Safe Simulation Of the Manipulator In the Presence Of Static and Dynamic Obstacles by Using Fuzzy System

Ali Davoodalhosseini¹, Saeed Behbahani¹

1 - Ph.D., Assistant Professor, School of Mechanical Engineering, Isfahan University of Technology, IRAN
a - ali_davoodalhoseini@yahoo.com

Keywords: path planning, manipulator, fuzzy controller, safe, Simulink, danger index.

ABSTRACT. Safe path planning is a necessary matter in human-robot interactions. This paper presents intelligent and safe path planning of the manipulator on a basis of the fuzzy system. This means that necessary commands for steering manipulator from departure to the destination are provided by the fuzzy controller. The fuzzy controller is a kind of controller that inspired by human decision on a basis of awareness and experience to steer the desired system. In this research, considered manipulator is located among the humans and static and dynamic obstacles. It should perform any state changing safely, in order to preserve health and tranquility of humans that work near the robot and also should avoid collision with obstacles. For this purpose, the velocity of changing of robot angle shouldn't exceed allowance. For achieving this goal, four factors are considered. These factors are known as a danger indexes including distance between human and robot, the velocity of the two links, human head orientation and changing face state. in the past researches, the researchers have not considered these danger factors in designing the manipulators. For monitoring these factors, the robot should be equipped with the different sensors. The proposed issue was simulated in Simulink environment in Matlab. Results show that the robot can avoid collision with the obstacles and arrives safely to a desirable position with the little acceptable error.

Introduction. In most existing work in autonomous navigation, a solution is attempted by separating the planning and control into two sequential stages. This may have some adverse effects. Robot motion planning with artificial potential field considers the problems of motion planning and control simultaneously. But in this research we don’t make ourselves involved in kinetic problems, i.e., the commands that our controller generates are the angle, not the momentum. Several autonomous systems have been developed using rule-based methods to control the motion of robot manipulator. Tsoukalas et al. [1] presented a neuro-fuzzy methodology for a robot to navigate in the dynamic environment. Ding and Li [2] solved the problem of obstacle avoidance for a redundant manipulator by using a fuzzy logic system. Mbede et al. [3] also designed neuro-fuzzy system in order to consider structured and unstructured uncertainty. Their robot can adapt itself by using neural network online learning. In aforementioned researches, no safety cases have been considered. Robots have been successfully employed in industrial settings to improve productivity and perform dangerous or monotonous tasks. Recently, research has focused on the potential for using robots to aid humans outside the strictly “industrial” environment, in medical, office or home settings. To this end, robots are being designed to perform homecare daily living tasks such as co-operative load carrying [4,5] and feeding [6] and to provide social interaction [7,8]. As robots move from isolated work cells to more unstructured and interactive environments, they will need to become better at acquiring and interpreting information about their environment [9]. One of the critical issues hampering the entry of robots into unstructured environments populated by humans is safety [10,11,12]. In particular, when the tasks of the interaction include manipulation tasks, such as picking up and carrying items [13] assisting with dressing, opening and closing doors, etc., large, powerful robots will be required. Such robots (e.g., articulated robots) must be able to interact with humans in a safe and friendly manner while performing their tasks. Industrial safety standards (RIA/ANSI [14]) focus on ensuring safety by isolating the robot away from humans, and are, therefore, not directly applicable to human-robot interaction applications. However, industrial experience has shown that eliminating hazards
through the mechanical design is often the most effective safety strategy. This approach has also been applied to interactive robots, for example, by applying a whole-body robot viscoelastic covering [15].

In order to prevent collisions, safeguarding type controllers execute a safety strategy if a person is detected within the work envelope of the robot. If a human is detected in the safeguarded zone, the default robot control sequence is altered to ensure the safety of the human [15,16]. These methods consider a fixed distance around the robot as the safeguarded zone, at which point the reactive controller performs a safety action. A more sophisticated approach is to develop a dynamically sized safeguarded zone, based on an implicit or explicit evaluation of the current danger, namely, a danger index. For example, Traver et al. [17] propose two human-friendly robotic designs. The “elusive robot” uses the distance between the robot and the human as the danger index. The “ergonomic robot” computes a danger factor based on the robot’s velocity and posture, the human’s direction of motion and eye gaze, and the rate of change of the distance between the robot and the human. The “ergonomic robot” is controlled to reduce the calculated danger index. Ikuta and Nokata [18] developed a danger evaluation method using the potential impact force as an evaluation measure. In their work, the danger index is defined as a product of factors that affect the potential impact force between the robot and the human, such as relative distance, relative velocity, robot inertia and robot stiffness. Several design examples are presented, but no control-based implementation of the danger index was presented. Both safeguarding and danger evaluation approaches propose that robot behavior is modified based on the human location and motion during human-robot interaction. The safeguarding approaches define discrete behaviors while the danger evaluation methods generate a continuum of behavior.

