The Development of a Low Code-based Planning Interface System for Coordinated Work of Multiple Construction Machines

Sheng Lian[1], Fumiya Matsushita[1], and Ryosuke Yajima[2]

[1]Civil Technology Division, Shimizu Corporation, Japan
[2]School of Engineering, University of Tokyo, Japan

s_lian@shimz.co.jp, f_matsushita@shimz.co.jp, yajima@i-con.t.u-tokyo.ac.jp

Abstract -

The construction industry faces labor shortages due to an aging workforce, driving the need for construction automation to boost productivity. Despite recent advances in construction robotics in Japan, the complexity of construction tasks requires coordinated operations among multiple robots. This necessitates not only advanced automation but also a sophisticated planning system to align workflows with construction plans. This study introduces a low-code Planning Interface System (PIS) using the Node-RED platform, enabling civil engineers to design and manage robot workflows without extensive programming expertise. The PIS translates construction plans into executable operational logic, leveraging Robot Operating System (ROS) middleware and the *rclnodejs* package for seamless communication. A prototype demonstrates the ability of the PIS to coordinate an excavator and dump truck in a simulated environment, performing tasks such as navigation, excavation, and dumping through ROS topics and actions. This study highlights the potential of the PIS to facilitate coordinated operations between multiple construction machines, paving the way for more autonomous and streamlined construction workflows.

Keywords -

ROS; Construction automation; Multi-robot coordination; Low-code platform

1 Introduction

The traditional construction industry has long been characterized by its labor-intensive nature, where the majority of construction activities heavily rely on human labor and manual operations [1]. However, the industry is now facing a significant labor shortage due to an aging workforce and difficulties in attracting younger workers [2]. To address this issue, the Ministry of Land, Infrastructure, Transport, and Tourism (MLIT) in Japan launched the i-Construction initiative in 2016, aiming to fully leverage information and communication technology to enhance efficiency across all stages of the construction process, from survey and design to construction and maintenance [3]. Building on this foundation, i-Construction 2.0, which was announced in April 2024, sets an ambitious goal to achieve automation of construction sites and increase productivity by 30% by 2040, a critical target considering the projected 20% decline in the working-age population by that year [4].

One of the key solutions to achieve site automation is the adoption of automated and autonomous construction machinery. In Japan, both construction machinery manufacturers and general contractors have made significant progress in developing their own construction robots, with a primary focus on earthwork projects. Examples include automated hydraulic excavators [5], dump trucks [6], and vibration rollers [7].

However, in actual construction sites, the complexity of tasks often necessitates the coordinated operation of multiple construction machines, as no single robot can handle the entire workload independently. Effective collaboration between various construction machines is therefore essential for smooth project delivery. Achieving this requires not only advanced automation for individual machinery, but also a robust planning system that is capable of designing and managing workflows in alignment with the construction plan.

This study addresses the need by focusing on the integration of construction planning with multi-robot coordination, aiming to provide civil engineers with an intuitive interface to efficiently plan and execute coordinated operations between construction machinery.

2 Research background

2.1 The Role of Construction Planning

Construction planning involves organizing and coordinating all activities and resources needed to complete a project effectively, on time, and within budget. A well-structured plan serves as the foundation for successful project delivery, considering constraints such as contract requirements, site conditions, project duration, and financial limitations [8]. Several key decisions are made dur-

