Evaluation Standard of Error Recovery Planning Focused on Revival Process from Failures in Robotic Manufacturing Plants

Akira Nakamura
Department of Information Systems, Faculty of Engineering, Saitama Institute of Technology
1690 Fusaiji, Fukaya, Saitama 369-0293, Japan

Kensuke Harada
Robotic Manipulation Research Group, Systems Innovation Department
Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama, Toyonaka 560-8531, Japan

Abstract

In recent times, intelligent robots have found applications across diverse fields. In scenarios demanding both repetitive and non-routine tasks, the likelihood of work errors increases. To address this, we have proposed methods encompassing both forward and backward recovery. Forward recovery suits minor modifications, while our focus is on backward recovery for substantial failures. Our study introduces a novel evaluation method to discern the optimal recovery path among various options.

Keywords: error recovery, task stratification, error classification, evaluation standard

1. Introduction

At the present day, information technology, particularly generative AI, is making rapid strides, a trend mirrored in the progress of robotic intelligence. As a consequence, robots face an increasing array of tasks during execution, often grappling with challenging assignments prone to errors and failures. This underscores the pressing need for effective methods to address these issues [1], [2], [3], [4], [5].

Over several years, our research [6], [7], [8], [9], [10] has focused on systematizing error recovery theory, resulting in a method based on task stratification and error classification concepts. The primary components of the robot system include sensing, modeling, planning, and execution sequences (Fig. 1). When an error occurs, the process transitions to the recovery phase. This section involves estimating the error's cause, classifying it, and correcting the original system. The refined process then operates on an enhanced, reliable system.

The proposed error recovery technology returns to the process before the step where the failure occurs and starts over from there. In practice, not only this type of backward recovery but also forward recovery, which moves forward after failure, is used. This study considers various paths from failure to recovery execution.

The concept of skills, which are motion primitives comprising a task, is described in Section 2. The fundamental technique for generating an error-recovery path is described in Section 3. Multiple possibilities for various recovery paths are presented in Section 4. Various evaluation standards and methods for selecting the most suitable path using the new evaluation method are considered in Section 5, and examples are presented in Section 6.

2. Concept of Skill

This section provides a brief overview of essential aspects of these skills [11], [12], [13].

2.1. Skill primitives

Motion primitives constituting tasks, termed "*skills*," were derived by analyzing the human behavior. Three crucial skills, "*move-to-touch*," "*rotate-to-level*," and "*rotate-to-insert*" (Fig. 2), are highlighted. A person's behavior, including representative and similar skills, serves as a model for the robot's motion primitives.

2.2. Stratification of tasks

The use of task hierarchies, as shown in Fig. 3, is effective for the execution of automated plants. The layer "$task^{(i+1)}$" occurs one tier above the layer "$task^{(i)}$," and the layer "*skill primitive*" is represented by the lowest layer "$task^{(0)}$."

3. Error Recovery

In an actual work environment, contrasting to an ideal case, various factors can lead to errors in the execution of a robot. This section outlines the error classification concept and error recovery technique [6], [7], [8], [9], [10].

3.1. Error classification

The errors can be categorized into several groups based on their possible causes. We considered four error groups: execution, planning, modeling, and sensing (Fig. 4).

3.2. Error recovery based on classification

First, if an error occurs, the cause is determined. Next, appropriate corrections are made to the system based on the tentative causes. The process returns to the previous step, and the task is executed again in this step (Fig. 4). Since a modified process was executed, the same error was less likely to occur. If the error scale is small, the process returns to the previous step in the lowest hierarchy (Figs. 4 and 5). Conversely, if the error scale is large, the process returns to the previous step in the highest-ranking layer of the hierarchy and is executed again from that step (Fig. 5).

4. Various recovery paths

The error recovery methods we proposed have primarily centered on rerunning the process by reverting to the step before an error occurs, constituting a backward error recovery process. However, alternative recovery procedures exist, including backward recovery for significant errors (failures) and forward recovery for minor errors. Additionally, errors may impact the environment, altering the arrangement or shape of surrounding objects, necessitating modified recovery processes. This section introduces several recovery procedures focusing on the degree of destruction of the environment surrounding an object according to [10].

