Contact States Estimation Algorithm Using Fuzzy Logic in Peg-in-hole Assembly*

Haeseong Lee\(^1\) and Jaeheung Park\(^{1,2}\)

Abstract—Peg-in-hole assembly is regarded as one of the essential tasks in the robotic assembly. To complete the task, it is required to estimate the Contact State (CS) of a peg relative to a hole and control motions of the peg in the environment. In this paper, we propose the estimation algorithm using fuzzy logic for the satisfaction of these requirements. Firstly, we describe a peg-in-hole environment, which has holes with several sizes on the surface with a fine area. Afterward, we classify the CS of the peg in the environment. Secondly, we explain and analyze the proposed algorithm and a motion control method. Using the proposed algorithm, we can estimate all the CS. After estimating the current CS, proper actions are commanded for the peg-in-hole assembly. To validate the proposed algorithm, we conducted an experiment using a 7 DOF torque-controlled manipulator and prefabricated furniture.

I. INTRODUCTION

Peg-in-hole assembly is one of the important applications of robotics that has been studied for decades [1]. The peg-in-hole assembly is found not only in the industrial area but also in daily life. It is still a difficult craft for a robot to insert a peg into a hole due to several uncertainties around the assembly environment. Therefore, there is a need for a strategy to accurately estimate CS of the peg in the peg-in-hole environment and control the robot to cope with uncertainties for the successful peg-in-hole assembly. To do this, CS estimation using fuzzy logic and a force control method prove to the appropriate solutions to overcome uncertainties during the peg-in-hole assembly. For instance, a peg-in-hole assembly was demonstrated using fuzzy logic control strategies [2], [3]. Based on 196 fuzzy rules related to force and moment, the relative distance between a peg and a hole is estimated [4]. To insert a peg into a hole, a strategy is suggested by estimating an angle error between the peg and the hole based on 16 fuzzy rules [5]. Five different CS of a peg were studied to insert the peg into a hole by four fuzzy rules [6].

There are different approaches to recognize the assembly environment using additional devices such as a camera. The relative location between a peg and a hole is recognized by a hand-eye system [7]. Since there exist blind spots when the sight of the camera is blocked by the end-effector, it is still required a search motion near the target hole. Researchers suggested a strategy to estimate the pose of the end-effector after detecting the target object [8], [9]. However, these vision systems are not always reliable especially when the environment is affected by illumination. Thus, it will be an adverse condition to estimate the CS of the peg during the peg-in-hole assembly without using physical information such as velocity and reaction force.

From other works([10], [11], [12], [13]), Some researches consider only a single pair of the peg and hole which are fit in size([2]-[13]). Also, they did not consider a case when the peg deviates from the search area. In this paper, we propose an estimation algorithm for the CS of the peg together with unintended failure situations such as jamming in a smaller-size hole or deviating from a top surface. Firstly, we describe a peg-in-hole environment and classify the CS between the peg and the environment. Also, we introduce fuzzy set parameters and fuzzy rules. These are useful in estimating the different CS of the peg. For robot control, the motion/force hybrid control method is used to create the motion that is similar to human motions in blind search. Using this force control, the end-effector of the manipulator generates a certain amount of force toward insertion direction while moving along a given trajectory on the top surface which is normal to the insertion direction. There we validate the proposed algorithm, by conducting experiments using a 7 DOF torque-controlled manipulator and wooden prefabricated furniture.

II. ANALYSIS OF CONTACT STATE

In this section, It describes the peg-in-hole environment. Also, six different CS between the peg and the environment is classified under the following assumptions:

A1. The peg and hole are cylinders and the peg is chamfered.
A2. The angle between the peg and the top surface is negligible.
A3. The clearance between the peg and hole is not tight.
A4. No relative motion between the end-effector and peg.
A5. The top surface area is finite.
A6. There exist possibilities to fail the peg-in-hole assembly.

A. Peg-in-hole Environment

Fig. 1(a) shows the peg-in-hole environment reflecting the aforementioned assumptions. There are several sizes of holes in the finite top surface. At the early stage of the assembly task, the peg moves along a given trajectory to search the target hole. Because the robot controller, which is explained in section IV, pushes the peg downward, the peg will go down to an empty space if the peg is located over any sinkage.
However, it does not guarantee that the sinkage is the desired hole since there are two additional situations. In other words, there exist 3 possible results in total. One is that the peg is inserted into a correct size hole due to A2 and A3. In another case, the peg is stuck in a smaller-size hole since the bottom part of the chamfered peg is narrower than the body of the peg (Fig. 1(a)). The other case is that a given trajectory deviates from the finite top surface so that the peg falls into unexpected space.

