

Technique of Recovery Process and Application of AI in Error Recovery Using Task Stratification and Error Classification

Akira Nakamura, Kazuyuki Nagata

Intelligent Systems Research Institute

National Institute of Advanced Industrial Science and Technology (AIST)

Central 2, 1-1-1 Umezono, Tsukuba, Ibaraki, 305-8568 Japan

Kensuke Harada

Robotic Manipulation Research Group

Systems Innovation Department

Graduate School of Engineering Science, Osaka University

1-3 Machikaneyama, Toyonaka 560-8531, Japan

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,
harada@sys.es.osaka-u.ac.jp, n-yamanobe@aist.go.jp
www.aist.go.jp*

Abstract

We have proposed an error recovery method using the concepts of task stratification and error classification. In this paper, the recovery process after the judgment of error is described in detail. In particular, we explain how to change the parameters of planning, modeling, and sensing when error recovery is performed. Furthermore, we apply artificial intelligence (AI) techniques, such as deep learning, to error recovery.

Keywords: error recovery, task stratification, error classification, manipulation, artificial intelligence

1. Introduction

Error recovery is an important research topic in robotic manipulation tasks in industrial production, plant maintenance, and housework. However, systematical methods of error recovery have not yet been developed. We propose error recovery using the concepts of task stratification and error classification. In the proposed method, the judgment of error is performed in the processes of the practice of a system. In this paper, the recovery process after the judgment of error is described in detail. In particular, we explain

how to change the parameters of planning, modeling, and sensing when error recovery is performed. This contributes to systematic error recovery.

Performing extensive error recovery leads to the accumulation of data about the recovery method. Furthermore, we apply artificial intelligence (AI) techniques, such as deep learning, to error recovery.

The rest of the paper is organized as follows: The concept of error recovery is described in Section 2. We modify the process of error recovery based on the size of error. The recovery processes in the case of a small error and a large error are explained in Sections 3 and 4,

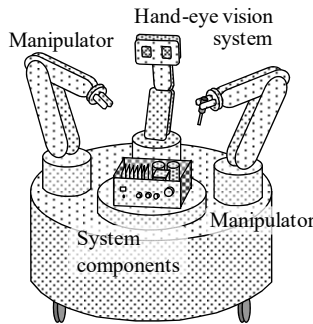


Fig. 1 Manufacturing and maintenance robot for audio-visual system components

respectively. We describe the usefulness of applying AI to error recovery in Section 5.

2. Concept of error recovery

This section describes the proposed error recovery method. The target task is the manipulation of a robot in an industrial factory and a power generation plant (Fig. 1). Concrete instances are assembly, disassembly, part replacement, and maintenance. In recent years, the role of robots in human life and industrial facilities has increased, and robotic manipulation during domestic tasks such as cleaning, washing and cooking is also a target of our error recovery system (Fig. 2).

2.1. Manipulation skill

First, we explain a fundamental unit and the composition of the manipulation task. Manipulation is considered based on a movement unit as a "skill".

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

In addition, composite tasks can be stratified, and a higher task possesses a lower task as a subtask; this is described later.

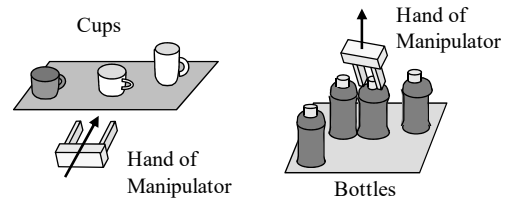


Fig. 2 Manipulators of robots working in daily life

2.2. Proposed error recovery method

Even though extensive research has been performed on techniques to recover an error that occurs in a manipulation system [4]–[14], systematization of error recovery has not yet been achieved. Most of these techniques are applied for the recovery of a small error that occurs in the trajectory of a manipulation robot. There are a few studies on recovery planning for a large error, which causes failure of achievement of an ordered task.

The proposed error recovery technique uses task stratification and error classification, which we proposed in Reference [15]. First, the cause of an error is estimated when it occurs. Next, the appropriate correction for the candidate of the cause is performed on a system. The manipulation returns to the previous step, and the task is re-executed successively (Fig. 3). The possibility of error occurrence would be small as the system is corrected.

