Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A tree-shaped motion strategy for robustly executing robotic assembly tasks*

Yusuke BOTA[1], Hajime MIZUYAMA[‡1], Akio NODA[2], Tatsuya NAGATANI[2], Ken-ichi TANAKA[2]

([1]*Department of Mechanical Engineering and Science, Kyoto University Yoshida-Honmachi, Sakyoku, Kyoto 606-8501, Japan*)

([2]*Advanced Technology R&D Center, Mitsubishi Electric Corporation 8-1-1, Tsukaguchi-Honmachi, Amagasaki, Hyogo 661-8661, Japan*)

E-mail: bota.yusuke@t04.mbox.media.kyoto-u.ac.jp; mizu@me.kyoto-u.ac.jp; noda.akio@dh.mitsubishielectric.co.jp; Nagatani.Tatsuya@aj.mitsubishielectric.co.jp; Tanaka.Kenichi@ah.mitsubishielectric.co.jp

Received Oct. 28, 2010; Revision accepted Oct. 29, 2010; Crosschecked Oct. 29, 2010

Abstract: An assembly robot needs to be capable of executing an assembly task robustly under various uncertainties. To attain this goal, we use a task sequence tree model originally proposed for manual assembly. This model regards an assembly task under uncertainties as a transformation of the contact state concept. The concept may contain several contact states with probabilities but these are transformed through a series of task elements into the contact state concept having only the goal state at the end. The transformed contact state concept can be classified according to the terminal condition of each task element. Thus, the whole assembly task can be designed as a tree-shaped contingent strategy called a task sequence tree. This paper proposes a systematic approach for reconfiguring a task sequence tree model for application to a robotic assembly task. In addition, by taking a 2D peg-in-hole insertion task to be performed by a robot equipped with a force sensor as an example, we confirm that the proposed approach can provide a robust motion strategy for the task and that the robot can actually execute the task robustly under bounded uncertainty according to the strategy.

Key words: Active compliant motion, Contact states, Motion strategy, Robotic assembly, Task sequence tree
doi:10.1631/jzus.A1001436 **Document code:** A **CLC number:** TP24

1 Introduction

An assembly robot needs to be capable of executing any assigned parts-mating task robustly under various uncertainties, for example, those in the geometric shape, dimension, position, and posture of the parts, in the robot control and in the sensory data. Traditional approaches for handling such uncertainties can be broadly classified into two groups: passive compliance and active compliant motion (Siciliano and Khatib, 2008).

Passive compliance deals with uncertainties using a hardware device. For example, remote center compliance (RCC) can absorb small lateral and orientation errors of the parts in complying with the contact forces applied during round peg-in-hole insertion (Whitney, 1982). However, RCC and other passive compliance devices, can be applied only to specified parts-mating tasks. Thus, in a flexible manufacturing environment, uncertainties should be handled using software rather than hardware. Active compliant motion, on the other hand, relies on force-based feedback control to deal with uncertainties (Lefebvre *et al.*, 2005). Its planning is often referred to as fine motion planning, and usually requires identifying the contact state between the parts. However, contact state identification itself is quite a difficult problem and it is still hard to find a general and reliable approach applicable to a practical situation. Therefore, a robust motion strategy which does not require completely identifying the contact state between the parts is needed.

To attain this goal, we use the task sequence tree

‡ Corresponding author
* Project (No. 19GS0208) supported by the Grant-in-Aid for Creative Scientific Research 2007–2011 funded by the Ministry of Education, Culture, Sports, Science and Technology, Japan

model, which was originally proposed for manual assembly (Murakami and Mizuyama, 2007; Kitagawa and Mizuyama, 2008). This model introduces the contact state concept, which is not limited to only one contact state but may contain several contact states with probabilities, and regards the assembly task to be executed as a process of transformation from an initial contact state concept to one having only the goal state through a series of task elements. Two types of conditional branches, i.e., decision branches and observation branches, emerge along the process. Through these branches, the whole assembly task under uncertainties can be designed as a tree-shaped contingent strategy called a task sequence tree.