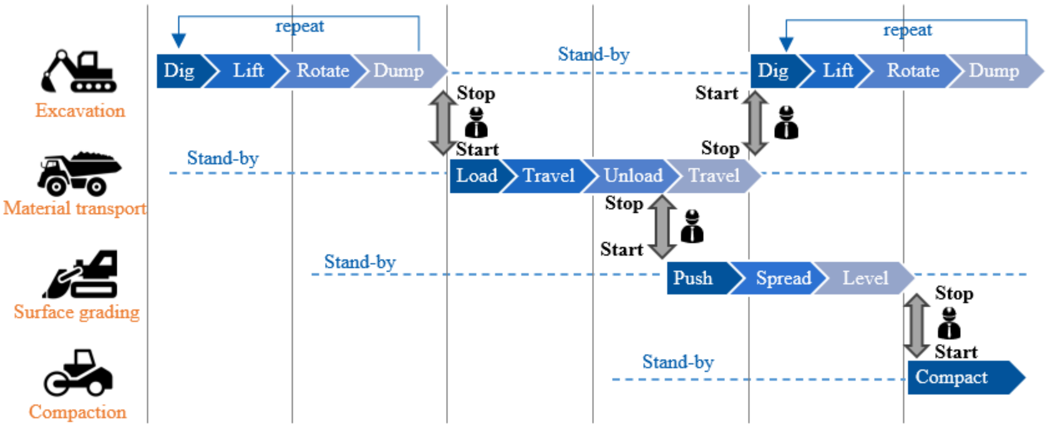

Figure 1. Conventional coordination of robot operations involving human intervention

ing this phase, including identifying specific construction tasks that need to be performed, determining the optimal number of construction machines required, and devising strategies for their arrangement and operation timings on-site. While construction robots have advanced in performing automated tasks such as excavation and material transportation, their operations often remain isolated rather than as part of an integrated construction workflow [9]. A construction plan, however, provides a framework for coordinating these robots to achieve more efficient task execution. By aligning construction operations with the overarching construction plan, isolated automated construction tasks can be transformed into a cohesive and collaborative workflow. This integration not only optimizes task execution but also facilitates smoother and more efficient project delivery.

2.2 Challenges in Multi-Robot Coordination

However, coordinating multiple robots poses significant challenges, as individual machines often lack awareness of others' positions and progress, making it difficult to determine their optimal operation timings. This limitation necessitates human intervention to monitor the overall status of construction and thus decide when to initiate or halt operations. As is demonstrated in Figure 1, in earthworks projects, construction machines such as excavators execute pre-programmed construction works (e.g., digging, lifting, dumping), but human intervention is still required to send "start" or "stop" signals for cooperative operation [10]. To achieve fully autonomous construction, robots must be able to exchange information and independently determine their operation timings. Existing methods often require complex code modifications involving conditionals, loops, and operators, demanding advanced programming expertise that civil engineers may not possess [11]. Therefore, there is a pressing need for an intuitive, accessible solution that allows engineers to define operational logic in a user-friendly manner, translating abstract ideas of collaborative robot operations into concrete, autonomous construction processes.

2.3 Objectives of This Study

The objective of this study is therefore to develop and prototype a Planning Interface System (PIS) that bridges the gap between construction planning and coordination of automated construction robots. The proposed system will allow civil engineers to translate construction plans into intuitive operational logic without requiring extensive programming skills. In addition, the PIS will provide comprehensive data visualization tools to enable civil engineers to monitor the status of construction activities and robot operations in real-time. Ultimately, this study aims to bring together all isolated automated construction tasks under a collaborative framework that aligns with the overarching construction plan, as well as to facilitate construction management process through real-time data visualization of the operational status of the site.

3 Specifications for PIS

3.1 System Requirements

The Public Works Research Institute of Japan has been advancing the Open Platform for Earthwork with Robotics and Autonomy (OPERA), an initiative designed to promote the development and social adoption of autonomous construction machinery [12]. OPERA achieves this by standardizing control signals for construction machinery, leveraging the widely adopted Robot Operating System (ROS) middleware to manage and transmit these signals. In light of the increasing trend among construction robots in Japan to adopt ROS-based control signals, it is essential that the proposed PIS is fully compatible with the ROS framework to ensure seamless interoperability and integration. The following is a list of requirements that the proposed PIS should meet -

1. *Low-Code Visual Programming Interface:* The PIS should provide a low-code, flow-based visual programming environment to enable intuitive configuration of operational logic reflective of construction plans.
2. *ROS Integration:* The PIS must ensure compatibility with ROS middleware, enabling seamless integration with existing ROS-based robotic systems. This functionality must support the exchange of control commands and feedback data between the PIS and construction robots on-site.
3. *Data Visualization:* The PIS should provide comprehensive data visualization tools to help users monitor the status of construction activities and robot operations. Real-time dashboards and reports can assist in making informed decisions and optimizing workflows for construction management purpose.
4. *Scalability and Flexibility:* The system must be scalable to handle projects of varying sizes and complexities, while also offering flexibility to accommodate different types and manufacturers of construction equipment. It should allow users to control construction equipment in various combinations to suit specific tasks, ensuring adaptability to diverse workflows and site conditions.

