CONTROL OF A FOUR-ROPE-DRIVEN LEVEL-ADJUSTMENT ROBOT BASED ON THE SIRMS DYNAMICALLY CONNECTED FUZZY INFERENCE MODEL

Jianhong Zhang, Jianqiang Yi and Xiangmin Tan

Institute of Automation
Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P. R. China
{ jianhong.zhang; jianqiang.yi; xiangmin.tan }@ia.ac.cn

ABSTRACT. *In the paper, a control scheme, based on the SIRMs (Single Input Rule Modules) dynamically connected fuzzy inference model, is designed for a four-rope-driven level-adjustment robot to level the eccentric payload and balance the rope tension. The control scheme is composed of four controllers separately controlling one independent rope's length. Each controller, composed of nine sub-controllers, selects different sub-controllers to deal with different limit switches' situations. Each sub-controller is set up such that the angular control of the payload takes higher priority than the rope tension. Experiment results show that the control scheme works very well and can handle all possible situations.*

Keywords: Fuzzy control, SIRM (Single Input Rule Module), Dynamic importance degree, Rope, Level-adjustment, Robot

1. **Introduction.** In industry, many large heavy payloads should be carried from one place to another when they are assembled and transported, such as assembling ship hulls in factories, loading and unloading containers in the dock, etc. Some payloads are so fragile and precise that they can absolutely not endure a point-to-point or line-to-line touch with the assembly platform or the transporting vehicles. Therefore, these eccentric payloads should be adjusted to level for the sake of safety. So far, there are mainly three kinds of mechanisms that can be used to regulate the payloads' posture: the weight-compensation mechanism [1], the link parallel platform [2,3], and the rope parallel platform [4]. Comparing their advantages and disadvantages, a four-rope-driven level-adjustment robot (Figure 1), which controls each rope by an independent actuator, was designed.

In [5], a type-2 fuzzy neural network controller was designed for the robot, which could adjust the payload to level without considering the rope tension. Then, a hierarchical fuzzy controller, which regulated the payload to level and adjusted the rope tension to be balanced simultaneously, was designed in [6]. However, none of the above-mentioned researches considered the actuators' moving ranges. Therefore, a fuzzy controller, which took the actuators' moving ranges into consideration, was given in [7]. Yet, the fuzzy controller could only handle some simple situations.

In the paper, a practical control scheme, considering all possible situations, is designed for the robot. Even though the four actuators all move to their limit positions, the control scheme levels the payload and balances the rope tension. The control scheme is composed of four controllers regulating four independent ropes' lengths respectively. Each controller includes nine sub-controllers based on the SIRMs dynamically connected fuzzy inference model (SIRMs model for short) [8,9] separately for one possible situation. Some experiments are done to verify the effectiveness of the control scheme.
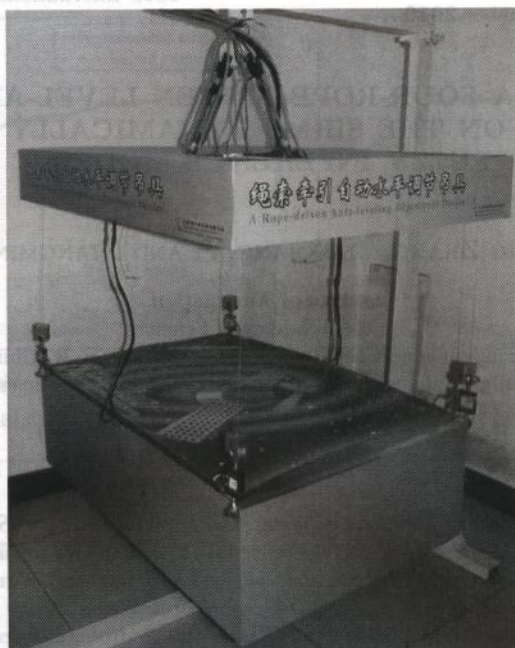
FIGURE 1. The robot

