# Operation Design Technology for Heterogeneous Robot System

*Hiroki Sugawara* [*1] *Yuya Okada* [*2] *Takeshi Hatanaka* [*2]

***In recent years, robots that can be deployed in process plants have begun to emerge and their use for on-site patrol inspections or as first responders in emergencies is being actively considered. However, current robots operate based only on pre-designed motions and pre-trained behaviors. This limits the tasks they can perform in real environments where diverse situations arise. Meanwhile, expanding the range of tasks through enhanced robot functionality and the development of low-cost robots is expected to increase the number of robots deployed. As these developments progress, robot operation will grow to a scale too massive for humans to handle. We aim to realize "Industrial Autonomy" and hypothesize that it is desirable for a robot management system to control all robots deployed in the field. To this end, we have been developing technology to design operation instructions for individual robots. In particular, we have developed a design algorithm that meets the computational time constraints required for real-world operations and is suitable for practical use.***

## INTRODUCTION

In recent years, various challenges have arisen at plant operation sites, including labor shortages, training of younger successors due to the retirement of experienced workers, and the need to strengthen competitiveness(1). Digital transformation (DX) has been promoted as a solution to these challenges, and various robots have been developed such as SPOT by Boston Dynamics, Inc., EX ROVR by Mitsubishi Heavy Industries, Ltd., ExR-2 by ExRobotics B.V., ANYmal by ANYBotics AG, and Taurob Inspector by Taurob GmbH. More new robots and further functional enhancements are expected in future, and the market for robots used in plant operations is projected to grow at an annual rate of approximately 10%(2).

Yokogawa Electric Corporation promotes industrial automation to industrial autonomy (IA2IA) as a pathway from industrial automation to industrial autonomy, in which plant facilities and operations themselves learn and adapt, and position robots as one of its enabling technologies(3). We have developed the OpreX Robot Management Core, which seamlessly connects plant operation systems with various types of robots, and we are working with customers to verify the implementation of robots on the plant floor. In plants where autonomous operation has been realized, as shown in Figure 1, diverse types of robots are expected to be deployed and to perform tasks autonomously in a complementary manner. However, robots that can currently be introduced into plants assume operation based on either manual control or automatic navigation through the replay of predefined motions(4)(5). Therefore, to enable the system to respond to various situations that were not anticipated at the time of predefined setup, it is necessary to develop several new functions.

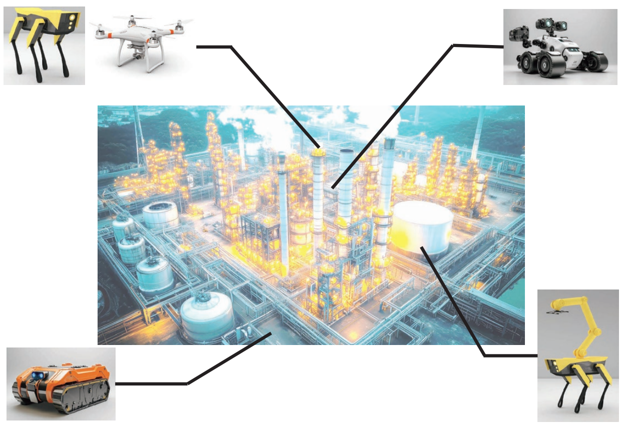

**Figure 1** Conceptual illustration of future robot operation in plants

In this study, we developed a technology for designing appropriate robot operation plans according to the situation as a key technology for achieving autonomous robot operation. This paper describes the developed technology.

## ISSUES AND TECHNICAL CHALLENGES IN ROBOT OPERATION IN PLANTS

### Issues in Robot Operation

Inspection tasks in plants are currently carried out by human workers using their superior five senses. In contrast, robots are equipped with limited functions, which means that

*1 DX Business Development Department, Innovation Center, Marketing Headquarters
*2 Hatanaka Lab, Department of Systems and Control Engineering, School of Engineering, Institute of Science Tokyo

there are some tasks they can perform and some they cannot. It is therefore necessary to assign robots according to their available functions. In the case of human workers, a single operator can inspect all items such as visual appearance, temperature, and abnormal sounds, and task are typically assigned by equipment or area (Table 1). In contrast, in the case of robots, it is assumed that robots equipped with the functions required for each task item share the work and carry out inspections of each piece of equipment in a mutually complimentary manner (Table 2).