Consider an indicative task sequence composed of n subtasks from the start to the goal (Fig. 6(a)). Suppose a failure occurs in the qth subtask during the process (Fig. 6(b)).

4.1. Recovery sequence I (RS-I): Complete restart

This method restarts work from the original starting point using the same process (Fig. 6(c)). If needed, the environment is restored to its original state, and if necessary, the original object or part is replaced with a new one and executed.

4.2. Recovery sequence II (RS-II): Restart from the middle of a prior process (without another process)

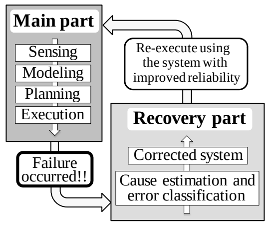
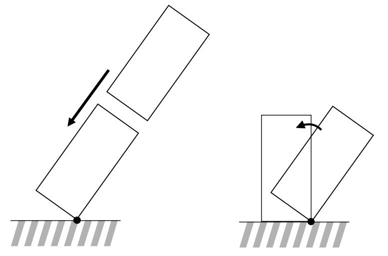


Fig. 1 Robot task system with an error recovery function



(a) Move-to-touch skill (b) Rotate-to-level skill

(c) Rotate-to-insert skill

Fig. 2 Three fundamental skills

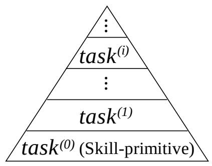


Fig. 3 Hierarchy of taskS

This method resumes work in the same process from subtask$_p$, the point in the middle of the process before the failure in the original process occurs (Fig. 6(d)).

4.3. Recovery sequence III (RS-III): Restart from the middle of a prior process (with another process)

This method resumes work from a point in the middle of the original process before a failure occurs (Fig. 6(e)). Unlike RS-II, a sequence from subtask$_t$ to subtask$_u$, not included in the original planning, restores the environment without causing problems for subsequent work.

4.4. Recovery sequence IV (RS-IV): Restart from the middle of a process that was supposed to come up later (with another process)

This method restarts work from the middle of the original process (Fig. 6(f)), differing from RS-III by reverting to its original process with subtasks scheduled after the original process failure. It includes a sequence from subtask$_v$ through subtask$_w$, not in the original plan.

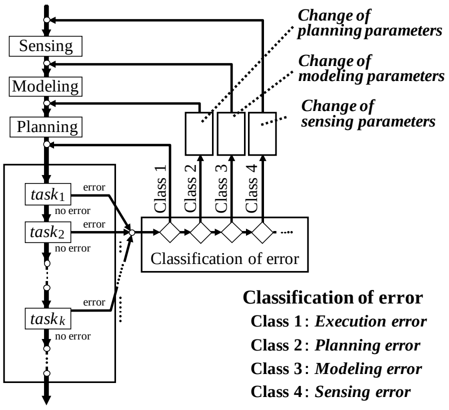


Fig. 4 Fundamental process flow with error recovery

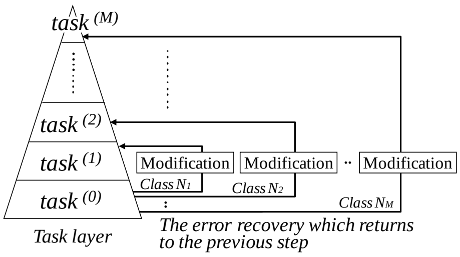


Fig. 5 The expression of task stratification and the process flow of the error recovery

Table 1 Degree of correlation for each criteria

	Manufacturer ／Operator	Consumer ／User
(i) Cost	△	◎
(ii) Time	○	○
(iii) Reliability	◎	△
(iv) Safety	◎	△
(v) Finishing	△	◎
(vi) Recovery data	◎	―
(vii) Tool	◎	―
(viii) Operator skill	◎	―
(ix) Work efficiency	◎	―
(x) Environmental impact	◎	―
(xi) Damege	○	○

