2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Fuzzy Control of a Four-rope-driven Level-adjustment Robot Considering all Constrained Situations

Jianhong Zhang, Jianqiang Yi, *Senior Member, IEEE,* Xiangmin Tan, and Yi Yu

Abstract—**In the paper, we aim to design a controller for the four-rope-driven level-adjustment robot to adjust eccentric payload to level and keep the rope tension balanced. As the robot's actuators can only move in a certain range, it is necessary for the controller to judge whether the actuators' movement becomes constrained or not. Further, different control strategy should be taken to handle different situations. The controller is composed of two control modules, each of which regulates one diagonal of the payload's upper surface. Each control module consists of nine fuzzy sub-controllers, and each sub-controller deals with a particular situation. By selecting appropriate sub-controllers automatically, each control module can deal with different situations. Experiment results show that the controller is effective and practical.**

I. Introduction

NOWADAYS, there is still some work that has to be done manually due to lacking of appropriate tools. For example, many 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 docks, etc. Some payloads are so heavy, precise, valuable and fragile that they cannot endure point-to-point or line-to-line touch with the assembly platform or the transporting vehicles. For the sake of safety, it is necessary to level these eccentric payloads whose density is uneven and centers of gravity are different from their centers of geometry. Currently, these payloads are mainly adjusted level by manual level-adjustment. The efficiency is rather low; there is also potential danger for the payload as well as the operators, because it is hard to keep the rope tension balanced, and a rope may snap if it bears too much tension. Hence, an automatic level-adjustment device is urgently necessary to level those valuable eccentric payloads in industry.

As far as our knowledge, there are mainly three kinds of mechanism that can be used to level the payloads, that is, the weight-compensation mechanism, the link parallel platform, and the rope parallel platform. The weight-compensation mechanism regulates the eccentric payload to level by adjusting the compensating weights' positions [1]. As long as the payload's center of gravity is obtained, it is easy to calculate the desired positions of the compensating weights. However, it is always hard to obtain its center of gravity because the payload is always eccentric. The typical link parallel platform is the Stewart platform. Many researchers have studied its kinematics [2], dynamics [3], and workspace analysis [4], etc. But the link parallel platform is not suitable to adjust the fragile payload to level, because its workspace is limited and its weight is relatively too heavy.

Relatively speaking, the rope parallel platform's dynamics and kinematics are more complicated because of the rope's flexibility and nonlinearity. However, the rope platform has simple structure, light weight, fast regulating speed, and large workspace. It has been widely studied and used in various fields. A novel rapidly deployable cable based robotic system, which is capable of accurate positioning within its 3D span, is presented in [5]. [6] designs a prototype of a planar cable robot and presents approaches to design positive tension controllers for the cable suspended robots. A robust point-to-point position control method in the task-oriented coordinates for completely restrained parallel wire-driven robots, which form translational systems under zero-gravity conditions, is proposed in [7]. Then a novel 2 DOFs cable-driven robot with self-calibration capabilities and online drift-correction capabilities for planar translation is presented in [8]. [9] presents a rapid computation of optimally safe tension distributions for parallel cable-driven robots using linear-program. All these researches are good reference for future study, and greatly promote the application of the rope parallel platform. In the above-mentioned literature, however, the payload is usually seen as a particle and its dimension is omitted. But, we have to take the payload's dimension into consideration in order to level the payload. Therefore, we've designed a level-adjustment robot based on the rope parallel platform, shown in Fig.1.

In order to study the robot's characteristics, a simplified static model is established in [10], and some useful features are deduced. However, it is too complicated to establish its precise mathematical model, because the payload's center of gravity is always hard to obtain and the posture of both the robot and the payload will change if any rope's length changes. Therefore, Yu and Yi [11] design a hierarchical fuzzy controller that can adjust the payload to level and regulate the rope tension to be balanced. However, the research doesn't take the actuators' moving ranges into consideration, and the actuators may be requested to move to an unreachable position, which is absolutely unacceptable. So, an intelligent controller taking the actuators' moving ranges into consideration is designed in [12]. Yet, the controller can only adjust the payload to level and regulate the rope tension

This work was supported by the National 863 Program (No.2007AA04Z 239) and the NSFC Projects under Grant No. 60975060, China.