Croft et al. [19] introduced three types of danger including long term, medium term and short term danger. After that, they investigated the effect of each one, separately. They proved the effectiveness of their new danger index by experiment.

1. Robot dynamics. The equation of the dynamic of a manipulator robot is given by the equation (1):

\[
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F(\dot{q}) + \tau_d = \tau
\]  

\(q, \dot{q}\) and \(\ddot{q}\) consecutively represent the angle, angular speed and angular acceleration. \(M\) is the inertia matrix, \(C\) is the Coriolis matrix, \(G\) is the gravitational acceleration vector and \(F\) represents the friction term. \(\tau\) is the joint torque and \(\tau_d\) represents any disturbances or torque that is caused by a dynamic which is not modeled.

2. Robot model. There are two methods to simulate a robot in Matlab’s Simulink environment; one method is to use the equations of motion for a robot, and the other is to simulate the robot in the Simmechanic environment. In this study, the second method is used.
As such the input for the links can be kinematic variables as well as torques. In this study, we used the kinematic variables as inputs.

To simulate the servo motor we used a transfer function to create the delay. The function is as follows;

\[ G(s) = \frac{3}{s^2 + 3s + 3} \]  

(2)

This transfer function causes the input angle to slowly adjust to the desired value, and also the input angle will have a settling time and maximum overshoot. In practice, the motor in charge of providing the torque cannot supply the input torque instantaneously. The properties of the robot are given in the table below.

**Table 1. Properties of robot**

<table>
<thead>
<tr>
<th>Property</th>
<th>Link 1</th>
<th>Link 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length(m)</td>
<td>1</td>
<td>1.3</td>
</tr>
<tr>
<td>Mass(kg)</td>
<td>1.09</td>
<td>1.12</td>
</tr>
<tr>
<td>Inertia(kg*m^2)</td>
<td>0.0908</td>
<td>0.1577</td>
</tr>
</tbody>
</table>

3. Danger index. Danger index specifies the current level of danger when interacting with the environment and it is used when the robot has to correct its performance. Analyzing typical collisions of robots in industry shows that most accidents occur when the operator is not aware of the movements of the robot. Therefore when the operator’s line of sight is not in the direction of the robot and he is not able to see the robot’s movements the danger index should increase.

Two indexes are used for this problem, head rotation angle, and operator’s face gesture. The equation used for head rotation is a sigmoid function based on the horizontal rotation of the head.

The rotation angle zero indicates that the operator is looking at the robot. The sigmoid function was used to ensure the gradual adjustment of this function. Also the changes in the function when the head rotation angle is between the angles of -15 and +15 degrees, doesn’t have a big effect on the function.

Furthermore; when the rotation angle of the head is between 60 and 90 degrees the value of the function must be at its maximum because the robot is hardly visible at this degree. The change in the values of the function between the angles of 60 and 90 degrees must be very fast.

\[ K_{OR} = 1 + \frac{M_{OR}}{1 + e^{-5S_{OR} (\theta_h - \theta_c)}} \]

(3)

\( K_{OR} \) is the head orientation scaling factor \( M_{OR} \) is the maximum increase in scaling due to the orientation \( S_{OR} \) controls the slope of the sigmoid function, \( \theta_h \) is the horizontal head orientation and \( \theta_c \) is the rotation of the head at the midpoint of the sigmoid.

\( M_{OR} = 2 \), \( S_{OR} = 0.2 \), \( \theta_c = 30 \)

Similarly, a sigmoid function was used for the affective state.
The excitation of the face has a direct relationship with an individual's anxiety and excitement. A sigmoid function is used so that it can filter the low excitation gestures, since estimating the excitation of the face is approximate.

\[
K_{AS} = 1 + \frac{M_{as}}{1+e^{S_{as}(a-c)}}
\]  

(4)

\(K_{as}\) is the affective state scaling factor \(\cdot\) \(M_{as}\) is the maximum increase in scaling due to the affective state \(\cdot\) \(S_{as}\) controls the slope of the affective state sigmoid function, \(ac\) is the midpoint of the arousal scale.

\(\text{Max} = 2\), \(S_{as} = 0.2\), \(a_{c} = 0.5\)

For each link, the closest point to the operator is called the critical point. Danger index is calculated for each critical point, this function also includes the distance between the robot and operator at a critical point. The distance coefficient is calculated by the following equation.

\[
F_D(s) = K_D \left( \frac{1}{s} - \frac{1}{D_{\text{max}}} \right)^2 \text{ for } S \leq D_{\text{max}}
\]

(5)

It is clear that if \(S \geq D_{\text{max}}\), \(F_D(s)\) equals to zero.