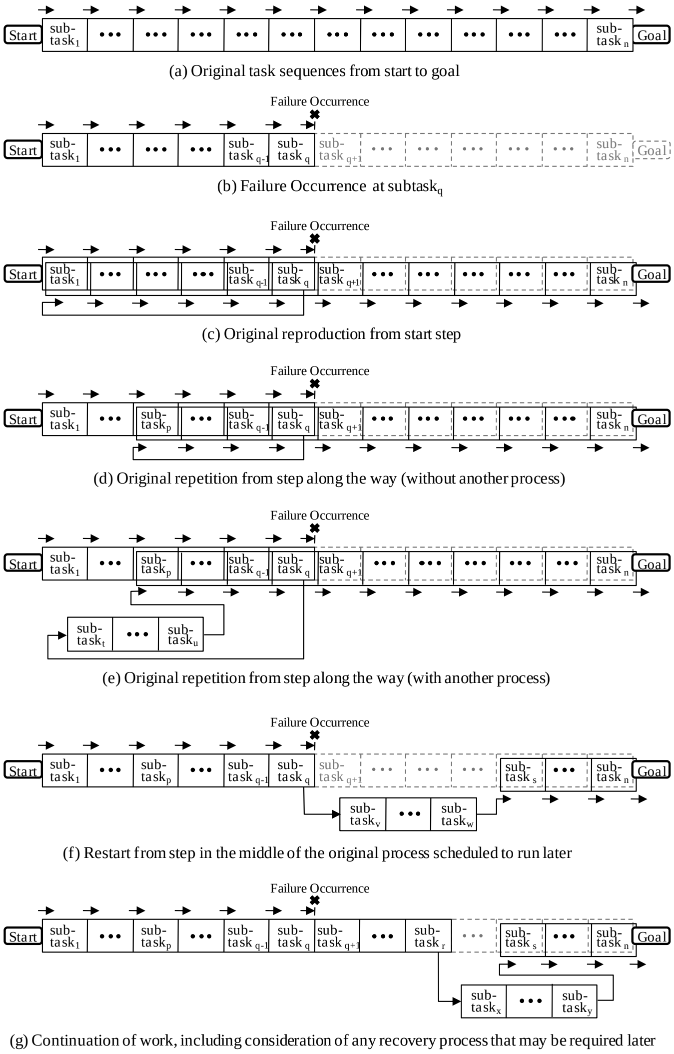


Fig. 6 Various error recovery sequences

4.5. Recovery sequence V (RS-V): Continuation of work including consideration of recovery needed later

In this method, work continues as long as possible after a failure. When further work becomes impossible, the environment is corrected, and work progresses (Fig. 6(g)). This modification involves a sequence from subtask$_x$ through subtask$_y$, not included in the original planning.

4.6. *As mentioned above, this study briefly categorizes the recovery sequences into five types.Recovery sequences RS-I through RS-III are backward recovery processes, whereas recovery sequence RS-V is a forward recovery process. Recovery sequence RS-IV is not a forward recovery process because it does not continue working after failure has occurred. However, because recovery sequences are subtasks scheduled to be executed after failure of the original process, this study included the recovery sequence RS- IV in the forward recovery process for convenience.*

5. Selection of a recovery path

As explored in Section 4, a single failure often presents multiple possible recovery processes. The study advocates the use of evaluation standards to discern the most suitable process from several candidates. This research incorporates 11 evaluation standards, expanding on the four to eight standards introduced in [9].

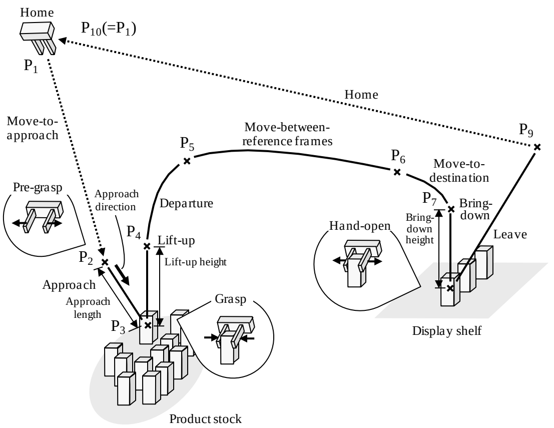


Fig. 7 Picking and placing task using a gripper

5.1. Various evaluation standards

(i) Cost

Cost is considered the most important evaluation standard. A recovery process with the minimum practical cost is selected.