J. Zhang, J. Yi, X. Tan, and Y. Yu are all with the Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China (phone:86-10-6255 0985-15, e-mail: { jianhong.zhang&jianqiang.yi&xiangmin.tan }@ia.ac.cn; yuyisusong@gmail.com).

Fig. 1. Four-rope-driven level-adjustment robot

to be balanced under some simple situations. For instance, the controller can not handle such situation that the two actuators in a diagonal both move to their limit positions. In fact, the four actuators in the robot may move to their limit positions at the same time. Then, what should the controller do?

In the paper, we aim to design a practical controller for the robot. No matter how many actuators (at most four) move to their limit positions, the controller should regulate the payload to level and keep the rope tension balanced. The new controller consists of two control modules, each of which regulates one diagonal of the payload's upper surface. And each module includes nine fuzzy sub-controllers designed to deal with different situations. According to current system state, each module selects an appropriate sub-controller. Finally, some experiments are done to verify the controller's performance.

II. Working Principle of the Robot

The four-rope-driven level-adjustment robot is mainly composed of an industrial PC, a motion control card, four linear actuators, two angle sensors, four tension sensors, four independent ropes, a supporting plane, and several power supplies. Each linear actuator includes one step motor, one power amplifier, one coupling, one flange, one linear motion unit, and two limit switches. The layout of the robot's actuators is schematically shown in Fig 2. Unlike the other rope parallel platforms, the linear actuator is adopted as the actuator instead of the motor/winch actuator combined with external encoders. Relatively speaking, the regulation speed of the linear actuator is slow, but the regulation accuracy of the linear actuator is much higher and the required maximum motor torque is much smaller. For the level-adjustment of the valuable eccentric payload, the regulation accuracy is much more important than the regulation speed, so the linear actuator is adopted in the robot.

Generally speaking, the payload's posture as well as the rope tension can be regulated by changing four ropes' lengths. Each tension sensor is installed between a rope and the payload. Thus, the rope tension can be detected by the tension sensor in real time. And the four tension sensors connect to the payload by four hanging points, which symmetrically

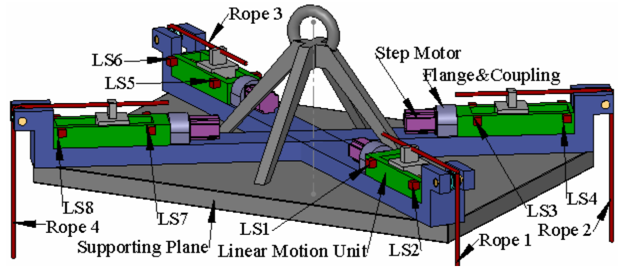

Fig. 2. Schematic diagram of the actuators' layout

distribute at the payload's upper surface and form a rectangular. Thus, the payload's posture can be detected by two angle sensors installed separately at the two diagonals of the rectangular. Hence, the payload is not necessarily a cuboid, and it can also be hexahedron, cylinder and other shapes. The industrial PC makes control commands according to the data of the angle sensors and tension sensors. And the motion control card, installed in the industrial PC's motherboard, transforms the control commands into electrical signals to drive the step motors with the help of power amplifiers. Afterwards, the linear motion units, grouping with the step motors by the couplings and flanges, transform the motors' rotation to each rope's linear movement. Thus, the rope length can be changed by the linear motion units' movement. As the linear motion unit's moving range is limited, two limit switches are fixed at two ends of each linear motion unit. Once the linear motion unit moves to one end, the corresponding limit switch will turn on. And the working principle of the robot is shown in Fig. 3.