\(S\) is the critical distance with the closest person. The constant \(K_D\) is used to increase this coefficient. When the distance between the critical point and the closest person is greater than the specified value \(D_{\text{max}}\), the value of the function is zero. When this distance is at its minimum the value of the function is one. The value of more than one indicates unsafe conditions [19].

The other index is the speed index. If the speed exceeds a certain value (15 degrees per second), as the output of index, the value of speed will be multiplied by 0.1. By multiplying the 4 factors and multiplying the resulted value by 0.01, the total danger index is calculated. This value will be reduced from the input torque. Depending on the level of safety which is needed the coefficient 0.01 is variable.

4. Different types of obstacles and their simulation. In this study two types of obstacles are used, dynamic and static. In practice, robot is equipped with several sensors which automatically send the distance to nearest obstacle to the control system in order to avoid hitting them. Sensors can also detect whether the obstacle is approaching or the robot has passed the obstacle and is getting further away.

However in the present study no tests were done and there was no actual robot involved. Therefore the distance between the two points of robot’s links and different obstacles are calculated at each instance using mathematical equations, and the nearest distance is used as the input for control system.

The both ends of robot’s links are the point that we tried to prevent from hitting any obstacles in the simulation. However because this assumption may cause other parts of the link to hit obstacles, the minimum allowed distance specified is about 0.5 meter. Also in this study, we differentiated between moving close short distance and moving away a short distance, considering the situation in which the distance is close, but the obstacle is moving away from the robot. In such situation, the distance between the obstacle and robot should not affect the control system’s decision. To nullify the effect of this situation we specified a large value for the distance (4.9 meters). To simulate obstacles ramp function was used. To accomplish this, a point which is moving away from the departure zero point is placed on the robot’s links movement path.

5. Control structure. Two fuzzy controllers with a slight difference were used for two links. Designed fuzzy controller includes two inputs and one output. First input is an error of angle (needed
angle to reach to the desired angle) and the second input is the distance to obstacles. Also, Output of the controller is the angle. The desired angle for the first link is 120 degrees and the desired angle for the second link is -60 degrees. Properties of inputs and output and table of rules are shown consecutively in table 2 and 3.

5.1. Fuzzification. The angle error $e$ is partitioned into nine fuzzy sets: right very very big (R2VB), right very big (RVB), right big (RB), right (R), zero (Z), left (L), left big (LB), left very big (LVB), and left very very big (L2VB). Its fuzzy membership functions are symmetric and shown in Figure 3(a). The distance to obstacle $d$ is partitioned into six fuzzy sets: VVnear, Vnear, near, good, far, Vfar. Its fuzzy membership functions are shown in Figure 3(b). The output angle $\theta$ is partitioned into eleven fuzzy sets: right very very big (R2VB), right very big (RVB), right big (RB), right (R), right small (RS), zero (Z), left (L), left small (LS), left big (LB), left very big (LVB), and left very very big (L2VB). Its fuzzy membership functions are symmetric and shown in Figure 3(c).

5.2. Rule Base. The rule base is generalized as follows:

$$R^i: \text{if } e(k) = \mu_1^i (e(k)) \text{ and } \ldots \text{ and } e(k-n+1) = \mu_{n-1}^i (e(k-n+1)) \text{ and } d(k) = \mu_1^i (d(k)) \text{ and } \ldots$$

$$\ldots \text{and } d(k-m+1) = \mu_{m-1}^i (e(k-m+1)) \text{ Then } F_i(k+1) = r_i.$$  

where $R^i (i = 1, 2, \ldots, i)$ denotes the $i$th implication, $1$ is the number of fuzzy rules, $r_i$ is the output from the $i$th implication, $n$ is the number of input variable $e$, and $m$ is the number of input variable $d$. Our fifty four rule bases are arranged into a look-up table and shown in Table 3. The two inputs $\mu(d)$ and $\mu(e)$ represent the fuzzy sets, which indicate the distance between robot and obstacle, and the position error, respectively. The outputs of the base are $F_j$ which describe the angle output. For example, the rule 1 is:

$$R^1: \text{If } d \text{ is VVnear and } e \text{ is R2VB Then } F_1 \text{ is LB}$$

---

**Fig. 2. Control structure**

Figure 2 shows the structure of the control in the Simulink environment.
Table 2. Properties of inputs and the output of the Fuzzy controller