(ii) Time

Time is an important evaluation standard. Priority is given to paths that require shorter recovery times.

(iii) Reliability

Reliability is considered a representative evaluation standard. The path with a high success rate is prioritized for accomplishing the recovery task.

(iv) Safety

Safety is considered a representative evaluation standard. Priority is assigned to paths that are less likely to harm people.

(v) Finishing

Finishing is considered an important evaluation standard. It prioritizes paths displaying excellent completion of the operation on the target object.

(vi) Recovery data

Recovery data is considered as the evaluation standard. Processes with a lot of useful data are prioritized.

(vii) Tool

The tool used for recovery is evaluated based on workability, with priority given to tools displaying high efficiency.

(viii) Operator skill.

Operator skill during the recovery process is an evaluation standard, favoring processes managed by a larger number of skilled experts.

(ix) Work efficiency.

Work efficiency is considered as an evaluation standard. Selecting an efficient work plan, that is, a route that is not unreasonable, wasteful, or uneven, is important. This index is common to (i) cost and (ii) time, but even if it is an efficient route, it may not necessarily be cheap or fast; therefore, it was set as a separate indicator.

(x) Environmental impacts

Environmental indicators, including noise levels, are considered, accounting for overall environmental problems such as air, ocean, and water pollution.

(xi) Damage

Damage is a measure of the difference between an object without and with errors. Although it may not be readily visible externally, the quality of an object could change owing to errors in the production process.

5.2. Selection of a recovery path using evaluation standards for each section

Reference [9] details how to choose the best recovery path for each evaluation standard. In contrast, this study proposes specifying indicators for each section of the recovery process and selecting the path with the superior total indicator. This prevents situations where a path chosen by a single indicator may be partially inappropriate.

Furthermore, evaluations may vary depending on the recipient of the standard. Table 1 delineates whether the evaluation is from the Manufacturer/Operator side or the Consumer/User side, indicating the degree of involvement with symbols ◎, ◯, △, and - in descending order. For instance, a process with high manufacturer/operator involvement may be chosen in the first half of the recovery stage, while a process with high consumer/user involvement may be selected in the second half. This ensures optimal restoration on the manufacturer/operator side and maximum satisfaction with the finished object on the consumer/user side.

6. Example of error recovery in product display

The task involves utilizing a manipulation robot to arrange products on shelves or in the display window of a convenience store. The objective is to position and posture the products optimally, as a tidy arrangement significantly boosts customers’ inclination to make purchases, as opposed to disorganized placements. Fig. 7 displays an image illustrating a pick-and-place task that transfers a single object from a product stock to a display space. Furthermore, Fig. 8 showcases the command sequence constituting a singular pick-and-place task. In this instance, the task revolves around evenly arranging 12 identical rectangular products in a layout of three rows