2. **Working Principle of the Robot.** The robot is mainly composed of an industrial PC with a motion control card, four independent ropes, a supporting plane, four tension sensors, two angle sensors and four actuators. Each actuator includes one step motor, one linear motion unit (LMU for short), and two limit switches (LS for short). The four actuators are schematically shown in Figure 2. By changing the ropes' lengths, the payload's posture is regulated, so is the rope tension. Each tension sensor is installed between a rope and the payload by a hanging point. The four hanging points symmetrically distribute at the payload's upper surface and form a rectangular. The two angle sensors are installed separately at the two diagonals of the rectangular to detect the angle of each diagonal with the horizontal plane. The motion control card controls the movement of the step motors. One LMU transforms the motor's rotation to the corresponding rope's linear movement. Because the LMUs' moving ranges aren't infinite, two limit switches are fixed at two ends of each LMU. If an LMU moves to one end, the corresponding limit switch will turn on. The working principle of the robot is shown in Figure 3.
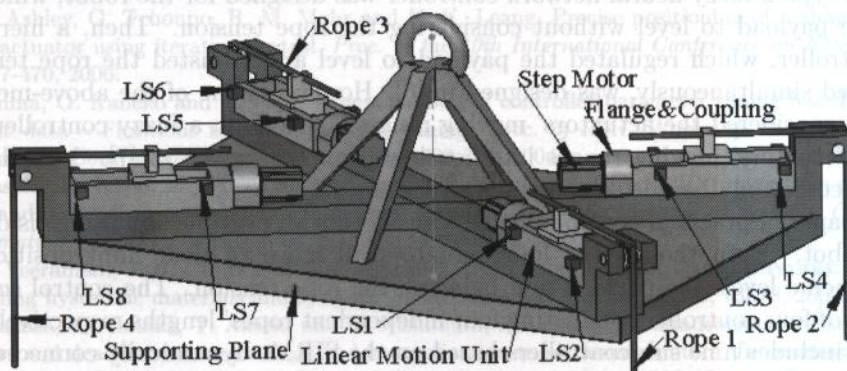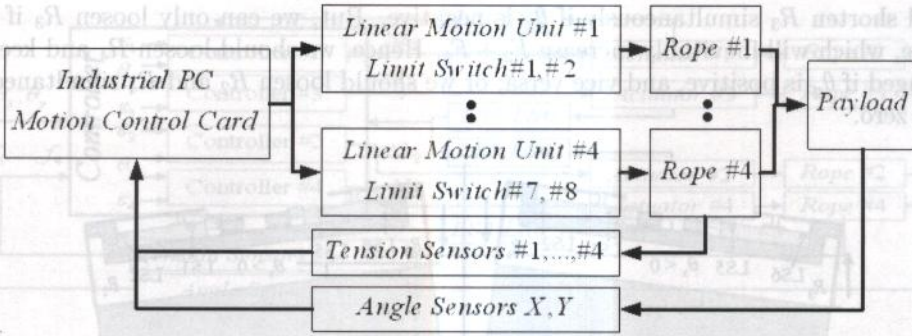

FIGURE 2. Sketch map of the actuators

FIGURE 3. Working principle of the robot

3. **Control Strategy.** Above all, we define some symbols to simplify the description. Each rope, whose length and tension are described as L_i and F_i respectively, is represented by R_i $(i = 1, ..., 4)$. The two angles detected by the angle sensors, are expressed as θ_x and θ_y respectively. In the case shown in Figure 4, θ_x is defined as positive. Furthermore, the average tension of the four ropes is defined as:

$$F_{AV} = \sum_{i=1}^{4} F_i/4 \tag{1}$$

Thus, the relative deviation of F_i and F_AV can be defined as:

$$\varepsilon_i = (F_i - F_{AV})/F_{AV} \quad (i = 1, ..., 4). \tag{2}$$

For the robot, there are two basic control objectives. Firstly, the payload should be regulated to level, which can be quantified as $|\theta_x| \leq 0.05°$ and $|\theta_y| \leq 0.05°$. Secondly, the rope tension should be balanced for the sake of safety, because a rope may snap if it bears too much tension. It can be quantified as $|\varepsilon_i| \leq 30\%$.

