Symbolic Representation of Trajectories for Skill Generation

Hirohisa Tominaga Jun Takamatsu Koichi Ogawara Hiroshi Kimura† Katsushi Ikeuchi

Institute of Industrial Science
The University of Tokyo
Tokyo, 106-8558, Japan

†Graduate School of Information Systems
University of Electro-Communications
Tokyo, 182-8585, Japan

Abstract

The completion of robot programs requires long development time and much effort. To shorten this programming time and minimize the effort, we have been developing a system which we refer to as "assembly-plan-from-observation (APO) system;" this system provides the ability for a robot to observe a human performing an assembly task, understand the task, and subsequently generate a program to perform that same task. One of the necessary tasks in APO is to create a trajectory of robot hand movement from observing human performance. The previous system developed a direct observation method based on the trajectory of a human movement. Though simple and handy, the system was susceptible to noise. This paper proposes a method to make the observation robust against noise by using symbolic representations of a trajectory based on contact analysis. The system divides the trajectory into small segments based on the contact analysis, then allocates an operation element referred to as a sub-skill to those segments; the result is a robust trajectory-based APO system.

1 Introduction

To enable a robot to program automatically, we propose an enhanced assembly-plan-from-observation (APO) method. This method enables a robot to: observe a human performing an assembly task; understand the task through the transition of the face contact relation in the task; associate necessary robot commands with such face transitions; and subsequently generate a robot program to achieve that same task. To execute the task, the APO uses motion templates, which consist of classes of operations along with beginning and ending configurations of the operations. The trajectory between these two configurations is obtained from the analysis of CAD models. In order to bypass the difficulty of designing trajectories, a technique to obtain a trajectory of a manipulated object from observation was proposed by [2]

However, when a robot attempts to execute a task, its performance occasionally fails due to observation errors and inconsistency between a real object and its internal model.

This paper proposes a method to overcome this problem by introducing symbolic representations of a trajectory and associated sub-skills, operation elements, with those symbolic representations of a trajectory. The system divides the observed trajectory into segments that have same class of contacts, and then allocates a sub-skill to those segments based on a transition from current to next contacts. This method makes a system robust because we can monitor the performance of the robot as transitions of contact states corresponding to sub-skills; when deviations from nominal transitions are detected, we can correct those unexpected contacts in one of the contacts along the nominal transitions. We can also decompose large motion templates employed in the previous APO system into sets of smaller sub-skills. The sub-skills are relatively simple in structure in comparison with the original motion templates, and are substantially easier to design.

2 Skill Issues

This section briefly reviews the previous APO system and explains issues regarding skills which are used by the APO system in order to generate a robot performance.

Fig. 1 denotes the outline of the APO system. An operator demonstrates an assembly task in front of cameras; the system observes the human performance and analyzes face contact relations between two objects at the beginning and the ending configuration of the task. Then, it chooses an appropriate skill, such as insert-into or put-on, according to the face contact transition. Finally, the skill generates a robot motion.

The advantage of the APO system is that it deals with the tasks using symbolic representations of contact-relation-transitions of objects. Since assem-

Fig. 1: Outline of APO

bly tasks contain a relatively small number of transitions, the number of corresponding skills to achieve such transitions is also limited; the APO system can perform a variety of tasks with a finite number of skills. Since the system has information about the purpose for which the skills are designed, it can correct errors when they occur during skill execution.

However, the current implementation of skills has two drawbacks. The first one is that it does not have information on the trajectory of the human performance. Therefore, from the analysis of CAD models of manipulated and environmental objects, a skill generates an appropriate trajectory. The second drawback is that a skill generates an entire robot action, such as the insert-into or put-on actions, from the beginning through the end of the task. Consequently, the size of a skill is too large, and sometimes too difficult, to implement.

In order to overcome these two problems, we adopt the method to obtain trajectory information from observing human performance, divide the trajectory into some segments according to the contact relation of objects, and assign sub-skills to those segments.

This method makes it possible to obtain information on the trajectory of human performance, and to decompose large skills into smaller sub-skills without losing the advantages of the APO system.

3 Obtaining assembly trajectory

The trajectory of human performance is recorded as a sequence of range data through a continuous stereo system. This system consists of three synchronized TV cameras and records three sequences of black-and-white images. Later, using those three image sequences, a sequence of range data is generated. In order to increase the accuracy, the stereo system projects patterns on the scene. From this range data, using 3DTM object recognition system, the system records the trajectory of a manipulated object in the space[11].

