Performance-Based Earthmoving Team Organization Algorithm Enabling Task Completion under Changing Conditions

Wakana Endo[1], Shota Chikushi[2], Yuichiro Sueoka[3], Yuki Kato[3], Kaito Kimura[3], Yusuke Ikemoto[4], Ren Komatsu[1], Keiji Nagatani[1], Atsushi Yamashita[5] and Hajime Asama[1]

Abstract— **In this study, we propose a methodology for automating earthmoving tasks using autonomous excavators and dump trucks, with an aim to enhance adaptability in response to environmental changes. Our methodology involves a general architecture that fosters self-organized swarm behavior among the robots. Furthermore, we present an illustrative algorithm that evaluates the performance of a team of robots and adjusts the team composition based on this performance evaluation. This algorithm was implemented in a simulation test in a dynamic environment. The results demonstrate that our methodology enables the coordination of excavators and dump trucks in environmental changes.**

I. INTRODUCTION

The use of automated robots in earthwork tasks, where they must adapt to changing situations, is highly anticipated [1]. In addition, earthmoving operations where machines flexibly respond to unexpected situations are needed both on construction sites and in emergency recovery from natural disasters, highlighting the importance to automate by introducing robots [2]. However, the full implementation of robots in these tasks has not yet been achieved due to the need for multiple robots to cooperate with each other and the dynamic nature of the environment, such as ground conditions, buried obstacles, and robot failures. Therefore, the objective of this research is to develop cooperative behaviors that allow swarms of construction robots to move the soil while adapting to dynamic changes in the environment.

This research focuses on self-organization in multirobot cooperation. It is defined as, “the mechanism or the process that enables a system to change its organization without an explicit external command during its execution time”, and the key features for a system to be self-organized are the realization of decentralized control and dynamic organizational change [3]. As these characteristics are essential for earthmoving work to be robust to environmental changes, it is necessary to develop a robot control method that enables dynamic team organization through decentralized control.

As for studies on the coordination of dump trucks and excavators, Schmidt et al. address the task of loading and transporting soil by an excavator [4]. They proposed a method for accurate dump truck path planning based on binary grid maps and pathfinding using an extended A* search algorithm. While their method has proven useful in guiding dump trucks to their destinations even in the presence of noisy localization, it only considered a pair of robots. Therefore, there is a need for further exploration of larger-scale robot coordination techniques in this area.

There have been several previous studies on distributed control, including one that determines robot teams based on threshold values [5] and another that performs distributed task assignment by voting while sharing task assignment status among robots [6]. However, in our study, the objective is robot coordination at a large-scale construction site. Therefore, it is desirable to share less information among robots, and a distributed decision-making method that can cope with dynamic organizational and environmental changes is necessary. Palmieri et al. proposed a swarm-robot coordination system that selected two tasks, field exploration and hazardous material removal, by stochastic state transitions using the concept of pheromones. However, the task selection weighting must be adjusted for each different environment and is not effective for dynamically changing environments [7].

Regarding adaptation to dynamic environmental changes, Zhang et al. automated a real excavator in a volatile environment using task planning algorithms that combined inverse reinforcement learning, data-driven imitation learning and optimization-based methods [8]. However, it is unclear whether these methods can scale to a large group of robots, such as dump trucks, while keeping computational costs feasible. Notomista et al. presented a task allocation method with multirobot navigation experiments that shared state equations among robots, monitored errors in motion prediction and actual motion, and recalculated task assignments to respond to environmental changes [9]. Moreover, Seraj et al. present a hierarchical framework for coordinating complementary robots in dynamic environments. Their framework demonstrated its efficacy through a wildfire-fighting

*This work is supported by JST [Moonshot Research and Development], Grant Number [JPMJMS2032].