If an error is small, the process returns to the lowest layer of a task hierarchy (Fig. 4). On the contrary, if an error is large, the process moves to a high-ranking layer of the hierarchy, and it is re-executed from the previous step (Fig. 4). Fig. 3 shows process flow for the former case, in which stratification is not considered, and the flow of the latter case is provided in Reference [16].

In Reference [15], we proposed error recovery using a backward correction process, which returns to the previous step. However, if an error is small, we perform error recovery using a forward correction process, in which an easy modification that shifts a trajectory is added to the system without returning to the previous step. In other words, in Reference [16], we propose an error recovery method in which it is possible to select a backward correction process or a forward correction process (Fig. 5).

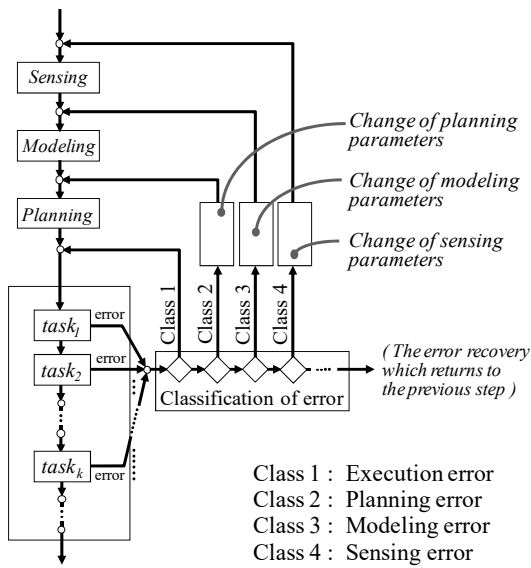


Fig. 3 Fundamental process flow with error recovery

3. Recovery process in the case of a small error

This section describes the error recovery technique for the case of a small error. In particular, the correction of an error is described in detail.

3.1. Conventional methods of controlling error

Conventional control methods have been used to correct a minute error in a trajectory. Techniques that use adaptive control, such as model reference adaptive control (MRAC) and self-tuning adaptive control, are the most popular. In such methods, the error in a trajectory can be minimized using a special control law while operating it. Therefore, it is a kind of error recovery using a forward correction process.

3.2. Concept of proposed error recovery method

We proposed an error recovery method using task stratification and error classification in Reference [15] (Fig. 3, Fig. 4). In this method, a system is revised based on the estimated cause of error, and the process is returned to the previous step and re-executed. In other words, the method can be interpreted as recovery through a backward correction process. We considered the following four kinds of errors: (Class 1) execution error, (Class 2) planning error, (Class 3) modeling error,

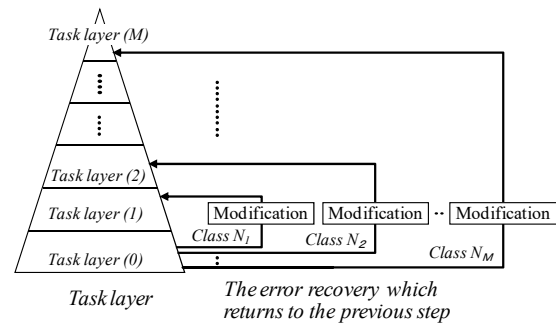


Fig. 4 The expression of task stratification and the process flow of the error recovery

and (Class 4) sensing error [15]. The process of handling an error is explained in detail. However, as a process is repeated without making modifications in (Class 1) execution error (Fig. 3), we consider the remaining three kinds of errors in this study.

3.2.1. (Class 2) Planning error

Planning error occurs in cases where the equations used to express a manipulation system do not express an actual system and the parameters used in planning are incorrect (Fig. 3). It includes the following corrections:

- Correction of the parameters of the equations used to express a manipulation system.
- Change in the initial condition for re-executing. For example, revision of the position and orientation of fingers.
- Change in the boundary condition and the threshold in the control process.
- Change in size, the direction of power, and the torque to be provided.
- Change in the planning technique to be used.

3.2.2. (Class 3) Modeling error

Modeling error occurs when a geometry model in software does not express the actual geometry accurately. A concrete correction (Fig. 3) includes the following: First, a change point about a single object is recorded.

