Online Supervisory Control for Safe and Adaptive Human-in-the-Loop Warehouse Logistic Automation

Makoto Kimura, Manato Fujimoto, and Kai Cai

Abstract— **In addition to labour shortage issues in the logistics industry, the recent impact of the new coronavirus pandemic has also led to an increase in online shopping usage worldwide. As a result, warehouse automation technology using robots has been attracting attention, and the increased efficiency is expected to reduce the burden on workers and cut labor costs. However, with current technology, it is difficult to fully automate a warehouse using only robots. In order to deal with this problem, we have adopted the concept of human-in-the-loop, in which a human intervenes in an automated system. Human-in-the-loop combines the strengths of both robots (accuracy, power, and speed) and humans (dexterity and problem-solving), enabling us to cope with unexpected problems without problems. Currently, when a problem occurs, the entire system has to be shut down for safety reasons when a human goes to deal with the problem, but by incorporating human-in-the-loop, a human can enter the warehouse and deal with the problem without shutting down the entire system. In addition, as we consider how to further improve efficiency, it is important to consider the following In thinking about further efficiency improvement, we also considered the case where robots, which are good at transporting goods, and humans, which are good at picking, work together by utilizing their respective strengths.**

I. Introduction

In recent years, the logistics industry has been under increasing strain due to labour shortages and the increasing use of online shopping. In addition, this problem has recently been exacerbated by the impact of the spread of the new coronavirus [1], [2]. As a result, warehouse automation systems using robotic warehouse automation technology have attracted much attention. The use of warehouse automation systems is expected to significantly increase efficiency in warehouses, reduce the workload of workers and cut labor costs.

Many methods exist to realise warehouse automation systems. Prior research using deep reinforcement learning [3], [4] has aimed to realise optimal route planning and improve operational efficiency by proposing Automated Guided Vehicle algorithms and conducting simulation verification. In addition, in previous research aiming to optimise product placement, the Analytic Hierarchy Process [5], which hierarchises products by determining their characteristics, and the Apriori algorithm [6], which pursues correlations between products, are used to pursue shorter working hours in the warehouse. The system is designed to reduce the amount of time spent in the warehouse. Blocking and deadlock conditions are two of the most common problems with warehouse automation systems. A blocking condition means that two or more robots collide with each other. A deadlock condition means that robots are stuck in a deadlock situation because they have met each other. Supervisor control [7], especially in discrete event systems [8], [9], is known to be an effective way to address these problems. Here, some previous work on supervisor control is presented [10]. First of all, there is a study that used computational software called TCT, which specialises in supervisor calculations, to build a logistics warehouse automation system using supervisor control [11]. In this study, the control is designed to avoid collisions and blocking between robots by creating supervisors. There is also research using online supervisor control [12]. In this study, the supervisor is calculated by limiting the number of states of each automaton to reduce computational complexity. Furthermore, there are studies [13] that automatically perform multiple task assignments in warehouse automation to consider further efficiency aspects [14], and studies that consider the case where priorities are assigned to each task. In these studies, online supervisor control has been used to deal with the unpredictable occurrence of tasks.

The most critical challenge in realising warehouse automation systems is the problem that it is difficult for robots alone to deal with problems that occur in warehouses. In fact, robots can accomplish the speedy transport of goods without any problems. However, when unexpected problems occur in the warehouse, such as when goods have fallen down the aisles in the warehouse or when the shape of the goods is too irregular for the robot to pick, the robot alone cannot cope with the problem. When such problems occur, the entire system is currently shut down and the problem is dealt with by humans. Such a process is by no means efficient. Here, this study introduces the concept of Human-in-the-loop in contrast to previous studies where only robots are the control target; Human-in-the-loop refers to the idea of building a system by daring to intervene humans in an automated system. Here, we present some previous research on Human-in-the-loop [15]. First, there is a study [16] that pursues human-robot interaction. This study proposes different human roles in smart factories. Second, there is a study [17] that seeks to minimise the time between the occurrence of a task and its classification by utilising the strengths of humans and robots. In this study, the path-finding problem is solved by analysing the information collected by the robot in the field and classifying the task by the human. In this study, the human-in-the-loop concept, which utilises the respective strengths of humans and robots, is also introduced to consider the case where humans and robots

The authors are with Department of Electrical and Information Engineering, Osaka City University. Emails: kimura@c.info.eng.osaka-cu.ac.jp (M. Kimura), kai.cai@eng.osaka-cu.ac.jp (K. Cai)

work cooperatively in a logistics warehouse to improve work efficiency.

