_2 & P_2 \end{bmatrix} \tag{39}$$

$$\hat{\boldsymbol{C}} = \begin{bmatrix} -\hat{P}_2\dot{q}_2\sin q_2 & -\hat{P}_2\dot{q}_1\sin q_2 - \hat{P}_2\dot{q}_2\sin q_2 \\ \hat{P}_2\dot{q}_1\sin q_2 & 0 \end{bmatrix} \tag{40}$$

where $\hat{P}_1 = 2, \hat{P}_2 = 4$. The number and the type of membership functions for each input variable are chosen to be consistent and comparable with the algorithms proposed by Lin and Hsu (2004). We define five membership functions for each input variable in (15) ($l = 1,\cdots,5$). The parameters of the adaptation laws in (24)-(28) are selected as $\eta_{j2} = \eta_{j3} = \eta_{j4} = 100$, $\eta_{j1} = 200 (j = 1,2)$ and $\boldsymbol{\sigma}_1(0) = \boldsymbol{\sigma}_2(0) = [1,1,1,1,1]^T$ by trial and error.

It can be seen from Figure 1 that the tracking error errors are with the range [-0.022,0.012] radians after 4 seconds. The tuning methodology utilized in the premise and consequence parts of the fuzzy rules allows the fuzzy system to approximate the control input. The drawback is the chattering phenomenon appearing in Figure 2.

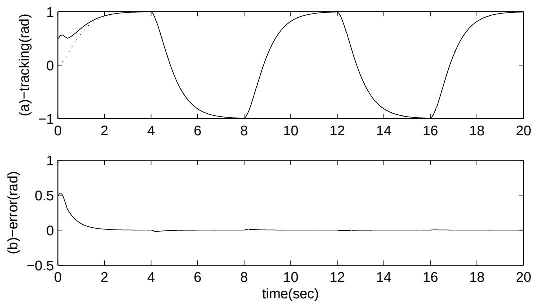

Figure 1: (a) Tracking and (b) errors of joint 1 in the revised Lin and Hsu's algorithm. Dash line:desired trajectory; solid line: actual trajectory.

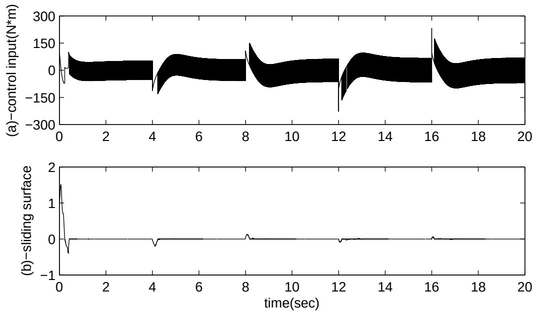

Figure 2: (a) Control input and (b) sliding surface of joint 1 in the revised Lin and Hsu's algorithm.

4 Stability issues for adaptive fuzzy sliding mode control algorithms as proposed in the literature

In this section, we briefly discuss some of the difficulties and drawbacks of the stability results achieved in much of the published literature on AFSMC.

During our investigation of the AFSMC algorithms proposed in the literature, a couple of key stability issues arose. One issue is associated with the need to know the bound in the universal approximation theorem in (7) and the other issue is based on the need to approximate the discontinuous sign function. The requirement to know the bound in (7) causes a practical problem in implementing and designing these controllers. The requirement to approximate the sign function results in a theoretical problem.

4.1 Case 1: Lyapunov stability in a practical sense

In Yoo and Ham (2000)'s method, to meet the requirement of Lyapunov stability, they need to choose a constant K_{Dj} to make $|s_j K_{Dj}| > |\omega_j| (s_j \neq 0)$ where ω_j is defined as the minimum approximation error. According to the universal approximation theorem in (7), there exists an optimal fuzzy system to estimate the given function and the approximation error ω_j is as small as possible. However, in real life, we can only define a fuzzy system with finite number of membership functions and fuzzy if-then rules. Therefore, we cannot find an optimal fuzzy system and the minimum approximation error ω_j which is as small as possible. The same situation happened in Guo and Woo (2003)'s algorithm where we need to choose a positive constant a_j

to make $a_j|s_j| > \gamma|s_j| \geq |\omega_j|(s_j \neq 0)$. Since ω_j can not be found in real applications, one cannot guarantee that the criterion $a_j|s_j| > |\omega_j|(s_j \neq 0)$ is satisfied.

Therefore, one must design the controller by simulation and experimentation trial and error. The algorithm proposed in this paper does not suffer from these practical problems.

4.2 Case 2: Problems on the proof of Lyapunov stability

To prove Lyapunov stability of the closed loop system, the following definitions are applied in the algorithms proposed by Wang et al. (2001) and Medhaffar et al. (2006).