The trajectory information obtained is represented in the configuration space (C-space), a 6-dimensional space which represents both the position and the orientation of an object[2]. It observes configuration of

objects at certain intervals through the system. This system represents object configurations as points in the C-space. In the C-space, as a result of the observation, a series of points, corresponding to the object configurations, is recorded.

A constraint of a manipulated object constitutes a manifold, referred to as a C-obstacle surface in a C-space. The configuration value obtained contains some observation errors. Due to these errors, observed points in the C-space jump around on the C-obstacle surface. In order to smooth out these observation errors, the system regards those points close enough to the surface as those on the constraining surface, and makes a smooth trajectory by connecting those observed points in the C-space. See Fig. 2 for an illustration of how to make the trajectory feasible.

A trajectory observed by the method can be achieved only under ideal situations. Often, a robot cannot follow the trajectory due to some discrepancy between CAD and real objects. It also true that a robot motion is often erroneous partly because of the difficulty of being dextrous enough to grasp an object firmly. Therefore, the system is required to have the capability of awareness of the current situation. For this, we will introduce contact relations to divide the trajectory into segments, and to assign sub-skills to the segments. Namely, we convert a continuous trajectory into a sequence of sub-skills. At each sub-skill, the system knows what kind of transition occurs; if an unexpected transition occurs, the system is aware of the error. And the system can recover such errors by performing a new sub-skill to convert the erroneous contact into a planned contact.

Fig. 2: Correcting path on C-surface

4 Analyzing contact relations

This section describes a method to analyze the contact relations along the trajectory of a manipulated object. Note that the previous APO system analyzes contact relations of a manipulated object at the beginning and the end of a task for generating skills, while the current system analyzes them throughout the tra-

jectory through the task performance for generating sub-skills.

4.1 Finding constraint inequalities

In order to analyze contact relations in a trajectory, we use constraint inequalities which represent feasible motions of a manipulated object.

Fig. 3: contact with screw representation

The screw theory is employed for representing constraint inequalities[4]. For example, when two objects contact each other at a point, as shown in Fig. 3, the feasible motion of the object B is constrained by the inequality (1). An equal part of the inequality (1) represents the motions which enable the contact relations to maintain, while the greater part does the detaching motions.

$$s_1 t_4 + s_2 t_5 + s_3 t_6 + s_4 t_1 + s_5 t_2 + s_6 t_3 \geq 0 \quad (1)$$

In the case of polyhedral objects, all kinds of contact relations can be represented by a combination of a vertex-face, a face-vertex, and an edge-edge contact. And these three contacts can be converted constraint inequalities.

$$\begin{pmatrix} a_{11} & \cdots & a_{16} \\ & \ddots & \\ a_{n1} & \cdots & a_{n6} \end{pmatrix} \begin{pmatrix} t_1 \\ \vdots \\ t_6 \end{pmatrix} \geq \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix} \quad (2)$$

4.2 Decomposition of dimensions

The previous APO system assigns the skill using such features that consist of maintaining, detaching, and constraining degrees-of-freedom (DOF) in translation [1]. We add three DOFs in rotation as shown in Fig. 4:

- Maintaining: The DOFs of axis directions to be able to rotate maintaining the contact relation.
- Detaching: The DOFs of axis directions not to be able to rotate maintaining the contact relation.
- Constraining: The DOFs of axis directions not to be able to rotate.

Fig. 4: Maintaining, detaching, and constraining DOFs in rotation

4.2.1 Analyzing contact relations

Using [8], the number of rank r , range of dimension of face of PCC $\{6 - r - p, \dots, 6 - r\}$, feasible motion of translation, the number of a both rotatable axis n_b , and the number of an either rotatable axis n_e are extracted from constraint inequalities (2). Maintaining m_t , detaching d_t , constraining c_t DOFs of translation are obtained from Table 1 and maintaining m_r , detaching d_r , constraining c_r DOFs of rotation are from the equation (3).

$$\begin{aligned} m_r &= 6 - r - m_t \\ d_r &= 3 - m_r - c_r \\ c_r &= 3 - (n_b + n_e) \end{aligned} \quad (3)$$

Table 1: Relation between feasible motion of translation and maintaining, detaching, and constraining DOF of translation

Translation	m_t	d_t	c_t	Translation	m_t	d_t	c_t
All	3	0	0	3D half	2	1	0
3D quarter	1	2	0	3D 1/8	0	3	0
2D space	2	0	1	2D half	1	1	1
2D quarter	0	2	1	Line	1	0	2
Half line	0	1	2	Point	0	0	3

4.2.2 Analyzing contact relations in “singular”

If a contact relation has any of the contacts shown in Fig. 5, we call them “singular contacts.”