This study considers the case of human intervention in a warehouse automation system with several robots carrying goods. The realisation of an on-line system in supervisor control was also considered. However, when realising human intervention in a warehouse where robots are present, the safety of humans must be given the highest priority. This problem was addressed using the supervisory control concept. The specific definition of safety is explained in detail in Section III-A.

This study incorporated human-in-the-loop into a warehouse automation system. This enables this study to achieve adaptability, which allows the system to respond appropriately to unforeseen problems, flexibility, which allows it to handle a wide range of task types, and efficiency, which allows it to handle tasks quickly. With regard to adaptability, by utilising human problem-solving skills, the system is able to deal with all kinds of troubles. For flexibility, by utilising human dexterity, it is able to handle all kinds of tasks. Regarding efficiency, by making good use of the robot's and human's strengths, the time required to pick up and deliver packages could be minimised. By solving these issues through this research, we believe we can contribute to solving the manpower shortage problem in the logistics industry.

II. Supervisor Control Theory

A discrete event system is a dynamic system in which the state transitions in a discrete manner. The basic model of a discrete event system, an automaton, is shown below.

$$G = (Q, \Sigma, \delta, q_0, Q_m)$$

Define Q as a finite set of states, Σ as a finite set of events, δ as the transition function of states, $q_0 \in Q$ as the initial state, and $Q_m \subseteq Q$ as the set of accepted states.The acceptance state is also called the marker state and represents the end state of the task.The transition function of the state δ can also be expressed as $\delta : Q \times \Sigma \to Q$. The transition from state $q \in Q$ to state $q' \in Q$ by the occurrence of event $\sigma \in \Sigma$ is denoted by $\delta(q, \sigma) = q'$. Furthermore, if an event $\sigma \in \Sigma$ can occur in state $q \in Q$, then $\delta(q, \sigma)$ is said to be defined, and is denoted by $\delta(q, \sigma)!$.

An arbitrary subset of Σ^* is called a language L of an event set Σ and denoted by $L \subseteq \Sigma^*$. For an automaton G, for the set of all event sequences that can be generated from an initial state q_0 , we define the following

$$L(G) = \{s_1 \in \Sigma^* \,|\, \delta(q_0, s_1)!\} \subseteq \Sigma^*$$

For the automaton G, the set of all event sequences that can occur from the initial state q_0 to the target state Q_m is defined as follows

$$L_m(G) = \{s \in L(G) \,|\, \delta(q_0, s) \in Q_m\} \subseteq L(G)$$

When $\overline{L_m(G)} = L(G)$, all event sequences in G can reach the receiving state, and G is said to be non-blocking. Synchronized composition is a method of computation that creates a new automaton from multiple automata; if two automata G_1 and G_2 are given as follows, the automaton G after synchronized composition of G_1 and G_2 is defined as follows.

$$G_1 = (Q_1, \Sigma_1, \delta_1, q_{0,1}, Q_{m,1})$$

$$G_2 = (Q_2, \Sigma_2, \delta_2, q_{0,2}, Q_{m,2})$$

$$G = G_1 || G_2 = (Q, \Sigma, \delta, q_0, Q_m)$$

To create an automaton in which three automata are synchronously synthesized, synchronously synthesize an automaton in which two automata are synchronously synthesized and a new automaton. Furthermore, by repeating this operation, it is possible to synchronously synthesize four or more automata. Therefore, the calculation method for synchronous synthesis of n automata can be expressed as follows.

$$G_1 || G_2 || G_3 || \cdots || G_n = (\cdots((G_1 || G_2) || G_3) || \cdots) || G_n$$

For any language $K \subseteq L_m(G)$ for the set $C(K)$ of all controllable sub-languages of K, the following relation holds.

