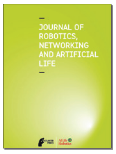
ATLANTIS PRESS

Journal of Robotics, Networking and Artificial Life
Vol. **6**(**4**); March (2020), *pp.* 225–230
DOI: https://doi.org/10.2991/jrnal.k.200222.004; ISSN 2405-9021; eISSN 2352-6386
https://www.atlantis-press.com/journals/jrnal



Research Article

Cost-oriented Planning for Error Recovery in an Automation Plant

Akira Nakamura[1,*], Natsuki Yamanobe[1], Ixchel Ramirez Alpizar[1], Kensuke Harada[1,2], Yukiyasu Domae[1]

[1]*Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology (AIST), Second Annex, AIST Tokyo Waterfront, 2-4-7 Aomi, Koto-ku, Tokyo 135-0064 Japan*
[2]*Robotic Manipulation Research Group, Systems Innovation Department, Graduate School of Engineering Science, Osaka University, 1-3 Machikaneyama, Toyonaka 560-8531, Japan*

ARTICLE INFO

Article History

Received 06 November 2019
Accepted 12 December 2019

Keywords

Cost-oriented planning
error recovery
task stratification
error classification
automation plant

ABSTRACT

In an automation plant, errors are more likely to occur during complicated tasks. In the case of a major error, the task is commonly re-executed after returning to the previous step. Therefore, deciding both the prior step that should be returned to and the recovery approach after return are important problems to consider. In this paper, cost-oriented planning of error recovery taking these two factors into account is proposed.

1. INTRODUCTION

In recent years, the automation of production plants has become a popular phenomenon. As high-mix low volume production is increasing, production methods differ from those in the case of large-scale mass production of the same products, and this has resulted in an increase in the cases of manufacturing process failure. Thus, the development of techniques to recover the processes is considered important.

Various studies on error recovery have been conducted in the manufacturing and robot research fields [1–6]. While several studies have been conducted on techniques to add a correction to the input of plants with a small error and perform recovery, there has been insufficient research on methods to return to the previous step of failure occurrence in order to perform recovery.

In this study, we investigate returning to a previous step for error recovery. We have considered methods of error recovery using task stratification and error classification [7–9]. In particular, we have shown techniques of plant correction which are performed using an error cause derived by the proposed method, and a method of planning for error recovery in consideration of cost has been proposed. When deriving the recovery process, multiple possible courses appear in several cases; in our method, the most suitable course is chosen by considering the cost of error recovery.

The remainder of this paper is organized as follows. The concept of skill is described in Section 2. The technique of failure recovery using error classification is explained in Section 3. A method of cost-oriented planning for error recovery is proposed in Section 4. Finally, in Section 5, a simple sample is shown using error occurrences in an assembly task.

2. CONCEPT OF SKILLS

In this paper, a “*skill*” implies the unit of motion. This section explains skills which are components of human behavior and machine motion [10–12].

2.1. Skill Primitives

We have derived the motion primitives which constitute tasks such as assembly and transport by analyzing a person’s behavior. The motion unit is defined as a “skill” [10–12]. Three basic skills, “move-to-touch,” “rotate-to-level,” and “rotate-to-insert,” are important (Figure 1). The representative person’s behavior can be composed of these three basic skills and similar ones. The units of machine motion can be considered to be similar to the rudiments of human behavior.

2.2. Stratification of Tasks

In general, it is possible to regard a task performed by a machine as following a hierarchical structure (Figure 2) [10–12]. One tier above the layer “$task^{(i)}$” is the layer “$task^{(i+1)}$”. The “*skill*” layer is located in the lowest layer, “$task^{(0)}$”.

