Error Recovery of Pick-and-Place Tasks in Consideration of Reusability of Planning

Akira Nakamura, Kazuyuki Nagata, Kensuke Harada and Natsuki Yamanobe
Intelligent Systems Research Institute
National Institute of Advanced Industrial Science and Technology (AIST)
Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan
E-mail: a-nakamura@aist.go.jp, k-nagata@aist.go.jp,
kensuke.harada@aist.go.jp, n-yamanobe@aist.go.jp
www.aist.go.jp

Abstract

Error recovery in robotic tasks is explored to enable robots to be used for complicated tasks. The authors' error recovery processes make use of the concepts of both task stratification and error classification. In this paper, the reusability of planning in error recovery is verified by using the typical pick-and-place tasks that are used in plant maintenance and industrial production.

Keywords: manipulation skill, error recovery, task stratification, error classification

1. Introduction

In recent years, studies on robotic manipulation for performing tasks in various fields have been conducted. We have conducted many research studies on the robotic manipulation used to perform plant maintenance tasks and manufacture industrial products (Fig. 1). These manipulation tasks tend to be complex, necessitating that composition rules be devised for the entire work process.

By analyzing the assembly and disassembly sequences performed by humans, we found that those tasks tend to be composed of several significant motion primitives. We call each motion primitive a "skill" and have shown that most maintenance and production tasks can be composed of a number of skills.[1–3]

Ideally, a robotic task has to be successfully completed as planned. However, in the actual tasks of complicated plant maintenance and industrial production, it is not rare for the execution of a task to terminate before completion. Therefore, error recovery is an important research theme for robots that need to perform such real-world tasks.[4, 5]

We have explored error recovery in robotic tasks to enable robots to be used for complicated tasks.[6–9] Our error recovery processes make use of the concepts of both task stratification and error classification, and these techniques are based on the concept of skills.

Considering the use of various recovery paths, the reusability of task planning may become an important aspect of research on error recovery. In this paper, the reusability of planning in error recovery is verified by using the typical pick-and-place tasks that are used in plant maintenance and industrial production.

Published by Atlantis Press

2. Stratification of Tasks

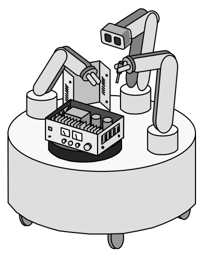

Fig. 1 Maintenance robot

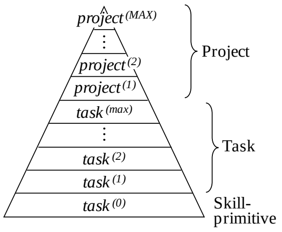

Fig. 2 Manipulation hierarchy

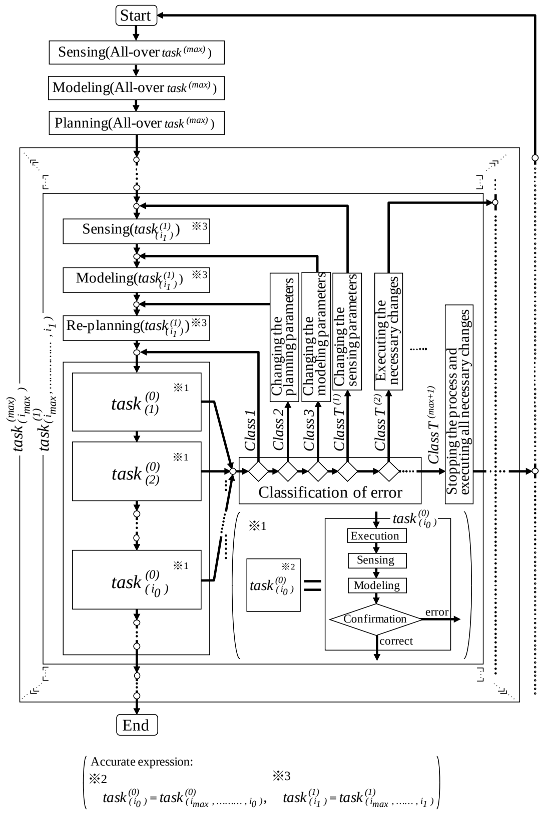

Fig. 3 Process flow with error recovery

This section explains our concept of skills and the stratification of tasks. See Refs. 1, 2 for more details.

2.1. *Concept of skills*

We analyzed human motions in such tasks as disassembly and reassembly and found that the movements consisted of several significant motion primitives. We call such motion primitives "skills [1, 2]". We considered three fundamental skills: move-to-touch, rotate-to-level and rotate-to-insert, all of which play an important part in those tasks. A specific task is composed of sequences of skill primitives such as these three skills. Moreover, many skills can be defined based on modified versions of these three fundamental skills[3].

