ISTANBUL UNIVERSITY –
JOURNAL OF ELECTRICAL & ELECTRONICS ENGINEERING

YEAR : 2003 (645-661)
VOLUME : 3
NUMBER : 1

ADAPTIVE FUZZY SLIDING MODE CONTROL FOR A CLASS OF BIPARTITE MODULAR ROBOTIC SYSTEMS

Mehmet Önder EFE[1] Lemi Dağhan ACAY[2] Cem ÜNSAL[3]
Pradeep K. KHOSLA[4]

[1]Collaborative Center of Control Science, Electrical Engineering Department
The Ohio State University, Columbus, OH43210, U.S.A, onderefe@ieee.org

[2]Electrical and Electronics Engineering Department, Bogazici University
Bebek 80815, Istanbul, Turkey, acay@boun.edu.tr

[3]ATOGA Systems Inc., 49026 Milmont Drive, Fremont
CA 94538, U.S.A., unsal@ieee.org

[4]Electrical and Computer Engineering Department, Carnegie Mellon University
Pittsburgh, PA, 15213-3890, U.S.A., pkk@ece.cmu.edu

ABSTRACT

One of the fundamental issues in the field of modular robotics is the design and implementation of robotic systems with low cost and high performance. Although the coordination of modules entails the minimization of several different performance criteria, the success of the evolution of a group of the modules is strictly dependent upon the fulfillment of control goals. The cost of the modules, on the other hand, is subject to the components used to measure the state, and the hardware used for actuation. Therefore, obtaining a good control performance with cheap hardware is a challenge for control specific issues. In this paper, we describe an adaptive fuzzy sliding mode control scheme implemented on the control of 3-DOF I-Cubes links, which operate in a highly information-limited environment due to the size constraints, and which are bipartite. The tuning law is justified both in continuous time and in discrete time cases of sliding mode control approach. The implementation results justify the theoretical foundations and strongly recommend the approach due to its low computational cost together with the robustness against disturbances and uncertainties.

Keywords: *Tuning Laws, Fuzzy Control, Robotics, and Sliding Mode Control*

I. INTRODUCTION

The idea of modular robotic systems has been inspired from the biological counterparts, which can adapt themselves to form topologically different objects with a single and massively interconnected entity. Several examples can be visualized in the cell level biology of microorganisms. However, at a larger scale with man made machines, the concept of coordination and cooperation of individual cells depends heavily on the control of the interactions between

Received Date : 12.10.2002
Accepted Date: 19.11.2002

the modules. At this stage, the low level control performance enters into the picture and plays a central role in achieving the precise realization of the task. One challenge that constrains the design is the cost of the design. Since the design is subject to the individual control of a number of modules, and since the ultimate goal of the design is to obtain autonomous reconfiguration with low cost, it becomes inevitable to come up with robust control schemes that handle uncertainties arising in a highly information-limited environment. More explicitly, the controller for each actuator must be capable of alleviating the drawbacks brought about by considerably low sampling rate, noise, low encoder resolution, unknown and nonlinear friction issues in gears and dead zone issues. Obviously, in such an environment, the adopted controller structure must be simple enough in order not to violate the computational requirements and must be versatile enough to fulfill the desired task.

Various applications analyzing the capabilities of self-reconfigurable or modular robotic systems are reported in the literature. Early studies on this topic report manual configuration [1], a modularly synthesized kinematics structure called *Tetrabot* [2], and cellular mobile robots with reconfiguration capability [3]. Later studies have reported examples in 2D such as *metamorphing hexagonal modules* [4], *self-repairing machines* [5], the *Cristalline robot* [6], which moves in a horizontal plane, and *Inchworm* [7], which moves in a vertical plane. The examples operating in 3D are *Polypod/Polybot* [8], *CONRO* [9], *robotic molecule* [10], self-reconfigurable structure [11], modular robot [12], *Proteo* [13] and *I-Cubes* [14]. A common property of what is presented in [8-14] is the capability of exploiting neighboring modules to fulfill the task.

When the modularity is achieved through mechanical actuation based on motors, the issues related to the control engineering expertise have to be studied carefully. The designed hardware and the adopted control strategy must be in good compliance to achieve the overall tracking accuracy, which is a prerequisite for the success of self-reconfiguration trials. Since the nature of the problem is involved with the alleviation of a number of difficulties, which are frequently encountered in practice, the solution should be sought in the domain of intelligent control systems. In the literature, many architectures of intelligent control schemes have been discussed. Artificial neural networks, fuzzy inference systems and those utilizing genetic algorithms or the hybrid variants of these are just to name a few [15]. In this paper, we utilize an adaptive Fuzzy Logic Controller (FLC) processing the error and the estimated rate of error in discrete time and producing a control signal to drive the module under control towards a predefined regime.

In physical implementations of intelligent controllers, the issues of parameter tuning must be contemplated with particular care. The reason for this is the fact that the parameters of the controller are to be refined at each sampling instant but the update mechanism will need a critic information containing the amount of discrepancy on the applied control signal. Apparently, the nature of the control problems does not allow the existence of such kind of supervision; rather, the designer is forced to extract the control error from the observed quantities. One important result presented in this paper addresses this problem from an analytic point of view. Furthermore, it is shown that the system under control is driven towards a predefined sliding regime by the suggested tuning law and error measure.

Sliding Mode Control (SMC) is a well-known nonlinear control approach, which is based on a two-sided decision mechanism. The approach introduces certain degrees of robustness against disturbances and uncertainties due to the stable and attracting invariant loci created in the phase space. The underlying idea is to force the tracking error vector towards the attracting loci, which is the switching hyperplane, and to maintain the motion on this multidimensional surface. The control strategy has extensively been studied for tracking control of uncertain systems, whose nominal governing dynamics are known with the bounds of uncertainties. A thorough investigation of the concept is presented in [17-19], and the relevance of the technique with intelligent control schemes is figured out in [20] from the point of how SMC could be used for intelligent control and how intelligence could be used to improve the performance of SMC schemes.

Although the SMC concept has been exemplified in various applications, when the discrete time implementations are considered, the results integ-rating the issues of intelligence and control in the context of a rigorous stability analysis are still in their way towards maturity. Several works focusing on the Discrete Time Sliding Mode Control (DTSMC) approach report the necessary and sufficient conditions for stability [21], use of reaching law approach in discrete time [22], convergence issues in quasi-sliding mode [23], and selection of the sampling time [24]. One of the examples demonstrating the feasibility and efficacy of such approaches in the field of DTSMC focuses on an exhaust measuring system [25], which introduces the use of a fuzzy supervisor utilizing triangular membership functions. The work presented by Xu *et al* [26] adopts a hybrid approach based on neuro-fuzzy integration, by the use of which the DTSMC task is achieved by tuning the parameters of a neuro-fuzzy identifier. The tuning is performed through error backpropagation technique, and the results have been discussed for computer control of a two link robotic manipulator. In [27] and [28], the design of DTSMC with recurrent neural networks and Gaussian radial basis function neural networks is presented respectively. In both studies, the neural networks are utilized for estimation purposes.

In the next section, we introduce the *I-Cubes* system and the control loop, next the fuzzy controller and its adaptation scheme is presented. In the fourth section, we present the implementation results, and the concluding remarks are given at the end of the paper.

2. I-CUBES

In this paper, we consider a class of self-reconfigurable modular robots to demonstrate the control performance of the proposed approach. Previous research in *I-Cubes* project have demonstrated that the concept of self-reconfiguration could be attained autonomously through a self decomposition of the task into subtasks and executing a planning strategy based on the extracted decisions in software domain [14].

I-Cubes system discussed in [14,29-31] is a bipartite robotic system composed of cubes forming a lattice and links (3-DOF manipulators) providing connections between the lattice elements. The components and the different CAD views of an *I-Cubes* link module are shown in Fig.1. In Fig. 1(a) the motor used for each joint is shown. The motor is Cirrus CS 21-BB Hi Performance Sub-Micro BB Servo. Its specifications are as follows: Size (cm) 2.1844× 1.9812×1.0922; Weight (gr) 9.0651; Speed (sec/60º) 0.09 under 4.8V, 0.078 under 6.0V; Torque (gr-cm) 1367.1387 under 4.8 V, 1654.9573 under 6.0 V. Fig. 1 (b) demonstrates how the motor is installed at the end of a single link. In Fig. 1 (c) and (d), the gearing that transfers rotary the motion of the motor to the link end is shown. A compact view is illustrated in Fig. 1(e) with encoders and two motors at the link ends and one motor in the middle of the whole system. Since the used encoders are 24-CPR and the worm gear reduction is 1:30, the resolution of the encoder feedback is 360º/(30×24)=0.5º. The cross-shaped connectors at the link ends are designed for attachment and detachment and are not under investigation in this paper. The last subplot of Fig. 1 demonstrates a single module of *I-Cubes* system used in the tests. As depicted in Fig. 1(e) and (f), the system has two links, each having six centimeters length, and three actuators to provide mobility in 3D space.