**Table 1** Example of task assignment for humans

| Equipment | Task item | Worker A | Worker B |
|---|---|---|---|
| X | Visual inspection | ✓ | |
| | Temperature | ✓ | |
| | Vibration | ✓ | |
| | Abnormal sound | ✓ | |
| | Liquid level | ✓ | |
| Y | Visual inspection | | ✓ |
| | Temperature | | ✓ |
| | Vibration | | ✓ |
| | Abnormal sound | | ✓ |
| | Instrumentation check | | ✓ |

**Table 2** Example of task assignment for robots

| Equipment | Task Item | Robot α | Robot β | Robot γ |
|---|---|---|---|---|
| X | Visual inspection | ✓ | | |
| | Temperature | | ✓ | |
| | Vibration | | | ✓ |
| | Abnormal sound | | ✓ | |
| | Liquid level | ✓ | | |
| Y | Visual inspection | ✓ | | |
| | Temperature | | ✓ | |
| | Vibration | | | ✓ |
| | Abnormal sound | | ✓ | |
| | Instrumentation check | ✓ | | |

In terms of available working time, human workers can respond at any time within their scheduled shift hours. In contrast, for robots, the duration of continuous operation is determined by battery specifications. If this constraint is ignored, problems may occur, such as the battery being depleted during operation and the robot stopping at the site. Therefore, as shown in Figure 2, it is necessary to allocate a shutdown period for charging within the shift.

In addition, robots are more strongly affected by operational constraints in plants. Examples of such constraints include cases where the time for field patrols is specified due to on-site safety rules, or where equipment is switched on or off during inspection. When equipment is switched on or off, the appropriate inspection timing is often predefined based on operational know-how. Even so, the execution time becomes a specified scheduling constraint.

To design appropriate robot operations while satisfying these constraints, it is necessary to solve the multi-robot task allocation (MRTA) problem, in which each constraint and robot operation policy is formulated explicitly. For the MRTA problem, the approaches shown in Table 3 have been proposed to date[6].

The market approach allocates tasks in a manner similar to an auction, in which each robot submits the cost it would incur to accept a given task, and the task is sequentially assigned to the robot with the most favorable bid. The optimization approach, in contrast, formulates all operational constraints mathematically and defines robot operation policies as an objective function. By solving the resulting optimization problem, a desired operation plan is obtained. Examples of robot operation policies include completing tasks as early as possible and minimizing the running costs of robots as much as possible. In general, the optimization approach is known to yield higher-performance results[7].

### Technical Challenges

The MRTA problem is classified as an NP-hard problem[8], and when formulated at the scale required for robot operation planning in plants, it is computationally expensive. In actual plant operations, however, various dynamic factors must be considered, such as additional tasks triggered by equipment alarms or status changes, handover items from the previous shift, and robot failures. Therefore, to achieve autonomous robot operation, it is necessary to develop an operation planning and design method that can respond to changes in the site environment and the addition of new tasks in a timely manner.

One possible way to handle such changes is to collect the status of robot operations at the time a change occurs and redesign the operation plan. However, this approach requires solving the computationally expensive MRTA problem each time a change arises. Since plan modifications must be carried out during actual operation without disrupting robot activities, it is necessary to adopt a computationally lightweight approach for solving the MRTA problem.

| Time | 8:00 | 8:30 | 9:00 | 9:30 | 10:00 | 10:30 | 11:00 | 11:30 | 12:00 | 12:30 | 13:00 | 13:30 | 14:00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Man | Working | | | | | | | | | | | | |
| Robot | Operating | | Charging | | | | Operating | | Charging | | | | |