B. Contact States

Considering the peg-in-hole environment and possible results, Fig. 1(b) shows the classified CS. CS-0 means the peg is moving downward along the insertion direction. CS-I means the peg keeps contact on the top surface. CS-II means a point contact situation between the hole edge and the peg’s chamfered surface. CS-III means two points contact situation between the hole edge and the peg’s chamfered surface. Because of the narrow bottom diameter by the chamfered shape, it shows that the peg is stuck in the smaller-size hole. CS-IV means the peg is properly inserted into the desired hole. CS-V means the peg deviates the top surface due to the limited top surface. Besides, CS-V is separated into two cases based on geometric properties. The one, which is defined as CS-V1, is a situation in which the displacement of the peg is less than or equal to the target depth. Whereas CS-V2 is the opposite case of CS-V1. Especially, CS-II and CS-III are characteristic contact conditions caused by the chamfered shape of the peg. Although the chamfered shape guides the peg to be inserted into the hole, it triggers unsuitable results such as CS-III. Also, CS-V is one of the fatal cases that leads the assembly to fail. In the next chapter, the proposed algorithm which copes with all the specified situations is presented.

Please note that the proposed algorithm does not consider the case when the peg is inserted into a larger-size hole since the problem could be prevented if an assembly sequence is composed in descending order of peg’s size.

III. PROPOSED ESTIMATION ALGORITHM

In the real assembly environment, it is difficult to precisely control the position of the peg due to uncertainties such as control error. If the controller can deal with a certain amount of error, it would be expected better estimation performances. Since the fuzzy logic is able to satisfy this requirement, it is one of the proper methods to estimate the state of the peg. To establish the proposed algorithm, it consists of three steps which are called fuzzification, fuzzy inference rule base, and defuzzification.

A. Fuzzy Set and Fuzzification

The fuzzification is a mapping that converts real-world information to fuzzy sets which are sets of interesting input or output variables. Fuzzified values are computed between 0 and 1 using membership functions. The membership functions are functions which represent the degree of membership of the real-world information in the fuzzy sets. In the proposed algorithm, there are four fuzzy sets, including three inputs and an output (Fig. 2). Each input subset is represented as PVS, PV, PM, PL, and PVL which are abbreviations as below:

PVS : positive very small
PS : positive small
PM : positive medium
PL : positive large
PVL : positive very large

The input fuzzy sets are the velocity of the end-effector, the displacement of the end-effector, and the reaction force applied to the end-effector which are denoted as by V, Z, and Fz, respectively. The first two parameters are the values along the insertion direction and the last is the value orthogonal to the insertion direction (Fig. 1(a)). In the output fuzzy set, all the CS are included. In Fig. 2, v_i, z_i, f_i, and θ_i are singleton values of each fuzzy set. And they divide the range of each subset. The velocity parameter plays a role of distinguishing a stationary state and a dynamic state of the peg along the insertion direction. All the CS are composed in descending order of peg’s size.
estimated under the stationary state except CS-0 and CS-V2. The displacement parameter is based on the geometric dimensions of the given assembly components. For instance, there is a rule to presume CS-V2 when considering only the displacement of the end-effector. Because of the geometry of the peg-in-hole environment, it is possible to detect CS-V2 if the peg moves downward over the target depth. The force parameter aims at compensating for insufficient information from the velocity and the displacement data. For example, it is difficult to separate CS-I and CS-II with only two parameters, velocity, and displacement. This because the velocity and displacement difference between CS-I and CS-II are not obvious (see Fig. 3). However, the force graph shows that there are distinguishable patterns between CS-I and CS-II. During CS-II, the peg is affected by the reaction force from a contact point when compared to CS-I, which causes an increment of the reaction force. Thus, all the CS are estimated more accurately with these three parameters.

Additionally, the overlapped area between two membership functions (Fig. 2) can be adjusted by a designer with respect to a given assembly task. The cusps of each fuzzy input set have 1 or 100% degree of membership when the velocity values are $v_1$ and $v_2$, the displacement values are $z_1$, $z_3$, $z_5$, $z_7$, and $z_9$, while the force values are $f_1$, $f_3$, and $f_5$. The singleton values at the cusps of the fuzzy displacement input set are referenced by the dimension of the given furniture part, but others are deduced experimentally. For the singleton values of fuzzy output set, $o_i$ is set by arbitrary integer numbers.