The block structure of the controlled system is depicted in Fig. 2, in which each motor is operating under an ordinary feedback loop. At each control instant, PIC sends 3 bytes to the computer. Two bytes of this information hold the measured position and the remaining byte is for the speed in terms of pulses and pulses/sec respectively. On the other hand, the computer sends two bytes to the PIC. First byte is for the direction (or sign) and the other is for PWM (Pulse Width Modulation) duty-cycle. The described process is achieved through a serial communication between the computer and the PIC. The two systems are synchronized at a rate of 160 msec (T_s), which is the adopted value of the control period. The process that takes place inside the computer include the interfacing, that has been designed on Java 2 platform, the control algorithm and parameter tuning scheme yet to be discussed, and the serial communication software. The first apparent challenge here is the width of control period.

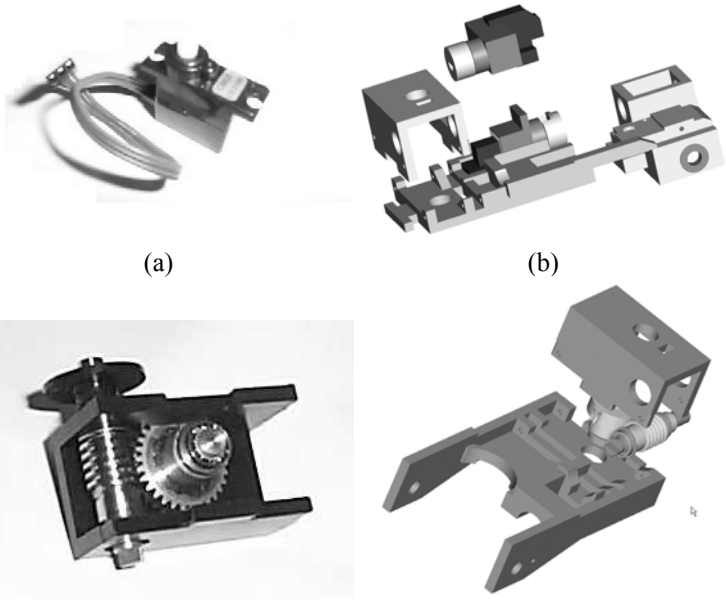
The system will be forced to find the desired state through a set of observations that are sparse in time. One might suggest increasing the frequency, but the practical difficulty is that the optical encoder does not provide sufficiently accurate readings when the sampling frequency is increased, and the observed positional data decreases to the order of a few pulses, which lead to an excessive noise when the speed is evaluated. On the other hand, changing the encoder and the relevant components with those providing better accuracy conflicts with the cost and size objectives of the I-Cubes system. In what follows, we describe the controller and the parameter tuning mechanism.

3. FUZZY LOGIC CONTROLLER AND THE ADAPTATION MECHANISM

3.1. The Controller

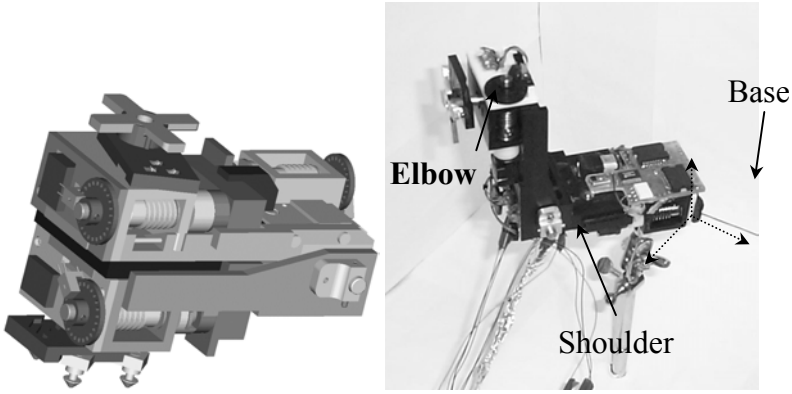
Contrary to what is postulated in the realm of predicate logic, representation of knowledge by fuzzy quantities can provide extensive degrees of freedom if the task to be achieved can better be expressed in words than in numbers. The concept of fuzzy logic in this sense can be viewed as a generalization of binary logic and refers to the manipulation of knowledge with sets, whose boundaries are unsharp [32]. Therefore the paradigm offers a possibility of designing intelligent controllers operating in an environment, in which the conditions are inextricably intertwined and subject to uncertainties and impreciseness.

Understanding the information content of fuzzy logic systems is based on the subjective judgements, intuitions and the experience of an expert. From this point of view, a suitable way of expressing the expert knowledge is the use of IF antecedent THEN consequent rules, which can easily evaluate the necessary action to be executed for the current state of the system under investigation.



(a) (b)

(c) (d)



(e) (f)

Fig. 1. The components and the physical allocation

Structurally, a FLC is comprised of five building blocks, namely, fuzzification, inference engine, knowledge base, rule base, and defuzzification. Since the philosophy of the fuzzy systems is based on the representation of knowledge in a fuzzy domain, the variables of interest are graded first. This grading is performed through the evaluation of membership values of each input variable in terms of several class definitions. According to the definition of a membership function, how the degree of confidence changes over the domain of interest is characterized. This grading procedure is called fuzzification. In the knowledge base, the parameters of membership functions are stored. Rule base contains the cases likely to happen, and the corresponding actions for those cases through linguistic descriptions, i.e. the IF-THEN statements. The inference engine emulates the expert's decision making in interpreting and applying knowledge about how the best fulfillment of the task is achieved. Finally, the defuzzifier converts the fuzzy decisions back onto the crisp domain [23].

The architecture utilized in this paper uses algebraic product operator for the aggregation of the rule premises and triangular membership functions denoted by μ. The overall representation of this structure is given in (1), in which R and m stand for the number of rules contained in the rule base and the number of inputs of the structure. The output of the FLC is evaluated by using weighted average defuzzification scheme. Throughout the paper, an underlined variable should be understood as a vector of appropriate dimensions.

$$u = \sum_{i=1}^{R} f_i \left(\frac{\prod_{j=1}^{m} \mu_{ij}(v_j)}{\sum_{l=1}^{R} \prod_{j=1}^{m} \mu_{lj}(v_j)} \right) = \underline{\phi}^{\mathrm{T}} \underline{\Omega} \qquad (1)$$

with i^{th} rule as: IF v_1 is $\exists_{i1}$ AND v_2 is $\exists_{i2}$ AND … AND v_m is $\exists_{im}$ THEN $f_i=\phi_i$. In the IF part of this representation, the lowercase variables denote the inputs and $\exists_{ij}$s stand for the fuzzy sets corresponding to the domain of each linguistic label. The THEN part is comprised of the prescribed decision, which is denoted by f_i, in the form of a scalar number denoted by ϕ_i. Generically, the adjustable parameters of the structure are comprised of the parameters of the membership functions together with the defuzzifier parameters ϕ_i. Since $\underline{\Omega}$ is the vector of fuzzy basis functions, an important feature of the representation in (1) is the linearity of the output in the defuzzifier parameters. In Fig. 3, (to which more reference will be made at a later stage, explaining the notations used on the figure) how the input space is covered by the membership functions is depicted.

