Manufacturing systems are affected by uncertainties, such as machine failure or tool breakage, which result in system downtime and productivity deterioration. In machining processes, system downtime must be reduced. This study aims to establish an automated scheduling technique that flexibly responds to unforeseen events, such as machine failure, based on adaptive operations of the handling manipulator instead of an operation schedule for the machine tools. We propose an “adaptive manipulation” procedure for establishing a reactive revision policy. The reactive revision policy modifies a portion of the manipulator operation sequence, followed by the machine operation sequence. We conduct a physical scheduling simulation on a material-handling manipulator system imitating a job-shop manufacturing system. Through simulations involving machine breakdown scenarios, the applicability of the reactive revision policy based on adaptive manipulation is demonstrated.

Keywords: reactive scheduling, material-handling manipulator system, automated scheduling system, machine failures, job-shop scheduling

1. Introduction

Machining manufacturing systems are affected by uncertainties, such as machine failure or tool breakage, which result in system downtime and productivity deterioration. In machining processes, system downtime must be reduced. The preventive maintenance of machine tools using IoT devices [1] has been developed over the last decade. However, it is difficult to prevent failures and unexpected downtime in advance. Therefore, it is essential to devise an automated scheduling method that responds to productivity deterioration such that manufacturing operations return to the original steady state [2]. In manufacturing, the problem above is recognized in flexible automation as implementing a flexible CNC machining manufacturing system integrated with a material-handling manipulator, which has been implemented by manufacturers including small-medium ones [3]. In studies pertaining to manufacturing systems equipped with material-handling manipulators, problems involving manipulator motions and machine scheduling [4] have been analyzed, where robotic cell problems were emphasized. The robotic cell problem is within the framework of a general permutation flow shop [5]. However, it is investigated in static and ideal scheduling situations in many studies.

This study aims to establish an automated scheduling technique that flexibly responds to unforeseen events, such as machine failure, based on the adaptive operations of the handling manipulator instead of job scheduling for machine tools. We first introduce a “regular manipulation,” which is typically used in practice, and a right-shift scheduling policy that creates a delayed schedule without any modifications to a job sequence. Subsequently, we propose an “adaptive manipulation” procedure to establish a reactive revision policy. The reactive revision policy modifies a portion of the manipulator operation sequence, followed by the machine operation sequence. We conduct a physical scheduling simulation on a material-handling manipulator system imitating a job-shop manufacturing system. We demonstrate the applicability and effectiveness of a reactive revision policy based on adaptive manipulation through scenario-based scheduling simulations using a material-handling manipulator system imitating a job-shop manufacturing system comprising two machines.

2. Related Studies

Numerous studies pertaining to industrial scheduling and production planning have addressed the various approaches used and applications developed in recent decades. Ogata et al. proposed a hierarchical problem-solving method for optimizing casting scheduling, where the problem was classified into a slab scheduling frame problem and a slab width decision problem [6]. Samukawa and Suwa provided an optimization solution for determining the optimal energy load profile with multiple processing/operation modes. They proposed a multi-start local search method to obtain an energy load profile [7]. Yang et al. proposed a scheduling model that considers degradation effects and preventive multi-maintenance activities as well as reject penalties to model production systems realistically; additionally,
they presented an unrelated parallel machine scheduling problem [8]. Ishigaki and Matsui proposed a two-step neighborhood algorithm that addresses changes in machine allocation and job order, where machine allocation and job orders are changed simultaneously but independently [9]. Sakaguchi et al. developed an optimized nesting and scheduling method based on a coevolutionary genetic algorithm, while considering the trade-off between nesting and scheduling problems in sheet metal processing [10]. Tanizaki et al. proposed a scheduling algorithm that combines meta-heuristics and operational simulation for production processes in which cranes interfere with each other [11]. Morita and Suwa developed a new approach for generating buffered schedules in critical chain scheduling and provided an exact optimization method for assigning time buffers to the baseline schedule of a project [12].