[1]W. Endo, R. Komatsu, K. Nagatani and H. Asama are with the Department of Precision Engineering, Graduate School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan {endo, komatsu, asama}@robot.t.u-tokyo.ac.jp, keiji@ieee.org

[2]S. Chikushi is with the Department of Robotics, Kindai University, 1 Takaya Umenobe, Hiroshima, 739-2116, Japan chikushi@hiro.kindai.ac.jp

[3]Y. Sueoka is with the Department of Mechanical Engineering, Osaka University, 2-1 Yamada-oka, Suita, Osaka, 565-0871, Japan sueoka@mech.eng.osaka-u.ac.jp, {kato.y, kimura.k}@dsc.mech.eng.osaka-u.ac.jp

[4]Y. Ikemoto is with the Department of Mechanical Engineering, Meijo University, 1-501 Shiogamaguchi, Tempaku-ku, Nagoya, Aichi, 468-8502, Japan ikemoto@meijo-u.ac.jp

[5]A. Yamashita is with Department of Human and Engineered Environmental Studies, Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8563, Japan yamashita@robot.t.u-tokyo.ac.jp

case study, uses multi-agent reinforcement learning for area surveillance, and a coordinated control and planning module for action decisions [10]. However, these two methods are not suitable for distributed control, which limits the information shared among robots, because each robot needs state models or detailed state information of other robots for coordination.

As introduced so far, there have been various studies on distributed coordination algorithms for robots and their adaptation to changes in the environment, but it has not yet been possible to achieve both of these goals. Therefore, the purpose of this study is to propose a distributed performance-based coordination algorithm for the self-organization of construction robots. To this end, we first propose an architecture for evaluating robot groups based on an internal model, and more specifically, we define a single team evaluation index called performance. We then construct an algorithm to determine the number of robots required for a team based on its predicted and measured values. The self-organizing effect of this algorithm is verified in a dynamic simulation environment where the velocity of dump trucks changes. The contributions of this paper are as follows:

1) An algorithm architecture that enables self-organization of heterogeneous robot groups has been proposed.
2) For teams of construction robots performing earthmoving work, a single index called performance has been defined. Using this metric, an algorithm has been constructed to incorporate the necessary robots in response to environmental changes and achieve tasks by changing the team's organization in a distributed manner.

II. DISTRIBUTED COORDINATION ALGORITHM BASED ON A PERFORMANCE PREDICTION MODEL

A. Scenario

In this study, we use earthmoving tasks with construction robots as a specific example to test whether self-organizing movements can be achieved with our proposed cooperation architecture. The task involves transporting earth from a loading site to a dumping site by a deadline using an excavator and a dump truck. To enable decentralized cooperation between different types of robots and adapt to environmental changes, we propose dividing the robots into teams and controlling them accordingly.

As shown in Fig. 1, we consider two roles of robots and assume a team consists of a Coordinator and several Cooperators. Here, the coordinator is defined as the robot that can decide the organization of the team, and the excavator plays that role, in this paper. The coordinator is the robot that the coordinator calls to join the team, and in this paper, the dump truck plays the role of the coordinator. It should be noted here that any type of robot, including excavators, can also join the team as Coordinators. There may also be idle excavators and dump trucks, which the Coordinator can add to the team to improve its performance. The Coordinator

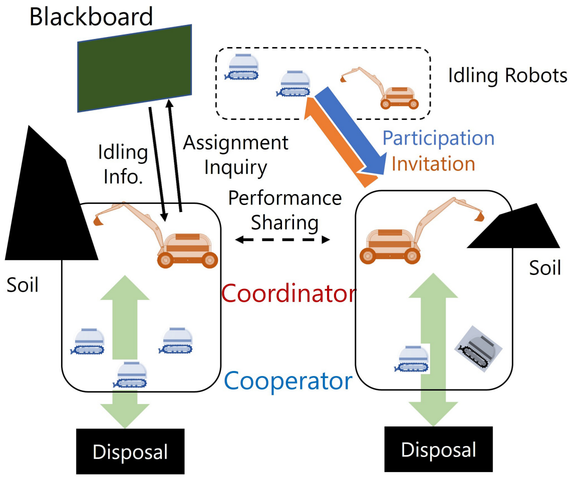

Fig. 1. Schematic of the earthmoving task as a case study for simulation experiments and the proposed team organization algorithm.

communicates with the necessary robots to organize the team and accomplish the task. The Coordinator has access to a virtual blackboard to view information such as each robot's assignment status and the number of idle robots. This allows the Coordinator to make changes to the team organization with minimal communication.