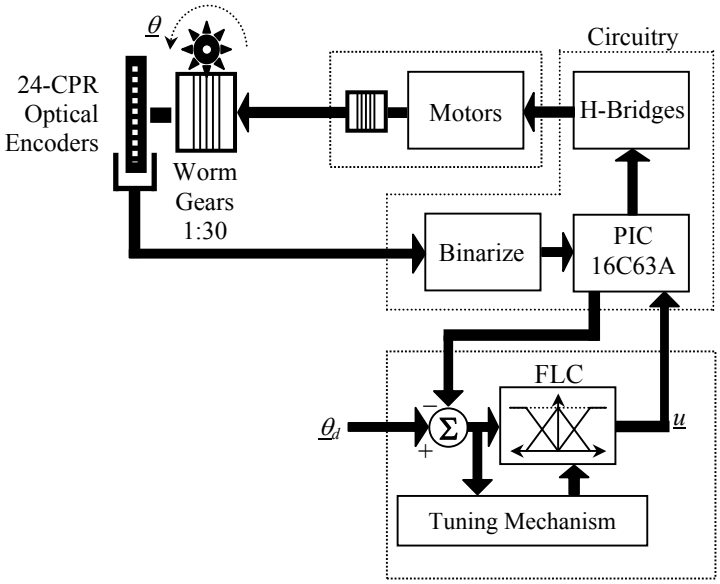


Fig. 2 Control system structure

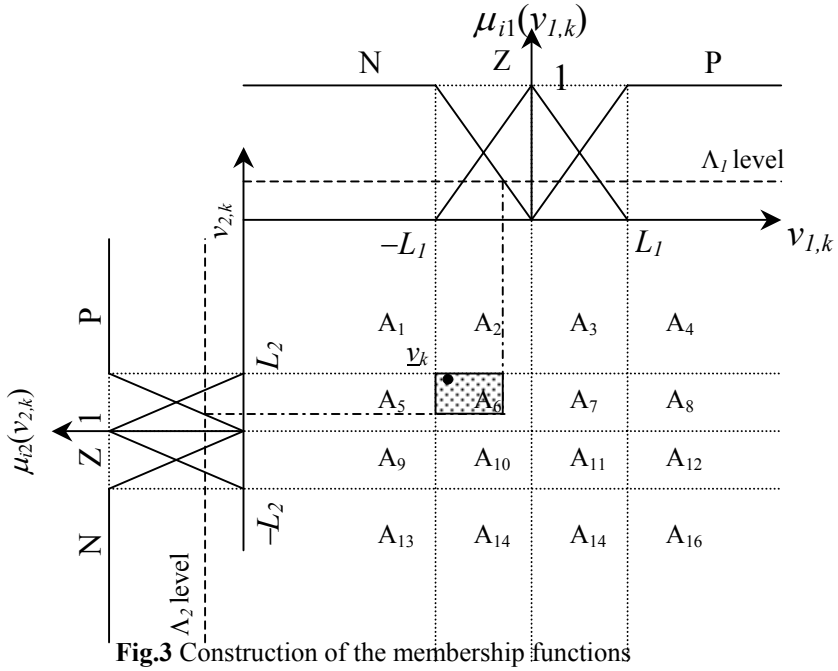


Fig.3 Construction of the membership functions

In this paper, each one of the motors in a single module is controlled by a separate FLC of structure given in (1). Since the controllers for all three motors have the same structure, we skip subsystem indexing in order to avoid confusion. The inputs to a FLC are the tracking error and its rate, i.e. $v_1=e$ and $v_2=\dot{e}$ measured at discrete instants of time. The inputs are in encoder pulses for errors and pulses per second for rates of errors, whereas the output of each FLC is a crisp integer number determining the duty-cycle of the Pulse Width Modulation (PWM) pulses to be produced by the PIC 16C63A shown in Fig. 2.

3.2. Tuning Mechanism for the Motors of an *I-Cubes* Module

Consider the system given as $\ddot{\underline{x}}(t) = \underline{\Psi}(\underline{x}(t), \dot{\underline{x}}(t)) + \underline{\Phi}(\underline{x}(t))\underline{u}(t)$, where $\underline{x}=(\theta_1, \theta_2, \theta_3)^T$ denote the angular position of each motor, and $\underline{u}=(u_1,u_2,u_3)^T$ is the vector of inputs for the three motors. Defining $\underline{\omega}=(\omega_1, \omega_2, \omega_3)^T$ as the angular velocities, and assuming the coupling effects as functions of time, the governing dynamics of each one of the actuators in the module resembles to the one given in (2).

$$\dot{\theta}(t) = \omega(t) \text{ and } \dot{\omega}(t) = P(\theta(t), \omega(t), t) + Q(t)u(t) \quad (2)$$

Let us drop the time variable and denote the desired angular positional behavior by θ_d, which is the command signal. The goal of the design is to force the error dynamics towards the sliding manifold described by $s_p = 0$, where $s_p = \dot{e} + \lambda e$, $e=\theta_d - \theta$, and $-\lambda$ is the slope of the sliding manifold. Obviously if the functions P and Q were known exactly, or the bounds of the uncertainties on these functions were known, one would suggest the sequence given as $u_{smc} = Q^{-1}\left(\dot{\omega}_d + \lambda\dot{e} + \alpha\,\mathrm{sgn}(s_p) + \beta\, s_p - P\right)$, with $\lambda > 0$, $\alpha > 0$ and $\beta > 0$. The formulated sequence enforces $\dot{s}_p = -\alpha\,\mathrm{sgn}(s_p) - \beta\, s_p$, which corres-ponds to constant plus proportional rate reaching law of SMC terminology [17]. It should be clear that the application of u_{smc} would enforce the error vector to hit the sliding manifold. The behavior during this phase is characterized by $\dot{s}_p = -\alpha\,\mathrm{sgn}(s_p) - \beta\, s_p$. Once the error vector gets trapped to the subspace characterized by $s_p = 0$, the behavior imposed along the locus $s_p = 0$ would bring the error vector towards the origin of the phase space.

Definition 1. The error on the control signal is defined to be the discrepancy between u_{smc} and the produced control (u). This quantity can explicitly be defined as $s_c = u_{smc} - u$.

Proposition 2. The error caused by the controller, i.e. $s_c \triangleq u_{smc} - u$, can be estimated as $s_c = \dot{s}_p + \alpha\,\mathrm{sgn}(s_p) + \beta\, s_p$, the minimization of which in magnitude ensures reaching and convergence to origin due to the dynamics of the sliding manifold.

The above proposition suggests that the tuning of the controller parameters are performed only if the error vector is not on the sliding manifold. Consider the two surfaces shown in Fig. 4. Most of the tuning schemes define the origin as the point at which the cost is minimal (See Fig 4(a)). However, the selection in the above proposition augments the set of points that minimize the cost as illustrated in Fig 4(b), hence, the convergence does not take a long time as in the case of traditional tuning schemes.

Remark 3. In this study, there are three FLCs, i.e. one for each motor, and the vector of the defuzzifier parameters ($\underline{\phi}$) of them are adjusted individually. In order not to be in conflict with the physical reality, for each FLC, it is assumed that the adjustable parameters, the time derivative of the signal exciting the controller and the time derivative of the target output of the controller remain bounded, i.e. $\|\underline{\phi}\| \le B_\phi$, $\|\dot{\underline{\Omega}}\| \le B_{\dot{\Omega}}$ and $\|\dot{u}_{smc}\| \le B_{\dot{u}_{smc}}$.

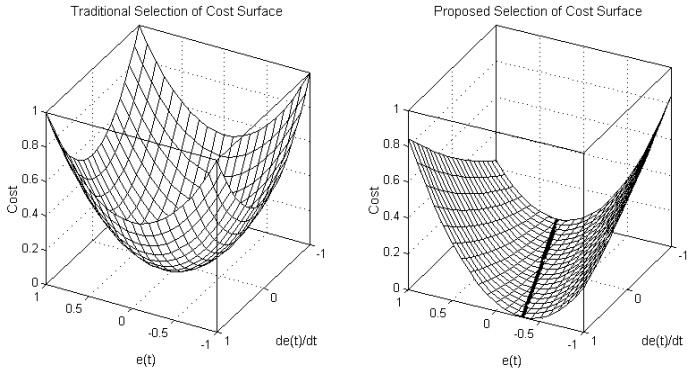


Fig. 4 Interpretation of the cost surface for proposed error measure

Theorem 4. For a system of structure (2) and a FLC of structure (1), the adaptation of controller parameters as described in (3) enforces the value of the control discrepancy (s_c) to zero.

$$\dot{\underline{\phi}} = \frac{\underline{\underline{\Omega}}}{\underline{\Omega}^{\mathrm{T}}\underline{\Omega}}\gamma\,\mathrm{sgn}(s_c) \tag{3}$$

where, γ is a sufficiently large positive constant satisfying

$$\gamma > B_{\phi}B_{\dot{\Omega}} + B_{\dot{u}_{smc}} \tag{4}$$

The adaptation mechanism in (3) drives an arbitrary initial value of s_c to zero in finite time denoted by t_h satisfying the inequality in (5).

$$t_h \le \frac{|s_c(0)|}{\gamma - (B_{\phi}B_{\dot{\Omega}} + B_{\dot{u}_{smc}})} \tag{5}$$

Proof: The proof of a similar case is presented in [34], which assumes the availability of the target outputs. □

Theorem 5: If the system enters the sliding mode $s_c = 0$ and remains in it thereafter, then the parameters of the *FLC*, $\underline{\phi}$, evolve bounded.