III. Control Strategy

A. Control Objectives

In order to clearly describe the robot, some symbols are defined. The rope is represented by R_i (*i=1, 2, 3, 4*), and its length and tension can be described as L_i and F_i respectively. The angles, formed by the horizontal plane with the two diagonals of the rectangular composed of the four hanging points, are described as θ_x and θ_y respectively. In the case shown in Fig.4, θ_x is defined as positive. θ_x, θ_y, and F_i can be detected by the sensors in real time. Furthermore, the four ropes' average tension can be defined as:

$$F_{AV} = \sum_{i=1}^{4} F_i / 4 \qquad (1)$$

The relative deviation between F_i and F_{AV} can be defined as:

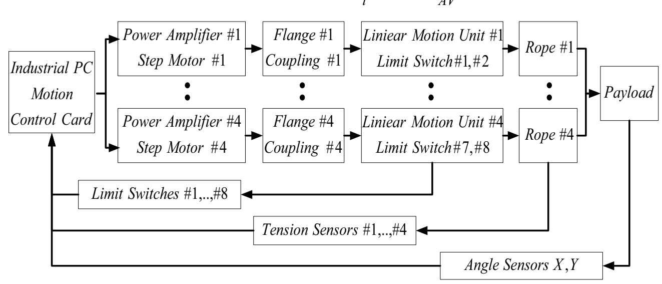

Fig. 3. Working principle of the robot

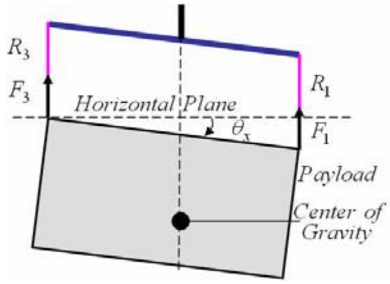

Fig. 4. Relationship of θ_x and the payload's center of gravity

$$\varepsilon_i = (F_i - F_{AV}) / F_{AV} \quad (i = 1, 2, 3, 4). \qquad (2)$$

Generally speaking, there are two basic control objectives for the level-adjustment robot. The first one is to regulate the payload to level. If both $|\theta_x|$ and $|\theta_y|$ are smaller than 0.05^o, the payload is considered as level enough. The second control objective is to keep the rope tension balanced, which can be quantified as $|\varepsilon_i| \le 30\%$. That is because a rope may snap if it bears too much tension. On the contrary, if a rope bears little tension, the other ropes will bear too much tension.

B. Normal Control Strategy

As we've mentioned before, the payload's posture can be regulated by changing L_i (*i=1,..,4*) according to θ_x, θ_y, and F_i (*i=1,..,4*). That is to say, the controller should determine how long each rope should be changed according to the six inputs. However, according to operating experience, θ_x (θ_y) can be independently regulated by mainly changing L_1 and L_3 (L_2 and L_4), although θ_x and θ_y are interactional. Thus, we can regulate the two diagonals of the payload's upper surface respectively. According to geometry knowledge, as long as both diagonals of the payload's upper surface are regulated to level, the payload will be level.

As shown in Fig.4, if θ_x is positive, we can either shorten R_1 or loosen R_3. If R_1 is shortened, F_1 will inevitably become bigger. Meanwhile, F_3 will also become bigger even though R_3 isn't changed. Namely, any change of one rope length will render the change of the other ropes' tension. According to regulating experience, $F_1 + F_3$ will be almost constant when both $L_1 + L_3$ and $L_2 + L_4$ don't change; $L_1 + L_3$ ($L_2 + L_4$) is almost inversely proportional to $F_1 + F_3$ ($F_2 + F_4$). Hence, we should shorten R_1 and loosen R_3 simultaneously to reduce their impact on $F_2 + F_4$, although F_1 (F_3) will inevitably become bigger (smaller). In other words, if L_1 decreases some length (i.e. ΔL), L_3 should increase ΔL synchronously to reduce their impact on F_2 and F_4. In addition, if R_1 and R_3 bear too much tension, we can loosen R_3 and keep R_1 unchanged to make R_2 and R_4 bear more tension.

C. Control Strategy with One Constrained Actuator in a Diagonal

However, the rope length can not be changed infinitely, because the linear motion unit's moving range is limited. Once a linear motion unit moves to one end, the corresponding limit switch will turn on and the linear motion unit can not move to the direction any further. How can we adjust the payload to level when some actuators move to their limit positions? For example, what should we do when limit switch #1(LS1) turns on?

