2009 American Control Conference
Hyatt Regency Riverfront, St. Louis, MO, USA
June 10-12, 2009

Control of a Rope-driven Self-leveling Device for Leveling Adjustment

Yi Yu, Jianqiang Yi, *Member, IEEE*, Chengdong Li, Dongbin Zhao, *Member, IEEE,* and Jianhong Zhang

Abstract—**To solve the level-adjusting problem of high accurate and costly payloads when loading and unloading, a rope-driven self-leveling device is developed, and a neuro-fuzzy controller is proposed. After a brief introduction of the configuration characteristics of the device and the fundamentals of neuro-fuzzy control, the construction of the neuro-fuzzy controller is set up, in which the angles of two diagonal inclinations which are measured from the two angle sensors are chosen as input variables, and the changes of two linear motion units' positions are the control variables. The neuro-fuzzy controller, whose rules are constructed based on human's regulating experience, was tuned by a hybrid algorithm, which is a combination of the least square estimate (LSE) method and the back-propagation (BP) algorithm. Experimental results show that the proposed neuro-fuzzy controller can achieve the control objective with high accuracy of regulation and short adjusting time, and is easily applied to the practical device.**

I. INTRODUCTION

IN many areas, such as industrial applications etc, high accurate and costly payloads (e.g., satellites, aircrafts, turbine engines) often need to be loaded and unloaded. Such costly payloads can't endure point-to-point or point-to-surface touch with the ground or the assembly platform. Besides, their centers of mass usually deviate from their geometric centers. This would cause inclination and lateral forces in the entire course, which usually leads to payloads' distortion, damage, or even complete destruction.

To avoid these pitfalls, various methods and mechanisms have been studied and applied, for example, link parallel platforms [1], cable parallel platforms [2], hybrid parallel platforms (combinations of link structures and cable structures) [3], and weight compensation mechanisms [4], etc. Theoretically, these adjusting methods above all could be used for leveling adjustment. However, link parallel platforms would inevitably increase the system weight, and are not easy to meet precision requirements and realize self-regulation; rope parallel platforms are too difficult to model and control, and also require large spaces to fix the ropes and increase the size of the levelling mechanism; these intrinsic shortcomings are also in some way, true for the hybrid parallel platforms; weight-compensating method is precise and smooth, and easy to operate, but it often can only accomplish one-dimensional regulation, meanwhile, heavier matching block and larger structure size are needed when regulating heavy and large payloads. Although rope based techniques own the above defects, corresponding parallel devices have large workspaces, strong pulling forces, and can regulate swiftly. Nowadays, research activities are mainly focused on the analysis of workspaces [5], the central position and pose of payloads [6], kinematics characteristics [7], and dynamic characteristics [8]. However, no studies have yet been reported on the leveling control of payloads' junction surface, and the equilibrium problem of ropes' pulling forces. To realize the leveling control for high-accurate and costly payloads, manual regulating devices using four ropes are universally adopted in practical applications. Disadvantages of these manual devices are obvious, such as great labor intensity, low efficiency, low precision, and hidden trouble in safety. How to improve the condition has become an urgent difficult problem, which needs to be completely solved.

Recently, based on the analysis of merits and drawbacks of the above methods and mechanisms, meanwhile, combining with the widely used manual regulating devices, we have developed a rope-driven self-leveling device, whose structure and configuration is shown in Fig. 1. A 2D model for the device is established in [9], but for the three dimensional payload, the accurate 3D model is quite hard to obtain for the reasons given in section II. Designing conventional controllers for the rope-driven self-leveling device is a complex and arduous job. Fuzzy logic control [10,11], which has been proved to be a practical alternative for a variety of challenging control applications that are difficult to be solved by classical methods, can sufficiently incorporate human knowledge or experience into system design. Therefore, it may be a quite good choice to use fuzzy logic controllers to regulate the payloads.

However, the conventional ways of designing fuzzy logic controllers face the difficulties to transform human experience into the rule base and to tune the parameters of the membership functions (MFs) so as to maximize (minimize) the performance index. In order to overcome these drawbacks, several approaches have been developed, one of which is fuzzy neural network controller (FNNC) or neuro-fuzzy controller [12-17]. Neuro-fuzzy controllers, which can combine the merits of fuzzy systems and neural networks, have been widely applied in many practical applications [16, 17]. Therefore, in this paper, a neuro-fuzzy

This work was partly supported by the NSFC Projects under Grant No. 60621001, National 863 Program(No.2007AA04Z239), China.