According to operating experience, although θ_x and θ_y are interactional, θ_x (θ_y) can be regulated by mainly changing L_1 and L_3 $(L_2$ and $L_4)$; F_i (F_1+F_3) is inversely proportional to L_i $(L_1 + L_3)$. As shown in Figure 4, if $\theta_x > 0$, we can either shorten R_1 or loosen R_3. If R_1 is shortened, however, F_1 will inevitably become bigger. Meanwhile, F_3 will also become bigger although R_3 isn't changed. Hence, we should shorten R_1 and loosen R_3 simultaneously to avoid changing $F_1 + F_3$ dramatically. On the contrary, if R_1 and R_3 bear too much tension, we can just loosen R_3 to reduce $F_1 + F_3$.
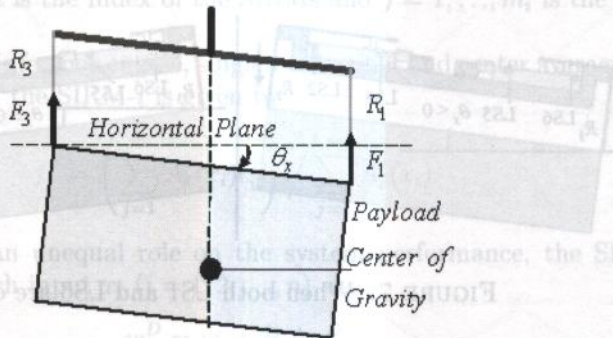

FIGURE 4. Positive situation of θ_x

However, if an LMU moves to one end and the corresponding limit switch turns on, it can only move towards the opposite direction. For instance, if the limit switch #1 (LS1) is on (Figure 5), R_1 cannot be shortened any more. In this case, we can still loosen

R_1 and shorten R_3 simultaneously if θ_x is negative. But, we can only loosen R_3 if θ_x is positive, which will inevitably increase $F_2 + F_4$. Hence, we should loosen R_4 and keep R_2 unchanged if θ_y is positive, and vice versa; or we should loosen R_2 and R_4 simultaneously if θ_y is zero.
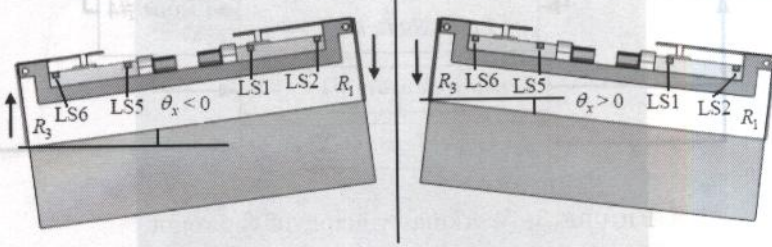

FIGURE 5. Schematic diagram when only LS1 turns on

Furthermore, neither R_1 nor R_3 can be shortened any more if both LS1 and LS5 are on (Figure 6). Consequently, if θ_x is negative, R_1 should be loosened; if θ_x is positive, R_3 should be loosened. Disappointingly, both policies will inevitably cause R_1 and R_3 to bear less tension. Then, R_2 and R_4 have to be regulated in order to keep the rope tension balanced.
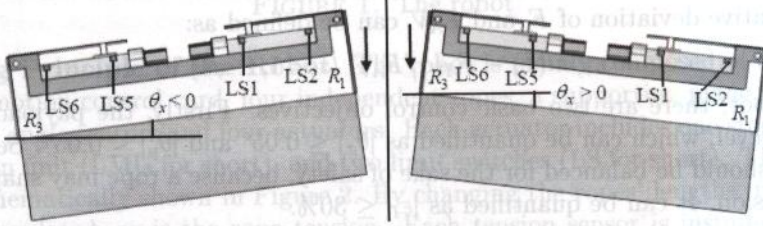