As shown in Fig. 5, although LS1 is on, θ_x can still be regulated to zero by loosening R_1 and shortening R_3 simultaneously if θ_x is negative. The control strategy is the same as the normal control strategy. But, if θ_x is positive, we can only loosen R_3 to regulate θ_x to zero, which will cause R_2 and R_4 to bear more tension. To reduce $F_2 + F_4$, we can loosen R_4 and keep R_2 unchanged if θ_y is positive, and vice versa; or we can loosen R_2 and R_4 simultaneously if θ_y is zero.

D. Control Strategy with Two Constrained Actuators in a Diagonal

In some situations, both actuators in a diagonal may simultaneously arrive to their limit positions. As shown in Fig.6, neither R_1 nor R_3 can be shortened any more when both LS1 and LS5 turn on. As a result, we can only loosen R_3 and keep R_1 unchanged if θ_x is positive, or loosen R_1 and keep R_3 unchanged if θ_x is negative. However, both measures will cause R_2 and R_4 to bear more tension. Thus, we have to adjust R_2 and R_4 to keep the rope tension balanced. The control strategy is similar when both LS2 and LS6 turn on.

However, the payload can not be regulated to level in some extreme situations. As shown in Fig.7, if θ_x is positive, there is no way to adjust the payload to level under the existing condition when both LS1 and LS6 turn on, because R_1 can not be shortened any more and R_3 can not be loosened any more. Namely, the payload is so slantwise that it is out of the robot's adjustable range. Fortunately, this kind of extreme situations rarely occur. However, if θ_x is negative, we can still regulate θ_x to zero by loosening R_1 and shortening R_3.

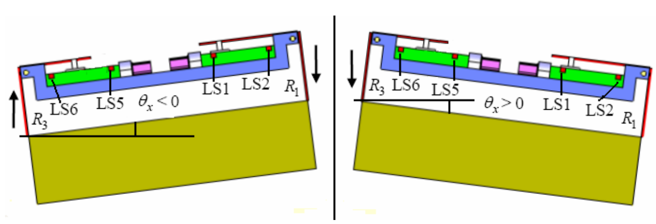

Fig. 5. Schematic diagram when only LS1 turns on

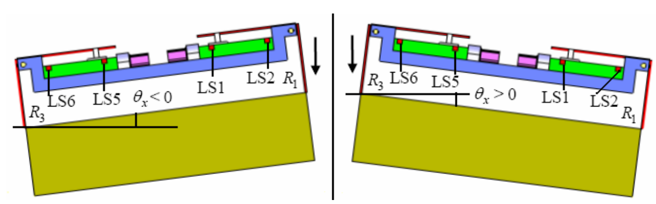

Fig. 6. Schematic diagram when both LS1 and LS5 turn on

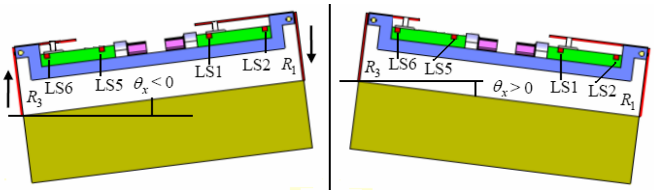

Fig. 7. Schematic diagram when both LS1 and LS6 turn on

IV. CONTROL DESIGN

As we've got much useful regulating experience, we try to regulate the robot with fuzzy logic. If the controller is designed directly, it will have six inputs (F_1 ,.., F_4 , θ_x and θ_y), and rule explosion is inevitable. However, according to the operating experience, even though θ_x and θ_y are interactional, θ_x (θ_y) could be independently adjusted by mainly changing L_1 and L_3 (L_2 and L_4). Thus, a controller, composed of two control modules, can be designed for the robot. According to θ_x , ε_1 , and ε_3 , control module X regulates θ_x as well as the rope tension by changing L_1 and L_3 , and the control module's outputs are ΔL_1 and ΔL_3 (change of the rope length L_1 and L_3). And control module Y, whose outputs are ΔL_2 and ΔL_4 , regulates θ_y according to θ_y , ε_2 , and ε_4 . Here, ε_i is adopted as the control module's input rather than F_i , because ε_i can clearly reflect the state of the rope tension. According to geometry knowledge, as long as both θ_x and θ_y are regulated to zero, the payload will be level. The control structure is shown in Fig.8. As the control module Y is similar to X, we take the design of the control module X as an example.