Y. Yu, J. Yi, C. Li, D. Zhao and J. Zhang are with the Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, P.R. China (phone: 86-10-62658815; fax: 86-10-62658815; e-mail: yuyisusong@gmail.com; {jianqiang.yi & chengdong.li & dongbin.zhao & jianhong.zhang} @ia.ac.cn.

978-1-4244-4524-0/09/$25.00 ©2009 AACC

controller is designed for the leveling adjustment of the rope-driven self-leveling device. And, we also use a hybrid algorithm, which is a combination of the least square estimate (LSE) method [18,19] and the back-propagation (BP) algorithm [12,13], to tune the parameters of the designed neuro-fuzzy controller to obtain better performance.

The paper is organized as follows: Section II presents the detailed structure of the proposed rope-driven self-leveling device for level adjustment. In Section III, the neuro-fuzzy controller for the rope-driven self-leveling device is developed and a hybrid algorithm, which is a combination of the least squares estimate (LSE) method and BP algorithm, is presented. Experimental results that show the performance of the proposed algorithm on the rope-driven self-leveling device are described in Section IV. Concluding remarks are given in Section V.

II. Structure of a Rope-driven Self-leveling Device

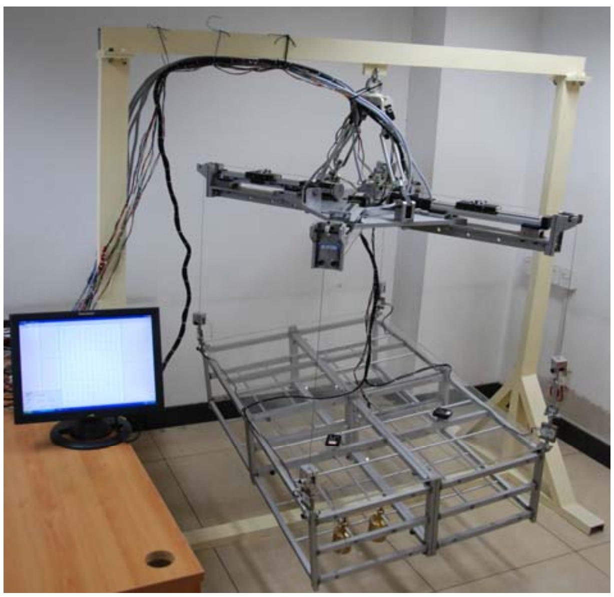

Fig. 1. Appearance of the prototype of rope-driven self-leveling device

In this section we present a prototype of the rope-driven self-leveling device. To coordinate the loading and unloading of rectangular payloads, manual regulating devices using four ropes are universally adopted in the practical applications. The desired performances are as follows: the junction surface angle with the horizontal is smaller than 0.2º, the adjusting time is less than 60 seconds, and for safety purpose the pulling force of each rope should almost be balanced, i.e., the deviation of each rope's pulling force should not be less or more than 30% of their average. We carry out the whole structure design of the device, in accordance with this demand. Below is a brief description of the various parts and their functions.

1） *Lifting Ring*: Installed in the center of rectangular worktable, it can implement horizontal and vertical motions when connected with lifting equipments.

2） *Rectangular Worktable*: It needs to bear all the weight of payloads and motor regulation devices, so rigid materials should be used. In addition, the rectangular worktable should not be eccentric, this not only requests that its center of mass should not deviate from its geometric center, but also that four regulation devices on it need to be fixed symmetrically along its two diagonal directions with certain size distribution.

3） *Four Regulation Devices*: As the core component of the device, each one consists of five parts: motor, motor driver, shaft coupling, connecting flange, and linear motion unit. The rope is connected with and moved by the linear motion unit which is driven by its motor via shaft coupling, and its moving direction can be changed by the crown block installed in the corner of the rectangular worktable. The payload is regulated by the four ends of such two ropes. Two regulation devices are installed symmetrically in the same diagonal direction, one of which is used to drive the rope, while the other is used to balance the rectangular worktable. When one regulation device driving the rope moves toward a certain direction, the other regulation device balancing the rectangular worktable will accordingly move toward the opposite direction.

4） *Two Ropes*: Each rope, which is installed between the two crown blocks fixed in the two ends of one diagonal direction, should be above the axial line of the linear motion units.

5） *Force Sensors*: One end of the force sensor is connected with one end of each rope, and the other is connected with one corner of the rectangular payload.