FIGURE 6. When both LS1 and LS5 are on

However, as R_1 (R_3) cannot be shortened (loosened) any more when both LS1 and LS6 are on, there is no way to adjust the payload to level under the existing conditions if θ_x is still positive (Figure 7). Luckily, this extreme situation rarely happens. Of course, if θ_x is negative, we can still loosen R_1 and shorten R_3 simultaneously.
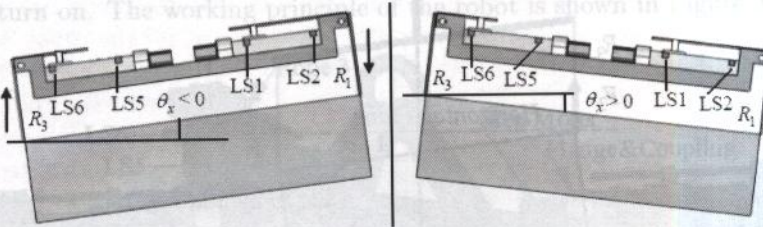

FIGURE 7. When both LS1 and LS6 are on

4. **Control Design and Experiment Results.** As it is hard to deduce a precise mathematical model for the robot, we try to control the robot with the SIRMs model. As θ_x (θ_y) could be adjusted by mainly changing L_1 and L_3 (L_2 and L_4), we can design four controllers for the robot, each of which regulates one rope's length. For instance, controller #1 outputs ΔL_1 (change of L_1) to regulate R_1 according to θ_x and ε_1. The control
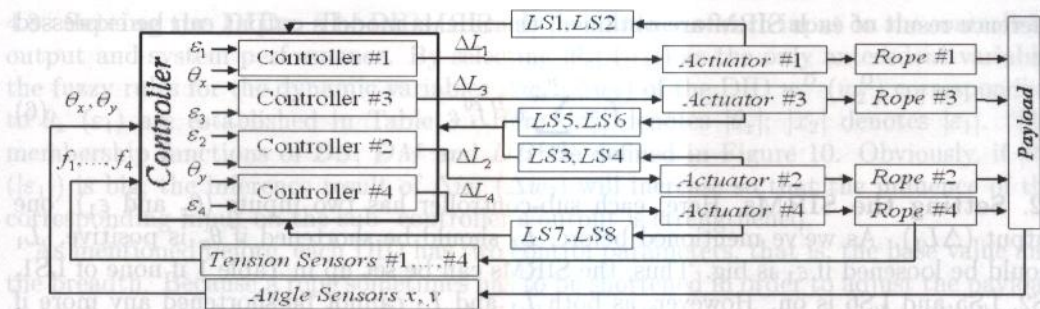
FIGURE 8. Control scheme

scheme is shown in Figure 8. In the paper, we mainly take the design of the controller #1 as an example.

When we design the controller #1, different control strategy should be taken if either the actuator #1 or #3 moves to its limit position. So the controller #1 must detect the states of LS1, LS2, LS5 and LS6 in every control circle. Totally, there are nine different situations, each of which is handled by a sub-controller. For instance, the simplest situation is that none of the limit switches turns on. Another four situations are that only one of LS1, LS2, LS5 and LS6 turns on. The others are that either LS1 or LS2 turns on when either LS5 or LS6 turns on. The nine sub-controllers have the same inputs and outputs, but different fuzzy rules to handle different situations.

4.1. **Introduction of the SIRMs model.** The SIRMs model, which can effectively avoid fuzzy rule explosion by designing a Single Input Rule Module (SIRM) and a dynamic importance degree (DID for short) for each input, has been applied in the stabilization control of parallel-type double inverted pendulum system [9], etc. Before presenting the design of the sub-controllers, let's briefly describe the SIRMs model. For systems of n inputs and 1 output, we firstly define a SIRM separately for each input as:

$$\text{SIRM-}i : \{R_i^j : \text{ if } x_i = A_i^j \text{ then } f_i = c_i^j\}_{j=1}^{m_i} \qquad (3)$$

where SIRM-i denotes the SIRM of the ith input, and R_i^j is the jth rule in the SIRM-i. The ith input x_i is the only antecedent variable; the consequent variable f_i is an intermediate variable corresponding to the output f. A_i^j is the membership functions of x_i. c_i^j is a real number. i is the index of the SIRMs and $j = 1, \ldots, m_i$ is the index of the rules in the SIRM-i.