3.2 Overall System Architecture

Figure 2 illustrates the overall system architecture, which comprises a construction planning system for defining detailed construction tasks, robot selection and operational instructions. This system interfaces with a centralized PIS that monitors overall vehicle statuses and issues control commands accordingly. Moreover, a group of automated construction robots are deployed at the construction site (i.e., the "edge"), where the operational data is generated and processed locally during task execution. Civil engineers utilize the PIS to translate construction plan details into operational logic that governs the collaborative behaviors of robots. By continuously acquiring and processing real-time data from edge robots regarding vehicle positions, conditions, and work progress, the PIS is able to gain a comprehensive understanding of the operational status of the site. This real-time insight allows the PIS to determine the optimal timings for each robot's tasks based on predefined logic. As a result, the PIS outputs a series of operational commands to individual robots, specifying task activation or deactivation, along with key operational parameters such as velocity and path trajectory for each robot. The output commands ensure a high level of coordination among various construction robots, aligning their operations with the overall construction plan.

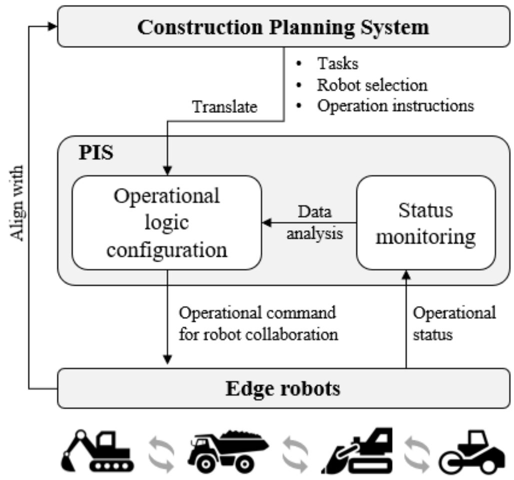

Figure 2. Overall system architecture

4 Prototype development

4.1 Platform Selection

Based on the system requirement in Section 3.1, Node-RED [13], an open-source, web-based visual programming tool, was selected as the development platform for the PIS to plan and control coordinated operation of construction robots. Its low-code, drag-and-drop interface enables civil engineers to design and configure operational logic without requiring extensive programming expertise. This allows civil engineers to focus on defining workflows that align with construction plans, efficiently translating project requirements into executable robotic behaviors.

Node-RED also excels in data visualization, providing a wide array of nodes to acquire, process, and display operational data. A node is an executable process that performs a specific computational task. By wiring up relevant nodes in Node-RED, users can quickly prototype dashboards featuring a variety of visual elements to monitor edge robots effectively and derive actionable insights into their operations. Additionally, the integration of Node-RED with ROS facilitated by the *rclnodejs* package [14] ensures seamless communication with ROS components such as topics, services, and actions. This capability allows users to visually develop and implement ROS-based control algorithms for multi-robot coordination. In this study, the proposed PIS leverages the Node-RED platform and an *rclnodejs*-based plugin software developed by EduArt Robotik [15] to directly interface with ROS.

4.2 Scenario Selection

The prototyping of the proposed PIS in this study focuses on automating the coordination of construction machines within earthworks projects, primarily due to the repetitive yet predictable nature of the workflow associated with earthwork construction. Earthwork operations, such as excavation, material transportation, and soil compaction, often require complex coordination among various construction machines, making them an ideal testing ground for collaborative automation. Additionally, the wide, open spaces with minimal external interference typical of earthwork sites further simplify the implementation of autonomous technologies, providing a controlled environment for testing and validation.

The workflow for earthworks construction can be broken down into a series of tasks, with the prototype in this study focusing on coordinating operations between an excavator and a dump truck using Node-RED and the relevant ROS client library. Figure 3 gives an example of the construction plan, specifying the detailed construction tasks involving the two machines, along with the timing as for each task to take place. To visualize and test their cooperative behavior in a controlled and risk-free environment, a Gazebo-based simulator was chosen for its simplicity and seamless integration with ROS. This ensures that the system's operational logic and coordination strategies are thoroughly validated prior to practical implementation in real-world construction scenarios.