2.2. *Stratification of tasks*

Figure 2 shows a hierarchy of manipulation tasks[6]. If we ignore the servo layer, the *skill* layer, which consists of elements such as the move-to-touch and rotate-to-level skills, is located in the lowest layer called the $task^{(0)}$ layer. One tier above the $task^{(0)}$ layer is the $task^{(1)}$ layer. Similarly, $task^{(i+1)}$ is composed of sequences of $task^{(i)}$ elements. The top layer, where the error recovery loop is closed, is called $task^{(max)}$ and one tier above $task^{(max)}$ is called the *project* layer. The *project* layer might also be hierarchized, but we will not discuss this here.

3. Error Recovery in Stratified Tasks

In an ideal environment, tasks are achieved without any errors occurring. In actual manipulation, however, errors often do occur from various causes. Our concept of error classification and process flow with error recovery in the task hierarchy are described in this section. See Ref. 6 for more details.

3.1. *Classification of errors*

The causes of manipulation failures can be attributable to several kinds of errors. We group the error states into several classes of errors: execution, planning, modeling and sensing, according to the possible causes[6].

Merely remedying the causes of these errors does not always solve the problem. It may be necessary to return to a previous step when the working environment is greatly changed by the error.

3.2. *Error recovery based on classification*

A generalized process flow of stratified tasks that takes error recovery into account has been shown in Ref. 6. Figure 3 is an illustration of the central portion of Fig. 10 in Ref. 6. This process is performed based on recovery through *a backward correction process*. At the Confirmation step in each skill primitive $task^{(0)}_{(i0)}$, the result is judged to be correct or a failure by an automatic process or by a human operator. Error recovery is performed using the following error classification.

Class 1: When the error is judged to be an execution error, $task^{(1)}_{(i1)}$ is executed again without correcting the parameter .

Class 2: When the error is judged to be a planning error, $task^{(1)}_{(i1)}$ is executed again with a change in the planning parameters .

Class 3: When the error is judged to be a modeling error, $task^{(1)}_{(i1)}$ is executed again with a change in the modeling parameters.

Class $T^{(1)}$: When the error is judged to be a sensing error, $task^{(1)}_{(i1)}$ is executed again with a change in the sensing parameters.

Class $T^{(2)}$: $task^{(2)}_{(i2)}$ is executed again after the execution of the necessary changes and returns to the start of one tier above the $task^{(1)}_{(i1)}$ layer.

:

:

Class $T^{(max)}$: $task^{(max)}_{(imax)}$ is executed again after the execution of the necessary changes and returns to the start of the (*max* - 1) tier above the $task^{(1)}_{(i1)}$ layer.

Class $T^{(max+1)}$: When it is judged that too many changes will be required, the process being executed is interrupted and the process returns to the start of the all-over task.

4. Expanded Processes in Error Recovery

4.1. *Forward correction process*

To correct the robot's motions at each step, a manual operation module for robot control can be inserted in the terminal processing of each primitive motion. For example, slight errors concerning the position and orientation of the object after transition and the condition of the grasped object can be corrected[8]. This process is recovery through *a forward correction process* which differs from recovery through *the backward correction process* described in 3.2.

4.2. *Additional tasks*

Additional tasks may be necessary in some cases to perform the corrections of Class 2, 3 and 4 errors (Fig. 4).[7] For example, additional geometry modeling of the working environment may be necessary for a Class 2 error, and additional geometry modeling of the object and the tool may be necessary for a Class 3 error. And additional geometry modeling of the working environment and calibration of the vision system may be necessary for a Class 4 error.

5. Planning with a View Toward Reusability

5.1. *About reusability*

Generally, a manipulation robot operates according to a computer program that is based on an ordered plan of operation. The tasks performed by a manipulation robot are varied and may involve such grasping tasks as gripper closing and opening, and transfer-related tasks such as lifting and approaching.

Similar tasks can performed using a similar computer program. As such, in systems with error recovery functions, similar tasks can be performed in the recovery portions. Therefore, it is possible to use a similar program for similar kinds of tasks for the total system, not only for the main operation part but for the additional recovery parts as well. A system with high reusability—one in which many of the same programs in the whole system could be reused including the recovery parts—would provide a very efficient approach to system configuration.

5.2. *How to improve reusability*

Let us explain a method for improving reusability by using the tasks involved in repacking objects from a large box into a small box in a physical distribution scenario as shown in Fig. 5. See Ref. 9 for more details on this task.

Failures may occur due to various causes when performing the given tasks, and in various situations such as grasping, lifting, carrying and packing.