Adopting the product inference engine, singleton fuzzifier and center average defuzzifier, the inference result f_i^0 of the SIRM-i is given by:

$$f_i^0 = \left(\sum_{j=1}^{m} A_i^j(x_i) c_i^j \right) / \sum_{j=1}^{m} A_i^j(x_i) \qquad (4)$$

As each input plays an unequal role on the system performance, the SIRMs model defines a DID w_i^D for each input x_i ($i = 1, 2, \ldots, n$) as:

$$w_i^D = w_i + B_i \Delta w_i^0 \qquad (5)$$

where w_i is the base value which guarantees the minimum weight of the corresponding input for a control process; and the dynamic value, product of the breadth B_i and the inference result Δw_i, tunes the influence degree of the input on system performance according to current system state. The base value and breadth are control parameters, and the dynamic variable can be expressed by fuzzy rules. Once each DID and the fuzzy

inference result of each SIRM are obtained, the SIRMs model's output can be expressed as:

$$f = \sum_{i=1}^{n} w_i^D f_i^0. \tag{6}$$

4.2. **Setting the SIRMs.** Here, each sub-controller has two inputs (θ_x and ε_1), one output (ΔL_1). As we've mentioned before, L_1 should be shortened if θ_x is positive. L_1 should be loosened if ε_1 is big. Thus, the SIRMs can be set up in Table 1 if none of LS1, LS2, LS5 and LS6 is on. However, as both L_1 and L_3 cannot be shortened any more if both LS1 and LS5 are on, the corresponding SIRMs are shown in Table 2. The SIRMs of the other seven situations aren't written down due to limit space. The membership functions of the antecedent variables θ_x and ε_1 are defined in Figure 9.

TABLE 1. SIRM setting for each input item when LS1, LS2, LS5 and LS6 are all off

Antecedent variable $x_1(\theta_x)$	Consequent variable f_1	Antecedent variable $x_2(\varepsilon_x)$	Consequent variable f_2
NB	−1.0	NB	1.0
ZO	0.0	ZO	0.0
PB	1.0	PB	−1.0

TABLE 2. SIRM setting for each input item when both LS1 and LS5 are on

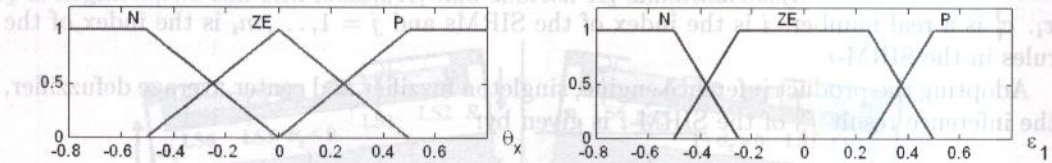| Antecedent variable $x_1(\theta_x)$	Consequent variable f_1	Antecedent variable $x_2(\varepsilon_x)$	Consequent variable f_2
NB	−0.5	NB	0.0
ZO	0.0	ZO	0.0
PB	0.0	PB	−1.0

FIGURE 9. Membership functions of θ_x and ε_1
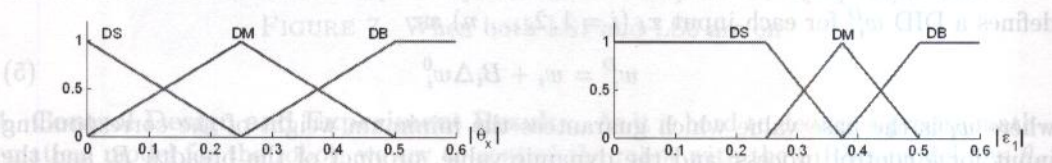

FIGURE 10. Membership functions of $|\theta_x|$ and $|\varepsilon_1|$ for each dynamic variables