The goal of the task is for a team, consisting of one Coordinator and multiple Cooperators, to secure the necessary number of Cooperators to complete the task. Cooperators are assigned to a team and sometimes transferred to another according to the performance prediction model described below.

In this problem setup, each robot has limited information. Specifically, a Cooperator only observes information from its own sensors, while Coordinators observe information shared by robots in their team. However, to enable cooperation among teams, Coordinators share each team's performance and information on the Coordinator to whom the robot is assigned, which is stored in Blackboard.

B. Self-Organization Architechture

Based on the above scenario, we propose a self-organizing architecture that can respond to changes in the environment, as shown in Fig. 2. This architecture is used by multiple Coordinators to form their respective teams. We believe that a swarm of robots can respond to unexpected changes in the environment by transitioning between centralized and decentralized structures, depending on the situation. For this purpose, we define two behavior modes, Cooperator and Coordinator, and propose the formation of multiple teams (see Figure 1) with an internal hierarchical structure. It is worth noting that while the teams have a centralized hierarchical structure, they cooperate with each other in a decentralized manner. We expect to be able to change the degree of dispersion of the entire flock by changing the number and size of the teams. In this paper, we conduct experiments in which the number of teams is fixed, but the

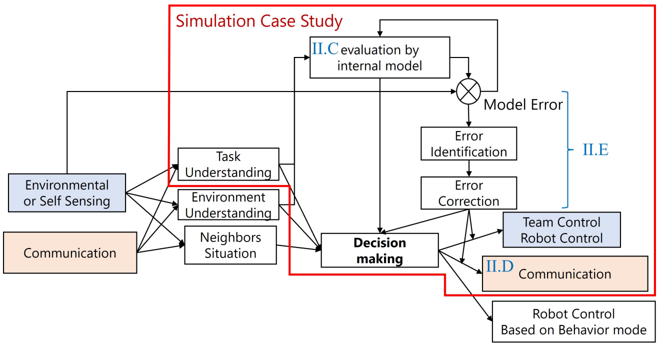

Fig. 2. Schematic of the proposed self-organizing collaborative architecture. Algorithm implementations for the simulation experiment were performed for the red frame part and are explained in the corresponding annotated sections.

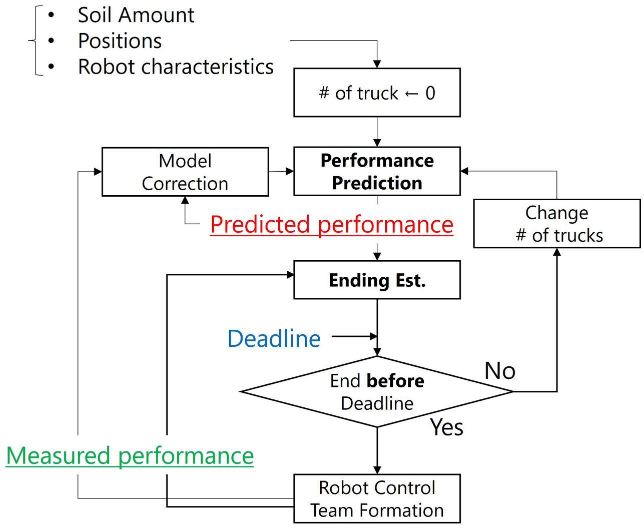

Fig. 3. Flowchart of the implemented team organization algorithm for excavators as Coordinator

size of the teams varies.