The control module X should judge if the linear motion unit #1 and #3 have moved to their limit positions by detecting the states of LS1, LS2, LS5, and LS6. This is because the linear motion unit may be damaged if it is forced to move to a position that is out of its moving range. For each specific situation, a fuzzy sub-controller should be designed. In fact, there are nine different situations to be managed, so nine fuzzy sub-controllers should be designed. The sub-controllers have the same inputs, same outputs, but different fuzzy rules to deal with different situations.

For example, the simplest situation is that neither linear motion unit #1 nor #3's movement becomes constrained. Another situation is either linear motion unit #1 or #3's movement becomes constrained (only one of LS1, LS2, LS5, and LS6 turns on). Other situation is the movements of both linear motion units become constrained (i.e. both LS1 and LS6 turns on). Obviously, different control strategy should be adopted to handle different situations.

Now, let's design the sub-controllers. Firstly, the membership functions of the sub-controllers' inputs and outputs should be defined. The fuzzy sets of θ_x are defined as: NB, NM, ZE, PM, PB, and their membership functions are shown in Fig.9. The fuzzy sets of ε_i (i=1, 3) are defined as: N, ZE, P, and their membership functions are shown in Fig.10. The

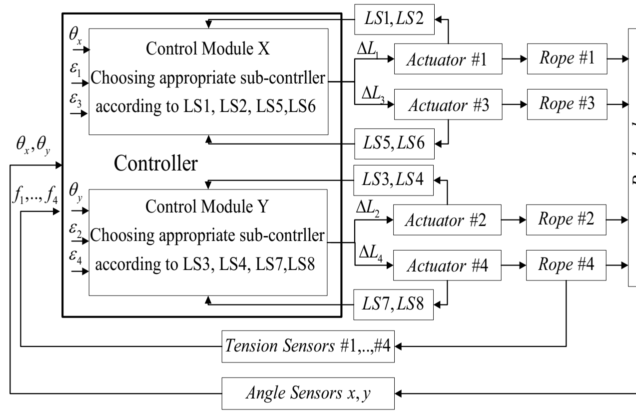

Fig. 8. Control structure

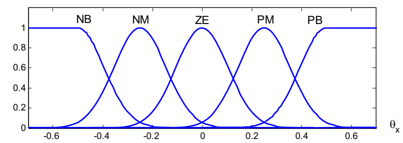

Fig. 9. Membership functions of θ_x 's fuzzy sets

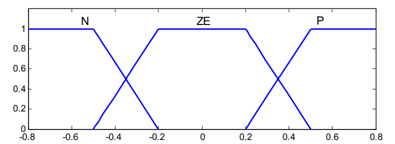

Fig. 10. Membership functions of ε_i 's fuzzy sets(i=1, 3)

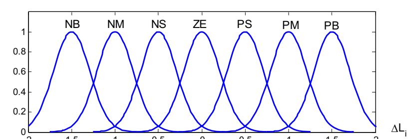

Fig. 11. Membership functions of ΔL_i 's fuzzy sets(i=1, 3)

fuzzy sets of ΔL_i (i=1, 3) are defined as: NB, NM, NS, ZE, PS, PM, PB, and their membership functions are shown in Fig.11.

Then, different fuzzy rules are designed to handle different constrained situations. There are three types of situations:

1) When the movement of neither linear motion unit #1 nor #3 becomes constrained, the normal control strategy can be adopted. In other words, they can both move to two directions. In this case, LS1, LS2, LS5, and LS6 are all off. The fuzzy rules (when θ_x is PB) are shown in Table I.