6） *Angle Sensors*: To measure the level inclination of the payload's junction surface, angle sensors need to be installed symmetrically in the payload's upper surface or lower surface. Ignoring the processing and deformation factors, upper surface and lower surface can be viewed as parallel, i.e., the dihedral angle of the payload's upper surface is equal to the dihedral angle of the payload's junction surface.

7） *Computer Control System*: Using the scheme of "PC + motion control board", it processes and analyses real-time data which are acquired from the force sensors and angle sensors, and then sends out commands to the motors to realize movement control.

8） *Electrical Sources*: They supply power for the regulation devices, force sensors, angle sensors, PC, etc.

III. Neuro-Fuzzy Controller for the Rope-driven Self-leveling Device

In this section, some problems and a control strategy for the rope-driven self-leveling device will be presented firstly; then, we will design a neuro-fuzzy controller for this device. To achieve better performance, we will also use the LSE method and BP algorithm to tune the parameters of the designed neuro-fuzzy controller.

A. Problems and Control Strategy

It is difficult to design a conventional controller for the rope-driven self-leveling device, because some problems are encountered when designing the practical control system. First, establishing the device's 3D model is really a tough problem to be solved, getting the numerical relationship between the lengths of ropes and the dihedral angle of payload's junction surface is not easy. Though a realistic 2D model of the device is proposed in [9] and its analytical solution can be obtained, it is so complicated that can't be applied directly and effectively in fact. What's worse, payload's exact weight and the accurate centroid position, which need to be used in the modeling, are usually unknown. Furthermore, coupling effect caused by the holistic physical structure exists in this real-world application, and is really difficult to be weakened. All these problems make it too hard to realize leveling adjustment through designing conventional controllers which are based on accurate mathematical model.

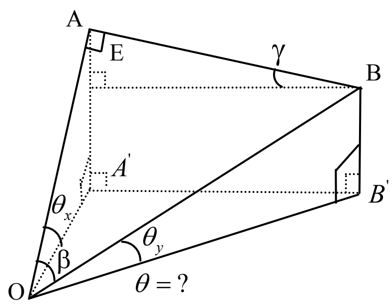

Fig. 2. Spatial relationship between payload's upper surface and the horizontal surface.

Fig 2 gives the spatial relationship between the payload's upper surface（OAB）and the horizontal surface（O A' B'）. Two angle sensors with the same precision, which can automatically detect the gradient, namely angle, between the lines they located in and the horizontal surface, are installed symmetrically along the two diagonal directions (OA and OB) of the payload's upper surface. Here, we define OA' and OB' as the projection of OA and OB to the horizontal surface, respectively. Therefore, the angles from the two angle sensors' measures, denoted as θ_x and θ_y, are the angles between the diagonal lines (OA and OB) and their respective horizontal projections (OA' and OB'). And, assume that θ is the dihedral angle between the payload's upper surface and the horizontal surface, and β (which is 67.3802° in the practical system) is the angle between the two diagonal lines (OA and OB). Suppose that $OA'=1$、$OB'=1$, from the following equation

$$\cos\theta = \frac{S_\Delta A'OB'}{S_\Delta AOB} \tag{1}$$

We can obtain that

$$\theta = \arccos\left(\frac{S_\Delta AOB}{S_\Delta A'OB'}\right) \tag{2}$$

$$= \arccos\left(\frac{1/2 \cdot OA \cdot OB \cdot \sin\angle AOB}{1/2 \cdot OA' \cdot OB' \cdot \sin\alpha}\right) \tag{3}$$

$$= \arccos\left(\frac{\sqrt{1-(sec\theta_x \cdot sec\theta_y \cdot \cos\beta - tg\theta_x \cdot tg\theta_y)^2}}{\sec\theta_x \cdot \sec\theta_y \cdot \sin\beta}\right) \tag{4}$$

From (4), we can deduce that, as long as θ_x and θ_y, the angles between the diagonal lines (OA and OB) and their respective horizontal projection (OA' and OB'), could both be adjusted to 0°, the plane (OAB) set by the two intersecting straight lines (OA and OB), that is, the junction surface of the payload, would be approximately horizontal. To achieve this objective, we can regulate the ropes through changing the positions, denoted as u_x and u_y, of the linear motor units.