	Scenario	Task	Trigger for the task
Task 1		**Dump truck**: Navigate to the excavation site **Excavator**: Stand-by	When the target position of dump truck is specified
Task 2		**Dump truck**: Stand-by **Excavator**: Digging, lifting, rotating, dumping (repeat)	When dump truck reaches its stand-by position at the excavation site
Task 3		**Dump truck**: Navigate to the dump site **Excavator**: Stand-by	When soil quantity reaches the weight limit of dump truck
Task 4		**Dump truck**: Soil dumping **Excavator**: Stand-by	When dump truck reaches the dump site

Figure 3. Example of a construction plan with detailed construction tasks and corresponding operation timings within an earthwork project

4.3 Prototype Development and Result Discussion

The prototype development thus involves translating the details of the construction plan above into equivalent flow-based operational logic that governs the work coordination of robots. Figure 4 shows the detailed communication flow between the Node-RED-based PIS and the edge robots. On the edge side, the robots execute high-level ROS control programs that govern their individual operations, such as navigation and excavation. Meanwhile, by incorporating the construction plan requirements into the operational logic, the PIS ensures the precise timing and conditions under which the edge control programs are executed, ensuring seamless collaboration among the machines. This operational logic is divided into three distinct parts: Part A, Part B, and Part C, each corresponding to a specific Node-RED flow, as illustrated in Figure 5. These parts collectively enable the PIS to orchestrate the coordination of tasks through intuitive flow-based programming. The following sections provide a detailed discussion of how the PIS facilitates task coordination.

4.3.1 Task 1 (Navigate to the excavation site)

Task 1 involves navigating the dump truck from its current position to reach a specified target position and orientation at an excavation site. This process implements path tracking control using the Pure Pursuit algorithm [16], which continuously calculates and adjusts the steering direction of the dump truck as it moves toward its target. This path tracking control is implemented using an action server–client communication framework in ROS. A ROS node on the edge robot initializes the action server that executes the Pure Pursuit algorithm and waits for goal requests from the Action Client a-1 in the PIS. The Action Client a-1 then sends the navigation goal to the running action server that triggers the navigation. The Node-RED configuration for this action client is detailed in Part A of Figure 5.

4.3.2 Task 2 (Excavation)

Task 2 automates the excavation process through a ROS node that integrates a subscriber and a publisher. The subscriber on the edge robot continuously monitors the **/flag_ex** topic to determine whether to proceed with excavation. Upon successful completion of Task 1 (the dump truck reaches its stand-by position at the excavation site), the action server sends the navigation result back to the PIS. This triggers the Publisher b-1 in the PIS to set the **/flag_ex** topic to "true", initiating the excavation process. The publisher as part of the excavation control program for Task 2 then sends joint position commands to control the excavator's movements. Simultaneously, the **/num_cycles** topic tracks the number of excavation cycles performed and thereby providing real-time feedback on the progress of the operation. The relevant Node-RED flow for coordinating Task 2 is shown in Part B of Figure 5.