Over the last two decades, efficient heuristics have been developed in studies associated with automated manufacturing systems involving FMSs heuristics [13–15]. Akturk et al. considered robot speed control for a robotic cell [16]. Sawada et al. proposed a modeling and scheduling algorithm for a transportation system in the semiconductor manufacturing industry, focusing on the maximization of throughput and minimum transit time of AGVs [17]. Similar to our research, Ham provided a simultaneous scheduling method for a production machine and material-handling manipulator. We considered a job-shop-type manufacturing system with several CNC machine tools (machines for simplicity) and a material-handling manipulator loading/unloading a workpiece into/from a machine. Here, \( n \) workpieces are to be machined during planning horizon \( T \). The processing of a workpiece is referred to as a job herein. Let \( O_{jk} \) denote the “machining operation” of job \( j \in J = \{1, \ldots, N\} \) on machine \( k \in M = \{1, \ldots, M\} \), denoted by \( M_k \). Here, \( N \) indicates the number of job types.

The manipulator performs two primary operations, i.e., loading and unloading, as described in the following.

![Illustrative example of Gantt chart of two machines and material-handling manipulator.](image)

1. **Loading operation**: The manipulator grasps a target workpiece placed in a designated area (e.g., a worktable) and transports it to a specific place (e.g., a pallet or fixed jig) based on a planned schedule.

2. **Unloading operation**: The manipulator grasps a finished workpiece at a specific location near the machine and shifts it to a predetermined location at every scheduled finish time of the processing operations.

The time required for loading (or unloading) can be modeled as the setup time before (or after) the corresponding machining operation. However, the machining operation might not begin immediately after loading, or unloading might not be conducted immediately after the machining operation is completed. Therefore, we implemented loading and unloading operations to be scheduled explicitly on a Gantt chart. Let \( A_{jk} \) and \( B_{jk} \) denote a “loading operation” before \( O_{jk} \) begins and a “unloading operation” after \( O_{jk} \) is completed, respectively.

\( A_{jk} \) must be completed prior to the start of \( O_{jk} \), and \( B_{jk} \) cannot begin before \( O_{jk} \) is completed. By introducing a precedence relation \( \prec \), those constraints are described as \( A_{jk} \prec O_{jk} \prec B_{jk} \). Next, we denote the processing times of operations \( O_{jk} \), \( A_{jk} \), and \( B_{jk} \) by \( p_{jk} \), \( p_{jk}^A \), and \( p_{jk}^B \), respectively. Based on the preliminaries above, we assume that job \( j \), denoted by \( J_j \), comprises three types of operations, i.e., \( A_{jk}, O_{jk}, \) and \( B_{jk} \). Additionally, we denote \( n_j \) as the production volume of \( J_j \).

**Figure 1** shows illustrative examples of a Gantt chart for two machines (\( M_k \) and \( M_k' \)) and a handling manipulator (MH). In **Fig. 1(a)**, both operations \( O_{jk} \) and \( O_{jk}' \) are to begin immediately after they are loaded into the cor-
Reactive Scheduling Based on Adaptive Manipulator Operations in a Job Shop Configuration with Two Machines

4. Schedule Revision Policies Based on Manipulations

4.1. Regular Manipulation and Right-Shift Policy

Provided that we understand the manner by which system integrators implement a series of loading and unloading manipulations and their fixed sequence into a robot controller, then all types of uncertainties can be dispelled because, in general, sufficient safety and appropriate motions are the most crucial objectives. Therefore, they might not implement an extra program against unforeseen events and simplify the control sequence. Manipulations based on this approach is referred to as, in this study, a “regular manipulation.”

Suppose no workpiece is identified in the designated location of machine trouble, e.g., no door-open signal from the machine; in this case, the manipulator returns to the standby position until the failure machine recovers. The delay incurred by the machine trouble is propagated to subsequent loading and unloading operations and other machines. Under the regular operation of the manipulator, a decision-maker applies a “right-shift scheduling policy” [22] from a scheduling perspective, described as follows.

**Right-shift scheduling policy (P1):** The current schedule is modified by shifting unprocessed operations to the right over time as much as possible based on the estimated downtime, thereby yielding a delayed but feasible schedule without any changes in the operation sequence.

Figure 2(a) shows an illustrative example of the mechanism policy P1 in response to a machine breakdown; Fig. 2(b) shows a right-shifted schedule after a machine failure occurs at the end of O32. In the regular manipulation, the manipulator returns to the standby position until it receives a recovered signal from M2; B_{22} is shifted to the right, followed by A_{32}, B_{41}, A_{11}, and B_{32}.

4.2. Adaptive Manipulation and Reactive Scheduling Policy

We propose an “adaptive manipulation” scheme that involves implementing an additional logic to reduce the idle time incurred by machine failures. The manipulator continues loading/unloading using the available machines. It verifies a completed workpiece in the designated location of the failed machine at a regular interval using a CCD camera and an image-recognition function.