In Wang et al. (2001)'s method, a fuzzy system $\hat{h}(s|\theta_h) = \theta_h^T\phi(s)$ is designed to estimate the switching control term u_{sw} and the optimal fuzzy system $\hat{h}(s|\theta_h^*)$ is defined as

$$\hat{h}(s|\theta_h^*) = (D + \eta_\Delta + \omega_{\text{max}})\text{sgn}(s) . \tag{41}$$

However, according to the universal approximation theorem in (7), we can only find an optimal fuzzy system to estimate any given real continuous function. Since the sign function in (41) is a discontinuous function, we cannot find an optimal fuzzy system $\hat{h}(s|\theta_h^*)$ to estimate the discontinuous function $(D + \eta_\Delta + \omega_{\text{max}})\text{sgn}(s)$.

In Medhaffar et al. (2006)'s algorithm, a fuzzy system $\hat{h}_i(s_i|\theta_{h_i}^*) = \theta_{h_i}^T\xi_{h_i}(\boldsymbol{s}_i)$ is designed to estimate the discontinuous function $(D_i + \eta_{i\Delta})\text{sgn}(s_i)$. Since the universal approximation theorem can only deal with the real continuous function, we cannot find the optimal fuzzy system $\hat{h}_i(s_i|\theta_{h_i}^*)$ based on (7).

Therefore, the above two algorithms have difficulty proving the convergence of the overall system based on the universal approximation theorem. The method proposed in this paper does not suffer from these theoretical difficulties.

5 Conclusion

Lin and Hsu (2004) created a methodology of learning both the premise and the consequence part of the fuzzy rules. Since this method is only designed for induction servomotor systems, we redesign this algorithm for robotic manipulators and the Lyapunov stability for the redesigned algorithm is proved. Simulation results demonstrate the effectiveness of the method. The Lyapunov stability in a practical application cannot be guaranteed in the algorithms proposed by Yoo and Ham (2000) and Guo and Woo (2003). According to the universal approximation theorem, a discontinuous function cannot be estimated by the fuzzy systems defined in the algorithms proposed by Wang et al. (2001) and Medhaffar et al. (2006).

References

Asada, H. and Slotine, J. J. (1986) *Robot Analysis and Control*, New York, NY: Wiley-Interscience.

Wang, L.X. (1997) *A Course in Fuzzy Systems and Control*, Upper Saddle River, NJ: Prentice Hall.

Yoo, B. and Ham, W. (1998) 'Adaptive fuzzy sliding mode control of nonlinear system', *IEEE Transactions on Fuzzy Systems*, Vol. 6, No. 2, pp.315–321.

Yoo, B. and Ham, W. (2000) 'Adaptive control of robot manipulator using fuzzy compensator', *IEEE Transactions on Fuzzy Systems*, Vol. 8, No. 2, pp.186–199.

Wang, J., Rad, A.B. and Chan, P.T. (2001) 'Indirect adaptive fuzzy sliding mode control: Part I: fuzzy switching', *Fuzzy Sets and Systems*, Vol. 122, No. 1, pp.21–30.

Guo, Y. and Woo, P.Y. (2003) 'An adaptive fuzzy sliding mode controller for robotic manipulators', *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, Vol. 33, No. 2, pp.149–160.

Ho, H.F., Wong, Y.K. and Rad, A.B. (2004) 'Adaptive fuzzy sliding mode control design: Lyapunov approach', *Proceedings of the 5th Asian Control Conference*, Melbourne, Australia, pp.1502–1507.

Wai, R.J., Lin, C.M. and Hsu, C.F. (2004) 'Adaptive fuzzy sliding-mode control for electrical servo drive', *Fuzzy sets and systems*, Vol. 143, No. 2, pp.295–310.

Lin, C.M. and Hsu, C.F. (2004) 'Adaptive fuzzy sliding mode control for induction servo motor systems', *IEEE Transactions on Energy Conversion*, Vol. 19, No. 2, pp.362–368.

Akbarzadeh-T, M.R. and Shahnazi, R. (2005) 'Direct adaptive fuzzy PI sliding mode control for a class of uncertain nonlinear systems', *IEEE International Conference on Systems, Man and Cybernetics*, Waikoloa, Hawaii, pp.2548–2553.

Shahnazi, R., Shanechi, H. and Pariz, N. (2006) 'Position Control of Induction and DC Servomotors: A Novel Adaptive Fuzzy PI Sliding Mode Control', *2006 IEEE Power Engineering Society General Meeting*, Montreal, Canada.

Medhaffar, H., Derbel, N. and Damak, T. (2006) 'A decoupled fuzzy indirect adaptive sliding mode controller with application to robot manipulator', *International Journal of Modelling, Identification and Control*, Vol. 1, No. 1, pp.23–29.

Wang, L.X. (1994) *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Englewood Cliffs, NJ: Prentice Hall.