Proof: Refer to [35]. □

3.3. Analysis of the Scheme from the DTSMC Perspective

The update law of (3) can be discretized by using first order Euler approximation and the stability issues can formally be analyzed as given next. The discrete time index is shown as a subscript with integer variable *k*.

Assumption 6. There exists a number Γ such that $(\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1})/(\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k) > \Gamma > 0$ is satisfied in $\Re^m$, where *m* is the number of FLC inputs and is equal to 2. The meaning of this and the conditions under which the above inequality holds true will be discussed later.

Theorem 7. For a discrete time system of structure (6), the use of a two input one output FLC described in (1) with a parameter adaptation rule as described in (7) leads to $s_{c_k}(s_{c_{k+1}} - s_{c_k}) < 0$, and the system is driven towards the quasi-sliding regime characterized by (8).

$$\begin{aligned}\theta_{k+1} &= \theta_k + T_s\omega_k \text{ and}\\ \omega_{k+1} &= \omega_k + T_s(P_k(\theta_k,\omega_k,k) + Q_k(k)u_k(k))\end{aligned} \tag{6}$$

$$\underline{\phi}_{k+1} = \underline{\phi}_k + T_s\frac{\underline{\underline{\Omega}}_k}{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k}\gamma\,\mathrm{sgn}(s_{c_k}) \tag{7}$$

$$s_{p_{k+1}} = s_{p_k} - T_s(\alpha\,\mathrm{sgn}(s_{p_k}) + \beta\, s_{p_k}) \tag{8}$$

where T_s is the sampling rate and γ is constrained to satisfy $\gamma\ T_s > \zeta/\Gamma$ with $\zeta = 2(B_{\phi}B_{\Omega} + B_{u_{smc}})$.

Proof: Define $\underline{\Omega}_{k+1} = \underline{\Omega}_k + \underline{\Delta\Omega}_{k+1}$ and $u_{smc,k+1} = u_{smc,k} + \Delta u_{smc,k+1}$. It should be clear that $\|\underline{\Delta\Omega}_k\| \le 2B_{\Omega}$ and $|\Delta u_{smc,k}| \le 2B_{u_{smc}}$ for ∀k≥0, where B_{Ω} and $B_{u_{smc}}$ are some positive constants satisfying $\|\underline{\Omega}_k\| \le B_{\Omega}$ and $|u_{smc,k}| \le B_{u_{smc}}$ for ∀k≥0 respectively. Furthermore, the parameters of the FLC are assumed to be bounded, i.e. $\|\underline{\phi}_k\| \le B_{\phi}$ with $B_{\phi} > 0$. Using these quantities, and referring to the definition and estimation of s_c in Proposition 2 with (1), one can proceed as follows:

As long as the inequality in Assumption 6 holds true, the obtained result implies that the adaptation mechanism enforces the FLC to synthesize u_{smc} which forces $s_c \to 0$ or equivalently the reaching law of (8), and leads to the achievement of the prescribed DTSMC task, and the theorem is proved. □

$$s_{c_k}\left(s_{c_{k+1}}-s_{c_k}\right)=s_{c_k}\left(u_{smc,k+1}-\underline{\phi}_{k+1}^{\mathrm{T}}\underline{\Omega}_{k+1}-s_{c_k}\right.$$

$$=s_{c_k}\left(u_{smc,k+1}-\underline{\phi}_k^{\mathrm{T}}\underline{\Omega}_k-\underline{\phi}_k^{\mathrm{T}}\underline{\Delta\Omega}_{k+1}\right)$$

$$-s_{c_k}\left(\gamma T_s\frac{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}}{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k}\mathrm{sgn}\left(s_{c_k}\right)+s_{c_k}\right)$$

$$=s_{c_k}\left(u_{smc,k+1}-u_{smc,k}+u_{smc,k}-\underline{\phi}_k^{\mathrm{T}}\underline{\Omega}_k\right)$$

$$-s_{c_k}\left(\underline{\phi}_k^{\mathrm{T}}\underline{\Delta\Omega}_{k+1}+\gamma T_s\frac{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}}{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k}\mathrm{sgn}\left(s_{c_k}\right)+s_{c_k}\right)$$

$$=s_{c_k}\left(\Delta u_{smc,k+1}-\underline{\phi}_k^{\mathrm{T}}\underline{\Delta\Omega}_{k+1}-\gamma T_s\frac{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}}{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k}\mathrm{sgn}\left(s_{c_k}\right)\right) \quad (9)$$

$$=-\gamma T_s\frac{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}}{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k}\left|s_{c_k}\right|+s_{c_k}\left(\Delta u_{smc,k+1}-\underline{\phi}_k^{\mathrm{T}}\underline{\Delta\zeta}\right.$$

$$\le-\gamma T_s\frac{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}}{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k}\left|s_{c_k}\right|+2\left(B_\phi B_\Omega+B_{u_{smc}}\right)\left|s_c\right.$$

$$\le-\left(\gamma T_s\frac{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}}{\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k}-\zeta\right)\left|s_{c_k}\right|$$

$$<-\left(\gamma T_s\Gamma-\zeta\right)\left|s_{c_k}\right|<0$$

Theorem 8. There exists a strictly positive Γ if the motion in the 2-dimensional input space of the FLC satisfies the conditions in (10) and (11).

$$\left|\mu_{i1}\left(v_{1,k+1}\right)-\mu_{i1}\left(v_{1,k}\right)\right|\le 1-\Lambda_1 \quad (10)$$

$$\left|\mu_{i2}\left(v_{2,k+1}\right)-\mu_{i2}\left(v_{2,k}\right)\right|\le 1-\Lambda_2 \quad (11)$$

where, $0<\Lambda_1<1$ and $0<\Lambda_2<1$.

Proof: Since $\gamma T_s > \zeta / \Gamma$ and we assumed that $(\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1})/(\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k)>\Gamma>0$ (Refer to Assumption 6), what we need is to evaluate the least value of $(\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1})/(\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k)$ and to show that it is strictly positive. Before going into the details, one should notice from (10)-(11) that a binary change in any of the membership functions is prohibited. For example, if $\mu_{i1}(v_{1,k})=1$ for some k, the value of $\mu_{i1}(v_{1,k+1})$ can decrease at most to the level $1-\Lambda_1$. Referring to Fig. 3, let the input vector perform a transition from region A_1 at time k to region A_6, at time $k+1$, and denote this transition by $A_1 \rightarrow A_6$. Clearly, the conditions in (10)-(11) require that the point $\underline{v}_k$ in Fig. 3 can reach points in the shaded area at time $k+1$, and this area is the largest area that can be reached from the region A_1 due to the conditions in (10)-(11). Having this in mind, we can claim that the least value of $\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}$ that can be observed from $A_1 \rightarrow A_6$ transition is $\Lambda_1\Lambda_2$. In obtaining this, one should note that it is sufficient to check the least value of $\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}$ because the supremum value of $\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_k$ is unity. Once the minimal least value of $\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}$ for all possible transitions is constructed, existence of a strictly positive Γ value can be claimed if the globally minimum value of $\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}$ is strictly positive. For this reason, consider the data given in Table 1, in which we summarize the results for all possible transitions. One should notice that there are equivalent transitions leading to the same least value. These are given in the third column of the table. In analyzing the results we enumerate the rules as follows: {Rule#1:NN, Rule#2:NZ, Rule#3:NP, Rule#4:ZN, Rule#5:ZZ, Rule#6:ZP, Rule#7:PN, Rule#8:PZ, Rule#9:PP}. Among what is given in the fourth column of Table 1, the transitions $A_6 \rightarrow A_6$, $A_7 \rightarrow A_7$, $A_{10} \rightarrow A_{10}$, and $A_{11} \rightarrow A_{11}$ are the most difficult ones as they require the evaluation of four rule outputs. For example in $A_6 \rightarrow A_6$ transition we get