Fig. 5: Three singular contacts

In the case of singular, constraint inequalities can be represented as (4). We use the different analysis method.

$$\begin{aligned}
A_{11} \cdot T \geq 0 \quad \text{or} \quad \cdots \quad \text{or} \quad A_{1m_1} \cdot T \geq 0 \\
\vdots \\
A_{n1} \cdot T \geq 0 \quad \text{or} \quad \cdots \quad \text{or} \quad A_{nm_n} \cdot T \geq 0 \\
(A_{ij} \in 1 \times 6 \text{ matrix}, T = {}^t(t_1 \cdots t_6))
\end{aligned} \tag{4}$$

However, it turned out that those singular DOFs can be treated as those without singular contact. In this case, we call these DOFs “singular maintaining”, “singular detaching”, and “singular constraining”.

5 Designing sub-skills

In the previous section, we defined three DOFs of translation and rotation. The change of contact relations lead to changing those DOFs. There are some transitions of DOFs as shown in Fig. 6. Among those possible transitions, following three transitions occur toward the direction of movements: ‘maintaining to detaching,’ ‘maintaining to singular maintaining,’ and ‘maintaining to singular detaching.’ These three transitions are important in the design of sub-skills.

Fig. 6: possible transition of DOFs

5.1 Maintaining to detaching in translation

The motion as shown in Fig. 7 leads to the transition from maintaining to detaching in translation. We call this motion “make-contact in translation.”

Fig. 7: Make-contact in translation

In order to implement the Make-contact-in-transition sub-skill, we use force sensors to detect when the manipulated object makes contact with the environmental object, and that the force value increases beyond a threshold. The system moves the manipulated object to the detaching direction of the next state until it achieves the contact.

5.2 Maintaining to detaching in rotation

The motion as shown in Fig. 8 leads to the transition from maintain to detaching in rotation. We call this motion “make-contact in rotation.”

Fig. 8: Make-contact in rotation

To implement this sub-skill, we use a force sensor as well as Make-contact-in-transition. The system can determine the direction of the rotation using the demonstration of an operator, and determine a rotation center by the analysis in C-space in advance. It rotates the manipulated object using this information until the force sensors detect the contact.

5.3 Maintaining to singular maintain in translation

The motion as shown in Fig. 9 leads to the transition from maintain to singular maintain in translation. We call this motion “slide in translation.”

Fig. 9: Slide in translation

When a manipulated object loses a contact with an environmental object, the force value decreases beyond a threshold value. Using the decrement of the force value, the system detects the point of singular contact. The direction to which the manipulated object be moved is acquired from the demonstration. The system moves the manipulated object toward the direction until the contact is lost.

5.4 Maintaining to singular maintain in rotation

The motion as shown in Fig. 10 leads to the transition from maintain to singular maintain in rotation. We call this motion “slide in rotation.”

Fig. 10: Slide in rotation

In order to achieve this motion, two controls should be done. One is the control to maintain the contacts,

while the other is to change the orientation of the manipulated object. Therefore, we decompose the motion into the two parts: at the first the system moves the manipulated object to a detaching direction at a slight distance, though one of two contacts is lost by this first motion; then the system rotates the object around the contact point until the manipulated object again makes contact with the environmental object at two points.

5.5 Assigning a “slide” sub-skill

The transition of maintaining to singular detaching in the moving direction usually leads to the same transition in another direction. Therefore, it is difficult to assign a “slide” sub-skill to it. In the end of a “slide” sub-skill, the number of “restricted DOF” [3] increases. Here, “restricted DOF” is defined as the sum of constraining and detaching DOFs. (Show Fig. 11.) It is defined by [3] in detail.

Fig. 11: Restricted DOF

If the number of “restricted DOF” in translation increases, a “slide in translation” sub-skill is assigned; if the number of “restricted DOF” in rotation increases, a “slide in rotation” sub-skill is assigned.