- Change in position and orientation.
- Change in size.
- Change in shape.

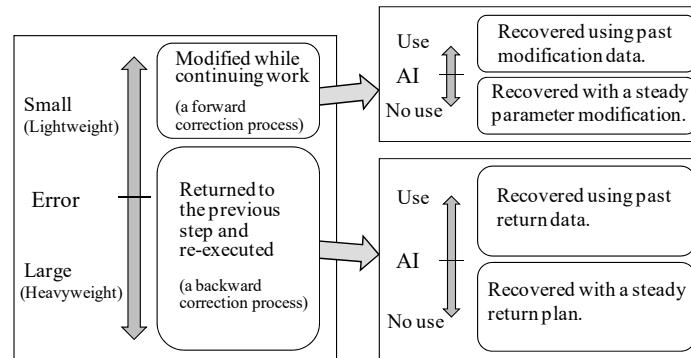


Fig. 5 Table about the error recovery technique

The models of all objects in a working environment are considered as targets for these corrections. However, in several cases, the corrections are performed only for objects directly related to work. For example, in work involving touching an object grasped in two fingers against a face of a board, the grasped objects, two fingers, and the face of the board are considered as the subjects of geometric models.

Next, it is important to be accurately model the relation between the objects. The following aspects should be considered for correctly modeling the relation between the two fingers and the grasped object when the task of touching the grasped object is performed:

- It is important that the location of the face touching the object is expressed correctly with respect to position and orientation. It is corrected if necessary.
- When the grasped object moves because of slipping, it is important to express it using a correction.

For example, the grasped object may slip between the two fingers owing to collision. Similarly, it is important to express the relations between multiple objects accurately.

3.2.3. (Class 4) Sensing error

Sensing error is caused by inaccurate sensing and insufficient calibration. If an error occurs during working practice, and its cause is judged to be a sensing error, the practice task is stopped and modification is performed (Fig. 3). A concrete correction includes the following:

- Change in various parameters of the sensing equipment.

- Change in the calibration parameters of the equations used to express the relation between a manipulator and a vision system.

We consider a correction process for the above-mentioned error. It is desirable to know the ranges of the values of the parameters to be revised beforehand. The value of a parameter is changed such that possibility of error occurrence decreases, and re-execution is performed. The values of parameters should be changed as strategically as possible. In addition, it is necessary to identify the interaction between parameters. If other parameters depend on a parameter, then it is necessary to change the values of these dependent parameters.

4. Recovery process in the case of a large error

In this section, we explain the process of error recovery by returning to a higher layer, as shown in Fig. 4. In Reference [15], we considered the repair task of audio-visual system components as an example and explained the stratification of the task in detail.

In the case of a large error, it is frequently necessary to return to the previous step. When practice fails completely, the process may have to return to the starting point, i.e., the highest layer. The working environment may change significantly because of failure. For example, multiple objects may move, fall, and break in the working environment because of failure of practice. In such a case, the contents of the re-execution task may have to be changed significantly or entirely. Even when not returning to the highest layer, it is better to reconsider a re-execution task in a few cases.

Plural passes may occur in a re-execution task. In such a case, it is suitable to select a pass based on an evaluation standard to be established. There are various evaluation standards; however, considering that an error has caused the re-execution, a method of selecting a pass that ensures the achievement of the final goal of the task is the most desirable.

5. Possibility of the application of AI

In recent years, it has been shown that AI techniques such as deep learning are effective for solving problems in several fields. In addition, the use of AI is efficient for error recovery.

5.1. Concept of the application of AI

In case of a small error, it is easy to apply AI for parameter correction (Fig. 5). It is easy to estimate which parameter should be changed and the degree by which it should be corrected because there is a large amount of past data. In case of a large error, the use of AI contributes equally to error recovery (Fig. 5). It is easy to estimate a higher layer to which a pass should be returned as there is a large amount of accumulated data.