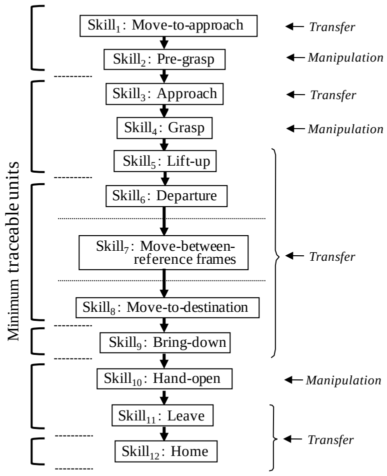


Fig. 8 Task sequence of picking and placing a product

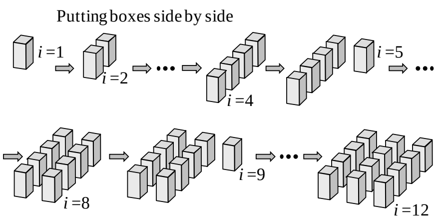


Fig. 9 Sequence of tasks related to display of products

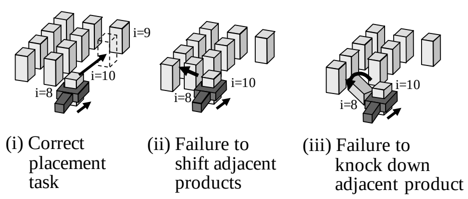


Fig. 10 Successes and Failures in tasks of display of products

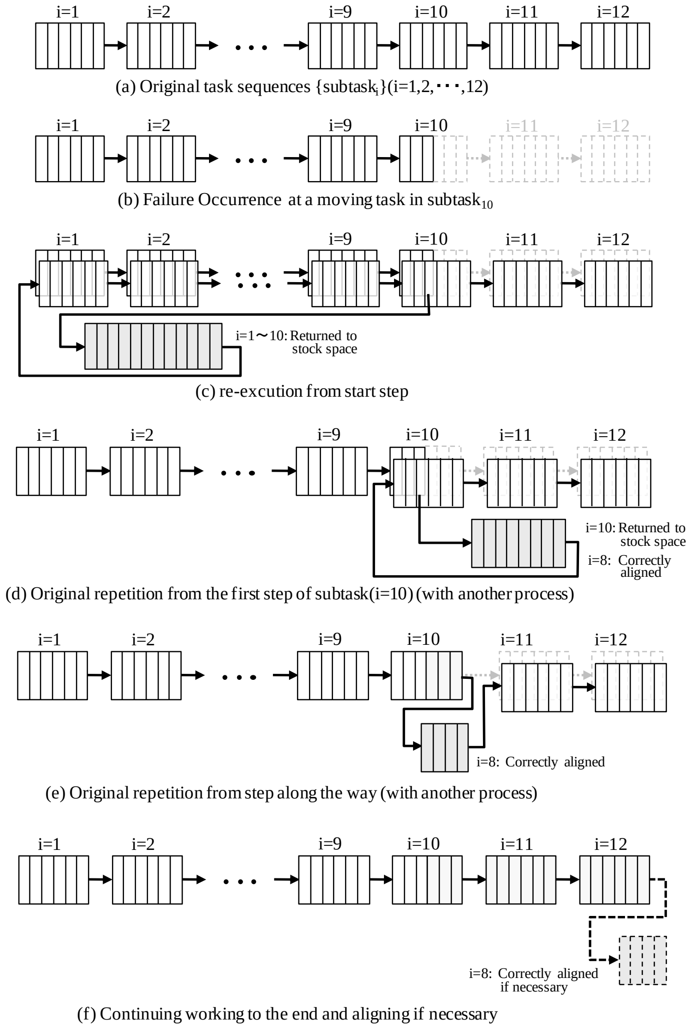


Fig. 11 Various recovery patterns

and four columns, as depicted in Fig. 9. Fig. 10 illustrates an error scenario; Fig. 10(i) demonstrates a carry-and-place operation for a pre-planned 10th object. Subsequently, Fig. 10(ii) portrays an instance where a hand interferes with the 10th object during the carry-and-place operation, causing the displacement of the 8th item from its original position. Finally, Fig. 10(iii) presents a scenario wherein the hand impacts the 8th item during the carry-and-place operations for the 10th item, resulting in the toppling of the 8th item.

Fig. 11 shows various recovery patterns. Fig. 11(a) illustrates normal planning in which no errors occur. Fig. 11(b) depicts the point at which the aforementioned failure occurred during the pick-and-place operation of the 10th object. Figs. 11(c)–11(f) show the respective recovery patterns. Fig. 11(c) demonstrates the error recovery of the backward type back to the start, in which the 1st object through the 8th object are returned to their original stock space, and the task is rerun from the beginning. Fig. 11(d) displays the error recovery of the backward type, where the recovery task is to return the 10th item to its original stock space, restore the 8th item to its original position and orientation, and pick and place the 10th item again. Fig. 11(e) shows the forward error recovery, which executes a recovery task that places the 10th item in its designated position even after an error occurs and then returns the 8th item to its original position and orientation, followed by the 11th pick-and-place. Fig. 11(f) shows a special type of forward error recovery, which is a recovery task that continues the work to the end even after an error occurs and returns the eighth product to its original position and orientation, if necessary. If a product is displayed in a store, it may not matter much if it is misplaced or even fallen over; that is how to deal with it. However, if it resonated with a customer's willingness to buy, it is executed. Although this is a particularly obvious example, it is sometimes desirable to perform the first half of the recovery task based on the operator's criteria and the second half of the recovery task based on the user's criteria.