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Number of membership functions</th>
<th>Membership functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input-angle error (e)</td>
<td>-180-180</td>
<td>9</td>
<td>R2VB-RVB-RB-R-Z-L-LB-LVB-L2VB</td>
</tr>
<tr>
<td>Input-distance to obstacle (d)</td>
<td>0-5</td>
<td>6</td>
<td>Collision-Vnear-near-good-far-Vfar</td>
</tr>
<tr>
<td>Output-angle(θ)</td>
<td>-180-180</td>
<td>11</td>
<td>RVVB-RVB-RB-RS-RVS-Z-LVS-LS-LB-LVB-LVVB</td>
</tr>
</tbody>
</table>

Table 3. Rules of the Fuzzy controller

<table>
<thead>
<tr>
<th>e/d</th>
<th>VVnear</th>
<th>Vnear</th>
<th>near</th>
<th>good</th>
<th>far</th>
<th>Vfar</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2VB</td>
<td>LB</td>
<td>LS</td>
<td>RSV</td>
<td>RVB</td>
<td>RVB</td>
<td>RVB</td>
</tr>
<tr>
<td>RVB</td>
<td>LB</td>
<td>LS</td>
<td>RSV</td>
<td>RB</td>
<td>RB</td>
<td>RB</td>
</tr>
<tr>
<td>RB</td>
<td>LB</td>
<td>LS</td>
<td>RSV</td>
<td>RS</td>
<td>RS</td>
<td>RS</td>
</tr>
<tr>
<td>R</td>
<td>LB</td>
<td>LS</td>
<td>RSV</td>
<td>RVS</td>
<td>RVS</td>
<td>RVS</td>
</tr>
<tr>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>L</td>
<td>RB</td>
<td>RS</td>
<td>LSV</td>
<td>LVS</td>
<td>LVS</td>
<td>LVS</td>
</tr>
<tr>
<td>LB</td>
<td>RB</td>
<td>RS</td>
<td>LSV</td>
<td>LS</td>
<td>LS</td>
<td>LS</td>
</tr>
<tr>
<td>LVB</td>
<td>RB</td>
<td>RS</td>
<td>LSV</td>
<td>LV</td>
<td>LV</td>
<td>LV</td>
</tr>
<tr>
<td>L2VB</td>
<td>RB</td>
<td>RS</td>
<td>LSV</td>
<td>LVB</td>
<td>LVB</td>
<td>LVB</td>
</tr>
</tbody>
</table>

First row and first column consecutively determine membership function of d (distance to obstacle) and e (error of angle). Other cells determine output of fuzzy system.
Fig. 3. Membership function plot for inputs and output, (a) – input-e, (b) – input-d, (c) – output-
6. Results and graphs.

Fig. 4. Angle of the links (Degree Vs. Second) (a)-First link, (b)-Second link

Fig. 5. Velocity of the links (Degree/S Vs. Second) (a)-First link, (b)-Second link

Fig. 6. Distance between links and obstacles (Meter Vs. Second), (a)-First link, (b)-Second link

As it can be seen in the diagrams, both of the links’ distances with obstacles are greater than the specified value (0.5 meters) and no accident occurred. Parts of the diagram which are fixed on 4.9 meters are points in which the distance between links and obstacles are more than 4.9 meters, but the fuzzy controller input send the value of 4.9; which is a relatively large distance; to the fuzzy controller
input. Because the values greater than 4.9 are outside of the range specified for this study. The number 1 and 2 links reach their specific angles in approximately 5 seconds that suggests robot’s high speed. Although the link number 1 has a 5 degrees error margin.

![Graph A](image1.png)

![Graph B](image2.png)

![Graph C](image3.png)

![Graph D](image4.png)

*Fig. 7. Danger indexes, (a)-Head orientation and Affective state, (b)-Human closing, (c)-Velocity of link1, (d)-Velocity of link2*

Because at each instance one of the danger indexes is zero the total danger index is always zero which means that the individuals in the vicinity of the robot are not in danger.

**Summary.** In this study, a manipulator robot is simulated using a fuzzy system. Also, dynamic and static obstacles were modeled. Furthermore to consider safety measures four different factors were considered as danger indexes which multiplying these four factors together and by 0.01 gives the total danger index.

The results show that the robot while avoiding static and dynamic obstacles reaches its desired angle with proper speed and low error margin. After about 5 seconds the link number one reaches the 125 degrees angle. Despite the fact that the desired angle for link number one was 120 degrees. Link number 2 reaches the desired angle although it has a slight oscillation at -60 degrees. Results show that the amplitude of the oscillation is reduced over time. Therefor, fuzzy system can be recommended in order to control the path of a robot. For future works, it is recommended to use a neural network in order to overcome uncertainty and also consider some conditions like using damping term to neutralize oscillation near desirable position.

**References**