**Figure 2** Envisaged operating status of humans and robots within a shift

**Table 3** Approaches to MRTA problems

| Approach | Overview |
|---|---|
| Optimization | • Formulates operational constraints, required task functions, and robot specification constraints, and derives appropriate assignments based on an objective function describing operation policies<br>• Tends to have high computational cost |
| Market | • Evaluates the execution cost of each robot for each task and sequentially determines the best assignment pattern<br>• Tends to have lower computational cost compared with optimization methods<br>• Requires an evaluation method that reflects robot operation policies and priorities |

## PROPOSED METHOD

In this study, we propose the computational algorithm shown in Figure 3 to reduce the computational cost[9]. In conventional MRTA problems, task assignment to robots and motion path planning interact with each other, which leads to a very large computational scale. To reduce the computational cost, we decoupled robot task assignment and path planning and solved them separately. In addition, we divided the plant into several areas in advance and performed task assignment calculations for each area in order to further reduce the computational cost. Since robots are shared resources across areas, we introduce a higher-level robot-to-area allocation to maintain consistency among area-level task assignment calculations, thereby enabling the design of a feasible overall plan. The details of this algorithm are described below.

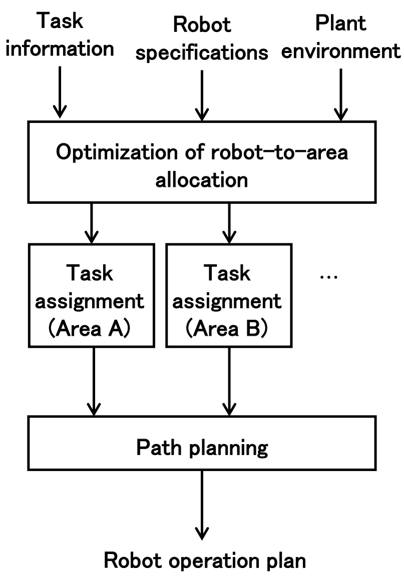


**Figure 3** Operation planning computational algorithm

### Problem Formulation

First, we describe the formulation of the robot operation design problem. Each task assigned to a robot is defined by [task location, required function, specified execution time]. The task location is defined so that it corresponds to the travel environment of the plant described below. Time is represented in discrete steps, with a step width of 30 minutes. For tasks with specified execution times, execution is specified at the corresponding time-step index.

The travel environment of the plant is represented as a graph, as shown in Figure 4, consisting of travel paths, task execution locations, and charging locations for each robot. Task execution locations and charging locations are represented as nodes, and edges are created between nodes when movement between them is possible, thereby representing travel paths. The travel distance between nodes is expressed by assigning weights to edges. The plant is divided in advance into arbitrary areas, and the area information is assigned to each node.

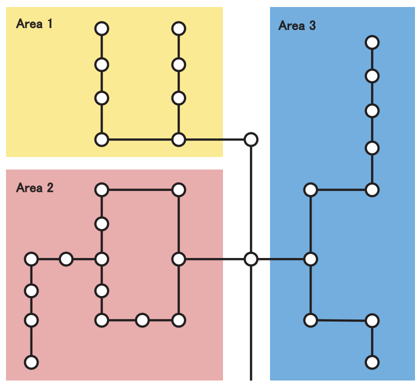


**Figure 4** Example graph representation of the plant environment

### Optimization of Robot-to-Area Allocation

Based on the problem formulation described in the section "Problem Formulation," an optimization calculation for allocating robots to areas was performed. In the area allocation, robots must be assigned so that all tasks within each area can be executed while satisfying temporal constraints. To this end, information was aggregated by area and required function, and the number of tasks that can be started at each time step and the number of tasks that must have been completed by each time step were defined as constraints. In addition, the number of tasks that must be completed within any given time interval was also included as a constraint.

The number of tasks that a robot can execute per time step was determined based on the location of its charging station and its distance to each area. The subsequent task assignment optimization within each area is made feasible by allocating robots in a manner that satisfies the above constraints. Furthermore, robots that were not assigned to any area were scheduled to remain at their charging locations for charging, thereby incorporating battery constraints into the formulation.