In this paper, we propose a systematic approach for reconfiguring the task sequence tree model for application to a robotic assembly task. Specifically, we show how to systematically computerize the most fundamental component of this model, i.e., observation branches. In addition, by taking a 2D peg-in-hole insertion task to be performed by a robot equipped with a force sensor as an example, we confirm that the proposed approach can provide a robust motion strategy for the task and that the robot can actually execute the task robustly under bounded uncertainty according to the strategy.

2 Task sequence tree model and its reconfiguration

2.1 Assembly task and robot system

In this study, a 2D peg-in-hole insertion task (Fig. 1) is used as an example to show how the proposed approach works. However, this approach is not limited to this model.

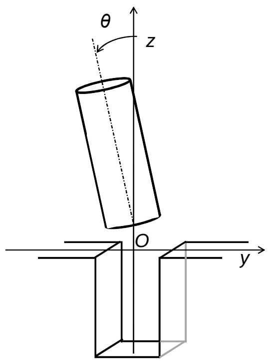

Fig. 1 Peg-in-hole insert task example

Further, we introduce the following assumptions about the assembly task and the robot system used to handle it.

1. A robot handles a cylindrical object called the part and assembles it into another object fixed in the environment called the work.

2. During the parts-mating process, only three degrees of freedom (y, z, θ) are considered and the robot should manipulate the part while keeping contact with the work.

3. Possible contact states (S0–S23) are classified as shown in Fig. 2.

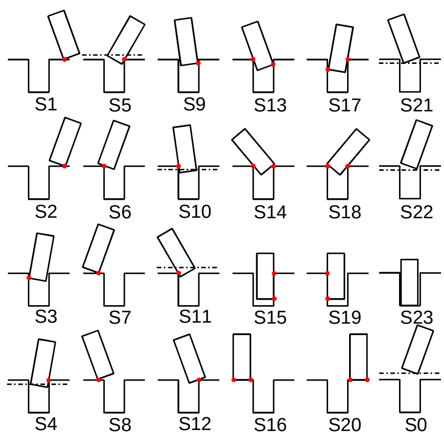

Fig. 2 Possible contact states

4. The robot has a force sensor and can measure the forces F_y and F_z and the moment M_θ.

5. A control mode can be selected for each degree of freedom from the following three modes: position control, force control, and compliant control.

2.2 Design of task elements

To keep the difference from the original task sequence tree model to a minimum, we decided to use the same five task elements and design them as close as possible to those used in the original model by using the available control modes. The designed task elements are as follows (Table 1): Left: to slide the part to the left; Right: to slide the part to the right; Downward: to move the part downward; Clockwise (CW): to rotate the part clockwise; Counter-clockwise (CCW): to rotate the part counter-clockwise.

Parameter values are adjusted on a trial-and-error basis. Each element stops when the specified step size is reached or a force/moment value higher than the preset threshold is observed.

Table 1 Control modes and parameter values of each task element

Task element		Right/Left	CW/CCW	Downward
Control mode	y	Force control	Compliant control	Compliant control
	z	Position control	Compliant control	Position control
	θ	Position control	Compliant control	Compliant control
Feedback gain	y	0.1	0.5	2
	z	0.1	0.01	0.1
	θ	0.01	0.01	0.01
Stiffness	y (N/m)	–	0.01	0.01
	z (N/m)	–	1	–
	θ (N·m/(°))	–	100	100
Target value	F_x (N)	5	–	–
	F_z	–	–	–
	M_θ	–	–	–
Threshold value	F_x (N)	10	10	10
	F_z (N)	10	10	10
	M_θ (N·m)	10	10	10
Step size	y (mm)	+5/−5	–	–
	z (mm)	–	–	−5
	θ (°)	–	+1/−1	–

2.3 Design of observation branch

2.3.1 Outline of approach

An observation branch is a process of classifying the transformed contact state concept into several categories based on the sensory information obtained while a task element is being applied. This online process itself can be formulated as a classification problem. However, there are no universally applicable categories. Rather, they should be defined to suit the sensory capability of the assembly system, and hence the categories used for manual assembly cannot be blindly accepted.

Thus, we take an inductive approach as outlined in Fig. 3 for designing an observation branch algorithm suitable for our system. Following preparatory experiments, the approach systematically takes three steps: extracting features from the data, clustering, and classification. Preparatory experiments are conducted several times for all combinations of an initial contact state and a task element. Each trial acquires the corresponding sensory time series data of F_y, F_z, and M_θ.