B. Design of Neuro-Fuzzy Controller

From above discussion, we can see that the controller for the rope-driven self-leveling device should have two inputs and two outputs. The input variables are the angles θ_x, θ_y of the two diagonal inclinations that are measured from the two angle sensors, and the output variables are the position changes u_x, u_y of the two linear motion units that are used to drive the two ropes.

The structure of the neuro-fuzzy controller for the rope-driven self-leveling device is shown in Fig. 3. This is a two-input-two-output neuro-fuzzy network which has four layers and 25 rules. The first layer of the controller is the input layer. The second layer is the membership function layer. Each node in this layer performs the function of a fuzzy set. And, the third layer is the rule layer. This layer corresponds to the inference process in fuzzy logic controllers. In this layer, each node corresponds to one fuzzy rule in the rule base. Lastly, the forth layer is the output layer. This layer is used to carry out the defuzzification process in fuzzy logic controllers [12, 13].

Assume that there are M (M=25) rules in the rule base, each of which has the following form

Rule k: IF θ_x is $\tilde{A}_x^k$ and θ_y is $\tilde{A}_y^k$,

THEN u_x is w_x^k and u_y is w_y^k

where $k = 1, 2, \cdots, M$, w_z^k ($z = x, y$) are consequent

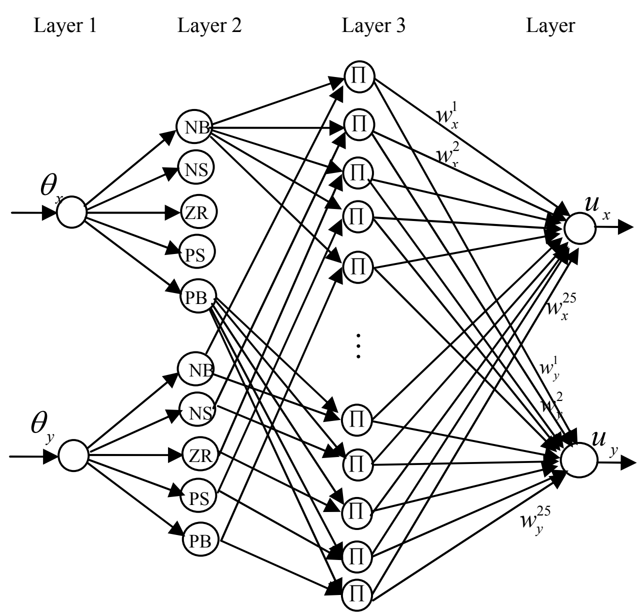

Fig. 3. Structure of the neuro-fuzzy controller.

weighting factors and $\tilde{A}_z^k$ ($z = x, y$) are Gaussian fuzzy sets NB, NS, ZR, PS, or PB, and

$$\mu_{\tilde{A}_z^k}(\theta_z) = \exp\left\{-\frac{1}{2}\frac{(\theta_z - m_z^k)^2}{(\sigma_z^k)^2}\right\}. \quad (5)$$

Once a crisp input $\Theta = (\theta_x, \theta_y)^{\mathrm{T}}$ is applied to the neuro-fuzzy controller, through the singleton fuzzifier and the inference process, the firing strength of the kth rule can be obtained from the third layer as

$$f^k = \mu_{\tilde{A}_x^k}(\theta_x) * \mu_{\tilde{A}_y^k}(\theta_y). \quad (6)$$

Then, using the weighted-average defuzzification method, the outputs of the neuro-fuzzy controller are

$$u_z(\Theta) = \frac{\sum_{k=1}^{M} w_z^k f^k}{\sum_{k=1}^{M} f^k}, \text{ where } z = x, y. \quad (7)$$

C. Training of the Neuro-Fuzzy Controller

To achieve better performance and meet the precision requirement of the rope-driven self-leveling device, in this subsection, we will use the LSE method [18,19] and BP algorithm [12,13] to tune the parameters of the designed neuro-fuzzy controller. Here, we use the LSE method to get the initial values of the consequent parameters at first several steps until the RMSE curve will not change obviously; and then, we start the BP training process till a satisfactory performance can be achieved.