The Coordinator uses onboard sensors and communication with other machines to grasp the status of the team and the surrounding environment, evaluates the team's functions based on this information, and controls the team formation to accomplish the task. When the environment changes, the evaluation based on the internal model may become inaccurate; therefore, it is important to correct errors by modifying the internal model as necessary.

C. Performance Prediction Model

Here, we consider a performance-based algorithm as a method to evaluate a group of robots using an internal model. We define team performance as the weight of soil that a team can carry per unit of time, and construct a model that calculates performance-based on pre-defined parameters. Then, to achieve our goal of cooperative behavior that can adapt to changes in the environment, we calculate the appropriate number of Cooperators to accomplish the task using two indices: Predicted Performance and Real Performance. A flowchart of the team formation algorithm based on this model is shown in Fig. 3. Considering two situations: one in which the number of dump trucks is small and the excavators have to wait for the dump trucks to return, and the other in which the number of dump trucks is large and there is a wait for the dump trucks to load, the Predicted Performance can be expressed as follows:

$$r^i_{\text{pred}} = \begin{cases} \dfrac{\bar{w}_i}{t_{ld}} & \text{if } n_i \geq \dfrac{(\frac{2D_i}{\bar{v}_i} + t_{ld} + t_{dp})}{n_i t_{ld}} \\ \dfrac{n\bar{w}_i}{\frac{2D_i}{\bar{v}_i} + t_{ld} + t_{dp}} & \text{otherwise.} \end{cases} \tag{1}$$

Each variable is defined in Table I. This performance can be calculated using the performance of the construction robot and the parameters given by the task, but each parameter must be updated when the environment changes. As described below, this parameter update is the role of the Coordinator and is assumed to be performed by the excavator in this paper.

TABLE I
DEFINITION OF VARIABLES

Symbol	Description
n_i	# of Cooperators in team i
i	ID of team
k	ID of dump truck
$\mathbb{S}_i$	Set of dump trucks belong to the team i
$\bar{w}_i$	Initial average loading amount [kg]
D_i	Distance between load / dump sites [m]
$\bar{v}_i$	Velocity of Cooperator [m/s]
T_{DL}	Deadline [s]
t	time [s]
t_{ld}	Average loading time [s]
t_{dp}	Average dumping time [s]
$w^k_{\text{last}}(t)$	Weight of loaded soil on dump truck k
t^k_{last}	Time of the last loading of dump truck k
$r^i_{\text{pred}}, r^i_{\text{real}}$	Predicted/Real Performance [kg/s]

Real Performance, on the other hand, is performance calculated using the amount of soil actually transported by the robot. It is assumed that the excavators and dump trucks are each equipped with appropriate sensors to measure the amount of material excavated and the amount of material loaded. Real Performance is calculated using the most recently dug soil volumes by all dump trucks in the team and the time taken to haul them, as follows:

$$r^i_{\text{real}} = \frac{1}{t - \min t^k_{\text{last}}} \sum_{k \in \mathbb{S}_i} w^k_{\text{load}}(t^k_{\text{last}}). \tag{2}$$

Because Real Performance requires the actual amount of soil transported, it cannot be measured until the dump truck has made one round trip and returned. Therefore, only the Predicted Performance can be used to calculate the end time of the work, but the Real Performance can be used after the first round trip is made. If the team composition changes, it is necessary to use the Predicted Performance again until the Real Performance is known.