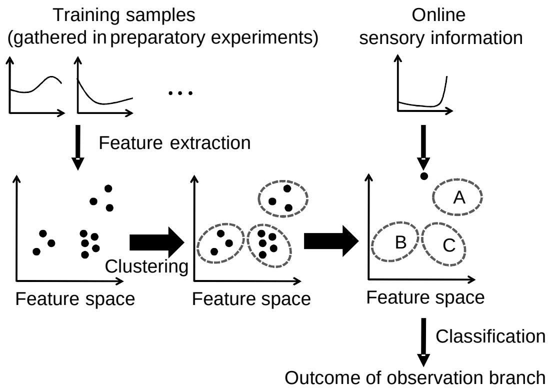

Fig. 3 Outline of observation branch design

2.3.2 Feature extraction step

Each of the training samples obtained by the preparatory experiments comprises sensory time series data of F_y, F_z, and M_θ, whose lengths are equal for a sample but different among samples. Thus, we pay attention to the distributions of their values. In particular, their means and standard deviations are considered as the candidate features, and the two most contributing features are chosen as the actual features for each task element. Thus, each training sample can be represented as a point in a 2D feature space (Fig. 3). Table 2 shows the number of training samples and the chosen features for each task element.

Table 2 Sample size and chosen features

Task element	Right/Left	CW/CCW	Downward
Sample size	61	28	46
Chosen features	std(F_y) mean(F_z)	std(F_z) mean(F_y)	std(F_y) mean(F_z)

* mean( ) and std( ) express the mean and standard deviation of the corresponding data

2.3.3 Clustering step

This step distributes the training samples for each task element into several categories. Ward's linkage hierarchical clustering algorithm is used for this purpose, where the similarities between objects are calculated using Euclidean distance. The number of clusters is decided according to the distinction rate, which represents the proportion of the samples acquired from the same contact state transition that are assigned to a single category. As a result, the number of clusters is determined as three for every task element, where the distinction rate is sufficiently high. The obtained clusters are named A, B and C, and they are then used as the classes into which an unknown object is classified at every observation branch.

Fig. 4a represents the possible state transitions realized by task element Right and the cluster to which a sample corresponding to each transaction should be assigned. Figs. 4b–4e express the same information for the other task elements. These figures provide basic information for designing a task sequence tree of the assembly task, as described in subsection 2.4.

2.3.4 Classification algorithm

This step assumes that the objects in each cluster are distributed according to a 2D Gaussian distribution and calculates the mean vector and variance-covariance matrix for each cluster. Then, when an unknown object is encountered, the Mahalanobis distance between the unknown object and each cluster is calculated and the object is classified into the cluster that has the shortest Mahalanobis distance to the object. The performance of this classification algorithm is confirmed through the leave-one-out cross-validation procedure.

2.4 Design of task sequence tree

Murakami and Mizuyama (2007) proposed how to choose a task element at a decision branch and thus to construct the whole task sequence tree for a manual assigned task. This approach evaluates a contact state concept with its ambiguity and distance-to-goal. The ambiguity is measured by the entropy of the contact state concept, and the distance-to-goal is quantified by the expected number of required state transition steps to the goal. At every decision branch, the approach chooses the task element which has the smallest value of a linear combination of these two features. We also apply this approach to the reconfigured task elements and observation branch.

For example, suppose that the robot takes Downward from the initial contact state S0 and the resultant contact state concept includes S1, S2, S4, and S12 with the same probabilities due to the uncertainties in the initial location and angle of the part. Then, the next contact state concept corresponding to each task element and observation branch outcome can be calculated based on the information in Fig. 4.