Given N input-output training data (θ_x^1 , θ_y^1 , u_x^1 , u_y^1), ($\theta_x^2, \theta_y^2, u_x^2, u_y^2$), …, ($\theta_x^N, \theta_y^N, u_x^N, u_y^N$), the BP algorithm should be used to adjust the parameters of the neuro-fuzzy controller to minimize the following square error function:

$$E^t = E_x^t + E_y^t \quad (8)$$

where

$$E_x^t = \frac{1}{2}(e_x^t)^2 = \frac{1}{2}(u_x(\Theta^t) - u_x^t)^2 \quad (9)$$

$$E_y^t = \frac{1}{2}(e_y^t)^2 = \frac{1}{2}(u_y(\Theta^t) - u_y^t)^2$$

(10) in which $u_x(\Theta^t)$ and $u_y(\Theta^t)$ are the output of the neuro-fuzzy controller for the inputs $\Theta^t = (\theta_x^t, \ \theta_y^t)$.

Now, let us present the BP update rules for the parameters of the neuro-fuzzy controller first.

(1) BP update rule for the weighting factors

$$w_z^k(t+1) = w_z^k(t) - \alpha_w \frac{\partial E^{(t)}}{\partial w_z^k}$$

$$= w_z^k(t) - \alpha_w e_z^{(t)} \frac{\partial u_z(\Theta^t)}{\partial w_z^k} \quad (11)$$

(2) BP update rule for the means of Gaussian FSs

$$m_z^k(t+1) = m_z^k(t) - \alpha_m \frac{\partial E^{(t)}}{\partial m_z^k}$$

$$= m_z^k(t) - \alpha_m (e_x^{(t)} \frac{\partial u_x(\Theta^t)}{\partial m_z^k} + e_y^{(t)} \frac{\partial u_y(\Theta^t)}{\partial m_z^k}) \quad (12)$$

(3) BP update rule for the widths of Gaussian FSs

$$\sigma_z^k(t+1) = \sigma_z^k(t) - \alpha_\sigma \frac{\partial E^{(t)}}{\partial \sigma_z^k}$$

$$= \sigma_z^k(t) - m_\sigma (e_x^{(t)} \frac{\partial u_x(\Theta^t)}{\partial \sigma_z^k} + e_y^{(t)} \frac{\partial u_y(\Theta^t)}{\partial \sigma_z^k}) \quad (13)$$

From (7)

$$\frac{\partial u_z(\Theta^t)}{\partial w_z^k} = \frac{f^k}{\sum_{k=1}^{M} f^k}, \ z = x, y \quad (14)$$

$$\frac{\partial u_x(\Theta^t)}{\partial m_z^k} = \frac{\sum_{m=1}^{M}(w_x^m - u_x(\Theta^t))\frac{\partial f^m}{\partial m_z^k}}{\sum_{m=1}^{M} f^m} \quad (15)$$

where $\frac{\partial f^m}{\partial m_z^k} = \begin{cases} \frac{\partial \mu_{\tilde{A}_x^m}(\theta_x) * \mu_{\tilde{A}_y^m}(\theta_y)}{\partial m_z^k} & m \in I(\tilde{A}_z^k) \\ 0 & m \notin I(\tilde{A}_z^k) \end{cases}$

in which $I(\tilde{A}_z^k)$ is the set of fuzzy rules whose antecedent FSs include $\tilde{A}_z^k$. For example $I(\tilde{A}_x^1) = \{1,2,3,4,5\}$.

In the same way, $\frac{\partial u_x(\Theta^t)}{\partial \sigma_z^k}$, $\frac{\partial u_y(\Theta^t)}{\partial m_z^k}$, $\frac{\partial u_y(\Theta^t)}{\partial (\sigma_z^k)^2}$ can be computed.

BP algorithm is sensitive to initial values. And, it is easy to set reasonable initial values of the antecedent parameters, but difficult to determine reasonable initial values of the consequent parameters. Therefore, in this study, we utilize the LSE method to get the initial values of the consequent parameters, as the outputs of the neuro-fuzzy controller are linear with the consequent weighting factors.

From (7)

$$u_z(\Theta) = \frac{\sum_{k=1}^{M} w_z^k f^k}{\sum_{k=1}^{M} f^k} = F(\Theta)^{\mathrm{T}} W_z, \quad (16)$$

where $F(\Theta) = [\frac{f^1(\Theta)}{\sum_{k=1}^{M} f^k(\Theta)}, \frac{f^2(\Theta)}{\sum_{k=1}^{M} f^k(\Theta)}, \ldots, \frac{f^M(\Theta)}{\sum_{k=1}^{M} f^k(\Theta)}]^{\mathrm{T}}$, (17)

$$W_z = [w_z^1, w_z^2, \ldots, w_z^M]^{\mathrm{T}}. \quad (18)$$

After the training data set {(θ_x^1 , θ_y^1 , u_x^1 , u_y^1), ($\theta_x^2, \theta_y^2, u_x^2, u_y^2$), …, ($\theta_x^N, \theta_y^N, u_x^N, u_y^N$)} is imported to the neuro-fuzzy controller, vectors $F(\Theta^1), F(\Theta^2), \ldots, F(\Theta^N)$ can be calculated.