[*] *Corresponding author. Email: a-nakamura@aist.go.jp*

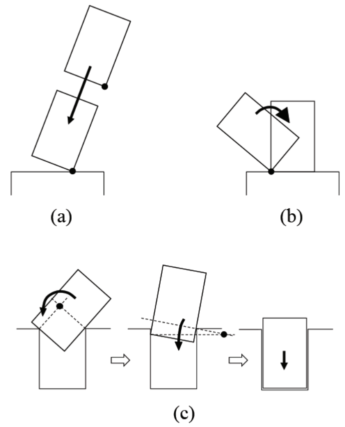


Figure 1 | Three fundamental skills. (a) Move-to-touch skill. (b) Rotate-to-level skill. (c) Rotate-to-insert skill.

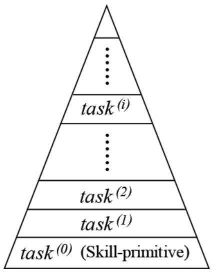


Figure 2 | Hierarchy of tasks.

3. ERROR RECOVERY

In the actual environment, unlike the ideal case, various factors can cause errors in machine performance. This section describes an error classification concept and our error recovery technique [7–9].

3.1. Error Classification

Errors can be classified into several groups based on the possible causes. We consider four error groups: execution, planning, modeling, and sensing (Figure 3) [7–9].

3.2. Error Recovery based on Classification

First, when the error occurs, the cause is estimated. Next, suitable correction of the system is performed based on the tentative cause. The process returns to the previous step, and the task is executed again from this step (Figure 3) [7–9]. The same error is less likely to occur because the corrected process has been executed.

If the scale of the error is small, the process returns to the previous step in the lowest layer of the hierarchy (Figures 3 and 4). Conversely, if the scale of the error is large, the process returns to the previous step in the highest-ranking layer of the hierarchy, and it is executed again from this step (Figure 4).

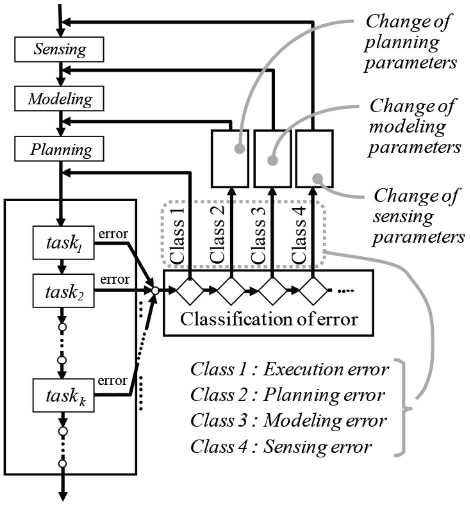


Figure 3 | Fundamental process flow with error recovery.

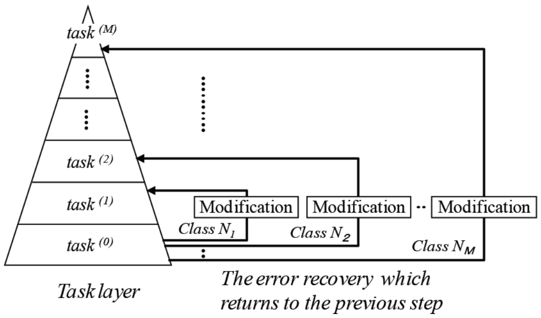


Figure 4 | The expression of task stratification and the process flow of the error recovery.

4. COST-ORIENTED ERROR RECOVERY

In this section, the candidate error recovery processes and the selection of the appropriate recovery process are explained.

4.1. Candidate Processes for Recovery

In the previous section, it was shown that the step to which the process returns after failure occurrences alters according to the scale of the error. However, it is possible to return further back than a single step of necessity minimum.

Figure 5 shows various possible error recovery processes for failure occurrences at $Subtask_m$ in Task $\boldsymbol{T}$ [from Start $\boldsymbol{S}$ to Goal $\boldsymbol{G}$], where Task $\boldsymbol{T}$ consists of sequence [$Subtask_1$, $Subtask_2$, $Subtask_3$, ..., $Subtask_m$, ..., $Subtask_n$] and $Subtask_k$ (k = 1, 2, 3, ..., n) consists of sequence [$Skill^k_1$, $Skill^k_2$, $Skill^k_3$, ..., $Skill^k_{n\_k(\max)}$]. $Subtask_m$ indicates the minimum traceable unit described in Nakamura et al. [8], i.e., the smallest unit in which it is necessary to return to the first node of a skill primitive sequence if a failure occurs. A skill where a failure occurs is part of the sequence [$Skill^m_1$, $Skill^m_2$, $Skill^m_3$, ..., $Skill^m_{n\_k(\max)}$]. When no problems are encountered, there is possibility that the execution of several skills has continued within $Subtask_m$, without returning to the previous step immediately after the failure occurrence.