It is effective to use AI in the following processes in the proposed error recovery technique:

- (i) Estimation of the cause of error.
- (ii) Parameter correction (which parameter should be changed and by how much)
- (iii) Selection of a point in a higher layer to return to (i.e., return to which step)
- (iv) Determination of the contents of a pass for re-execution (by what type of task)

5.2. Possibility of the application of AI in manipulation task

Let us consider the possibility of error recovery by

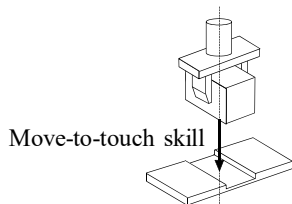


Fig. 6 Task of sticking a grasped object on a board with a dent

sticking a grasped object to a board with a shallow dent

as shown in Fig. 6. This task fundamentally requires only a move-to-touch skill for transferring the object, if other parts for the grasping motion are not considered. Two cases of a small error and a large error will be described.

5.2.1. Possibility of using AI in case of a small error

In this case, the manipulation process at the recovery step continues to be executed without returning or returns to the lowest layer of a task hierarchy (Fig. 4).

In the former case, there is a possibility that recovery can be achieved by minute correction without carrying out the return operation (Fig. 7). This type of error recovery has been studied for a long time. For example, MRAC, shown in 3.1, is a typical method. The method reduces the error in a trajectory by adjusting some parameters of the equation that expresses a manipulation system.

In the latter case, error recovery is performed using the concept of error classification. For example, an error in which a grasped object does not fit into the dent of board at the end of the task (Fig. 8) is considered. The system parameter is adjusted based on the estimated cause of error, and the process is returned to the previous step in the same layer and re-executed (Fig. 3). In practice, it is necessary to consider the following types of errors

(I) (Class 1) Execution error

This is a mechanical error caused in the manipulator mechanism such as a gear backlash. When it is judged to be an execution error, a manipulation task is executed again without correcting the parameters. AI may make it clear whether the manipulation task can be achieved by re-execution without correction.

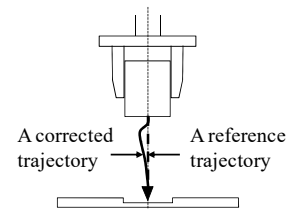


Fig. 7 A corrected trajectory derived by adjusting a parameter in a system equation

(II) (Class 2) Planning error

This is an error caused by inaccurate parameter values in planning. When it is judged to be a planning error, the manipulation task is executed again with a change in the planning parameters, as described in 3.2.1. AI may make it clear which planning parameters such as the system parameters, the initial conditions, the boundary conditions, and the threshold need to be adjusted.

(III) (Class 3) Modeling error

This is an error caused by differences between the real object and the geometric model used in the software. When it is judged to be a modeling error, the manipulation task is executed again by changing the modeling parameters. In this manipulation task, the margins of many errors can be considered as follows.

- i) An error caused with respect to shape, size, position, and orientation of the grasped object
- ii) An error caused with respect to the shape, size, position, and orientation of the manipulator hand
- iii) An error caused with respect to shape, size, position, and orientation of the board

Furthermore, an error may occur for the relations of the relative position and orientation between the grasped object and hand. The relation between the grasped object and board is similar. That is, it is necessary to consider the errors of the relative relation of several objects.

There are many parameters that can be changed. Even in such a situation, AI can be used to choose parameters that need to be changed. The choice of data and the amount of change are derived by collating the errors that occurred in the real system during the analysis and the data stored in computer software.

(IV) (Class 4) Sensing error

This is an error that occurs during visual sensing.

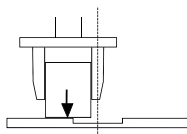


Fig. 8 An error in which a grasped object does not fit into a dent at the contact step

When it is judged to be a sensing error, the manipulation task is executed again by changing the sensing parameters, as described in 3.2.3. The use of AI is effective in this case as mentioned above. AI may choose a modification method of a parameter of vision system and a calibration parameter during each equipment in the working environment, and it should be derived effectively.

5.2.2. Possibility of using AI in case of a large error

In this case, the manipulation process at the recovery step returns to the upper layer of a task hierarchy (Fig. 4). For example, an error in which a grasped object detaches and lies besides the goal is considered. Figure 9 shows that an object is lying out of a goal by opening of the hand before achieving the task. In addition to this, the grasped object may be destroyed or a machine might break down. Figure 10 shows that an object is crushed as contact happened before the expected. In this type of error, a process has to return to the previous step again, that is, the upper layer of a task hierarchy (Fig. 4). The cause of the error varies in this case and is the same as that mentioned in 5.2.1 (I) (Class 1) - (IV) (Class 4). The important problems in this case are deciding which layer the process has to be returned to and the contents of the recovery process.