Denote $Q = \begin{bmatrix} q^{1^T} \\ q^{2^T} \\ \vdots \\ q^{M^T} \end{bmatrix} = \begin{bmatrix} F(\Theta^1)^T \\ F(\Theta^2)^T \\ \vdots \\ F(\Theta^N)^T \end{bmatrix}$,

$$D_z = \begin{bmatrix} u_z^1 \\ \dots \\ u_z^N \end{bmatrix}, \; z = x, y .$$

To minimize $\left\| Q^T W_z - D_z \right\|^2 = 2\sum_{t=1}^{N} E_z^t$, using the pseudo-inverse, the LSE of $W_z (z = x, y)$ can be written as [12,13,18,19]

$$W_z^* = (Q^T Q)^{-1} Q^T D_z . \tag{19}$$

Because of the expensive computation in coping with the matrix inverse and the possibility that $Q_z^T Q_z$ may be singular, we need to adopt recursive formulas which are presented below to compute the LSE of W [12,13,18,19].

$$W_z^{t+1} = W_z^t + \gamma_{t+1} P^t q^{t+1} \left[d_z^{t+1} - q^{t+1^T} W_z^t \right] \tag{20}$$

$$P^{t+1} = P^t - \gamma_{t+1} P^t q^{t+1} q^{t+1^T} P^t \tag{21}$$

$$\gamma_{t+1} = \frac{1}{1 + q^{t+1} q^{t+1^T} P^t} \tag{22}$$

where $t = 0,1,\dots,N-1$, P^t is the covariance matrix, and W_z^N equals to the least squares estimate W_z^*. The initial conditions are $W_z^0 = 0$ and $P^0 = \gamma I$, where I is the identity matrix of dimension $M \times M$ and γ is a positive larger number.

Based on the discussion above, we can use the hybrid algorithm which combines LSE method and BP algorithm together to tune the neuro-fuzzy controller for the rope-driven self-leveling device. At first, we use the LSE method to train the neuro-fuzzy controller; after the RMSE curve does not change obviously, then we can start the BP training process until a satisfactory performance can be achieved.

IV. Experimental Results

In our experiment, we use 492 experiential data pairs to train the neuro-fuzzy controller. The initial membership functions of NB, NS, ZR, PS, PB for θ_x, θ_y are shown in Fig. 4 (dashed black line). And, W_z $(z = x, y)$ is set to be zero. Fig. 5 demonstrates the RMSE (root mean squared error) curve in the training process. The first 5 steps uses LSE method, and the last 20 steps utilize the BP algorithm, in which the learning rate α_w =0.1; α_m =0.001; α_σ =0.0001.

After being trained, the rule tables for u_x and u_y are shown in Table I and II, respectively. And, the trained membership functions of NB, NS, ZR, PS, and PB for θ_x, θ_y are also shown in Fig. 4 (solid red line). The control surfaces of the neuro-fuzzy controller for u_x and u_y are shown in Fig. 6 (a)-(b). Then, the trained neuro-fuzzy controller is used for the leveling adjustment of the rope-driven self-leveling device. Fig. 7 gives control results of the neuro-fuzzy controller from two experiments where θ_x and θ_y have the different initial angles (3.39°/-1.85° and 5.61° / 4.05°) with the opposite sign and the same sign, respectively. From Fig.

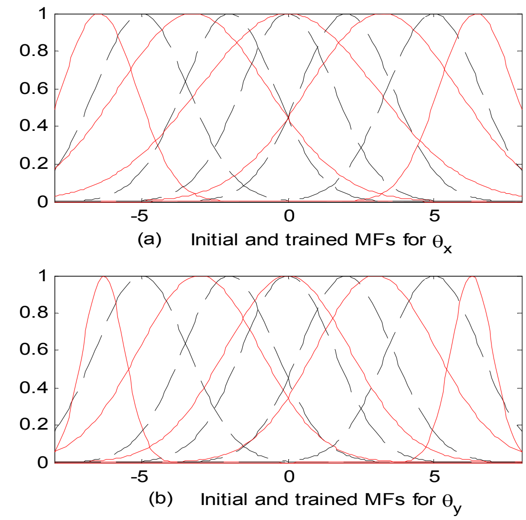

Fig. 4. Initial and trained membership functions for θ_x and θ_y.