$$C(K) = \{K' \subseteq K \,|\, \overline{K'}\Sigma_u \cap L(G) \subseteq \overline{K'}\}$$

From $\emptyset \in C(K)$, $C(K)$ is not an empty set. Also, for the largest element $\sup C(K)$ that contains all of the elements of $C(K)$, it can be defined as follows.

$$\sup C(K) = \cup\{K' \,|\, K' \in C(K)\}$$

This $\sup C(K)$ is called the maximally controllable sub-language of K. Furthermore, when the language of the control request $K \subseteq L_m(G)$ is not controllable and $\sup C(K) \neq \emptyset$, there is always a non-blocking supervisor $V_{\sup}$ that satisfies the following condition.

$$L_m(V_{\sup}/G) = K_{\sup}$$

This supervisor $V_{\sup}$ permits as many occurrences of $L_m(G)$ as possible to satisfy the control specification. Such a supervisor $V_{\sup}$ is also called a maximum allowable supervisor.

III. Setting up the environment in Human-in-the-Loop

A. Warehouse Structure

In this study, we use a warehouse with the structure shown in Figure 1. The warehouse consists of four parts: waiting area, aisle, shelf, and destination. Tasks are stored in the shelves, and it is impossible for robots to pass each other in the aisles. We consider the case where a human and a robot coexist and work together in this warehouse. The warehouse is divided into 70 squares, and a state number is assigned to each square. The waiting area at the top of the warehouse is in state "0", the aisles and shelves in the warehouse are in states "1" to "70", and the destination at the bottom of the warehouse is in state "71". All tasks cannot be generated. Since the occurrence of all tasks is uncontrolled and the tasks are managed in a queue structure, the maximum number of tasks that can be assigned to a single robot is 1. For all robots, the only movements are forward, right, and left. They are not allowed to enter the shelf except when loading goods, and when processing a task, they always enter from the top of the shelf (north) and exit from the bottom of the shelf (south). When a task is assigned to a robot, the robot waits in the waiting area, and when the task is assigned, the robot proceeds to the location where the package is located, retrieves the package, proceeds to the destination at the bottom of the warehouse, and delivers the package to complete the task. The human is working in the lower part of the warehouse, and will go to the destination location only when a problem occurs in the warehouse or when the human's cooperation is needed.

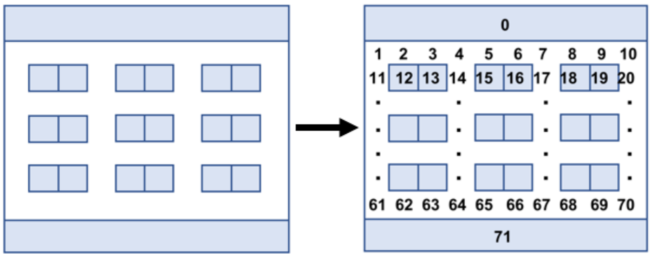

Fig. 1. Warehouse Structure

When there are multiple robots and humans in a warehouse, the following four issues arise.
(1) Safety: Prevent contact (collision) between robots or between humans and robots.
(2) Accuracy: Completing all tasks
(3) Efficiency: All agents should complete their tasks in the shortest possible time.
(4) Adaptability: all randomly occurring tasks are handled

As for safety, in the case of robot-only control, the conditions that prohibit two robots from being in the same place are defined for all combinations of robots and set as control requirements. However, in the case where humans coexist in the warehouse, additional safety must be required. This issue will be discussed in detail in the modelling methods (III-C).For accuracy, consideration should be given to avoiding blocking and deadlock states (Fig. 2). The deadlock state is a state in which both robots block each other's path and cannot move. This can be solved by requiring the supervisor to be non-blocking. Efficiency is addressed by calculating the shortest path using the $A*$ algorithm. Adaptability is achieved by making the supervisor control online.