The proposed method uses the Predicted Performance to

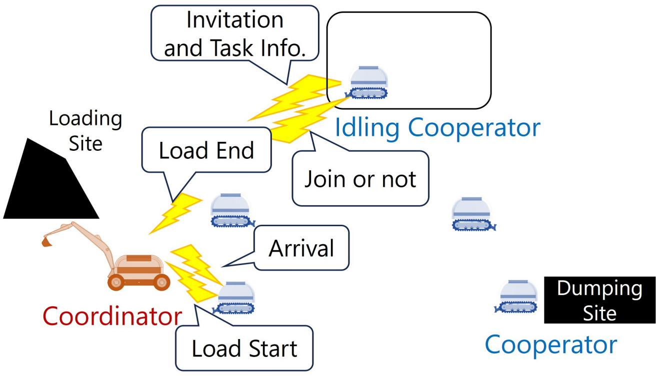

Fig. 4. Schematic of the communications between the Coordinator and Cooperator

calculate the time required to complete the task when the amount of soil remaining in the loading site to be excavated by the team is set to $X(t)$, and compares it with the delivery date to determine whether the team composition needs to be changed or not. The conditions that necessitate a change in team composition are as follows:

$$\begin{cases} t + \dfrac{X(t)}{r_{\text{real}}} > T_{DL} & \text{if } r_{\text{real}} \text{ available} \\ t + \dfrac{X(t)}{r_{\text{pred}}} > T_{DL} & \text{otherwise.} \end{cases} \tag{3}$$

In the proposed method, the number of Cooperators that can accomplish the task by the deadline is calculated for each team based on the above two performances and the expected time of work completion, and then the Coordinator organizes the team.

D. Local communication between Robots

As shown in Fig. 1, the proposed cooperative algorithm organizes multiple construction robot teams when there are multiple loading sites, and the Coordinator shares information among the teams. In the simulations,a virtual blackboard was used to identify idle dump trucks and to invite them to different teams. The blackboard can only be accessed by the Coordinator, who records the team assignments of all available dump trucks in the field. When adding a dump truck to a team, the Coordinator first retrieves a list of idle dump trucks from the blackboard, determines which one to add, and updates the assignment information on the blackboard. In this paper, we only manage IDs because all dump trucks have homogeneous performance. However, when dealing with a group of heterogeneous robots with different payloads or other performance characteristics, the performance of each dump truck as well as its assignment information should be written on the blackboard. This allows the Coordinator to select dump trucks according to the performance required by each team.

Regarding the communication between the Coordinator and the Cooperator, Fig. 4 shows all implemented interactions for the cooperative earth moving. Following the assumption that the robots do not have any global information of the scenario, inter-robot communications are limited to short distances. We also assume that the Coordinators can detect obstacles ahead, which eliminates the need for communication between them to avoid collisions.

E. Error Correction and Team Formation Change Using Performance

As shown in Eq. (1), the model used to calculate the Predicted Performance of the Coordinator is based on changeable parameters, which are the average velocity of the dump trucks, the amount of soil loaded and the average loading/dumping time. Since these parameters may change as the work environment changes, the method used in this study is to check individual parameters and update the parameters that have changed beyond the threshold when the Real Performance exceeds the threshold value (set at 10 percent of the previous value). Among the parameters used in Eq. (1), the amount of soil loaded and the loading time are numbers that can be measured by the Coordinator, so the Coordinator can determine changes by himself. The average velocity of the dump truck can be calculated from the round-trip time of the dump truck, t_{circ}, under the assumption that the soil discharge time and hauling distance remain unchanged.

$$\bar{v}_i = \frac{2D_i}{t_{\text{circ}} - t_{ld} - t_{dp}}. \tag{4}$$

When Real Performance changes beyond a threshold value, the actual values for each parameter are checked to determine which parameters should be overridden.

Here, in the performance prediction model, $\bar{w}, t_{ld}$ are the parameters affected by the excavator loading operation and v is the parameter affected by the travel of a dump truck, and if the Real Performance changes according to changes in the environment and the work end time is later than the delivery date, which robot should be added by the parameter to be updated is the one that is updated. If the Real Performance changes according to changes in the environment and the job completion time is later than the deadline, the updated parameters can determine which robot should be added.