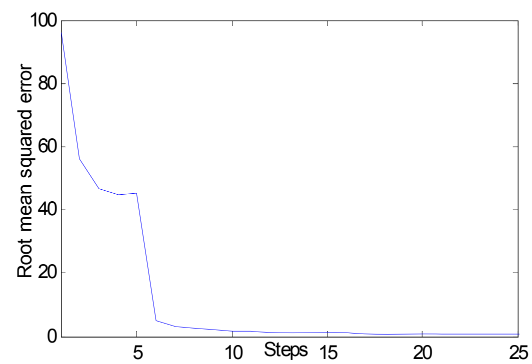

Fig. 5. RMSE curve in the training process.

TABLE I RULE TABLE FOR u_x

u_x(mm)		θ_x				
		NB	NS	ZR	PS	PB
θ_x	NB	99.61	74.99	21.84	72.92	112.78
	NS	65.86	79.87	28.74	80.29	62.86
	ZR	4.13	1.51	-0.94	-0.85	-1.62
	PS	-66.41	-80.66	-27.26	-80.31	-65.25
	PB	-103.4	-71.47	-20.95	-74.66	-112.62

TABLE II RULE TABLE FOR u_y

u_y(mm)		θ_y				
		NB	NS	ZR	PS	PB
θ_y	NB	96.14	75.38	1.62	-70.75	-110.66
	NS	65.61	77.33	-0.11	-78.26	-66.13
	ZR	17.31	10.73	-0.22	-11.78	-14.35
	PS	68.22	79.08	0.93	-76.47	-66.02
	PB	100.61	72.32	-0.96	-77.62	-108.85

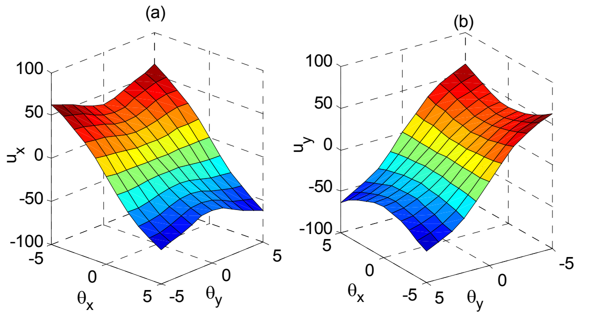

Fig. 6. Control surfaces of the neuro-fuzzy controller for u_x (a) and u_y (b).

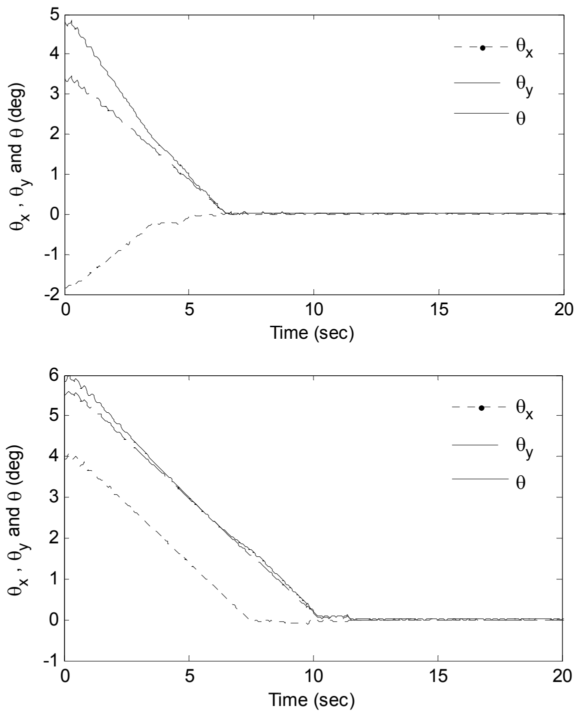

Fig. 7. Control results of the neuro-fuzzy controller for θ_x , θ_y , and θ .

7, we can see that the precision of the junction surface angle θ can approach almost to 0° when θ_x and θ_y are adjusted to be very small, and the adjusting time is much less than 60 seconds, i.e., the trained neuro-fuzzy controller can achieve the control objective—leveling adjustment of the rope-driven self-leveling device.