2) When the movement of either linear motion unit #1 or #3 becomes constrained, the control strategy should be modified comparing with the normal control strategy. In the case, only one of LS1, LS2, LS5, and LS6 turns on. For instance, R_1 can not be shortened when only LS1 is on. If θ_x is negative, we can still loosen R_1 and shorten R_3 simultaneously to regulate θ_x . Thus, if θ_x is NB or NM, the fuzzy rules are the same as the rules under normal situation. But if θ_x is positive, we can only loosen L_3 because R_1 can not be shortened any more. As a result, the fuzzy rules have to be modified when θ_x is ZE, PM, and PB. The modified rules (when θ_x is PB) are shown in Table II. Similarly, the fuzzy rules should

be also modified comparing with the normal situation when only LS2, LS5, or LS6 turns on.

3) When the movement of both linear motion unit #1 and #3 become constrained, the control strategy is different from the control strategy mentioned above. For instance, neither R_1 nor R_3 can be loosened when both LS2 and LS6 are on. The modified fuzzy rules (when θ_x is PB) are shown in Table III. Besides, because neither R_1 nor R_3 can be shortened when both LS1 and LS5 are on, the fuzzy rules is similar to the rules when both LS2 and LS6 are on. Further, when both LS1 and LS6 are on, R_1 (R_3) can not be shortened (loosened). In the case, if θ_x is positive, there is no way to adjust the payload to level. Yet, if θ_x is negative, we can still loosen R_1 and shorten R_3 . Thus, when θ_x is NB (NM), the fuzzy rules are the same as the rules when LS1, LS2, LS5, and LS6 are all off. However, when θ_x is ZE (PM, PB), both ΔL_1 and ΔL_3 should be ZE. When both LS2 and LS5 are on, the fuzzy rules are similar with the rules when both LS1 and LS6 are on.

Adopting the fuzzy rules, the fuzzy sub-controllers with product inference engine, singletom fuzzifier, and center average defuzzifier can be expressed as [13]:

$$\Delta L_i = \frac{\sum_{l=1}^{M} \overline{\Delta L_i}^{l} \left(\prod_{j=1}^{n} \mu_{A_j^l}(z_j) \right)}{\sum_{l=1}^{M} \left(\prod_{j=1}^{n} \mu_{A_j^l}(z_j) \right)} \quad (i = 1, 3), \tag{3}$$

where, M is the number of the fuzzy rules (M=45); n is the number of the fuzzy sub-controllers' inputs(n=3); z_j denotes the fuzzy sub-controllers' inputs(z_j represents θ_x , ε_1 , ε_3 respectively); ΔL_i denotes the fuzzy sub-controllers' outputs (ΔL_i represents ΔL_1 , ΔL_3 respectively); A_j^l denotes the fuzzy sets of the *j*th input variable in the *l*th fuzzy rule; $\overline{\Delta L_i}^{l}$ is the center of the output variable's fuzzy set in the *l*th fuzzy rule.

Additionally, $\Delta L_i > 0$ means L_i should decrease ΔL_i , and $\Delta L_i < 0$ means L_i should increase $|\Delta L_i|$. After designing the nine sub-controllers, the control module X can regulate θ_x under different situations by choosing appropriate sub-controllers.

Then, the complete control flow is shown in Fig.12. The controller consists of the control modules X and Y, each of which regulates one diagonal of the payload's upper surface as well as the rope tension. Firstly, the controller judges whether the payload needs to be regulated or not. If the payload is slantwise or the rope tension is not balanced, the controller judges whether the actuators' movement becomes constrained or not by detecting the limit switches' states. If any limit switch is on, the controller should judge whether the payload can be regulated or not. If not, the regulating process fails and the payload cannot be regulated to level under the existing conditions. If the payload can be regulated, the diagonal where no actuators' movement is constrained is regulated until the rope tension is balanced or some limit switches turn on in the diagonal. Then the other diagonal is regulated. As each control module can handle one diagonal of the payload's upper surface under all possible situations, the complete controller can deal with all constrained situations.