### B. Fuzzy Inference Rule Base

Fuzzy rules are represented by the form of if A then B where A means conditions and B means results. There are AND and OR operations in the fuzzy inference mechanism. If $u_{and}$ and $u_{or}$ are the results of each operator, then these results can be computed as

$$u_{and} = \min(x_0, x_1, x_2, \cdots, x_j)$$

$$u_{or} = \max(x_0, x_1, x_2, \cdots, x_j)$$

where $x_i$ is a fuzzified value.

Considering the number of membership functions in each input parameter, there will be $2 \times 5 \times 3 = 30$ possible fuzzy rules. To enumerate all of the possible rules, similar rules are grouped and expressed into 12 rules as follows.
R1 : if V is PS AND Z is PVS AND Fk is PS then CS-I
R2 : if V is PS AND Z is PVS AND Fk is PM then CS-II
R3 : if V is PS AND Z is PVS AND Fk is PL then CS-III
R4 : if V is PS AND Z is PS AND Fk is PS then CS-I
R5 : if V is PS AND Z is PS AND Fk is PM then CS-II
R6 : if V is PS AND Z is PS AND Fk is PL then CS-III
R7 : if V is PS AND Z is PS AND Fk is PS then CS-I
R8 : if V is PS AND Z is PM AND Fk is PM then CS-II
R9 : if V is PS AND Z is PM AND Fk is PL then CS-IV
R10 : if V is PS AND Z is PL AND Fk is PS then CS-IV
R11 : if V is any AND Z is PVL AND Fk is any then CS-V
R12 : The other cases are CS-0

Using the fuzzy inference mechanism with the defined rules, the output membership value is

\[ u_i = \min(\mu_1(v), \mu_2(z), \mu_3(f_k)) \]  

where \( i \) means the \( i \)-th rule, \( \mu_1, \mu_2, \) and \( \mu_3 \) are the membership functions of each fuzzy input set, and \( v, z, \) and \( f_k \) are the velocity, displacement, and force inputs, respectively.

C. Defuzzification

The defuzzification is performed using Center of Gravity (C.O.G) method[14] which is defined as

\[ y = \frac{\sum_{i=1}^{j} u_i r_i}{\sum_{i=1}^{j} u_i} \]  

where \( y \) is the defuzzified output value, \( j \) is the number of fuzzy rules, \( u_i \) is from (3), and \( r_i \) is the singleton value of the output fuzzy set in the \( i \)-th rule.

To ensure robust and accurate results, the final output is determined using the following two steps. Firstly, a possible estimation output is selected if the current defuzzified value \( y_{cur} \) keeps similar value when compared to the previous value \( y_{prev} \) during a certain amount of time. Meanwhile, it is 0.02 s in this paper.

\[ |y_{cur} - y_{prev}| \leq \varepsilon \]  

where \( \varepsilon \) means a small error between the current defuzzified value and the previous defuzzified value.

However, it is not guaranteed that the possible estimation output is exactly the same as the one of singleton values in the fuzzy output set. Thus, as the second step, the acceptable error boundary \( \beta \) is defined. When the estimated output is located in the error boundary, the final output is updated. Otherwise, the previous result is kept.

\[ |y - y_i| \leq \beta \]  

where \( \beta \) is 0.2 in this paper. At the beginning of the algorithm, CS-1 is assumed that the initial CS.

IV. ROBOT CONTROL STRATEGY

A. Control Input Wrench

Regarding the proposed algorithm, the robot generates force toward the insertion direction and moves the peg on the top surface. The control input wrench is designed using the motion/force hybrid control as in[11], [15]).