III. SIMULATION EXPERIMENT

To validate our proposed dynamic collaborative architecture and performance-based team formation algorithm, we conducted simulations using two different platforms: ROS 2 and Vortex Studio. Vortex Studio is a high-fidelity simulation platform from CM Labs Simulation Inc, which can simulate the operation of mechanical systems and earthworks in real-time [11]. It was used to model the physical components of the robot, such as its sensors and actuators, as well as its work environment. On the other hand, ROS 2 was used to simulate robot path planning, excavation, loading, and other control functions. These simulators were run on separate computers as shown in Table II and communicated with each other using UDP (User Datagram Protocol) communication.

Each team had only one excavator as a Coordinator. The number of dump trucks assigned to a team could be

TABLE II
COMPUTER SPECIFICATIONS

	Robot Control	Physics Simulation
OS	Ubuntu 20.04	Windows 11
CPU	Intel Core i7-10710U	Intel Core i7-11700K
GPU	Intel UHD Graphics	NVIDIA GeForce RTX 3070
Memory	16GB (16GB×1)	32GB (16GB×2)
Application	ROS 2 Foxy	Vortex Studio 2022.10

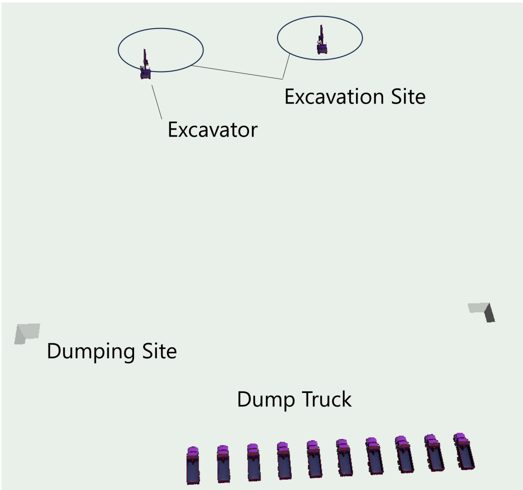

Fig. 5. Simulated environment for two teams of construction robots. At the beginning of the simulation, all robots are aligned in the waiting area on the lower side of the figure.

changed based on the estimated task completion time. For this purpose, an Idling Cooperator that did not belong to any team was available in a waiting area.

As shown in Fig. 5, we assigned earthmoving tasks to two loading sites and two sand dumping sites. As an initial condition, there were $12\,000\,\mathrm{kg}$ and $8000\,\mathrm{kg}$ of soil in the loading sites. We initially placed 10 dump trucks and two excavators in the waiting area. We started the scenario by giving the following task information to the excavators, and they first headed for the loading sites to assess the situation.

- Coordinates of the loading site for each excavator
- Coordinates of the two dumping sites
- Number of available dump trucks in the field
- Deadline

For simplicity, only one excavation per load is assumed in the simulations in this paper. It is also assumed that the two excavators and the 10 dump trucks all have the same performance, with the velocity of the dump trucks for each team. However, it should be noted that the excavation volume of the excavators varied each time due to the characteristics of the earthwork simulator.

The actual travel velocity of the Cooperator was set slower than the specification to simulate environmental conditions where the Predicted Performance is not available. Other specific parameters are shown in Table III. The initial average loading amount, the distance between the loading site and dump site, and the average velocity of the Cooperator are identical between the two teams.

TABLE III
NUMERICAL VALUES FOR SIMULATION

Symbol	Value
$\bar{w}_1, \bar{w}_2$	$950\,\mathrm{kg}$
D_1, D_2	$150\,\mathrm{m}$
$\bar{v}_1, \bar{v}_2$	$4.0\,\mathrm{m/s}$
T_{DL}	$900\,\mathrm{s}$
t_{ld}	$35\,\mathrm{s}$
t_{dp}	$30\,\mathrm{s}$

The simulation was set to end when the remaining soil in both loading sites was emptied or when the deadline had been reached. When one team finishes moving all the soil, all robots in the team are to return to the waiting area and stand by in case the other team calls an additional cooperator.