The implementation of adaptive manipulation enables us to establish a reactive schedule revision in response to...
machine failure. From a scheduling decision perspective, we can avoid the deterioration of machine utilization by modifying the current manipulator operations sequence. Therefore, we propose a “reactive revision policy” based on the adaptive manipulation, as follows.

**Reactive revision policy (P2):** The handling manipulator is never resumed unless the other machine is not available, which indicates that jobs on the available machine are preferentially processed until the failure machine recovers.

The optimization of schedule revision is omitted herein, and the manipulator does not manage its operation schedule (sequence). This is because the adaptive manipulation in this study operates myopically, and the manipulator controller is not equipped with a scheduling function (scheduler). Fig. 3 illustrates the reactive revision based on adaptive manipulation, in which unloading B_{32} and loading operations A_{32} move backward because they cannot execute. Operations B_{31} and A_{11} are conducted prior to them.

If the recovery of M$_2$ prolongs, then the loading operation related to the machining operation assigned immediately after O$_{11}$ will move forward before B$_{32}$ and A$_{32}$, followed by its unloading operation. The revision policy P$_2$ performs this myopic revision sequentially and repeatedly based on the regular inspection of a completed workpiece on the failed machine, M$_2$.

5. Scheduling Experiments

We performed a series of scheduling simulations using a real manipulator system to evaluate the proposed reactive revision policy by comparing it with the right-shift policy. A manufacturing situation in which a two-machine (M = 2) job-shop system manufactures four different types of jobs (N = 4) was considered.

5.1. Hardware Configurations

Figure 4 shows an experimental manipulator station imitating a two-machine job-shop manufacturing system. A six-axis robotic manipulator (FANUC CR-7iA/L) was used to load and unload the workpieces. A force sensor and CCD camera were installed on the manipulator hand to enable flexible operations based on image recognition. Every operation of the handling manipulator began from a predetermined standby (or home) position. The handling station comprised loading and unloading worktables on both sides of the manipulator. Each worktable can hold 15 workpieces at most, and is conveniently divided into five areas to handle four jobs: M (J$_j$) for raw materials, W (J$_j$) for work-in-process, and F (J$_f$) for the final product obtained through processing job$_j$. Each area on the worktable can hold three workpieces at most.

M$_1$ and M$_2$ in Fig. 4 represent CNC machine tools; however, they are not physical tools in this study because our focus is to investigate the efficacy of adaptive operation planning of the material-handling manipulator. Areas M$_1$ and M$_2$ in Fig. 4 contain two small plates that imitate the pallets. A workpiece on the plate indicates that cutting is completed and that the workpiece is ready to be removed. The absence of a workpiece on the plate at its planned completion time indicates that the machine is still operating or in the idle state but is holding a workpiece inside. In this case, a certain duration is intentionally substituted for the processing time of the machine such that

1. the workpiece is removed by hand immediately after the manipulator has placed the workpiece on the plate; subsequently,
2. the workpiece is returned to the plate after the predetermined time, which corresponds to the operation processing time to be introduced in the next section.

We determined the processing time using a stopwatch. This was conducted each time the manipulator shifted the workpiece to M$_1$ or M$_2$.

5.2. Simulation Schemes

5.2.1. Problem Instances and Baseline Schedule

Table 1 shows a problem instance involving four jobs, each of which entails different job routing, and processing
times. Job types J1 and J2 have only one task processed on M1 and M2, denoted by O_{11} and O_{22}, respectively. J3 has two processing operations: O_{31} on M1 and O_{32} on M2, where O_{31} ≺ O_{32}. Similarly, J4 is composed of two processing operations, O_{42} and O_{41}, where O_{42} ≺ O_{41}. We measured the actual loading and unloading operation times $p^A_{jk}$ and $p^B_{jk}$ via preliminary experiments and set their respective average times. The actual processing time ranges were approximately $p^A_{jk} ± 1$ and $p^B_{jk} ± 1$ s. Therefore, we expect a difference between the planned and realized schedules.

Next, we propose the following procedure to generate a baseline schedule for J1, J2, J3, and J4 including sequencing of manipulator operations.

Step 1 (Sequencing): Set a sequence on each machine with the shortest makespan (the minimum makespan $c_{max} = \max c_{jk} \{ c_{jk} \} = 150$ s).