$$\begin{aligned}\underline{\Omega}_k^{\mathrm{T}}\underline{\Omega}_{k+1}&=\left(2v_{1,k}v_{1,k+1}+\left(v_{1,k}+v_{1,k+1}\right)L_1+L_1^2\right)*\\&\quad\left(2v_{2,k}v_{2,k+1}-\left(v_{2,k}+v_{2,k+1}\right)L_2+L_2^2\right)/\left(L_1^2L_2^2\right)\\&=\left(L_1L_2\right)^{-2}F\left(v_{1,k},v_{1,k+1}\right)G\left(v_{2,k},v_{2,k+1}\right)\\&\ge\left(L_1L_2\right)^{-2}\inf_{v_{1,k};v_{1,k+1}\in A_6}F\left(v_{1,k},v_{1,k+1}\right)*\\&\quad\inf_{v_{2,k};v_{2,k+1}\in A_6}G\left(v_{2,k},v_{2,k+1}\right)\end{aligned} \quad (12)$$

which is to be evaluated over a domain satisfying the following conditions:

i) $v_{1,k}, v_{1,k+1} \in [-L_1, 0]$

ii) $v_{2,k}, v_{2,k+1} \in [0, L_2]$

iii) $\left|v_{1,k} - v_{1,k+1}\right| \le (1-\Lambda_1)L_1$

iv) $\left|v_{2,k} - v_{2,k+1}\right| \le (1-\Lambda_2)L_2$

In Fig. 5, we portray the hexagonal domains over which the inframum values are to be evaluated. More explicitly, $\inf_{v_{1,k}; v_{1,k+1} \in A_6} F(v_{1,k}, v_{1,k+1})$ is evaluated over the hexagon $O\sigma_1\sigma_2\sigma_3\sigma_4\sigma_5$ and $\inf_{v_{2,k}; v_{2,k+1} \in A_6} G(v_{2,k}, v_{2,k+1})$ is evaluated over $O\sigma_6\sigma_7\sigma_8\sigma_9\sigma_{10}$. The function F takes it minimal values at the centers of the line segments σ_1-σ_2 and σ_4-σ_5, and the function G takes its minimal values at the centers of the line segments σ_6-σ_7 and σ_9-σ_{10}. When the corresponding values are solved, the least value of $\underline{\Omega}_k^T \underline{\Omega}_{k+1}$ is obtained as given in $A_6 \rightarrow A_6$ row of Table 1. In finding the expressions in the fourth column of Table1, one should keep in mind that $\sum_{l=1}^{R} \prod_{j=1}^{m} \mu_{lj}(v_j) = 1$ due to the selection of membership functions depicted in Fig. 3. In accordance with this, the entries of $\underline{\Omega}_k$ vector contain the multiplication of relevant membership grades for the two inputs.

If the fourth column of the table is combined, we get the final result given below:

$$\frac{\underline{\Omega}_k^T \underline{\Omega}_{k+1}}{\underline{\Omega}_k^T \underline{\Omega}_k} > \Lambda_1\Lambda_2 \min\left(\left(1-\frac{\Lambda_1}{2}\right)\left(1-\frac{\Lambda_2}{2}\right); \Lambda_1\Lambda_2\right) \quad (13)$$

The result above ensures that a $\Gamma > 0$ exists in $\Re^2$ (or in $v_{1,k} \times v_{2,k}$ for $k \ge 0$) and it satisfies the inequality in (14).

$$0 < \Gamma < \Lambda_1\Lambda_2 \min\left(\left(1-\frac{\Lambda_1}{2}\right)\left(1-\frac{\Lambda_2}{2}\right); \Lambda_1\Lambda_2\right) \quad (14)$$

The result given above proves Theorem 8 and confirms the convergence claim of Theorem 7. □

Remark 9. A system of structure (2) in continuous time, or equivalently of structure (6) in discrete time with a FLC of input-output relation (1) with membership functions set as depicted in Fig. 3 can be driven towards the predefined sliding regime by tuning the defuzzifier parameters through the adoption of (3) for continuous time realizations, or (7) for sampled data systems.

Considering the inequality in (4) and the last line of (9), one can explicitly suggest the following relation for uncertainty bound parameter.

$$\gamma > \max\left(B_\phi B_{\dot{\Omega}} + B_{\dot{u}_{smc}}; \frac{\zeta}{\Gamma T_s}\right) \quad (15)$$

The relation above merges the results of continuous time case and discrete time case.

4. EXPERIMENTATION AND OBSERVATIONS

In this section, we discuss the results obtained through real-time experimentation of the I-Cubes module. Several difficulties related to the experimentation are emphasized, and the way in which the high performance with low cost is observed is clarified.

The first issue in the implementation stage is the time sparseness of the observed data and the necessity of filtering. Denote the Laplace variable by S and discretize the filter structure H(S)=□/(S+□) by forward Euler approximation. During the tests, one major problem alleviated by such a simple filtering is the noise contained in the calculated angular velocities, and the evaluated FLC outputs.

Table 1. Possible Regional Transitions and the Least Value of $\underline{\Omega}_k^{\mathrm{T}} \underline{\Omega}_{k+1}$

Regional Transition	Contributing Rule(s)	Equivalent Transitions	Least value of $\underline{\Omega}_k^{\mathrm{T}} \underline{\Omega}_{k+1}$
$A_1 \rightarrow A_1$	3	$A_4 \rightarrow A_4$, $A_{13} \rightarrow A_{13}$, $A_{16} \rightarrow A_{16}$	1
$A_1 \leftrightarrow A_2$	3	$A_3 \leftrightarrow A_4$, $A_{13} \leftrightarrow A_{14}$, $A_{15} \leftrightarrow A_{16}$	Λ_1
$A_1 \leftrightarrow A_5$	3	$A_9 \leftrightarrow A_{13}$, $A_4 \leftrightarrow A_8$, $A_{12} \leftrightarrow A_{16}$	Λ_2
$A_1 \leftrightarrow A_6$	3	$A_4 \leftrightarrow A_7$, $A_{10} \leftrightarrow A_{13}$, $A_{11} \leftrightarrow A_{16}$	$\Lambda_1\Lambda_2$
$A_2 \rightarrow A_2$	3,6	$A_3 \rightarrow A_3$, $A_{14} \rightarrow A_{14}$, $A_{15} \rightarrow A_{15}$	1
$A_2 \leftrightarrow A_3$	6	$A_{14} \leftrightarrow A_{15}$	Λ_1^2
$A_2 \leftrightarrow A_5$	3	$A_3 \leftrightarrow A_8$, $A_9 \leftrightarrow A_{14}$, $A_{12} \leftrightarrow A_{15}$	$\Lambda_1\Lambda_2$
$A_2 \leftrightarrow A_6$	3,6	$A_3 \leftrightarrow A_7$, $A_{10} \leftrightarrow A_{14}$, $A_{11} \leftrightarrow A_{15}$	$2\Lambda_1^2\Lambda_2$
$A_2 \leftrightarrow A_7$	6	$A_3 \leftrightarrow A_6$, $A_{10} \leftrightarrow A_{15}$, $A_{11} \leftrightarrow A_{14}$	$\Lambda_1^2\Lambda_2$
$A_5 \rightarrow A_5$	2,3	$A_8 \rightarrow A_8$, $A_9 \rightarrow A_9$, $A_{12} \rightarrow A_{12}$	1
$A_5 \leftrightarrow A_6$	2,3	$A_7 \leftrightarrow A_8$, $A_9 \leftrightarrow A_{10}$, $A_{11} \leftrightarrow A_{12}$	$2\Lambda_1\Lambda_2^2$
$A_5 \leftrightarrow A_9$	2	$A_8 \leftrightarrow A_{12}$	Λ_2^2
$A_5 \leftrightarrow A_{10}$	2	$A_6 \leftrightarrow A_9$, $A_7 \leftrightarrow A_{12}$, $A_8 \leftrightarrow A_{11}$	$\Lambda_1\Lambda_2^2$
$A_6 \rightarrow A_6$	2,3,5,6	$A_7 \rightarrow A_7$, $A_{10} \rightarrow A_{10}$, $A_{11} \rightarrow A_{11}$	$\Lambda_1\Lambda_2(1\text{-}0.5\,\Lambda_1)(1\text{-}0.5\,\Lambda_2)$
$A_6 \leftrightarrow A_7$	5,6	$A_{10} \leftrightarrow A_{11}$	$2\Lambda_1^2\Lambda_2^2$
$A_6 \leftrightarrow A_{10}$	2,5	$A_7 \leftrightarrow A_{11}$	$2\Lambda_1^2\Lambda_2^2$
$A_6 \leftrightarrow A_{11}$	5	$A_7 \leftrightarrow A_{10}$	$\Lambda_1^2\Lambda_2^2$