\[ F = [f \ m]^T \]  

\[ f = k_p \Omega(x_d - x) + k_v \Omega(\dot{x}_d - \dot{x}) + \Omega \Gamma f_{asm} \]  

\[ m = k_{wp} \delta \Phi + k_{ww}(\delta \theta - \dot{\theta}) \]

where \( F \in \mathbb{R}^6 \) is an input wrench vector, \( f \in \mathbb{R}^3 \) and \( m \in \mathbb{R}^3 \) are a force vector and a moment vector, respectively. \( x_d, x, \dot{x}_d, \) and \( x \in \mathbb{R}^3 \) are the end-effector’s target position, current position, target and current velocities, respectively. \( k_p \) and \( k_v \) are control gains. \( \Omega \in \mathbb{R}^{3x3} \) is the task specification matrix which is a diagonal matrix and whose elements are binary numbers and \( \Omega = I - \Omega([15]) \). When the binary numbers are assigned as 1, motions are controlled along the selected axes. Otherwise, the force is commanded. \( f_{asm} \in \mathbb{R}^3 \) denotes a constant assembly force applied toward the insertion direction during the whole process. As the force, \( f_{asm} \), is applied all the time, the peg will be pushed into the empty place if the peg is moved over any sinkage during search phase. \( \theta_d \) and \( \theta \in \mathbb{R}^3 \) denote the end-effector’s target and current angular velocities, respectively. \( k_{wp} \) and \( k_{ww} \) are control gains. \( \delta \Phi \in \mathbb{R}^3 \) denotes the angular rotation error vector [11].

To search the hole, Archimedean spiral trajectory is generated by

\[ \rho(t) = \frac{p}{2\pi} \psi(t) \]  

\[ \psi(t) = \sqrt{\frac{4\pi v_i}{p}} \]  

where \( \rho \) is the radius of the spiral, \( \psi \) is the polar angle of the spiral, \( p \) is the spiral pitch, and \( v_i \) is the linear velocity of the end-effector on the given trajectory.

To move the peg along the trajectory with constant speed, \( \psi \) is driven as (11). In this paper, the spiral trajectory is computed on the X-Y plane of the global frame, which is the motion control part. The assembly force in the force control part is generated along the Z-axis. Also, the moment from the input wrench plays a role to firmly hold the current orientation of the end-effector.

B. Force sensing

Since any F/T sensors are not used in this paper, contact force and moment are computed using values from joint torque sensors considering the redundancy of the robot. The joint torques are mapped into the force/moment in the operational space by

\[ F = \bar{J}(\theta)^T \Gamma \]
where $\bar{J}(q) \in \mathbb{R}^{n \times 6}$ is the dynamically consistent generalized inverse of Jacobian matrix, $\Gamma \in \mathbb{R}^n$ is a joint torque vector from the joint torque sensors, and $n$ is the degree of freedom of the robot.

To use the result of (12) as the force fuzzy input set, it is represented with respect to the end-effector’s frame.

$$f_e = R_e^T f$$

where $f_e \in \mathbb{R}^3$ is a force vector with respect to the end-effector’s frame, and $R_e \in \mathbb{R}^{3 \times 3}$ is a end-effector’s rotation matrix with respect to the global frame.

V. EXPERIMENT AND RESULT

A. Experiment Setup

To verify the proposed algorithm, the 7 DOF torque-controlled manipulator, Panda, manufactured by FRANKA EMiKA and IKEA STEPAN chair are used (Fig. 4(a)). Fig. 4(b) shows the dimension of the peg-in-hole environment which is a part of the IKEA STEPAN chair. There are three holes on the finite top surface. The chamfered shape and the body diameter of the peg are 5 mm and 7.8 mm, respectively. The chamfered angle is approximately 55°. Considering the diameter of the correct holes as 8 mm, there is 0.2 mm clearance. The diameter of the smaller-size hole is 6 mm, which implies that the peg might be stuck into the smaller-size hole during the peg-in-hole assembly. In the experiments, the linear velocity of the end-effector, $v_s$ is 1 cm/s, the spiral pitch, $p$ is 1 mm, and the assembly force, $f_{asm}$ is set to 6 N, and singleton values are selected as shown in Table I.

B. Fuzzy Versus Crisp

To validate the proposed algorithm, the proposed algorithm is compared to the corresponding crisp algorithm. Meanwhile, the crisp algorithm means the binary logic so that the result is always 1 or 0. To organize the crisp algorithm, thresholds of the velocity and force are determined by intersecting points between membership functions, as shown in Fig. 2. The thresholds of the displacement are determined by the geometric dimensions of the given furniture part. When the peg-in-hole assembly is finished, the final contact states can be classified by 4 kinds of states, CS-III, CS-IV, CS-V1, and CS-V2. Thus, the result of the experiments is evaluated by how appropriate the algorithms estimate the final situation as reported in Table II. When the two algorithms fail to estimate in common, an unexpected large reaction force is measured so that the current CS is miscalculated as CS-III.