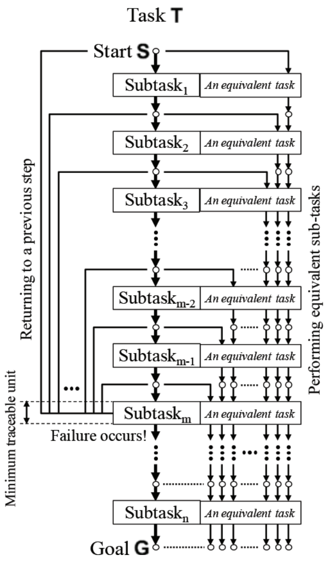


Figure 5 | Various processes of error recovery considered for a failure occurred in $Subtask_m$.

When the process returns to the node before $Subtask_j$ (j is one of 1, 2, 3, …, m), the process [$Subtask_j$, $Subtask_{j+1}$, ..., $Subtask_n$] is executed in the corrected system (the center bold arrows of Figure 5). However, re-execution using the same sequence of subtasks is not generally possible owing to several factors such as the transformation of the target object and the change of arrangement of the objects. In such a case, the rerun is performed by using an equivalent task of the original sequence of subtasks (the right-side arrows of Figure 5). The equivalent task is derived by large-scale re-planning.

4.2. Selection of the Lowest-cost Process

It is necessary to select one out of the multiple possible error recovery processes. Let us consider that it is most suitable to choose the process with minimum practical costs, where costs include material charges, parts charges, electricity bills, and planning expenses. If failure occurs, error recovery is planned, several task sequences for recovery are derived, these costs are calculated, and the most suitable recovery sequence is then chosen.

5. COST-ORIENTED RECOVERY IN ASSEMBLY TASKS

In this section, the cost-oriented error recovery discussed in Section 4 is observed through a simple sample exercise.

5.1. Candidate Processes for Recovery

Let us consider an assembly task in which a nameplate is stuck to a product by four precision screws. Figure 6 shows a sequence of the task. First, a tacking task is performed (Figure 6a–6d) in which these four screws are fastened temporarily to the nameplate, placed levelly; second, an erecting task is performed (Figure 6e and 6f) in which the nameplate with four screws stands up; third, a touching task is performed (Figure 6g and 6h) in which the nameplate is moved to the fixing location; and finally, an installation task is performed (Figure 6i–6m) in which the nameplate is fixed on a plane of the product.

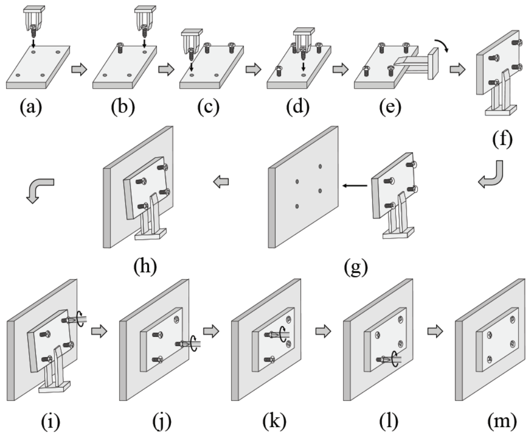


Figure 6 | An assembly task in which a nameplate is stuck to a product by four precision screws. (a–d) a tacking task, (e–f) an erecting task, (g–h) a touching task, (i–m) an installation task.