4.3. **Setting the DIDs.** The DIDs indicate the influence of each input on the controller's output and system performance. By selecting $|\theta_x|$ $(|\varepsilon_1|)$ as the only antecedent variable, the fuzzy rules for the dynamic variables Δw_1 (Δw_2) of the DID w_1^D (w_2^D) corresponding to θ_x (ε_1) are established in Table 3. Here, $|x_1|$ denotes $|\theta_x|$; $|x_2|$ denotes $|\varepsilon_1|$. The membership functions of DS, DM and DB are defined in Figure 10. Obviously, if $|\theta_x|$ $(|\varepsilon_1|)$ is big, the inference result of Δw_1 (Δw_2) will increase so that the influence of the corresponding input on the sub- controller's output is strengthened.

As mentioned before, each DID has two control parameters, that is, the base value and the breadth. Because a rope sometimes has to be shortened in order to adjust the payload to level even though this may disappointingly cause the rope tension to be unbalanced, the control parameters are set up so that the sum of the base value and the breadth of θ_x is larger than that of ε_1. After many times of trial and error, the control parameters are given as follows: the base value $w_1 = 2$ $(w_2 = 1)$ and the breadth $B_1 = 1$ $(B_2 = 1)$.

TABLE 3. Fuzzy rules for the two dynamic variables

Antecedent variable $|x_i|$	Consequent variable Δw_i
DS	0.0
DM	0.5
DB	1.0

4.4. **Experiment results.** Some experiments have been done to test the control scheme's performance and a set of experiment results is shown in Figure 11. At the very beginning, the payload is quite slantwise and the rope tension is extremely unbalanced (i.e. F_1 is even zero). What's worse, R_1, R_3 and R_4 cannot be loosened further because the limit switches LS2, LS6 and LS8 are on. However, the control scheme levels the payload and balances the rope tension satisfactorily. From Figure 11, we can easily find that the four controllers' outputs are closely related to the actuators' states. When some actuators move to their limit positions, the controllers give out reasonable outputs to avoid damaging the actuators.
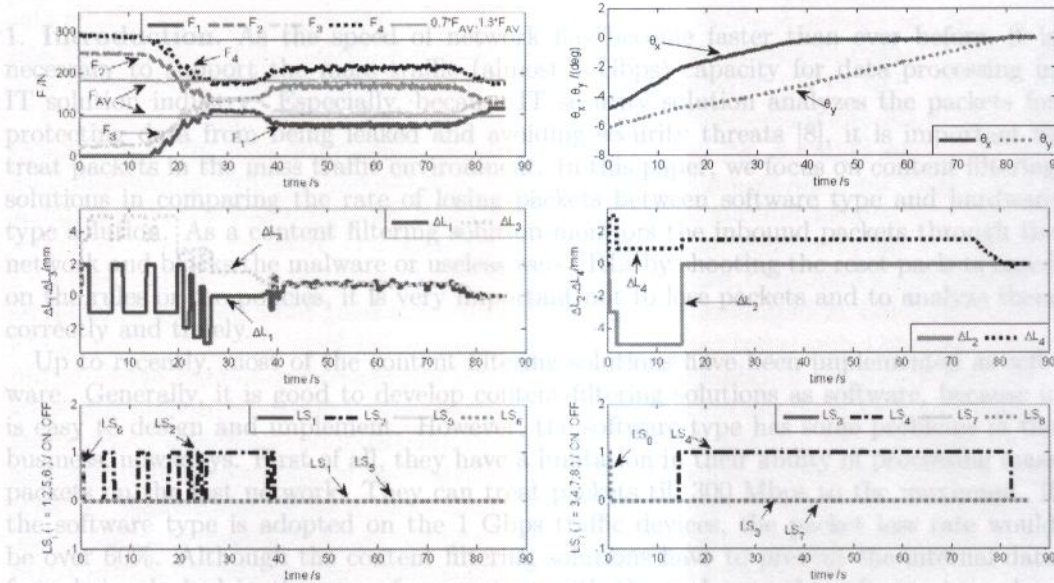

FIGURE 11. Experiment results

5. **Conclusions.** In the paper, a control scheme, based on the SIRMs model, is designed for the four-rope-driven level-adjustment robot. The control scheme includes four controllers, each of which consists of nine fuzzy sub-controllers and regulates one rope's length. According to the states of the limit switches, each controller switches the suitable sub-controller among the nine sub-controllers. Experiment results show that the control scheme can deal with all possible situations and the control scheme is effective.