**Assignment of Tasks to Robots within Each Area**

Based on the results presented in the section "Optimization of Robot-to-Area Allocation," task assignment calculations were performed to assign the tasks within each area to the robots allocated to each area. As described in the section "Problem Formulation," we formulated the required functions for each task and any specified execution times as constraints, and we performed an optimization calculation to determine the task assignments.

**Robot Path Planning**

Path planning was performed for each robot at each time step. According to the area allocation results described in the section "Optimization of Robot-to-Area Allocation," we focused on one area at a time when designing the path. Based on the allocation status of the preceding and subsequent time steps, we determined the start point within the area under consideration as the location closest to the area assigned in the previous step, and the end point as the location closest to the area assigned in the next step. The execution locations of the tasks assigned in the section "Assignment of Tasks to Robots within Each Area" were then defined as intermediate waypoints. In this way, the problem was formulated as a path planning problem that requires traveling from the start point to the end point while passing through all intermediate waypoints.

This path planning problem is also NP-hard, and obtaining an exact solution is not practical in terms of computational cost. In this work, we performed path planning using an approximation algorithm, namely, the cheapest insertion method, together with Dijkstra's algorithm for finding the shortest path between any two specified nodes in a graph[10]. The cheapest insertion method is an algorithm that repeatedly inserts waypoints into a robot's route, considering the insertion order so that the increase in path length is minimized, until all waypoints have been inserted. Using Dijkstra's algorithm, we compute the actual path resulting from each candidate waypoint insertion and calculate the corresponding increase in path length. By evaluating all candidate waypoints-position pairs based on this increase, we perform path planning using the cheapest insertion method.

## SIMULATOR VERIFICATION

To verify the effectiveness of the computational flow presented in the previous section, we conducted a simulator-based verification experiment. For the optimization calculations in the verification, we used Gurobi Optimizer (Gurobi Optimization, LLC) as the optimization solver. The computations were executed on an Intel (R) Xeon (R) Gold 5317 CPU.

**Verification Setup**

Figure 5 shows a virtual field simulating a plant composed of eight areas and 511 tasks. The functions required to execute tasks comprised five types: {camera, thermal camera, microphone, camera + arm, microphone + arm}. The robot specifications and configurations were defined as shown in Table 4. Using these robots, calculations to design an operation plan that completes all tasks within 8 h were performed, and the required computation time was evaluated. In the area allocation, the objective function was designed with the policy of early execution of tasks and avoiding the assignment of robots to distant areas. In the task assignment within each area, the objective function was defined with the policy of early execution of tasks.

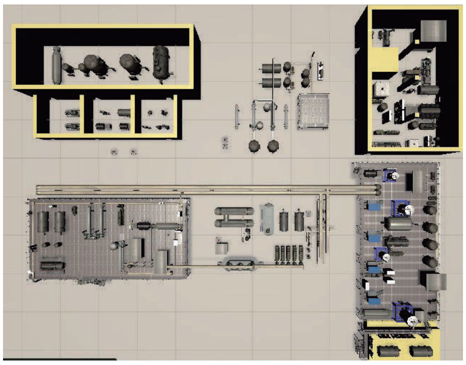

**Figure 5** Simulated environment for the simulator

**Table 4** Robot specifications and configuration

| | Drone | Quadruped robot | Tracked robots |
|---|---|---|---|
| Equipped functions | • Camera | • Camera<br>• Microphone<br>• Arm | • Camera<br>• Thermal camera<br>• Microphone<br>• Arm |
| Charging time required [min] | 30 | 120 | 60 |
| Continuous operating time [min] | 90 | 180 | 120 |
| Number of deployed units [no.] | 4 | 3 | 3 |

Since the computational methods implemented in the optimization solver are dependent on initial values, differences in the initial settings can lead to variations in both the computed solutions and the required time for optimization. In this verification, we evaluated the method using five different random seeds to generate the initial settings, thereby varying the initial conditions.

**Computational Results**

Figure 6 shows an example of the number of tasks executed in each area under the robot operation plan designed in this verification. The cumulative number of completed tasks in each area is shown by the green bars, while the time constraints on task execution in each area are represented by blue and red lines. Some tasks have specified execution time steps. Therefore, even if a robot is assigned to such a task, it cannot be executed until the designated time step is reached.