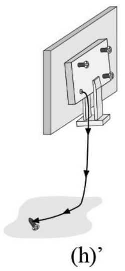


Figure 7 | An error in which a screw is dropped at (h) in Figure 6.

Next, we will consider an error, as shown in Figure 7, in which a precision screw is dropped when a failure occurs at Figure 6h. It is possible to derive various processes for error recovery of this failure. In this example, we consider three types of recovery courses as follows.

- [ER-I] Error Recovery I

First, we will consider a recovery process that it is easy to understand. The method is rerun from Start ***S***. The nameplate from which one screw in four screws was omitted is discarded, and the re-start is executed using a new nameplate and four new screws, that is, the tacking task (Figure 6a–6d) is performed using new parts.

- [ER-II] Error Recovery II

Now, we will consider a recovery process carried out by returning to the previous step of a skill primitive (Figure 6d) in the tacking task. Figure 8 shows the process with respect to recovery. The sequence from Figure 8p–8r is an additional process required to return to the previous step. The de-touching task (Figure 8h–8p) and leveling task (Figure 8q and 8r) are performed. It is noted that the sequence from Figure 8d–8m is essentially the same process as that from Figure 6d–6m.

There are two types of methods for use of a precision screw at the hole of the fallen screw. One is the use of new screw from the parts box, and another is the reuse of the fallen screw by searching and lifting. The former is shown with [ER-II(N)] and the latter with [ER-II(F)].

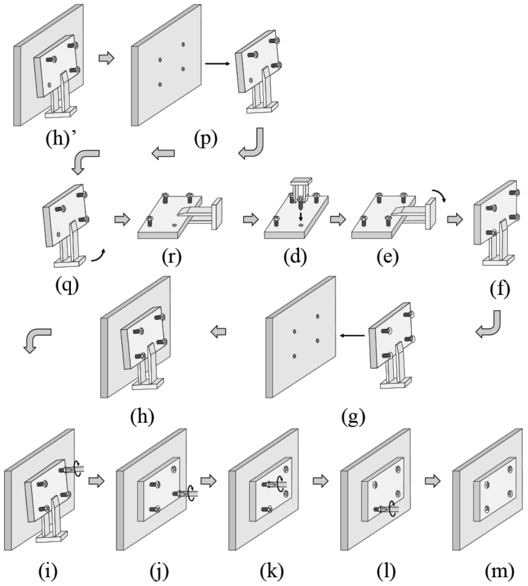


Figure 8 | [ER-II] Error recovery II.

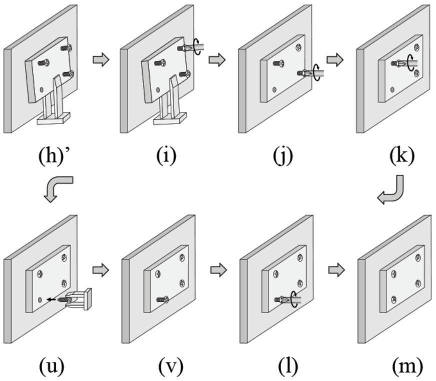


Figure 9 | [ER-III] Error recovery III.

- [ER-III] Error Recovery III

For the third process, we will consider the method for a very small scale of return, that is, a recovery process carried out by returning to a skill primitive in *Subtask*$_m$ of the failure occurrence. The method to immediately rerun the tacking task of a screw at the hole for the fallen screw (Figure 7) in order to regenerate the state of Figure 6h is the easiest to understand. However, we will consider the steady method in which the process of error recovery is carried out after a section *Subtask*$_m$ of the installation task (Figure 6i–6k) is finished. Figure 9 shows the process with respect to recovery after failure occurrence (Figure 7). Figure 9i–9m corresponds to Figure 7i–7m, respectively. The sequence from Figure 9u and 9v is an additional process to tack a screw into the hole of the nameplate and is performed after the sequence from Figure 9i–9k because of task reliability. In particular, the task to temporarily fasten a screw to the hole of the nameplate is more difficult when the nameplate is fixed perpendicularly than when it is placed levelly.