1. $M_1 : \quad J_1(O_{11}) \rightarrow J_3(O_{31}) \rightarrow J_4(O_{41})$
2. $M_2 : \quad J_2(O_{42}) \rightarrow J_2(O_{21}) \rightarrow J_3(O_{32})$

Step 2 (Loading first rule): In Fig. 5, the starting time $s_{jk}$ of O_{jk} is set to the completion time of A_{jk}, denoted by $c^A_{jk}$, i.e.,

$$s_{jk} = c^A_{jk} = s^A_{jk} + p^A_{jk},$$

where $s^A_{jk}$ denotes the starting time of A_{jk}. Loading operation (A) is performed prior to unloading operation (B) such that every processing operation begins at the earliest time, i.e., $s^B_{jk} \geq c_{jk}$.

Step 3 (Cyclic scheduling): The sequences above are appended to the current schedule individually without an unforced idle time, yielding the following cyclic schedule:

1. $M_1 : \quad J_1 \rightarrow J_3 \rightarrow J_4 \rightarrow J_1 \rightarrow J_3 \rightarrow \cdots$
2. $M_2 : \quad J_4 \rightarrow J_2 \rightarrow J_3 \rightarrow J_4 \rightarrow J_2 \rightarrow \cdots$

Based on the preliminary results above, we assume that the manufacturing system processes 40 workpieces ($n = 40$) within the planned horizon $T = 2425$ s where the production volume of each job type is $(n_1,n_2,n_3,n_4) = (10,10,10,10)$.

### 5.2.2. Machine Failure Scenarios

We considered a situation in which any one of the machine tools was subjected to breakdown. Let $D_l$ denote machine failure scenario $l$. In scenario $D_l$ ($l = 1, \ldots, L$), either $M_1$ or $M_2$ is intended to be broken down at a point in time $t_l$ with an estimated downtime $t_l$. We employed five characteristic scenarios ($L = 5$), as summarized in Table 2 in the physical simulations stated later.

The occurrence time of failure $b_l$ exhibited an exponential distribution with mean $1/\lambda = 180$ s, and the downtime length $t_l$ was randomly assigned. Scenarios $D_1$ and $D_2$ with the same downtimes involved two machine failures: earlier or later during planning horizon $T$. In $D_3$ and $D_4$, a machine failure with a longer downtime occurred in the middle of $T$. Scenario $D_5$, which involved frequent machine failures, was an extreme scenario; however, the difference in flawed schedules between the two policies can be significant.

### 5.3. Simulation Results

We performed a series of scheduling simulations involving machine failure scenarios on the experimental manipulator system, as depicted in Fig. 4. We measured the cycle time and the number of finished jobs, denoted by $n_f$ within $T$.

Figure 6 depicts the changes in the number of completed jobs over $[0,2400]$, in which an increasing tangent indicates the production speed [pcs/s]. Scenario $D_1$ with a slight delay yielded similar results between policies $P_1$ and $P_2$. The production speed decreased immediately after machine failure; however, it maintained the ideal production speed. The same tendency was observed in scenario $D_2$. In the remainder of the scenario, the production speed by $P_1$ seriously slows down in comparison with that by $P_2$. The results indicate that the production speed can be adjusted to the original production speed by applying reactive revision in a delayed schedule. Performing right-shifting only is insufficient when large-scale or frequent machine failures occur.

Figure 7 shows the average cycle time over time. Under an ideal circumstance (i.e., no machine failures), the average cycle time converges to a theoretical cycle time, i.e., $60.6$ s ($= 2425/40$). The measured average cycle time was $61.1$ s. After machine failure occurred, the cycle
Table 2. Machine breakdown scenarios.

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Number of machine breakdowns</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>b_1</td>
</tr>
<tr>
<td>broken machine</td>
<td>M_1</td>
</tr>
<tr>
<td>b_4</td>
<td>998</td>
</tr>
<tr>
<td>t_4</td>
<td>680</td>
</tr>
<tr>
<td>broken machine</td>
<td>M_1</td>
</tr>
<tr>
<td>b_5</td>
<td>109</td>
</tr>
<tr>
<td>t_5</td>
<td>95</td>
</tr>
</tbody>
</table>

Fig. 6. Production speed.

The cycle time increased for both policies. By applying reactive revision P_2, the cycle time recovered sooner compared with applying P_1. The cycle time increased significantly immediately after large-scale or frequent machine failures occurred, and a significant amount of time was required for recovery, particularly in scenarios D_3 and D_5. The right-shift and reactive revision policies provide similar results if machine failures are insignificant. However, the schedule may be completely flawed if downtime occurs frequently.