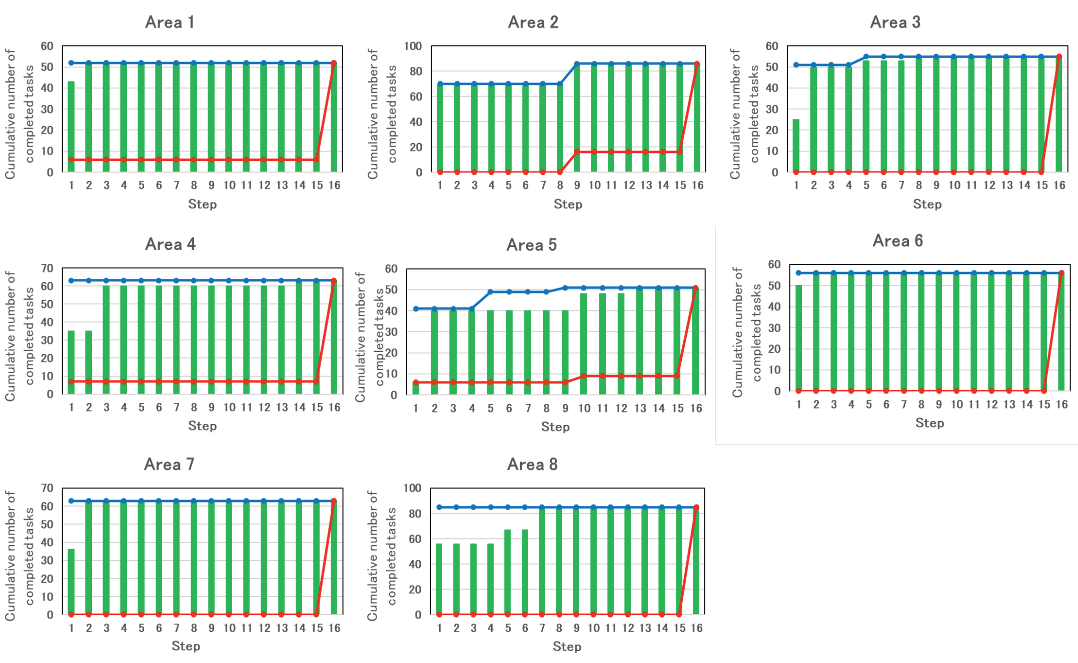


**Figure 6** Example of task execution progress in each area according to the designed plan

The blue line represents the transition in the number of tasks that are executable at each time step. In addition, all tasks must be completed within the shift or within their specified time windows. The red line represents the required cumulative number of completed tasks. The constraints are satisfied if the number of executed tasks in the plan for each area is no greater than the executable number (below the blue line) and no less than the required completion number (above the red line). As shown in Figure 6, the number of executed tasks in the generated plan satisfies the constraints in all areas. These results confirm that the computational algorithm proposed here is capable of generating feasible plans.

Table 5 presents the required computation time of the planning algorithm developed in this paper, as well as the total travel distance of each robot in the designed operation plan. The response time for the problem scale considered in this study was approximately 1 min, which was determined to be within a practical range.

The travel distance varied depending on the random seed used. This is considered to result from differences in the area allocation and task assignment outcomes under different seeds, leading to differences in the tasks assigned to each robot and the resulting travel routes. These variations in allocation are considered to stem from the decoupling of task assignment and path planning in particular. In conventional methods, task assignment is performed while taking into account path planning including the movement required to execute each task. In contrast, in the proposed algorithm, these problems are separated. As a result, differences in task execution locations within an area are not considered during task assignment, and differences in travel distance arise even when task assignment results are evaluated as equivalent in the calculation process.

However, the standard deviation of the total travel distance indicates that this variation is very small. This is considered to result from restricting the operating range per time step to a single area. In addition, in none of the cases were the operational constraints of the robots violated. Therefore, we concluded that the impact of the variation in travel routes is negligible and does not affect practical operation.