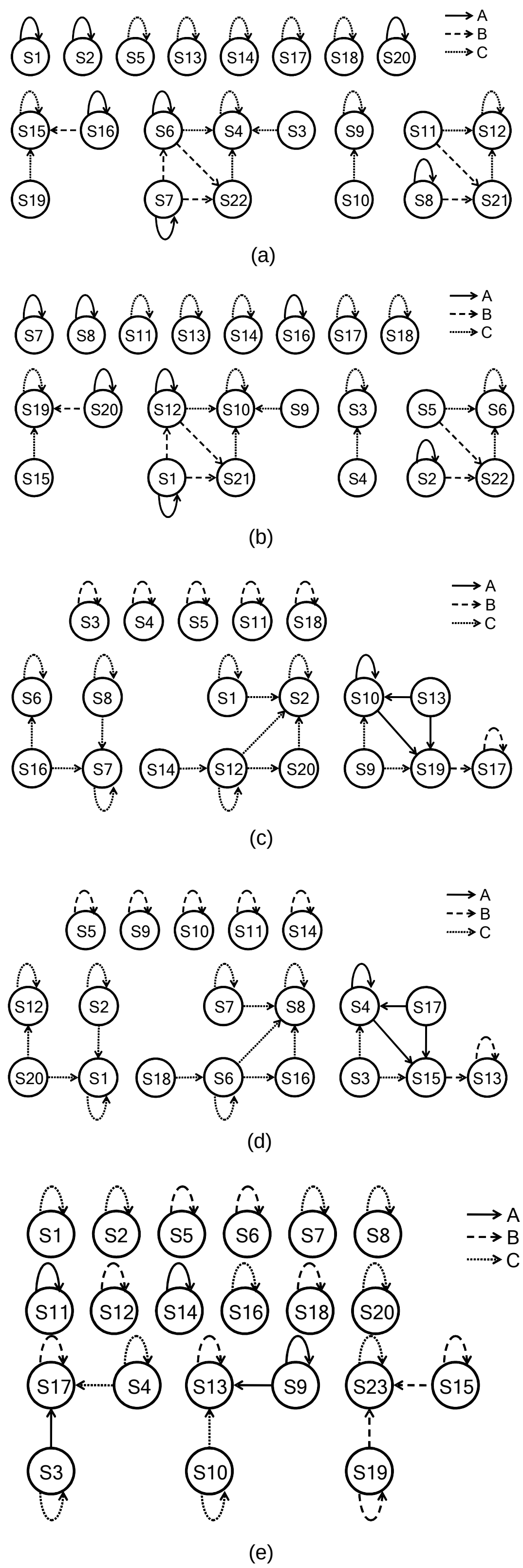

Fig. 4 Possible state transitions and corresponding clusters for task element (a) Right; (b) Left; (c) CW; (d) CCW; and (e) Downward

Thus, the ambiguity and distance-to-goal of each possible contact state concept can be evaluated, and taking expectations of them over the possible observation branch outcomes of a task element gives a measure of the element. According to this measure, we can choose the second task element to take. Then, the same procedure can be applied to each observation branch outcome of the second task element. Repeating this until all the observation branch outcomes reach the concept including only the goal state will give us a whole task sequence tree (Fig. 5).

3 Validation experiment

We conducted a validation experiment to confirm whether a robot could actually execute the example task robustly under bounded uncertainty according to the task sequence tree designed above. Since we assumed that the first contact state concept contains S1, S2, S4, and S12, several trials were executed accordingly by altering the initial location and angle of the part. The part and work used in this experiment were made of plastic material and metal material, respectively. Fig. 6 shows the scene of the experiment.

The results confirmed that the robot could execute the peg-in-hole insertion task shown in Fig. 1 robustly under the bounded uncertainty according to the strategy expressed by the task sequence tree shown in Fig. 5.