Applying AI is effective for error recovery in case of both small and large errors. However, as a large amount of data is necessary, it is required that there is much outbreak of the error, and demand contradicts ideal performance in a sense. When there are only a few errors, there is less experience; thus, it is important to plan and perform recovery steadily, and not to the extent to which AI can be used for recovery, in case of both small and large errors (Fig. 5).

6. Conclusion

Theoretical systematization of error recovery has not

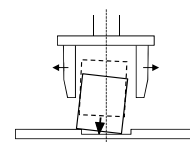


Fig. 9 An error in which a liberated object from a hand does not fit into a dent

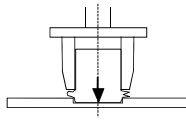


Fig. 10 An error in which a grasped object is crushed by colliding with a board

yet been developed. We proposed error recovery using task stratification and error classification. In this paper, we described the recovery process carried out after the judgment of error in detail. In particular, we explained how to change the parameters that express various conditions in planning, modeling, and sensing errors when recovery is performed. Then, we changed the process of error recovery based on the size of an error. The technique was derived based on task stratification. If an error is large, the recovery process may move to a higher layer. This implies returning to the previous step. Our recovery method can be applied to various sizes of errors. We explained the possibility of applying AI techniques, such as deep learning, to error recovery. Abundant available data on error recovery contribute to the usefulness of AI. Future studies will include the application of our method to an actual system. Moreover, in error recovery using AI, we will study a technique of modifying system parameters in the case of a small error and a method of constructing the re-execution task in the case of a large error.

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 skillbased 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. E. Z. Evans and C. S. George Lee, Automatic generation of error recovery knowledge through learned reactivity, in *Proc. IEEE Int. Conf. Robot. Autom.*, (San Diego, USA, 1994), 2915-2920.
6. C. M. Baydar and K. Saitou, Off-Line error prediction, diagnosis and error recovery using virtual assembly systems, in *Proc. IEEE Int. Conf. Robot. Autom.*, (Seoul, Korea, 2001), 818-823.
7. Y. Yamada, T. Morizono, Y. Umetani and T. Yamamoto, Human error recovery for a human/robot parts conveyance system, in *Proc. IEEE Int. Conf. Robot. Autom.*, (Washington, DC, USA, 2002), 2004-2009.
8. K. Yamazaki, M. Tomono, T. Tsubouchi and S. Yuta, Motion planning for a mobile manipulator based on joint motions for error recovery, in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, (Beijing, China, 2006), 7-12.
9. M. Scheutz and J. Kramer, Reflection and reasoning mechanisms for failure detection and recovery in a distributed robotic architecture for complex robots, in *Proc. IEEE Int. Conf. Robot. Autom.*, (Roma, Italy, 2007), 3699-3704.
10. J. C. Himmelstein, E. Ferre and J.-P. Laumond, Teleportation-based motion planner for design error analysis, in *Proc. IEEE Int. Conf. Robot. Autom.*, (Kobe, Japan, 2009), 914-920.
11. 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.
12. J. Rojas, K. Harada, H. Onda, N. Yamanobe, E. Yoshida and K. Nagata, Early failure characterization of cantilever snap assemblies using the PA-RCBHT, in *Proc. IEEE Int. Conf. Robot. Autom.*, (Hong Kong, China, 2014), 3370-3377.
13. E. Krabbel, E. Kristiansen¹, L. Hansen and D. Bourne, Autonomous optimization of fine motions for robotic assembly, in *Proc. IEEE Int. Conf. Robot. Autom.*, (Hong Kong, China, 2014), 4168-4175.
14. J. S. Laursen, U. P. Schultz and L-P. Ellekilde, Automatic error recovery in robot assembly operations using reverse execution, in *Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst.*, (Hamburg, Germany, 2015), 1, 1785-1792.
15. A. Nakamura and T. Kotoku, Systematization of error recovery in skill-based manipulation, *Artificial Life and Robotics*, Springer, 14(2) (2009) 203-208.
16. 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.