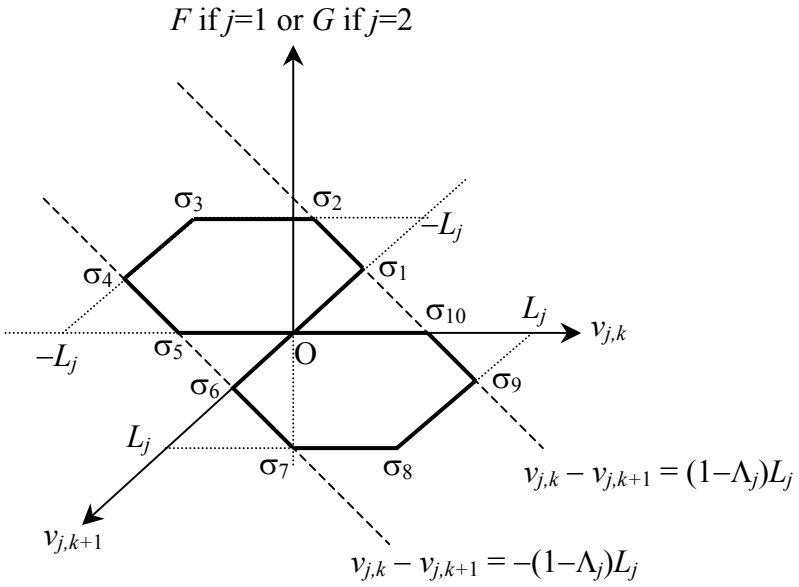


Fig. 5 Regions of Interest for $A_6 \rightarrow A_6$ Transition or its Equivalents

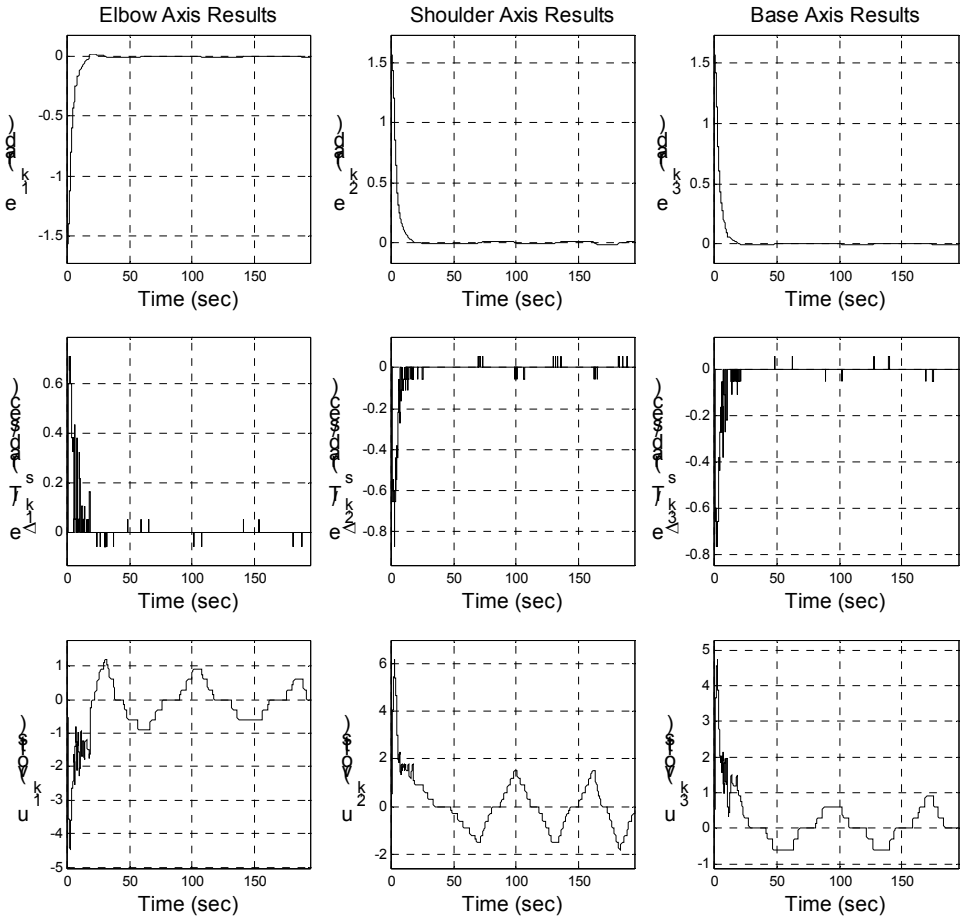


Fig. 6 Tracking errors and applied control signals

Table 2. Real-Time Experimentation Parameters for All Three FLCs

α	Reaching Law Parameter	1
β	Reaching Law Parameter	5.9375
γ	Uncertainty Parameter	4
λ	Slope parameter of the Sliding Manifold	0.3
τ_u	FLC Output Filtering	3.125
τ_v	FLC Velocity Input Filtering	3.125
L_1	FLC parameter (pulses)	100 (5π/36 rad)
L_2	FLC parameter (pulses/sec)	20 (π/36 rad/sec)
u_{max}	Maximum applicable input value (volts)	6
u_{min}	Minimum applicable input value (volts)	-6

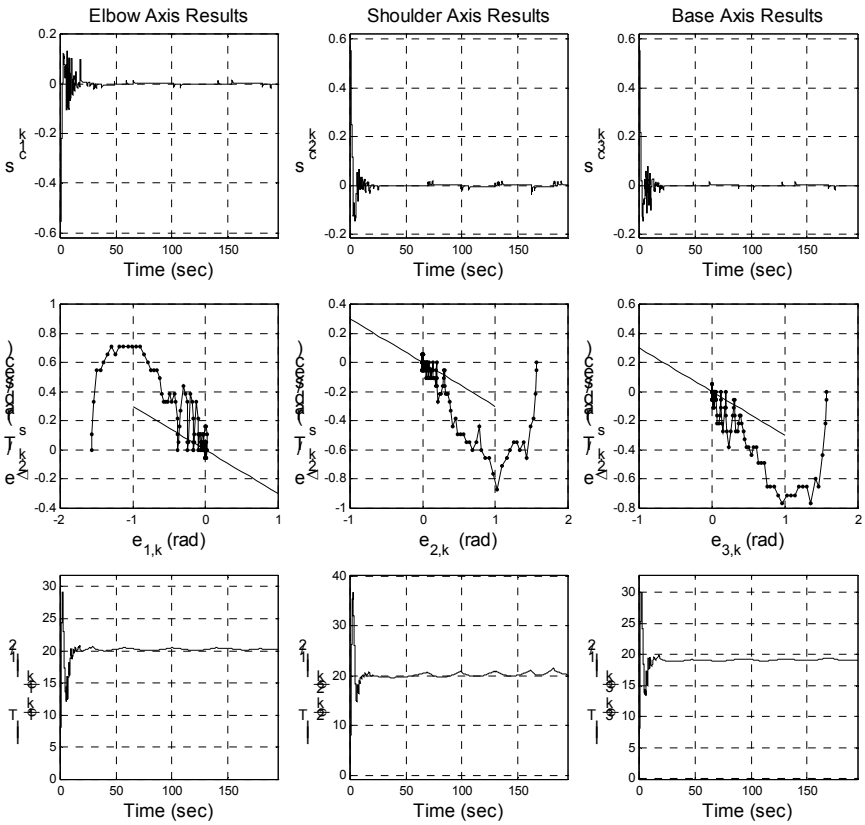


Fig. 7 Extracted control signal errors, phase space behaviors and parametric evolutions

For this purpose, we filter the velocity error before applying it to the FLCs and the output voltages before driving the motors. The filter structures for the velocity error and the control input are described in (16) and (17) respectively.

$$\hat{v}_{2,k+1} = (1 - \tau_v T_s)\hat{v}_{2,k} + (\tau_v T_s) v_{2,k} \text{ with } \hat{v}_{2,0} = 0 \quad (16)$$

$$\hat{u}_{k+1} = (1 - \tau_u T_s)\hat{u}_k + (\tau_u T_s) u_k \text{ with } \hat{u}_0 = 0 \quad (17)$$

where τ_v and τ_u denote the corresponding parameter of $H(S)$. The used values of these variables and the other parameters of the implementation are tabulated in Table 2. For each one of the motors in a single module, a dedicated FLC is utilized, the membership functions are kept static and only the parameters of the defuzzifiers (ϕ) are adjusted.