The failure in which a plastic (PET) beverage bottle is dropped during transfer is considered here. In most cases, the dropped object ends up in some sort of disordered position, for instance falling over to a horizontal posture from its original vertical orientation. While an object with the shape of a PET bottle topples in most cases, sometimes it remains standing upright.

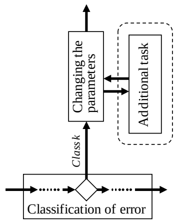

Fig. 4 Additional task

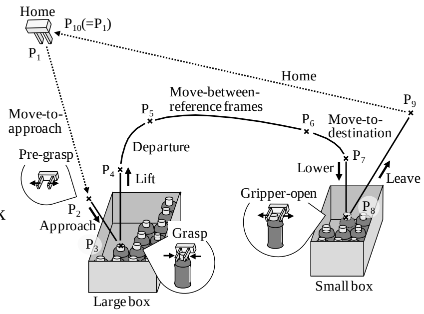

Fig. 5 Picking and placing task using a gripper

Error recovery involves grasping the object and moving it to an objective destination. A movement about the postures in the task means keeping the object vertical when it is already vertical and standing upright, and righting it back to the vertical from a toppled horizontal position. In the original scenario, most pick-and-place tasks will not require restoring to vertical posture from an existing vertical orientation. In other words, the task is performed by a pick-and-place program that expresses a change in three degrees of freedom from the position without changing the orientation. The process of the error recovery when the failure occurred proceeds by planning the flow including a change in the orientation of grasping, carrying with that change of posture, and packing. That is, the task is performed by a pick-and-place program that expresses a change in six degrees of freedom of the position and orientation.

It is undesirable to have many programs involved even in a transfer task. This program represents one of pick-and-place that expresses a change in six degrees of freedom of the position and orientation. The change from the original scenario can be considered as change in three degrees of freedom of the position without a change in orientation.

Furthermore, it is desirable for the program to be one that is usable in a wide area as well as in a local domain. That is why calibration needs to be performed in a wide area. Global calibration means that not only the pick-and-place task of the original scenario but also the processes of error recovery can be performed by the same program. Reusability increases when as many common programs as possible can be used for the total manipulation tasks.

6. Conclusions

In this paper, we have discussed the reusability of planning in manipulation tasks including error recovery processes. The techniques and validity of the reusability of such programs have been presented and verified using pick-and-place tasks. Programming of the total task including error recovery will be simplified by using the same planning in the path of the original task and in multiple recovery paths.

In the future, we will conduct further research on optimum adjustment methods for the various parameters used in the error recovery paths and for selecting common programs. We will attempt to apply our techniques to actual maintenance robots.

References

1. T. Hasegawa, T. Suehiro and K. Takase, A model-based manipulation system with skill-based execution, *IEEE Trans. Robot. Autom.*, **8**(5) (1992) 535-544.
2. A. Nakamura, T. Ogasawara, T. Suehiro and H. Tsukune, Skill-based backprojection for fine motion planning, in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, (Osaka, Japan, 1996), **2**, 526-533.
3. A. Nakamura, T. Ogasawara, K. Kitagaki and T. Suehiro, Using robust and simplified geometric models in skill-based manipulation, in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, (Hawaii, USA, 2001), **1**, 138-145.
4. B. R. Donald, Planning multi-step error detection and recovery strategies, *Int. J. Robot. Res.*, **9**(1) (1990) 3-60.
5. P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou and S. Schaal, Skill learning and task outcome prediction for manipulation, in *Proc. IEEE Int. Conf. Robot. Autom.*, (Shanghai, China, 2011), 3828-3834.
6. A. Nakamura and T. Kotoku, Systematization of error recovery in skill-based manipulation, *Artificial Life and Robotics*, Springer, **14**(2) (2009) 203-208.
7. A. Nakamura and Y. Kawai, Recovery technique from classified errors in skill-based manipulation, in *Proc. 15th Int. Symp. on Artificial Life and Robotics (AROB 15th '10)*, (Oita, Japan, 2010), 1010-1013.
8. A. Nakamura, K. Nagata, K. Harada, N. Yamanobe, T. Tsuji, T. Foissotte and Y. Kawai, Error recovery using task stratification and error classification for manipulation robots in various fields, in *Proc IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, (Tokyo, Japan, 2013), 3535-3542.
9. A. Nakamura, K. Nagata, K. Harada and N. Yamanobe, Error recovery using task stratification and error classification for manipulation robots in physical distribution, *Trans. Control and Mechanical Syst.*, **3** (2014) 122-131.

Published by Atlantis Press