5.6 Maintaining to singular detaching

The motion as shown in Fig. 12 leads to the transition from maintain to singular detaching in translation. This motion looks like the motion combining “make-contact” and “slide” in translation. But this motion can perform the similar method of “make-contact,” so we do not treat this motion.

Fig. 12: maintaining to singular detaching

6 Examples

In this experiment, we constructed a test bed as shown Fig. 13 that consists of a dual arm with a pair of dextrous hands and a real-time stereo system.

Consider the peg-in-hole operation shown in Fig. 14. In the first transition, a maintaining DOF in trans-

Fig. 13: A test bed

Fig. 14: Maintaining, detaching, and constraining DOFs of translation and rotation, and Restricted DOFs in translation and rotation

lation changes to a detaching DOF. A “make-contact in translation” sub-skill is assigned.

In the second transition, maintaining DOFs in translation and rotation change to singular maintaining DOFs. A restricted DOF in translation increases. A “slide in translation” sub-skill is assigned.

In the fourth transition, a maintaining DOF in translation changes to a detaching DOF. A “make-contact in translation” sub-skill is assigned.

In the fifth transition, a maintaining DOF in rotation changes to a singular maintaining DOF and a restricted DOF in rotation increases. A “slide in rotation” sub-skill is assigned.

In the seventh transition, a maintaining DOF in translation changes to a detaching DOF. A “make-contact in translation” sub-skill is assigned.

Fig. 15 represents the sequence of the robot executing a peg-in-hole task using a sequence of assigned sub-skills. We confirmed that sub-skills work effectively.

7 Conclusions

This paper proposed a system which has the ability to observe a human motion, divide the trajectory obtained from the observation into several states according to the contact relation, and assign a sub-skill to each contact transition. We proposed a new method to classify the contact relations, implemented the sub-skills, and verified the behavior of the system with the sub-skills. Our enhanced system has the advantages of both the contact-state-based system (the APO system) and the trajectory-based system.

Acknowledgment

This work is partly supported by the Japan Society for the Promotion of Science under the grant JSPS-RFTF 96P00501.

References

- [1] K.Ikeuchi and T.Suehiro, "Toward an Assembly Plan from Observation Part I : Task Recognition With Polyhedral Objects," *IEEE Trans. Robotics and Automation*, Vol.10, no.3, pp.368-384, June 1994.
- [2] G.V.Paul and K.Ikeuchi, "Modeling Planar Assembly Paths from Observation," *Proc. of IEEE Int. Conf. on Intelligent Robots and Syst., IROS'96*, Osaka, Japan, pp.520-525, 1996.
- [3] T.Suehiro, "Study of an advanced manipulation system," *Researches of the Electrotechnical Laboratory*, No.912, June, 1990. (in Japanese)
- [4] B.Roth, "An Extension of Screw Theory," *Journal of Mechanical Design*, Vol.103, pp.725-735, 1981.
- [5] J.Miura and K.Ikeuchi, "Task-Oriented Generation of Visual Sensing Strategies in Assembly Tasks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.20, No.2, Feb, 1998.
- [6] H. W. Kuhn and A. W. Tucker, "Linear Inequalities and Related Systems," *Annals. of Mathematics Studies*, Vol. 38, 1956
- [7] J. Xiao and L. Zhang, "Contact Constraint Analysis and Determination of Geometrically Valid Contact Formations from Possible Contact Primitives," *IEEE Trans. on Robotics and Automation*, Vol.13, No.3, pp. 456-466, June 1997.
- [8] J. Takamatsu, H. Kimura, and K. Ikeuchi, "Classifying Contact States for Recognizing Human Assembly Tasks," *IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp 177-182, Aug 1999.
- [9] S. Hirai, H. Asada, and H. Tokumaru, "Kinematic Analysis of State Transitions in Assembly Operations and Automatic Generation of Transition Network," *SICE*, Vol.24, No.4, pp.84-91, 1988. (in Japanese)
- [10] B. J. McCarragher and H. Asada, "A Discrete Event Approach to the Control of Robotic Assembly Tasks," *IEEE Int. Conf. Robotics and Automation*, pp.331-336, 1997.
- [11] M. D. Wheeler and K. Ikeuchi, "Sensor Modeling, Probabilistic Hypothesis Generation, and Robust Localization for Object Recognition," *IEEE trans. Pattern Analysis and Machine Intelligence*, Vol.17, pp.252-265, 1995.

Fig. 15: peg-in-hole task