Initially, the motors are at rest, i.e. at zero angle and at zero angular velocity for each, and the desired angular position for all three of them has been set as $\pi/2$ radians. The rationale behind this selection is due to the philosophy of *I-Cubes* project, which presumes that each actuator performs $\pi/2$ radians [30-31], so that at a larger scale, the motion planning algorithm could exploit the information coming from the lowest level of the design, and this is strictly dependent upon the control performance. Although the motion of multiple modules is possible, the application of *I-Cubes* constrains the link level motion to one-at-a-time basis for each one of the motors. In order to demonstrate the performance of the proposed control scheme, we test the scheme while all three motors are actuated. By this means, the concept presented in the previous section could fairly be assessed.

In the implementation stage, we observed that the gearing illustrated in Fig. 1 introduces

nonlinear friction terms and the angular velocity of the motors even without the link load changes under the same input voltage. Furthermore, again because of the inevitable gearing of the design, there exists a considerable dead zone in the motor actuation. Approximately between ±0.8 volts of average control input magnitude, the motors do not respond at all.

These difficulties apparently require an intelligent controller meeting the desired performance specifications. In Fig. 6, the tracking errors and controller outputs for the base, shoulder and elbow motors are illustrated in the units of radians, radians/second and volts respectively. Parallel to what the presented design prescribes, clearly from the first two rows, the error in angular positions and velocities tend to zero. The bottom row depicts the evaluated control signals. As mentioned before, the existing dead zone in the system dynamics for all three links causes an oscillatory behavior in the applied motor voltages. When a tiny nonzero error is measured, the controller tries to improve it, however, this requires the excess of the dead zone level. If this causes a nonzero error having the opposite sign, the procedure repeats. Therefore, an oscillatory behavior is observed in the control signals. The performance in terms of maintaining the precision with the presence of such a nonlinearity is satisfactory and the smoothness of the control signal is another important observation that should be highlighted.

Fig. 7 illustrates the values of s_c postulated in Proposition 2, the behavior in the corresponding phase space defined for each motor, and the evolution of the adjustable parameter vector norms of the three FLCs. For the parametric evolution, we plot the value of $\left\| \underline{\phi}_k \right\| = (\underline{\phi}_k^{\mathrm{T}} \underline{\phi}_k)^{1/2}$, which is an appropriate measure for evaluating the stability of the tuning mechanism. As claimed in Theorem 5, the trends of the evolutions seen in the bottom subplots suggest that the excited parameters are tuned until the goal described for each one of the motors is reached. The extracted values of the control discrepancies converge to zero, and the tuning becomes inactive during the sliding mode.

In order to avoid adverse effects of the sign function appearing in the reaching law and the tuning law, $\mathrm{sgn}(s) \approx s/(|s|+\delta)$ approximation is used with δ=0.25. This significantly reduces the chattering magnitudes due to the discontinuity of the original sign function, and helps the smoothly inactivation of the tuning scheme in the vicinity of the sliding manifold.

For the presented results, for all three FLCs, there exists a $0<\Lambda_1<0.92$ and $0<\Lambda_2<0.8375$ enabling the existence of a strictly positive Γ satisfying the inequality $0<\Gamma<0.2418$. Apparently, the numbers obtained justify the theoretical background discussed in the third section.

The computational burden of the proposed strategy for each FLC and its tuning mechanism requires 113 floating point operations and 12 comparisons per control period. This result is in good compliance with the needs of the modularity concept of distributed robotics.

5. CONCLUSIONS

This paper discusses the issues related to real-time adaptive fuzzy sliding mode control of a class of bipartite modular robotic systems called *I-Cubes*. Since one of the core issues in the design and implementation of the *I-Cubes* entity is to keep the modularity with cheap components, the control precision becomes a tedious objective due to the time-sparse observations of data, numerical differentiations, noise, nonlinear friction effects, dead zone effects and limited mobility in applicable voltage range for the motors used. In order to address all these difficulties in an appropriate manner, we suggest the use of adaptive fuzzy sliding mode control scheme with triangular and stationary membership functions. The defuzzifier parameters of the FLCs are adjusted according to a rule that drives the corresponding subsystem dynamics into a predefined sliding regime. The analysis of the approach has been presented to address the issues described in continuous time and discrete time, and the stability conditions are derived. The results observed justify the theoretical claims, both in terms of performance in operational space and in terms of the stability in adjustable parameter space. The mentioned difficulties are alleviated through the use of the proposed adaptation mechanism by assuming that the plant under control belongs to a certain class, and with significantly low computational

burden. When the addressed difficulties are considered with what have been obtained, it can fairly be claimed that the proposed technique constitutes a good solution for admissibly precise control with such cheap hardware, and is therefore a good candidate for control applications in modular robotics.

ACKNOWLEDGEMENTS

The work described in this manuscript was supported in part by Defense Advanced Research Projects Agency under contract DABT-63-97-1-0003, and by the Pennsylvania Infrastructure Technology Alliance, a partnership of Carnegie Mellon University, Lehigh University, and the Commonwealth of Pennsylvania's Department of Economic and Community Development.

The authors would like to thank Dominic Muren, Will Hein, Kevin Peterson and Mike Vande Weghe for their efforts in various phases of the realization of this project.

REFERENCES

[1] Paredis, C. J.-J. and Khosla P. K., "Kinematic Design of Serial Link Manipulators from Task Specifications," *Int. J. of Robotics Research*, Vol.12, No.3, pp.274-287, 1993.

[2] Neville, B. and Sanderson, A., "Tetrabot Family Tree: Modular Synthesis of Kinematic Structures for Parallel Robotics," *Proc. IEEE / RSJ Int. Symposium of Robotics Research*, pp.382-390, 1996.

[3] Fukuda, T. and Kawaguchi, Y., "Cellular Robotic System as One of the Realization of Self -Organizing Intelligent Universal Manipulator," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.662-667, 1990.

[4] Pamecha, A., Chiang, C.-J., Stein, D. and Chirikjian, G. S., "Design and Implementation of Metamorphic Robots," *Proc. ASME Design Eng. Tech. Conf. and Comp. in Engineering Conf.*, Irvine CA, 1996.

[5] Yoshida, E., Murata, S., Tomita, K., Kurokawa, H. and Kokaji, S., "Experiments of Self-Repairing Modular Machine," *Distributed Autonomous Robotic Systems 3*, H. Lueth, R. Dillmann, P. Dario, H. Wörn, (eds.), Springer-Verlag, pp.119-128, 1998.

[6] Vona, M. and Rus, D. L., "A Physical Implementation of the Self-reconfiguring Cristalline Robot," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp.1726-1733, 2000.

[7] Kotay, K. and Rus, D. L., "The Inchworm Robot: A Multi-Functional System," *Autonomous Robots*, Vol.8, No.1, pp.53-69, 2000.

[8] Yim, M., Duff, D. and Roufas, K. D., "PolyBot: A Modular Reconfirurable Robot," *Proc.IEEE Int. Conf. on Robotics and Automation*, pp.514-520, 2000.

[9] Castano, A., Shen, W.-M. and Will, P., "CONRO: Towards Deployable Robots with Inter-Robots Metamorphic Capabilities," *Autonomous Robots*, Vol.8, No.3, pp.309-324, 2000.

[10] Kotay, K. and Rus, D.L., "Motion Synthesis for the Self-Reconfiguring Robotic Molecule," *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vol.2, pp.843-851, 1998.

[11] Murata, S., Kurokawa, H., Yoshida, E., Tomita, K. and Kokaji, S., "A 3-D Self-Reconfigurable Structure," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.432-439, 1998.

[12] Yoshida, E., Murata, S., Kaminura, A., Tomita, K., Korokawa, H. and Kokaji, S., "Motion Planning of Self - reconfigurable Modular Robot," *Proc. of the 7th Int. Symp. on Experimental Robotics*, pp.375-384, 2000.

[13] Bojinov, H., Casal, A. and Hogg, T., "Emergent Structures in Modular Self-reconfigurable Robots," *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.1734-1742, 2000.

[14] Ünsal, C. and Khosla, P. K., "A Multi-Layered Planner for Self-Reconfiguration of a Uniform Group of I-Cube Modules," *Proc. 2001 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2001.

[15] Jang, J.-S. R., Sun, C.-T., and Mizutani, E., *Neuro-Fuzzy and Soft Computing*, PTR Prentice-Hall, 1997.

[16] Wang, L. X., *A Course in Fuzzy Systems and Control*, PTR Prentice-Hall, 1997.