V. Conclusion

The paper proposes a rope-driven self-leveling device for leveling adjustment, then a neuro-fuzzy controller whose rules are constructed based on human's regulating experience is designed. We use a hybrid algorithm, which is a combination of the least square estimate (LSE) method and the back-propagation (BP) algorithm, to tune the parameters of the designed neuro-fuzzy controller to obtain better performance. Experimental results have demonstrated that the neuro-fuzzy controller can achieve the leveling adjustment of the rope-driven self-leveling device, and is easily applied to practical device. Balance control of the pulling forces has not yet been considered in this paper. How to realize the synchronization control of the leveling adjustment and the pulling force balance is a significant challenge and will be a focus of future work.

References

[1] K. Etemadi-Zanganeh and J. Angeles, "Real-time direct kinematics of general six-degree-of-freedom parallel manipulators with minimum sensor data," *Journal of robotic systems*, vol. 12, pp. 833–844, 1995.

[2] A. Ming, M. Kajitanti and T. Higuchi, "On the design of wire parallel mechanism," *International Journal of the Japan Society for Presion Engineering*, vol. 29, pp. 337–342, 1995.

[3] T. Arai, K. Yuasa, Y. Mae, K.Inoue, K. Miyawaki and N. Koyachi, "A hybrid drive parallel arm for heavy material handling," *IEEE Robotics & Automation Magazine*, vol. 9, 2002, pp. 45–54.

[4] S. Tamaki and M. Kato, "Cargo hanging device for crane," JP. Patent 19910297680, Oct. 18, 1991.

[5] C. B. Pham, S. H. Yeo, G. L. Yang, et al, "Force closure workspace analysis of cable-driven parallel mechanism," *Mechanical and Machine Theory*, vol. 41, pp. 53-69, 2006.

[6] J. W. Jeong, S. H. Kim, Y. K. Kwak, "Design and kinematic analysis of the wire parallel mechanism for a robot pose measurement", *Proceedings of 1998 IEEE Int. Conf. on Robotics and Automation*, 1998, 4: 2941–2946.

[7] R. L. Willams, P. Gallina, "Planar cable-direct-driven robots. part i: kinematics and statics," *Proceedings of ASME Design Engineering Technical Conferences*, PA, 2001.

[8] R. L. Willams, P. Gallina, "Planar cable-direct-driven robots. part ii: dynamics and control," *Proceedings of ASME Design Engineering Technical Conferences*, PA, 2001.

[9] J. Zhang, D. Zhao, J. Yi and Y. Yu, "Modeling of a rope-driven self-levelling crane," *in 3rd International Conference on Innovative Computing Information and Control*, China, pp. 489-489, 2008.

[10] L.-X. Wang, *Adaptive Fuzzy System and Control: Design and Stability Analysis*. NJ: Prentice-Hall, Englewood Cliffs, 1994.

[11] L. A. Zadeh, G. J. Klir and B. Yuan, *Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems*. World Scientific Pub Co, 1996.

[12] J.-S. R.Jang, C. T. Sun, and E.Mizutani, *Neuro-Fuzzy and Soft Computing*. NJ: Prentice-Hall, Englewood Cliffs, 1997.

[13] J.-S. R.Jang, "Anfis: Adaptive-network-based fuzzy inference system," *IEEE Trans. Syst., Man, Cybern*, vol. 23, pp. 665–684, 1993.

[14] G. Leng and G. Prasad, "Design for self-organizing fuzzy neural networks based on genetic algorithms," *IEEE Trans. Fuzzy Syst.*, vol. 14, pp. 755–766, 2006.

[15] G. Serra and C. Bottura, "An IV-QR algorithm for neuro-fuzzy multivariable online identification," *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 2, pp. 200–210, Apr. 2007.

[16] J. Zhang, "Modeling and optimal control of batch processes using recurrent neuro-fuzzy networks", *IEEE Trans. Fuzzy Syst.,* vol. 13, no. 4, pp. 417–427, Aug. 2005.

[17] Jeong, Kwang Son Hong, You Sik Park, Chong Kug, "Neuro-fuzzy control of a robot manipulator for a trajectory design," *IEEE International Workshop on Robot and Human Communication*, pp. 260–265,1997.

[18] H. Simon, Adap*tive filter theory (3rd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[19] O. Nelles, *Nonlinear system identification.* New York: Springer-Verlag, 2001.