7. Conclusion

When an error arises during the primary task, the process transitions into the recovery phase. This section explores diverse recovery paths utilizing our proposed error recovery method, grounded in both task stratification and error classifications. We introduced a method to systematically derive the optimal recovery path, aligning with evaluation standards for each section.

As discussed, numerous recovery paths exist, and choosing the right one poses challenges. While this paper introduced a method for optimal path selection, it does not delve into defining evaluation standards for each section. Addressing this aspect becomes imperative for future research endeavors.

References

1. B. R. Donald, Planning multi-step error detection and recovery strategies, Int. J. Robot. Res., 9(1) (1990) 3-60.
2. T. Niemueller, G. Lakemeyer and S. S. Srinivasa, A Generic Robot Database and its Application in Fault Analysis and Performance Evaluation, in Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst., (Vilamoura, Portugal, 2012), 364-369.
3. E. D. Lello, M. Klotzbucher, T. D. Laet and H. Bruyninckx, Bayesian Time-Series Models for Continuous Fault Detection and Recognition in Industrial Robotics Tasks, in Proc. IEEE/RSJ Int. Conf. on Intell. Robots Syst., (Tokyo, Japan, 2013), 5827-5833.
4. E. Krabbe, E. Kristiansen, 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.
5. A. S. Wang and O. Kroemer, Learning Robust Manipulation Strategies with Multimodal State Transition Models and Recovery Heuristics, in Proc. IEEE Int. Conf. Robot. Autom., (Montreal, Canada, 2019), 1309-1315.
6. A. Nakamura, K. Nagata, K. Harada, N. Yamanobe, T. Tsuji, T. Foissotte and Y. Kawai, Error recovery using task stratification and error classification for manipula-tion robots in various fields, in Proc IEEE/RSJ Int. Conf. on Intell. Robots Syst., (Tokyo, Japan, 2013), 3535-3542.
7. A. Nakamura, K. Nagata, K. Harada and N. Yamanobe, Technique of Recovery Process and Application of AI in Error Recovery Using Task Stratification and Error Classification, J. Robotics, Networking and Artificial Life, 5(1), 2018, pp. 56-62.
8. A. Nakamura, N. Yamanobe, I. R. Alpizar, K. Harada and Y. Domae, Cost-oriented Planning for Error Recovery in an Automation Plant, J. Robotics, Networking and Artificial Life, 6(4), 2020, pp. 225-230.
9. A. Nakamura, N. Yamanobe, I. R. Alpizar, K. Harada and Y. Domae, Selection of Optimal Error Recovery Process using Evaluation Standards in Automated Plants, J. Robotics, Networking and Artificial Life, 8(3), 2021, pp. 211-217.
10. A. Nakamura and Harada, Error Recovery Patterns Focusing on the Revival Process from Failures in Manipulation Tasks, J. Advances in Artificial Life Robotics, 3(4), 2023, pp. 230-237.
11. T. Hasegawa, T. Suehiro and K, Takase, A robot system for unstructured environments based on an environment model and manipulation skills, in Proc. IEEE Int. Conf. Robot. Autom., (Sacramento, USA, 1991), 916-923.
12. A. Nakamura, T. Ogasawara, T. Suehiro and H. Tsukune, Fine motion strategy for skill-based manipulation, Artificial Life and Robotics, Springer, 1(3), 1997, pp. 147-150.
13. A. Nakamura, K. Kitagaki and T. Suehiro, Using simplified geometric models in skill-based manipulation, Advanced Robotics, 18(8), 2004, pp. 835-858.

Authors Introduction

Prof. Akira Nakamura

He received the Ph.D. degree in Electrical Engineering from Keio University in 1991. From 2021, he has been working as a Professor at Faculty of Engineering of Saitama Institute of Technology. His research interests include robot planning, vision and control system.

Prof. Kensuke Harada

He received his Doctoral degrees in Mechanical Engineering from Kyoto University in 1997. From 2016, he has been working as a Professor at Graduate School of Engineering Science, Osaka University.