[17] Hung, J. Y., Gao, W. and Hung, J. C., "Variable Structure Control: A Survey," *IEEE Trans. on Industrial Electronics*, Vol. 40, No. 1, pp. 2-22, 1993.

[18] Young, K. D., Utkin, V. I. and Özguner, Ü., "A Control Engineer's Guide to Sliding Mode Control," *IEEE Trans. on Control*

Systems Technology, Vol. 7, No. 3, pp. 328-342, 1999.
[19] Utkin, V. I., *Sliding Modes in Control Optimi-zation*, Springer Verlag, New York, 1992.
[20] Efe, M. Ö., "Variable Structure Systems Theory Based Training Strategies for Computationally Intelligent Systems," *Ph.D. Dissertation*, Bogazici University, 2000.
[21] Sarpturk, S. Z., Istefanopulos, Y. and Kaynak, O., "On the Stability of Discrete-Time Sliding Mode Control Systems," *IEEE Trans. on Automatic Control*, Vol. 32, No. 10, pp. 930-932, 1987.
[22] Gao, W., Wang, Y. and Homaifa, A., "Discrete-Time Variable Structure Control Systems", *IEEE Trans. on Industrial Electronics*, Vol. 42, No. 2, pp. 117-122, 1995.
[23] Sira-Ramirez, H., "Non-linear Discrete Variable Structure Systems in Quasi-Sliding Mode," *Int. Journal of Control*, Vol. 54, No. 5, pp. 1171-1187, 1991.
[24] Chen, X. and Fukuda, T., "Computer-Controlled Continuous-Time Variable Structure Systems with Sliding Modes," *Int. Journal of Control*, Vol. 67, No. 4, pp. 619-639, 1997.
[25] Palm, R. and John, R., "Supervisory Fuzzy Control of an Exhaust Measuring System," *Proc. of the Fifth IEEE Int. Conf. on Fuzzy Systems*, Vol. 1, pp. 479-485, 1996.
[26] Xu, H., Sun, F. and Sun, Z., "The Adaptive Sliding Mode Control based on a Fuzzy Neural Network for Manipulators," *IEEE Int. Conf. on Systems, Man and Cybernetics*, Vol. 3, pp. 1942-1946, 1996.
[27] Fang, Y., Chow, T. W. S. and Li, X. D., "Use of a Recurrent Neural Network in Discrete Sliding-Mode Control," *IEEE Proc. Control Theory Appl.*, Vol. 146, No. 1, pp. 84-90, 1999.
[28] Muñoz , D. and Sbarbaro, D., "An Adaptive Sliding-Mode Controller for Discrete Nonlinear Systems," *IEEE Trans. on Industrial Electronics*, Vol. 47, No. 3, pp. 574-581, 2000.
[29] Ünsal, C., Kiliççöte, H., Patton M. and Khosla, P. K., "Motion Planning for a Modular Self-reconfiguring Robotic System," *Distributed Autonomous Robotic Systems* 4, Springer-Verlag, pp.165-175, 2000.
[30] Ünsal, C., Kılıççöte, H. and Khosla, P. K., "A Modular Self-Reconfigurable Bipartite Robotic System: Implementation and Motion Planning," *Autonomous Robots*, v.10, no.1, pp. 23-40, 2001.
[31] Ünsal, C. and Khosla, P. K., "Solutions for 3-D Self-reconfiguration in a Modular Robotic System: Implementation and Motion Planning," *Proceedings of SPIE, Sensor Fusion and Decentralized Control in Robotic Systems III*, November 2000.
[32] Yen, J. and Langari, R., *Fuzzy Logic*, PTR Prentice-Hall, New Jersey, 1999.
[33] Passino, K. M. and Yurkovich, S., *Fuzzy Control*, Addison-Wesley, California, 1998.
[34] Sira-Ramirez, H. and Colina-Morles, E., "A Sliding Mode Strategy for Adaptive Learning in Adalines*," IEEE Trans. on Circuits and Systems – I: Fundamental Theory and Applications*, Vol.42, No.12, pp.1001-1012, 1995.
[35] Yu, X., Zhihong, M. and Rahman, S. M. M., "Adaptive Sliding Mode Approach for Learning in a Feedforward Neural Network," *Neural Computing and Applications*, Vol.7, pp.289-294, 1998.

Mehmet Önder Efe received the B.Sc. degree from Electronics and Telecommunications Engineering Department, Istanbul Technical University (Turkey) in 1993, and M.S. degree from Systems and Control Engineering Department, Bogazici University (Turkey), in 1996. He has completed his Ph.D. study in Bogazici University, Electrical and Electronics Engineering Department in June 2000. Between August 1996- December 2000, he was with Bogazici University, Mechatronics Research and Application Center as a research assistant. During 2001, Dr. Efe was a postdoctoral research fellow at Carnegie Mellon University, Electrical and Computer Engineering Department, and he was a member of the Advanced Mechatronics Laboratory group. Since January 1st, 2002, he has been with The Ohio State University, Electrical Engineering Department as a postdoctoral research associate. He is working at the Collaborative Center of Control Science. Dr. Efe is the author of more than 50 technical publications focusing on the applications of computational intelligence and control theory and is a member of IEEE, AIAA and Soft Computational Intelligence Society of Turkey.

Lemi Dağhan Acay received the B.Sc. degree from Bogazici University, Electrical and Electronics Engineering Department (Turkey) in 2001. After graduation, he spent six months in the Institute for Complex Engineered Systems (ICES) of Carnegie Mellon University as a visiting research associate. During this period, Mr. Acay was involved with the design and implementation of I-Cubes system. He is currently with Bogazici University, Electrical and Electronics Engineering as a research assistant, and is expecting to obtain M.Sc. degree from the same department.

Cem Ünsal received the B.Sc. degree from Electrical and Electronics Engineering Department, Bogazici University, Istanbul, Turkey, and his M.Sc. and Ph.D. degrees from Bradley Department of Electrical and Computer Engineering, Blacksburg, VA, USA, in 1993 and 1997 respectively. From January 1997 to July 2001, he was a Postdoctoral Fellow at the Robotics Institute, and worked as a Project Scientist at the Institute for Complex Engineered Systems, in Carnegie Mellon University, Pittsburgh, PA. He is currently working as a Software Engineer at Atoga Systems, Inc., Fremont, CA, where he participates in the development of client-server architectures for network management systems. Dr. Ünsal is the author and co-author of several articles on artificial intelligence, intelligent control, robotics, self-organization and self-reconfiguration. Dr. Ünsal is a member of IEEE, AAAI. You can contact him at unsal@ieee.org..

Pradeep Kumar Khosla received his B. Tech. (Hons.) from IIT (Kharagpur, India) in 1980, and both MS (1984) and Ph.D. (1986) degrees from Carnegie Mellon University. He served as Assistant Professor of ECE and Robotics (1986-90), Associate Professor (1990-94) and Professor (1994 -), and Founding Director (1/97-6/99) of the Institute for Complex Engineered Systems (ICES) (which includes the former Engineering Design Research Center - a NSF ERC). During his tenure ICES grew to a total budget of more than $10M per year. He achieved this by strategically positioning ICES to pursue interdisciplinary projects in Embedded Systems, Tissue Engineering, Design and Manufacturing, and Networking. He is currently the Philip and Marsha Dowd Professor of Engineering and Robotics, and since July 1999 has been Head of the Electrical and Computer Engineering Department at Carnegie Mellon. Prior to joining Carnegie Mellon, he worked with Tata Consulting Engineers and Siemens (1980- 82) in the area of real-time control. Professor Khosla's research interests are in the areas of internet-enabled distributed and collaborative design, collaborating autonomous systems, agent-based architectures for embedded control, software composition and reconfigurable software for real-time embedded systems, reconfigurable and distributed robotic systems, distributed information systems, and intelligent instruments for biomedical applications. His research is multidisciplinary and has focused on the theme of "creating complex embedded systems and information systems through composition of and collaboration amongst building blocks." He is involved in Electrical and Computer Engineering, Design, and Robotics education both at the graduate and the undergraduate levels. He was a member of the committee that formulated a curriculum for the multidisciplinary Ph.D. program in Robotics at Carnegie Mellon. He was also a member of the Wipe the Slate Clean Committee that created a new four year undergraduate ECE degree curriculum at CMU. In support of the new curriculum he developed the Introductory Freshman level course "Introduction to Electrical and Computer Engineering" that emphasizes the notion of Teaching in Context. He is the co-author of a text book and a laboratory manual related to this course.