CONTROL OF A FOUR-ROPE-DRIVEN LEVEL-ADJUSTMENT ROBOT BASED ON THE SIRMs DYNAMICALY CONNECTED FUZZY INFEERENCE 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

Received May 2010; accepted July 2010

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.
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.
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 \( R_i \) \((i = 1, \ldots, 4)\). The two angles detected by the angle sensors, are expressed as \( \theta_x \) and \( \theta_y \) respectively. In the case shown in Figure 4, \( \theta_x \) is defined as positive. Furthermore, the average tension of the four ropes is defined as:

\[
F_{AV} = \frac{1}{4} \sum_{i=1}^{4} F_i \quad (1)
\]

Thus, the relative deviation of \( F_i \) and \( F_{AV} \) can be defined as:

\[
\epsilon_i = \frac{F_i - F_{AV}}{F_{AV}} \quad (i = 1, \ldots, 4).
\]

For the robot, there are two basic control objectives. Firstly, the payload should be regulated to level, which can be quantified as \( |\theta_x| \leq 0.05^\circ \) and \( |\theta_y| \leq 0.05^\circ \). 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 \( |\epsilon_i| \leq 30\%\).

According to operating experience, although \( \theta_x \) and \( \theta_y \) are interactional, \( \theta_x \) (\( \theta_y \)) can be regulated by mainly changing \( L_1 \) and \( L_3 \) \((L_2 \text{ and } L_4)\): \( F_1 \) (\( F_1 + F_3 \)) 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 \( F_1 + F_3 \).

![Positive situation of \( \theta_x \)](image)

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

![Figure 5. Schematic diagram when only LS1 turns on](image)

Furthermore, neither $R_1$ nor $R_3$ can be shortened any more if both LS1 and LS5 are on (Figure 6). Consequently, if $\theta_x$ is negative, $R_1$ should be loosened; if $\theta_x$ 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](image)

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 $\theta_x$ is still positive (Figure 7). Luckily, this extreme situation rarely happens. Of course, if $\theta_x$ is negative, we can still loosen $R_1$ and shorten $R_3$ simultaneously.

![Figure 7. When both LS1 and LS6 are on](image)

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 $\theta_x$ ($\theta_y$) 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 $\Delta L_1$ (change of $L_1$) to regulate $R_1$ according to $\theta_x$ and $\varepsilon_1$. The control
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 SIRM model. The SIRM 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 SIRM model. For systems of n inputs and 1 output, we firstly define a SIRM separately for each input as:

\[ SIRM-i : \{ R^j_i : \text{if } x_i = A^j_i \text{ then } f_i = c^j_i \}^m_{j=1} \]  

(3)

where SIRM-i denotes the SIRM of the ith input, and \( R^j_i \) 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 \( f \). \( A^j_i \) is the membership functions of \( x_i \), \( c^j_i \) is a real number. i is the index of the SIRM 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 \( f^0_i \) of the SIRM-i is given by:

\[ f^0_i = \left( \frac{\sum_{j=1}^{m} A^j_i(x_i)c^j_i}{\sum_{j=1}^{m} A^j_i(x_i)} \right) \]  

(4)

As each input plays an unequal role on the system performance, the SIRM model defines a DID \( w^D_i \) for each input \( x_i \) (i = 1, 2, ..., n) as:

\[ w^D_i = w_i + B_i \Delta w_i \]  

(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 \( \Delta w_i \), 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:

$$f = \sum_{i=1}^{n} w_i^D f_i^0.$$  \hspace{1cm} (6)

4.2. Setting the SIRMs. Here, each sub-controller has two inputs ($\theta_x$ and $\varepsilon_1$), one output ($\Delta L_1$). As we’ve mentioned before, $L_1$ should be shortened if $\theta_x$ is positive. $L_1$ should be loosened if $\varepsilon_1$ 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 $\theta_x$ and $\varepsilon_1$ are defined in Figure 9.

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

<table>
<thead>
<tr>
<th>Antecedent variable</th>
<th>Consequent variable</th>
<th>Antecedent variable</th>
<th>Consequent variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1(\theta_x)$</td>
<td>$f_1$</td>
<td>$x_2(\varepsilon_1)$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>NB</td>
<td>-1.0</td>
<td>NB</td>
<td>1.0</td>
</tr>
<tr>
<td>ZO</td>
<td>0.0</td>
<td>ZO</td>
<td>0.0</td>
</tr>
<tr>
<td>PB</td>
<td>1.0</td>
<td>PB</td>
<td>-1.0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Antecedent variable</th>
<th>Consequent variable</th>
<th>Antecedent variable</th>
<th>Consequent variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1(\theta_x)$</td>
<td>$f_1$</td>
<td>$x_2(\varepsilon_1)$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>NB</td>
<td>-0.5</td>
<td>NB</td>
<td>0.0</td>
</tr>
<tr>
<td>ZO</td>
<td>0.0</td>
<td>ZO</td>
<td>0.0</td>
</tr>
<tr>
<td>PB</td>
<td>0.0</td>
<td>PB</td>
<td>-1.0</td>
</tr>
</tbody>
</table>

**Figure 9.** Membership functions of $\theta_x$ and $\varepsilon_1$

**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 $|\theta_2| (|\varepsilon_1|)$ as the only antecedent variable, the fuzzy rules for the dynamic variables $\Delta w_1$ ($\Delta w_2$) of the DID $w_{\text{D}}$ ($w_{\text{F}}$) corresponding to $\theta_2$ ($\varepsilon_1$) are established in Table 3. Here, $|x_1|$ denotes $|\theta_2|$; $|x_2|$ denotes $|\varepsilon_1|$. The membership functions of DS, DM and DB are defined in Figure 10. Obviously, if $|\theta_2|$ ($|\varepsilon_1|$) is big, the inference result of $\Delta w_1$ ($\Delta w_2$) 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 $\theta_2$ is larger than that of $\varepsilon_1$. 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**

<table>
<thead>
<tr>
<th>Antecedent variable $x_i$</th>
<th>Consequent variable $\Delta w_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS</td>
<td>0.0</td>
</tr>
<tr>
<td>DM</td>
<td>0.5</td>
</tr>
<tr>
<td>DB</td>
<td>1.0</td>
</tr>
</tbody>
</table>

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.
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.

Acknowledgment. This work is partially supported by the National 863 Program (No. 2007AA04Z239) and the NSFC Projects under Grant No. 60975060, China.

REFERENCES