REFERENCES

[1] X. Zhang, J. Yi, D. Zhao and G. Yang, Modelling and control of a self-leveling crane, *Proc. of the 2007 IEEE International Conference on Mechatronics and Automation*, Harbin, China, pp.2922-2927, 2007.

[2] K. Liu, J. M. Fitzgerald and F. L. Lewis, Kinematic analysis of a Stewart platform manipulator, *IEEE Trans. on Industrial Electronics*, vol.40, no.2, pp.282-293, 1993.

[3] W. Khalil and S. Guegan, Inverse and direct dynamic modeling of Gough-Stewart robots, *IEEE Trans. on Robotics and Automation*, vol.20, no.4, pp.754-762, 2004.

[4] J. Pusey, A. Fattah, S. Agrawal and E. Messina, Design and workspace analysis of a 6-6 cable-suspended parallel robot, *Mechanism and Machine Theory*, vol.39, no.7, pp.761-778, 2004.

[5] C. Li, J. Yi, Y. Yu and D. Zhao, Inverse control of cable-driven parallel mechanism using type-2 fuzzy neural network, *Acta Automatica Sinica*, vol.36, no.3, pp.459-464, 2010.

[6] Y. Yu, J. Yi, C. Li, D. Zhao and J. Zhang, Fuzzy logic based adjustment control of a cable-driven auto-leveling parallel robot, *Proc. of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, pp.2102-2107, 2009.

[7] J. Zhang, J. Yi and Y. Yu, Intelligent control of a four-rope-driven level-adjustment device with constrained outputs, *Proc. of the 2009 IEEE International Conference on Robotics and Biomimetics*, Guilin, China, pp.1860-1865, 2009.

[8] H. Seki, H. Ishii and M. Mizumoto, On the generalization of single input rule modules connected type fuzzy reasoning method, *IEEE Trans. on Fuzzy Systems*, vol.16, no.5, pp.1180-1187, 2008.

[9] J. Yi, N. Yubazaki and K. Hirota, A new fuzzy controller for stabilization of parallel-type double inverted pendulum system, *Fuzzy Sets and Systems*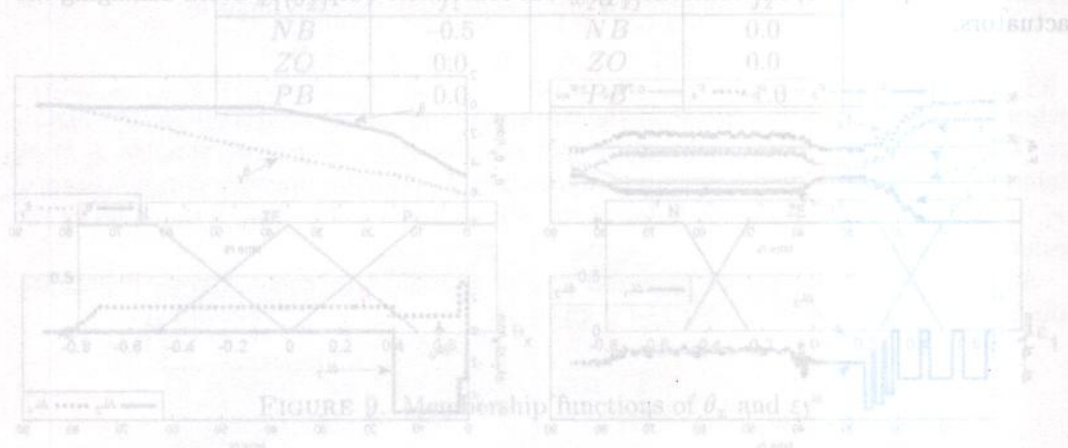, vol.126, no.1, pp.105-119, 2002.