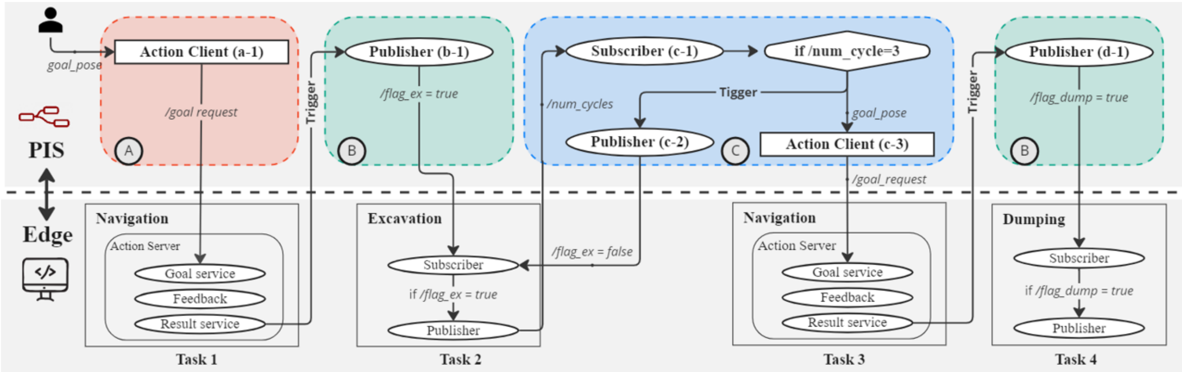

Figure 4. Overall communication flow between the PIS and edge side

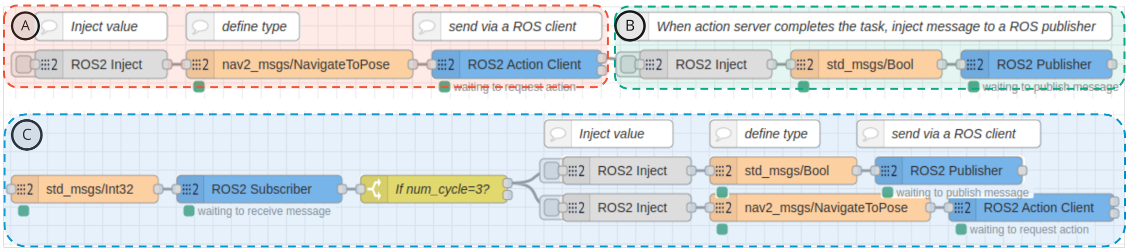

Figure 5. Node-RED flow configuration for work coordination

4.3.3 Task 3 (Navigate to the dump site)

Task 3, similar to Task 1, but this time involves navigating the dump truck to a designated dump site. In practice, excavation halts when the truck reaches its weight limit, whereas in the case of a simulated environment, this condition is approximated by counting the number of excavation cycles. The PIS sets up a Subscriber (c-1) to listen on the **/num_cycles** topic. After three cycles of excavation, the PIS triggers a Publisher (c-2) to set the **/flag_ex** topic to "false", effectively pausing the excavation work from Task 2. At the same time, the PIS sends a goal request via the Action Client c-3 to initiate the dump truck's navigation towards the dump site as per Task 3. This operational logic ensures that the dump truck starts moving only after the required amount of soil has been excavated, maintaining a smooth transition between the cessation of excavation and the initiation of material transportation. The detailed flow configuration for Task 3 is illustrated in Part C of Figure 5.

4.3.4 Task 4 (Soil dumping)

Task 4 oversees the soil dumping operation, managed by a ROS node to control the movement of the dump truck's vessel. On the edge robot side, the ROS node first starts a subscriber to monitor the **/flag_dump** topic to determine the timing for soil dumping, which then instructs the publisher to send joint control commands to the vessel actuator. Upon receiving the action results from Task 3, meaning when the dump truck has reached its designated position at the dump site, the PIS triggers the Publisher d-1 to set the **/flag_dump** topic to 'true' and hence initiates the dumping process. The setup for Publisher d-1, similar to Publisher b-1 and d-1, is detailed in Part B of Figure 5.

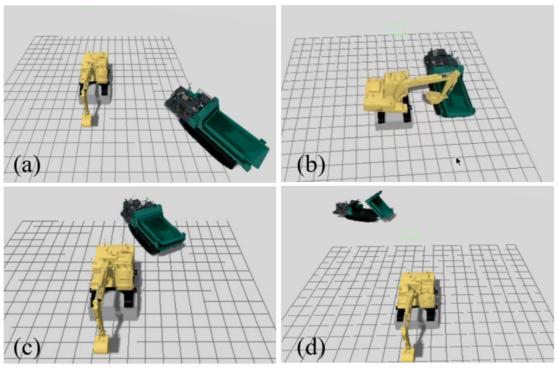

Figure 6. A snapshot of the cooperative behaviors of machines in Gazebo simulation: (a) Navigate to the excavation site; (b) Excavation and loading; (c) Navigate to the dump site; (d) Soil dumping