There are two types of methods for use of a screw at the hole intended for the fallen screw similar to [ER-II]; [ER-III(N)] and [ER-II(F)] show the methods using a new screw and reusing the fallen screw, respectively.

5.2. Selection of Recovery Process based on Cost

Next, the calculation of cost is performed for each recovery process. We consider three types of cost: recovery planning cost $C(R)$, object searching cost $C(S)$, and parts purchasing cost $C(P)$. Here, a unit of cost is introduced for simplification; we assume that the fixed number “U” indicates a unit of cost. However, in practical terms, the cost cannot always be related to a fixed value.

First, we consider $C(R)$, which represents the cost generated by the process added with a recovery. The $C(R)$ value of [ER-I] is smaller because it follows the same sequence to the original, and conversely, that of [ER-III] is larger because it follows a sequence with a difficult skill primitive. We suppose that $C(R)$ of [ER-I] = U, $C(R)$ of [ER-II] = $2U$, and $C(R)$ of [ER-III] = $3U$.

Second, we consider $C(S)$, which represents the cost generated by the process added for finding a screw. Then, let us assume that [Sch-N] indicates the process of finding a new screw in the parts box, and [Sch-F] indicates the process of finding the fallen screw in the working environment. The $C(S)$ value of [Sch-N] is smaller than that of [Sch-F], and thus, we suppose that $C(S)$ of [Sch-N] = $2U$, and $C(S)$ of [Sch-F] = $5U$.

Third, we consider $C(P)$, which represents the cost generated by the purchase of parts. Then, let us assume that [Pch-S] indicates the purchase expense of a new screw, and [Pch-P] indicates the purchase expense of a new nameplate with four new screws. The $C(P)$ value of [Pch-P] is significantly larger than that of [Pch-S], and thus, we suppose that $C(P)$ of [Pch-S] = U, and $C(P)$ of [Pch-P] = $8U$.

We now calculate a cost for each case, considering the total cost TC = $C(R) + C(S) + C(P)$.

$$\begin{aligned}\text{TC [ER-I]} &= C(R)\text{ of [ER-I]} + 0 + C(P)\text{ of [Pch-P]}\\ &= U + 0 + 8U\\ &= 9U\end{aligned} \tag{1}$$

$$\begin{aligned}\text{TC [ER-II}(N)] &= C(R)\text{ of [ER-II]} + C(S)\text{ of [Sch-N]}\\ &\quad + C(P)\text{ of [Pch-S]}\\ &= 2U + 2U + U\\ &= 5U\end{aligned} \tag{2}$$

$$\begin{aligned}\text{TC [ER-II}(F)] &= C(R)\text{ of [ER-II]} + C(S)\text{ of [Sch-F]} + 0\\ &= 2U + 5U + 0\\ &= 7U\end{aligned} \tag{3}$$

$$\begin{aligned}\text{TC [ER-III}(N)] &= C(R)\text{ of [ER-III]} + C(S)\text{ of [Sch-N]}\\ &\quad + C(P)\text{ of [Pch-S]}\\ &= 3U + 2U + U\\ &= 6U\end{aligned} \tag{4}$$

$$\begin{aligned}\text{TC [ER-III}(F)] &= C(R)\text{ of [ER-III]} + C(S)\text{ of [Sch-F]} + 0\\ &= 3U + 5U + 0\\ &= 8U\end{aligned} \tag{5}$$

The following relation is derived from Equations (1) by (5).

$$\begin{aligned}&\text{TC [ER-II}(N)] < \text{TC [ER-III}(N)] < \text{TC [ER-II}(F)]\\ &< \text{TC [ER-III}(F)] < \text{TC [ER-I]}\end{aligned}$$

Thus, TC [ER-II (N)] will have the smallest price, indicating that the case of [ER-II] Error Recovery II with use of a new screw instead of the dropped screw has the lowest cost.

6. CONCLUSION

In this study, a cost-oriented planning for error recovery was proposed. In general, more than one sequence is derived when process of error recovery is considered. This paper explains the method of selection from the multiple possible courses. In our method, the most suitable sequence is chosen based on cost.