We summarize the resulting number of finished jobs and the average cycle time with machine failure(s), together with an ideal result where manufacturing was executed as planned without any machine failures (“No failures” in Table 3).
Reactive Scheduling Based on Adaptive Manipulator Operations in a Job Shop Configuration with Two Machines

Table 3. Results of physical simulations (P₁: right-shift policy, P₂: reactive revision policy).

<table>
<thead>
<tr>
<th>Policy</th>
<th>Number of finished jobs</th>
<th>Cycle time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>D₁</td>
<td>P₁</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>P₂</td>
<td>35</td>
</tr>
<tr>
<td>D₂</td>
<td>P₁</td>
<td>35</td>
</tr>
<tr>
<td></td>
<td>P₂</td>
<td>36</td>
</tr>
<tr>
<td>D₃</td>
<td>P₁</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>P₂</td>
<td>37</td>
</tr>
<tr>
<td>D₄</td>
<td>P₁</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>P₂</td>
<td>37</td>
</tr>
<tr>
<td>D₅</td>
<td>P₁</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>P₂</td>
<td>32</td>
</tr>
<tr>
<td>No failures</td>
<td></td>
<td>40</td>
</tr>
</tbody>
</table>

6. Concluding Remarks

We proposed a reactive scheduling technique for a machining manufacturing system that involved a material handling manipulator based on adaptive manipulation. We first introduced regular manipulation and adaptive manipulation to realize a right-shift scheduling policy and a reactive revision policy, respectively. Subsequently, we performed scenario-based scheduling simulations on a real manipulator system testbed. The simulation results and discussions can be summarized as follows:

- As shown in Fig. 6, the final production speed of the P₁ and P₂ scenario was similar to that of the D₁ and D₂ scenarios, whereas in D₃ and D₄, the production speed when P₁ was applied was much lower than that when P₂ was applied. The same trend was observed in the graph of the average cycle time depicted in Fig. 7.

- The right-shift scheduling policy and reactive scheduling policy provided similar schedule performances when machine failure was slight, thereby resulting in a slight delay in the schedule.

- When machine failure was significant or occurred frequently, the reactive revision policy significantly outperformed the right-shift policy.

- By applying the reactive revision policy, the deteriorated production speed and cycle time recovered sooner compared with applying the right-shift policy.

The adaptive operations of the manipulator can enhance autonomous and automated scheduling if the manipulator controller has a scheduler function. In future studies, the following can be considered: (i) develop a theoretical model for schedule revision problems derived from a dynamic manufacturing environment with uncertainty;
(ii) implement a scheduling function into the robot control program to enhance adaptive manipulations for flexible material handling.

Acknowledgements
This study was partially supported by Grants-in-Aid for Scientific Research of the Japan Society for the Promotion of Science (JSPS), No. 18K11740. The author would like to thank Tomoya Yudanii and Masahiko Matsumura for developing the simulation programs and providing fundamental data.

References:

Name: Ryo Yonemoto
Affiliation: Graduate School of Science and Engineering, Setsunan University

Address: 17-8 Ikeda-naka-machi, Neyagawa, Osaka 572-8508, Japan

Brief Biographical History:
2017 Received Bachelor of Engineering degree from Setsunan University
2017 Received Master of Engineering degree from Graduate School of Science and Engineering, Setsunan University
2017- Doctoral Student, Graduate School of Science and Engineering, Setsunan University

Membership in Academic Societies:
- Japan Society of Mechanical Engineers (JSME)

Name: Haruhiko Suwa
Affiliation: Professor, Department of Mechanical Engineering, Setsunan University

Address: 17-8 Ikeda-naka-machi, Neyagawa, Osaka 572-8508, Japan

Brief Biographical History:
1997 Received Ph.D. in Engineering from Kobe University
1997- Assistant Professor, Setsunan University
2001- Associate Professor, Setsunan University
2010- Professor, Setsunan University
2012-2013 Visiting Scientist, Massachusetts Institute of Technology

Main Works:

Membership in Academic Societies:
- Japan Society of Mechanical Engineers (JSME)
- American Society of Mechanical Engineers (ASME)
- Institute of Systems, Control and Information Engineers (ISCIE)
- Japan Society for Precision Engineering (JSPE)