**Table 5** Computational results

| | Computation time [s] | Total travel distance [km] |
|---|---|---|
| Run 1 | 55.0 | 5.06 |
| Run 2 | 57.1 | 5.06 |
| Run 3 | 56.6 | 4.97 |
| Run 4 | 55.5 | 5.07 |
| Run 5 | 59.2 | 5.04 |
| Average | 56.7 | 5.04 |
| Standard deviation | 1.5 | 0.03 |

## CONCLUSION

In this paper, we proposed an operation planning flow for achieving autonomous robot operation in process plants and reported on the development of the associated planning technology. Toward realizing industrial autonomy as advocated by Yokogawa, we confirmed that the proposed method can generate feasible plans with a computational cost low enough to respond flexibly to various changing situations.

In recent years, labor shortages and the transfer of technical expertise have become significant issues at plant sites. The introduction of robots is one approach to addressing these challenges, and as robot functionality continues to advance, the scope of robot activities in the field is expected to expand further. As a result, the workload required in order to manage robot operations and the burden on on-site personnel may increase. However, by applying the proposed operation planning technology to achieve autonomous robot operation, the range of operations in which robots can be applied can be expanded while minimizing the burden on plant personnel.

Two key future directions can be identified for achieving autonomous robot operation. The first is 3D modeling of the plant environment. The proposed technology requires accurate physical environmental information. Although drawings from the time of plant construction are available, discrepancies may arise between those drawings and the current state due to equipment updates and additional installations after operations have begun. In such cases, accurate environmental information can be created by constructing a 3D model based on current on-site measurements. The second is robot motion control based on designed motions. Current robots do not have the capability to execute motions designed outside of their predefined functions. Therefore, motion control functions that can reliably execute the actions designed by the proposed technology are required. Yokogawa will continue research and development toward realizing plant operation through robots.

## REFERENCES


(1) Ministry of Economy, Trade and Industry; Ministry of Health, Labour and Welfare; Ministry of Education, Culture, Sports, Science and Technology, 2025 White Paper on Manufacturing Industries, https://www.meti.go.jp/report/whitepaper/mono/2025/pdf/all.pdf, (accessed 2025-7-22, in Japanese)

(2) Tim Shea, "Robotics for the Process Industries Global Market Research 2023-2028," ARC Advisory Group, 2024

(3) Yokogawa Electric Corporation, IA2IA- Industrial Automation to Industrial Autonomy, https://www.yokogawa.com/solutions/featured-topics/ia2ia/, (accessed 2025-7-31, in Japanese)

(4) Boston Dynamics, Inc., SPOT SDK, https://dev.bostondynamics.com/, (accessed 2025-7-31)

(5) Mitsubishi Heavy Industries, Ltd., EX ROVR "ASCENT", the Autonomous, Explosion-proof, Plant Inspection Robot, https://www.mhi.com/business/products-services/energy-environment/nuclear-power-generation/robot-mechatronics/explosion-proof-plant-inspection-robot-ex-rovr, (Accessed 2025-7-31)

(6) Alaa Khamis, Ahmed Hussein, et al., "Multi-robot Task Allocation: A Review of the State-of the-Art," In: Cooperative Robots and Sensor Networks 2015, Springer, 2015, pp. 31-51

(7) Mohamed Badreldin, Ahmed Hussein, et al., "A Comparative Study between Optimization and Market-Based Approaches to Multi-Robot Task Allocation," Advances in Artificial Intelligence, Vol. 2013, 256524, 2013

(8) Brian P. Gerkey, Maja J. Mataric, "A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems," The International Journal of Robotics Research, Vol. 23, No. 9, 2004, pp. 939-954

(9) Yuya Okada, Hiroki Sugawara, et al., "Feasibility-aware hierarchical task assignment for equipment inspection robots with heterogeneous work efficiency," Advanced Robotics, Vol. 39, No. 2, 2025, pp. 114-126

(10) Masao Fukushima, Introduction to Mathematical Programming, New Edition, Asakura Publishing, 2011 (in Japanese)