In the future, we expect to conduct further research on the derivation of the error recovery process. A selection method besides the cost will be considered, such as working hours and available tools, which are important in automation plants.

CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

REFERENCES

[1] B.R. Donald, Planning multi-step error detection and recovery strategies, Int. J. Robot. Res. 9 (1990), 3–60.

[2] T. Fukuda, M. Nakaoka, T. Ueyama, Y. Hasegawa, Direct teaching and error recovery method for assembly task based on a transition process of a constraint condition, Proceedings 2001 ICRA, IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), Seoul, IEEE, South Korea, 2001, pp. 1518–1523.

[3] M. Scheutz, J. Kramer, Reflection and reasoning mechanisms for failure detection and recovery in a distributed robotic architecture for complex robots, Proceedings 2007 IEEE International Conference on Robotics and Automation, IEEE, Roma, Italy, 2007, pp. 3699–3704.

[4] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, S. Schaal, Skill learning and task outcome prediction for manipulation, 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 2011, pp. 3828–3834.

[5] T. Matsuno, J. Huang, T. Fukuda, Fault detection algorithm for external thread fastening by robotic manipulator using liner support vector machine classifier, 2013 IEEE International Conference on Robotics and Automation, IEEE, Karlsruhe, Germany, 2013, pp. 3443–3450.

[6] A.S. Wang, O. Kroemer, Learning robust manipulation strategies with multimodal state transition models and recovery heuristics, 2019 International Conference on Robotics and Automation (ICRA), IEEE, Montreal, QC, Canada, 2019, pp. 1309–1315.

[7] A. Nakamura, T. Kotoku, Systematization of error recovery in skill-based manipulation, Artif. Life Robot. 14 (2009), 203–208.

[8] A. Nakamura, K. Nagata, K. Harada, N. Yamanobe, T. Tsuji, T. Foissotte, et al., Error recovery using task stratification and error classification for manipulation robots in various fields, 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, Tokyo, Japan, 2013, pp. 3535–3542.

[9] A. Nakamura, K. Nagata, K. Harada, N. Yamanobe, Technique of recovery process and application of AI in error recovery using task stratification and error classification, J. Robot. Netw. Artif. Life 5 (2018), 56–62.

[10] T. Hasegawa, T. Suehiro, K. Takase, A model-based manipulation system with skill-based execution, IEEE Trans. Robot. Autom. 8 (1992), 535–544.

[11] A. Nakamura, T. Ogasawara, T. Suehiro, H. Tsukune, Skill-based backprojection for fine motion planning, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS ‘96, IEEE, Osaka, Japan, 1996, pp. 526–533.

[12] A. Nakamura, T. Ogasawara, K. Kitagaki, T. Suehiro, Using robust and simplified geometric models in skill-based manipulation, Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No.01CH37180), IEEE, Maui, HI, USA, 2001, pp. 138–145.

AUTHORS INTRODUCTION

Dr. Akira Nakamura

He received the PhD degree in Electrical Engineering from Keio University in 1991. He is currently a research staff at National Institute of Advanced Industrial Science and Technology (AIST). His research interests include robot planning, vision and control system.

Dr. Natsuki Yamanobe

She received the PhD degree from the University of Tokyo in 2007. She is currently a Senior Research Scientist with Artificial Intelligence Research Center of Advanced Industrial Science and Technology (AIST).

Dr. Ixchel Ramirez Alpizar

She received her PhD degree in Mechanical Engineering from the Graduate School of Engineering, Osaka University, Japan, in 2013. Since 2019, she has been working as a senior researcher at the Artificial Intelligence Research Center of AIST. Her research interests include robot learning from demonstration, robotic manipulation and human-robot collaboration. She is a member of IEEE and RAS.

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.

Dr. Yukiyasu Domae

He received his PhD degree from Hokkaido University in 2014. He is currently a team leader with Artificial Intelligent Research Center (AIRC), National Institute of Advanced Industrial Science and Technology (AIST).