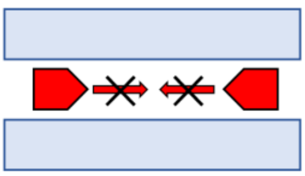

Fig. 2. deadlock

B. Agent modeling

The $n(\geq 1)$ robots present in the warehouse are defined by the automaton $G_i(i = 1, 2, \ldots, n)$.

$$G_i = (Q_i, \Sigma_i, \delta_i, q_{0,i}, Q_{m,i})$$

Q_i is the set of all states on the shortest path of the i robot, Σ_i is the set of events of the i robot, δ_i is the state transition function of Q_i, $q_{0,i}$ is the initial state of Q_i, and $Q_{m.i}$ is the set of accepted states of Q_i. All events are controllable events, and we consider the case of moving in four directions: east, west, north, south, and south (TABLE. I). m people are defined in the same way. All robots initially exist in the waiting area, and after the task assignment, they move in the shortest path through the task to the goal (Fig. 3).

event	Event number
Go North	$i \times 10 + 1$
Go East	$i \times 10 + 3$
Go South	$i \times 10 + 5$
Go West	$i \times 10 + 7$

TABLE I
ROBOT $i \in \{1, \ldots, n\}$ EVENT NUMBER

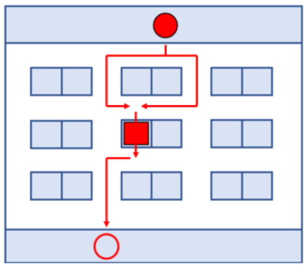

Fig. 3. Shortest path for the robot assigned the task

C. Methods in modelling

In a human-in-the-loop situation, where a human and a robot are present in the warehouse at the same time, the safety of the human has to be given top priority. The efficiency of the robot's task processing should also be considered. There is a trade-off between the safety of the human and the efficiency of the task, and different methods exist depending on which is more important. In this study, we propose two methods for modelling robots and humans.
(1) Modelling 1

Human safety should be the top priority. There is a difference between humans and robots in terms of uniformity of speed. Robots are able to move at a set and uniform speed, while humans are not uniform in speed due to fatigue and impatience. Therefore, it is necessary to ensure safety regardless of the speed of the human. Therefore, we propose a method to prevent robots from intruding on the path of humans (Fig. 4). Specifically, the idea is to make all the paths that humans are scheduled to pass through as prohibited areas for robots, and when the humans finish passing through, the robots will be allowed to pass through. One way to realize this idea is to define human automata using uncontrollable events. The reason for this is that humans cannot be controlled as strictly as robots, and there may be individual differences among humans. By making the human behavior an uncontrollable event, the human path is perceived as an obstacle from the robot's point of view, and the supervisor is calculated.

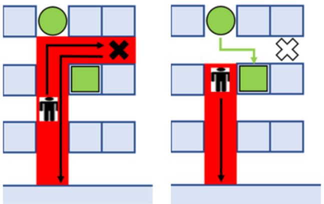

Fig. 4. Modelling 1

(2) Modelling 2

The safety aspect of humans should be ensured, but the efficiency aspect should also be considered. Robots can be controlled accurately because they are based on strict control, but humans have indeterminate delays in reaction. In addition, the first method is not as efficient because it considers the human safety aspect too much and is quite conservative. Therefore, it is necessary to pursue efficiency while dealing with the indeterminate delay in human response. Therefore, we propose a method of setting a no-go area around the human (Fig. 5). Specifically, the prohibited area around the human is designed to move dynamically according to the human's movement.

As for the control of collision avoidance, in the case of robots, when there is another robot in the direction of travel of one of the robots, collision is avoided by prohibiting the action of either robot. However, when considering a collision between a human and a robot, it is very dangerous to prohibit actions after the robot comes in front of the human, because an error may occur and lead to a major accident. Therefore, it is necessary to maintain a certain distance between humans and robots.

In order to be able to arbitrarily determine the minimum distance that must be kept between the human and the robot to ensure safety, the range of the no-entry area around the human was defined by the variable d. This makes it possible to flexibly respond to various robot specifications and warehouse sizes.

D. Online Supervisor Control

In an actual warehouse, it is impossible to know when and where a problem will occur. Therefore, by introducing online supervisor control, it is possible to flexibly respond to "time" and "place" and to conduct simulation verification in a manner closer to that of an actual warehouse. The algorithm of online supervisor control is shown in the following five steps.