IV. RESULT

Using the simulation environment described in the previous section and the parameters listed in Table III, a single robot team performed the earthmoving task. Since Real Performance can only be calculated after all dump trucks have completed one round trip, the value remains zero for a certain period of time (about 200 seconds in this case) after the start of the task and after changing the number of teams. However, if only some of the dump trucks return, team performance can be estimated by assuming that all dump trucks contribute equally to the team performance. This estimated Real Performance is shown with a dashed line as a reference value.

The graph shows that the Predicted Performance increases or decreases around $t = 200$, indicating that the performance prediction model has been modified. This modification is due to inaccuracies in the pre-defined parameters (Table I) and is a phase in which the prediction model was adjusted to be closer to reality. The excavation volume of the excavator and the round-trip route and time of the dump truck can change due to slight differences in simulation results. The reason why measured performance values are not constant is due to variations in the excavation volume of excavators.

Since the velocity of the dump truck changed at $t = 400$, the error correction algorithm reflected this event in the performance prediction model, leading to a decrease in Predicted and Real Performance around $t = 600$. Later, the number of dump trucks in Team 2 was increased to avoid missing the deadline due to the lower performance of the dump trucks.

It can also be seen that the number of dump trucks was increased for Team 1 before the model was modified. This occurred when the estimated task completion time exceeded the deadline before the Real Performance was updated, triggered by the fact that it was taking longer for the dump trucks to return due to their reduced velocity. This can be explained with Eq. (3) because it shows that the left-hand side with t may exceed the deadline at the end of the work, even if performance remains the same.

As a result, we were able to confirm that team evaluation based on internal models and model modification based on

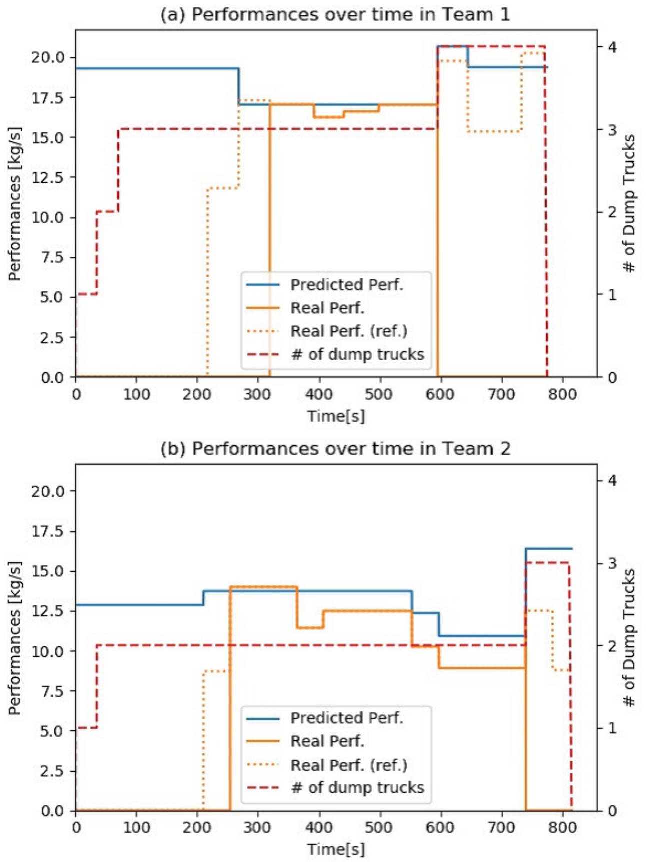

Fig. 6. The performance transition for each team (Coordinator) during the simulation experiment. The number of dump trucks in the team is also indicated. Since the velocity of the dump trucks changed at $t = 400$, it can be seen that both Predicted and Real Performance decreased around $t = 600$. Later, the number of dump trucks in team 2 was increased in order to avoid breaking the deadline due to the lower performance of the dump trucks.

local sensing, which are part of our proposed architecture, are functional. The distributed coordination algorithm implemented as a case study was shown to be able to use only the necessary number of dump trucks from a pool of 10 for organizing each team and completing the task within the deadline.