TABLE I
FUZZY RULES
WHEN LS1, LS2, LS5 AND LS6 ARE ALL OFF (θ_x IS PB)

ε_1 \ ε_3	N		ZE		P	
N	PB	NM	PB	NM	PM	NM
ZE	PB	NM	PB	NB	PM	NB
P	PB	NB	PM	NB	PM	NB
	ΔL_1	ΔL_3	ΔL_1	ΔL_3	ΔL_1	ΔL_3

TABLE II
FUZZY RULES WHEN ONLY LS1 IS ON (θ_x IS PB)

ε_1 \ ε_3	N		ZE		P	
N	ZE	NM	ZE	NM	ZE	NM
ZE	ZE	NM	ZE	NB	ZE	NB
P	ZE	NB	ZE	NB	ZE	NB
	ΔL_1	ΔL_3	ΔL_1	ΔL_3	ΔL_1	ΔL_3

TABLE III
FUZZY RULES WHEN BOTH LS2 AND LS6 ARE ON (θ_x IS PB)

ε_1 \ ε_3	N		ZE		P	
N	PB	ZE	PB	ZE	PM	ZE
ZE	PB	ZE	PM	ZE	PM	ZE
P	PM	ZE	PM	ZE	PS	ZE
	ΔL_1	ΔL_3	ΔL_1	ΔL_3	ΔL_1	ΔL_3

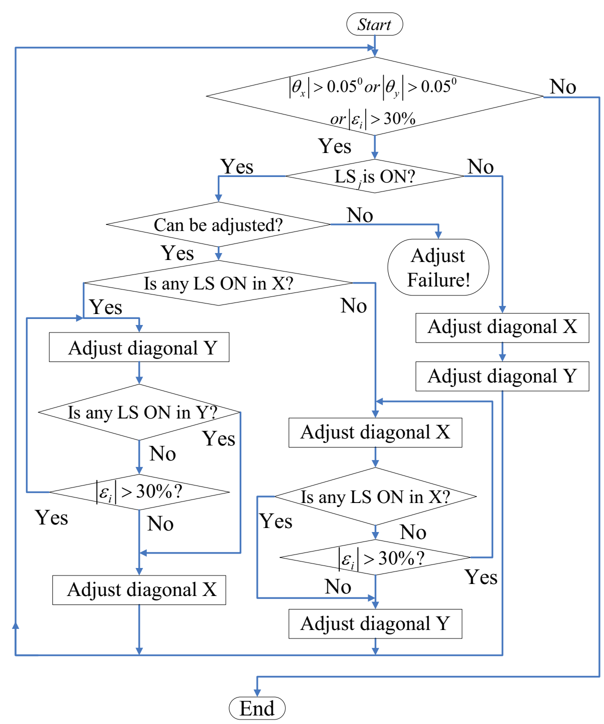

Fig. 12. Complete control flow

V. Experiment Results

Finally, some experiments are done to test the controller's performance, and a set of results are shown in Fig. 13. At the beginning, the payload is slantwise and the rope tension is extremely unbalanced - F_3 is even zero. Meanwhile, limit switches LS2, LS4 and LS8 are on, which means R_1, R_2 and R_4 cannot be loosened. In order to adjust θ_x to zero, we should shorten R_1 by ΔL_{R1} and loosen R_3 by ΔL_{R3}. However, in order to regulate the rope tension to be balanced, ΔL_{R3} should be smaller than ΔL_{R1}. R_1 and R_3 are continuously regulated until LS6 turns on, then R_2 and R_4 are regulated. Afterwards, two diagonals of the payload's upper surface are adjusted alternately. From Fig.13, we can easily find that the controller's outputs are closely related to the limit switches' states. Namely, once a linear motion unit moves to its limit position, the controller will give out a reasonable output to avoid damaging the linear motion unit. Within less than a minute, the payload is regulated to level and the rope tension is also adjusted to be balanced.