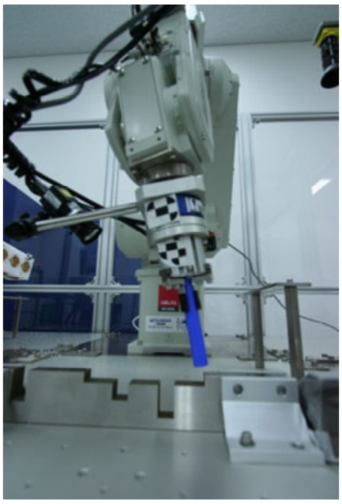

Fig. 6 Scene of validation experiment

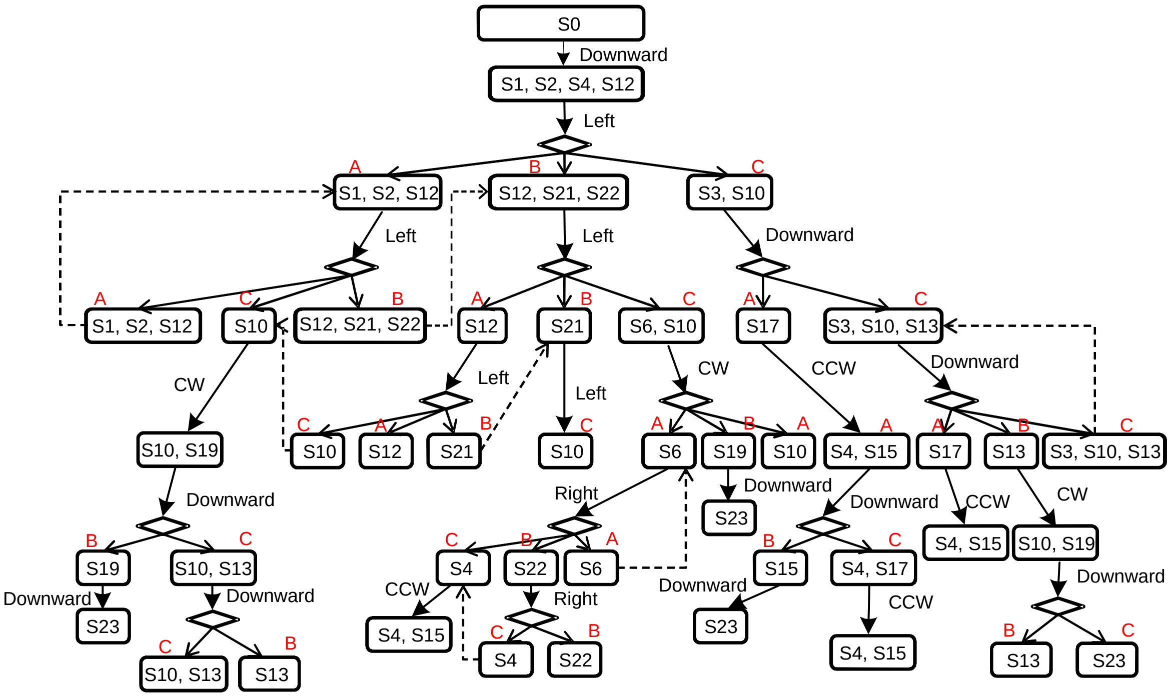

Fig. 5 Task sequence tree designed for 2D peg-in-hole insertion under uncertainties

4 Conclusions

In this paper, we presented a systematic approach for reconfiguring the task sequence tree model, originally proposed for designing and analyzing a manual assembly task under uncertainties, into a robotic assembly task. We also confirmed that the proposed approach can provide a robust motion strategy for a robotic assembly task and that a robot could actually perform the task robustly under bounded uncertainty according to the strategy.

However, the current approach requires many hours of work (i.e., for the preparatory experiments) to obtain a task sequence tree. In addition, the proposed strategy may be vulnerable to phenomena which are not expected from the preparatory experiments, e.g., a misclassification at an observation branch. Ongoing research should focus on these problems.

References

Kitagawa, Y., Mizuyama, H., 2008. Modeling and Supporting the Process of Learning Skills for a Manual Assembly Task. Proceedings of 9th APIEMS, Bali, Indonesia, p.1018- 1029.

Lefebvre, T., Xiao, J., Bruyninckx, H., Gersem, G.D., 2005. Active compliant motion: a survey. *Advanced Robotics*, **19**:479-499. [doi:10.1163/156855305323383767]

Murakami, Y., Mizuyama, H., 2007. Detailed Design of a Manual Assembly Task Incorporating how to Efficiently Handle Uncertainties. Proceedings of the 8th APIEMS, Kaohsung, Taiwan.

Siciliano, B., Khatib, O., 2008. Springer Handbook of Robotics. Springer-Verlag, Berlin Heidelberg.

Whitney, D.E., 1982. Quasi-static assembly of compliantly supported rigid parts. *Journal of Dynamic Systems, Measurement and Control*, **104**:65-77.