4.3.5 Validation of results in Gazebo simulation

Figure 6 provides a snapshot of the resulted machine behaviors in the simulation with respect to different tasks,

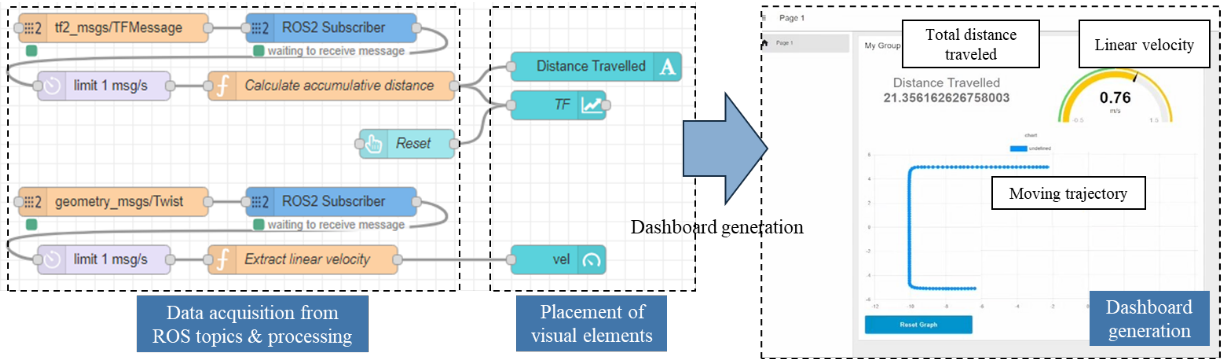

Figure 7. Sample Node-RED flow and its output dashboard for monitoring vehicle velocity, travel distance and trajectory

illustrating how the PIS orchestrates the excavator and dump truck to perform their respective tasks in a coordinated manner. The entire workflow from Task 1 (navigation to the excavation site) to Task 4 (soil dumping) is executed autonomously with no human intervention, with the PIS determining the optimal operation timings for each task based on predefined logic reflective of the construction plan. The simulation result demonstrates the potential of the PIS to design intuitive, flow-based operational logic for coordinating multiple construction tasks seamlessly.

4.3.6 Dashboard development

In order to facilitate the construction management process, a Node-RED dashboard was developed to provide real-time monitoring of the operational status of the construction machines throughout the task execution. By subscribing to relevant ROS topics, the system collects and processes data to generate insightful visualizations, offering a clear overview of machine performance. Figure 7 provides a simple example of the Node-RED flow and its corresponding dashboard, which tracks the linear velocity, travel distance and trajectory of the dump truck. By integrating with additional ROS topics, the dashboard is designed to expand its monitoring capabilities to include more construction management-oriented metrics, such as excavated soil quantity for tracking work progress and machine utilization rates for optimizing resource allocation.

5 Conclusion and future improvement

This study demonstrates the development of a low-code-based PIS leveraging Node-RED platform to integrate construction planning with the coordinated operation of multiple construction robots. The PIS simplifies configuration of operational logic for civil engineers through an intuitive drag-and-drop interface, reducing the need for extensive programming knowledge. By ensuring compatibility with ROS middleware through the *rclnodejs* package, the system enables seamless integration with existing ROS-based robotic systems. Additionally, the PIS provides robust data visualization capabilities, offering real-time insights into the operational status of construction activities and robot performance. The prototype demonstrated the coordination of excavator and dump truck operations in an earthwork scenario using ROS topics and actions. This showcases the PIS's ability to unify isolated automated tasks into a collaborative framework that aligns with the overarching construction plan with minimized human intervention.

Future work will expand the system to include a bulldozer for soil leveling and a vibration roller for compaction to allow for a more completed configuration of work coordination among all machines involved in earthwork operations. This will enable the PIS to handle more complex scenarios and edge cases that account for real-world construction scenarios. The dashboard will also be improved by integrating additional relevant ROS topics to monitor more construction management-oriented metrics for quality and quantity control. Additionally, a more advanced physics engine-based simulation environment will be adopted to accurately model soil properties, material interactions and machine dynamics, ensuring robust testing before field deployment. Finally, the PIS is expected to be deployed on actual construction machines to validate its effectiveness and practical applicability in real-world settings.