VI. Conclusion

In the paper, a controller is designed for the four-rope-driven level-adjustment robot. The controller consists of two control modules, each of which is composed of nine fuzzy sub-controllers and can regulate one diagonal of the payload's upper surface as well as the rope tension. As the linear motion units' moving ranges are limited, the controller should judge whether any linear motion unit's movement is constrained or not by detecting the limit switches' states, because different control strategy should be taken to deal with different constrained situations. By choosing different sub-controllers, each control module can deal with different constrained situations. Thus, the controller can manage all possible constrained situations. Finally, experiment results show that the adjustment process is smooth, and the controller is effective and practical.

References

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

[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, Apr. 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. Bessala, P. Bidaud, and F. B. Ouezdou, "Analytical study of Stewart platforms workspaces," *Proceedings of the 1996 IEEE International Conference on Robotics and Automation,* USA, 1996, pp.3179-3184.

[5] P. H. Borgstrom, M. J. Stealey, M. A.Batalin, and W. J. Kaiser, "NIMS3D: A novel rapidly deplyable robot for 3-Dimensional applications," *Proceedings of the 2006 IEEE/RJS International Conference on Intelligent Robots and Systems,* China, 2006, pp.3628-3635.

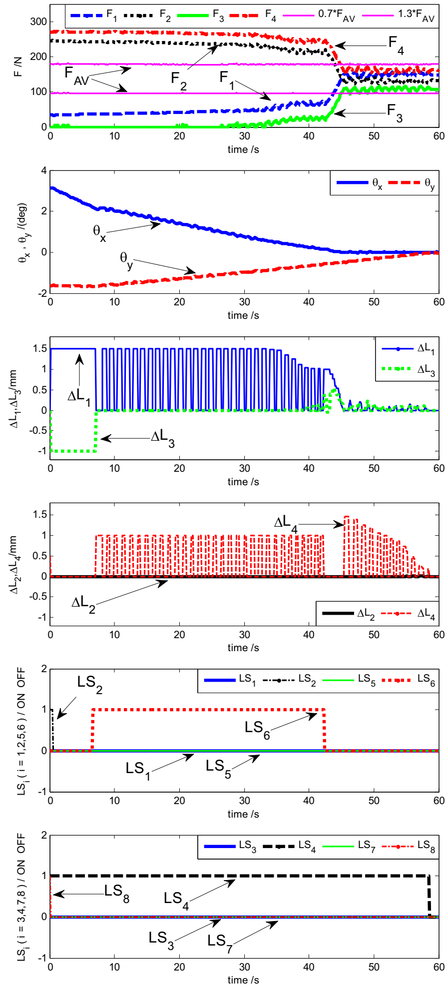

Fig. 13. Experiment results

[6] S.-R. Oh, and S. K. Agrawal, "Cable suspended planar robots with redundant cables: Controllers with positive Tensions," *IEEE Trans. on Robotics,* vol.21, no.3, pp.457-465, Jun. 2005.

[7] H. Kino, T. Yahiro, F. Takemura, and T. Morizono, "Robutst PD control using adaptive compensation for completely restrained parallel-wire driven robots: Translational systems using the minimum number of wires under zero-gravity condition," *IEEE Trans. on Robotics,* vol.23, no.4, pp.803-812, Aug. 2007.

[8] P. H. Borgstrom, B. L. Jordan, B. J. Borgstrom, M. J. Stealey, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser, " NIMS-PL:A cable-driven robot with self-calibration capabilities," *IEEE Trans. on Robotics*, vol.25, no.5, pp.1005-1015, Oct. 2009.

[9] P. H. Borgstrom, B. L. Jordan, G. S. Sukhatme, M. A. Batalin, and W. J. Kaiser, "Rapid computation of optimally safe tension distributinos for parallel cable-driven robots, " *IEEE Trans. on Robotics,* vol.25, no.6, pp.1271-1281, Dec. 2009.

[10] J. Zhang, D. Zhao, J. Yi, and Y. Yu, "Modeling of a rope-driven self-levelling crane," *Proceedings of the 3rd International Conference on Innovative Computing Information and Control,* China, 2008, pp.489-492.

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

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

[13] L. Wang, "Fuzzy systems are universal approximators," *IEEE Trans. on Fuzzy Systems,* vol.1, no.2, pp.1163-1170, 1992.