The crisp algorithm only allows the exact value as true and considers the other values as false whereas the proposed algorithm admits a certain amount of errors near each threshold. This difference affects the success rate when the CS is estimated during the peg-in-hole assembly. For example, as shown in Fig. 5, the two algorithms generate different estimation results when the displacement input is 3.1 mm at 3.5 s. Because the threshold between PS and PM in the displacement variables is 3 mm, the crisp algorithm computes the estimation result as CS-V1 based on R8. However, in the proposed algorithm, the degree of membership has more weight for PS than PM in the displacement fuzzy set. Thus, the output of the proposed algorithm is CS-II based on R5. At 3.8 s, reflecting the large reaction force, the proposed algorithm estimates the CS as CS-III and the final estimation coincides with the actual situation. Similar situations like Fig. 5.
Fig. 5: An example of how the proposed algorithm and the corresponding crisp algorithm estimate contact states. The crisp algorithm generates a wrong estimation at 3.5 s, but the proposed algorithm generates the correct estimation at 3.8 s.

Fig. 6: An example of how the proposed algorithm and the corresponding crisp algorithm estimate contact states. The crisp algorithm generates a wrong estimation at 0.9 s, but the proposed algorithm generates the correct estimation at 4.7 s.

5 occurs frequently in the crisp algorithm. Consequently, the success rates in the case of CS-III, CS-IV, and CS-V2 are lower in the crisp algorithm when compared to the proposed algorithm. Although the crisp algorithm fails to estimate the CS correctly, it is less critical since the failure case is misjudged as a different failure case.

However, Fig. 6 shows that the misjudgment of CS-IV can be considered as a serious situation. The two algorithms compute different estimation results when the displacement input is 1.2 mm at 0.9 s. The crisp algorithm determines the estimation result as CS-III according to R6 because the threshold between PVS and PS in the displacement variables is 0.5 mm. On the other hand, in the proposed algorithm, the degree of membership has more weight for PVS than PS in the displacement fuzzy set. Thus, the output of the proposed algorithm is CS-II according to R2. At 4.7 s, the proposed algorithm estimates the final situation as CS-IV after going through CS-0, CS-I, and CS-II. From this comparison, since the crisp algorithm does not admit a certain amount of errors near the threshold, it is indicated that the peg-in-hole assembly is possible to be misjudged by the crisp algorithm even though the actual result succeeded by inserting the peg into the correct hole as shown in Fig. 6.

It will be a chance to enhance the performance of the crisp algorithm if more conditions are included. However, it will make the crisp algorithm more complex. The proposed fuzzy algorithm performs better with relatively fewer conditions. This is one of the major benefits of using the proposed fuzzy algorithm to estimate the CS of the peg.

C. Execution Result

Fig. 7 shows a snapshot of the results of the experiment. In the experiment, the manipulator is commanded to execute proper motions after estimating the contact state of the peg. For example, in Fig. 7(a), the peg is escaped from the smaller-sized hole after detecting the peg is jammed, CS-III. In Fig. 7(b), the manipulator detects CS-IV and inserts the peg into the correct hole. In Fig. 7(c) and Fig. 7(d), the peg deviates from the top surface and moves down to unwanted space. Estimating these conditions as CS-V1 and CS-V2, respectively, the peg moves up to recover the failure situations.
VI. CONCLUSION

In this paper, the proposed algorithm using fuzzy logic is presented to estimate the CS of the peg in the peg-in-hole environment. Firstly, the peg-in-hole environment is introduced, which has several sizes of holes on the finite top surface. Then, possible CS between the peg and the environment, including failure cases of the peg-in-hole assembly, are classified. To cover all of these CS, the velocity of the end-effector, the displacement of the end-effector, and the reaction force applied to the end-effector are selected as fuzzy input parameters. With these inputs, the fuzzy rules are composed and the defuzzification is conducted. The proposed algorithm is validated through the experiments using a 7 DOF torque-controlled manipulator executed by motion/force hybrid control and a wooden prefabricated chair. Also, the proposed algorithm is compared to the corresponding crisp algorithm. In comparison, it is indicated that the proposed algorithm produces better estimation results when compared to the crisp algorithm. In our future work, we intend to discuss estimating the CS under a tight clearance or a chamfer-less condition.

REFERENCES


361