V. CONCLUSION

In this study, a performance-based distributed cooperative architecture was proposed, capable of making self-organized action decisions to perform automatic earthmoving tasks in a variable environment. A concrete algorithm, in line with the architecture, was also proposed and implemented. This algorithm uses Predicted and Real Performance to adjust the organization of the robot team. The key elements of this algorithm are dynamic team formation using performance as a single metric and model modification with cause identification when the performance changes. To verify whether these elements work in a self-organized manner, we conducted an experiment using the Vortex Simulator and ROS 2, which confirmed that when the velocity of dump trucks decreased, the performance prediction model was modified and the number of dump trucks in the team increased to compensate for the decrease in performance.

Thus, under the conditions of the experiment, it was found that the Coordinator Algorithm could autonomously recruit necessary Cooperators and adaptively organize the team according to the situation, without needing to contact any centralized supervisor. Although only the case in which dump trucks were added was tested in this experiment, further various situations such as adding excavators to the team should also be validated in future studies. Additionally, proposing an algorithm that can determine which robots are most deficient in each type of robot that constitutes the team based on performance will be an important future direction.

REFERENCES

[1] H. A. D. Nguyen and Q. P. Ha, "Robotic autonomous systems for earthmoving equipment operating in volatile conditions and teaming capacity: A survey," *Robotica*, vol. 41, no. 2, pp. 486–510, 2023.

[2] K. Nagatani, M. Abe, K. Osuka, P.-j. Chun, T. Okatani, M. Nishio, S. Chikushi, T. Matsubara, Y. Ikemoto, and H. Asama, "Innovative technologies for infrastructure construction and maintenance through collaborative robots based on an open design approach," *Advanced Robotics*, vol. 35, no. 11, pp. 715–722, 2021.

[3] G. D. M. Serugendo, M. P. Gleizes, and A. Karageorgos, "Self-organization in multi-agent systems," *The Knowledge Engineering Review*, vol. 20, no. 2, pp. 165–189, 2005.

[4] D. Schmidt and K. Berns, "Construction site navigation for the autonomous excavator Thor," in *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, 2015, pp. 90–97.

[5] W. Agassounon, A. Martinoli, and K. Easton, "Macroscopic Modeling of Aggregation Experiments using Embodied Agents in Teams of Constant and Time-Varying Sizes," *Autonomous Robots*, vol. 17, no. 2, pp. 163–192, 2004.

[6] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-Based Decentralized Auctions for Robust Task Allocation," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[7] N. Palmieri, X.-S. Yang, F. D. Rango, and A. F. Santamaria, "Self-adaptive decision-making mechanisms to balance the execution of multiple tasks for a multi-robots team," *Neurocomputing*, vol. 306, pp. 17–36, 2018.

[8] L. Zhang, J. Zhao, P. Long, L. Wang, L. Qian, F. Lu, X. Song, and D. Manocha, "An autonomous excavator system for material loading tasks," *Science Robotics*, vol. 6, no. 55, p. eabc3164, 2021.

[9] G. Notomista, S. Mayya, Y. Emam, C. Kroninger, A. Bohannon, S. Hutchinson, and M. Egerstedt, "A Resilient and Energy-Aware Task Allocation Framework for Heterogeneous Multirobot Systems," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 159–179, 2022.

[10] E. Seraj, L. Chen, and M. C. Gombolay, "A Hierarchical Coordination Framework for Joint Perception-Action Tasks in Composite Robot Teams," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 139–158, 2022.

[11] CM Labs, "Vortex Studio," accessed: Aug. 2023.