References

[1] T. Bock. The future of construction automation: Technological disruption and the upcoming ubiquity of robotics. *Automation in Construction*, 59:113–121, 2015. doi:https://doi.org/10.1016/j.autcon.2015.07.022.

[2] T. Elbashbishy and I. H. El-adaway. Skilled worker shortage across key labor-intensive construction trades in union versus nonunion environments. *Journal of Management in Engineering*, 40(1), 2024. doi:10.1061/JMENEA.MEENG-5649.

[3] Ministry of Land, Infrastructure, Transport, and Tourism. Status of the i-Construction initiatives (in Japanese). On-line: https://www.mlit.go.jp/tec/i-construction/pdf/06.7_kikaku_siryou6.pdf, Accessed: 12/17/2024.

[4] Ministry of Land, Infrastructure, Transport, and Tourism. i-Construction 2.0 - the automation of construction sites (in Japanese). On-line: https://www.mlit.go.jp/tec/constplan/content/001738240.pdf, Accessed: 12/17/2024.

[5] H. Suzuki, G. Yamauchi, D. Endo, and T. Hashimoto. The efforts of the public works research institute to promote the development of autonomous construction technology. *Journal of Control, Measurement, and System Integration*, 61(9):651–655, 2022. doi:10.11499/sicejl.61.651.

[6] T. Akegawa, K. Ohno, S. Kojima, K. Yamada, W. Go, T. Suzuki, S. Kiribayashi, T. Komatsu, N. Miyamoto, T. Suzuki, Y. Shibata, K. Asano, K. Nagatani, and S. Tadokoro. Path planning of the turning back of an autonomous large-scale six-wheeled dump truck for loading/leaving sediment based on backhoe work. In *2022 IEEE/SICE International Symposium on System Integration*, pages 511–518, United States, 2022. doi:10.1109/SII52469.2022.9708879.

[7] K. Kikuchi, K. Nagatani, T. Komatsu, S. Kiribayashi, K. Asano, Y. Shibata, K. Ohno, T. Suzuki, and Y. Hirata. Position estimation using environment-installed laser range finders and traveling control for autonomous surface compression work of vibration roller. *Journal of the Robotics Society of Japan*, 38 (9):872–881, 2020. doi:10.7210/jrsj.38.872.

[8] J. Hinze. *Construction Planning and Scheduling*. Pearson, 2011. ISBN 9780132473989.

[9] S. Lian and F. Matsushita. System design of an integrated controller for coordinated operation of autonomous construction machinery. *Intelligence, Informatics and Infrastructure*, 3(3):10–14, 2022. doi:10.11532/jsceiii.3.3_10.

[10] K. Sano, S. Sato, and R. Nakamura. Demonstration of a cooperative control system for a group of automatic construction machines. *Construction Machinery*, 75(7):46–51, 2023.

[11] F. Matsushita, R. Yajima, K. Ozawa, and K. Nagatani. Development of civil engineering / robotics integrated education program aimed at creating innovation in ict earthwork. *Journal of Japan Society of Civil Engineers (Education)*, 78(1):38–52, 2022. doi:10.2208/jscejeep.78.1_38.

[12] G. Yamauchi, E. Daisuke, H. Suzuki, and T. Hashimoto. Proposal of an open platform for autonomous construction machinery development. In *Proceedings of the 40th International Symposium on Automation and Robotics in Construction*, pages 186–191, India, 2023. doi:10.22260/ISARC2023/0027.

[13] Contributors OpenJS Foundation. Node-RED. On-line: https://nodered.org, Accessed: 12/04/2024.

[14] RobotWebTools. rclnodejs package. On-line: https://github.com/RobotWebTools/rclnodejs, Accessed: 12/10/2024.

[15] EduArt-Robotik. edu-nodered-ros2-plugin. On-line: https://github.com/EduArt-Robotik/edu_nodered_ros2_plugin, Accessed: 12/10/2024.

[16] C. Coulter. *Implementation of the Pure Pursuit Path Tracking Algorithm*. Camegie Mellon University, Pittsburgh, 1992.