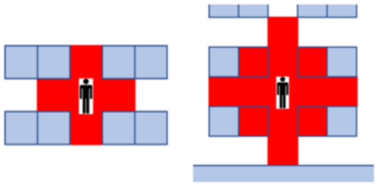

Fig. 5. Modelling 2

(1) Set the initial state, transit point, and destination for each agent, and calculate the shortest path using the A* algorithm method.
(2) Create an automaton for each agent and perform synchronous composition to create the automaton to be controlled.
(3) Create an automaton of the control request.
(4) Calculate the supervisor such that the control target satisfies the control request.
(5) If there are unprocessed tasks, return to (1).

By making the supervisor online, it is possible to automatically recalculate the route. The timing of recalculation includes when a trouble occurs, when there is an unprocessed trouble when a trouble is processed, and when a new trouble occurs when no trouble is assigned (Fig. 6).

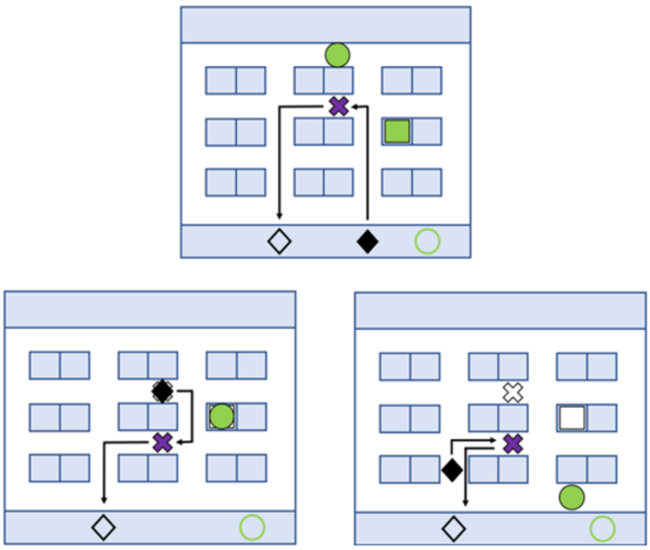

Fig. 6. Online Supervisor Control

IV. Case Study

In a Human-in-the-Loop, where a human and a robot are simultaneously present in the warehouse, the proposed method is used to model the agents, online supervisor control is introduced, and the following simulation validation is performed for the case of two robots and one human.

A. In case of trouble

We consider the case of a warehouse with two tasks for the robots and several problems to be handled by the human. By putting the supervisor control system online, it is possible for a human to handle troubles whose timing of occurrence is random.

(1)Adopting modelling 1

In order to make the entire human path a prohibited area, the robot's behavior is defined by controllable events and the human's behavior by uncontrollable events. Fig. 7 (left) shows the prohibited area of a robot immediately after trouble is assigned to a human. The red and green circles represent robots 1 and 2, respectively, and the squares of the same color and the unfilled circles represent the task and destination of each robot. The black and unfilled diamonds represent the human and the human's goal point, respectively, and the X mark represents trouble. The red-colored area is a corridor through which humans may pass, and is a prohibited area for robots. The state after six transitions from Fig. 7 (left) is shown in Fig. 7 (right). Since the human passed through state 47 toward the trouble and transitioned to state 48, state 47 is excluded from the prohibited area, and robot 1 transitions to state 47 by transition 15. Robot 2 is waiting in state 33, and since the prohibited area exists in state 43, it is still prohibited from taking the next action. The human has arrived at the destination and has taken care of the problem.

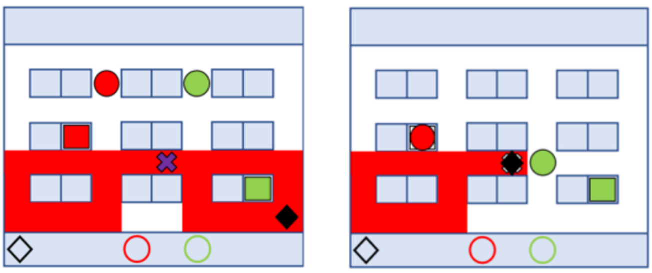

Fig. 7. Adopting modelling 1

(2)Adopting modelling 2

In order to set up a prohibited area around the human, the *mutex*[1] function is used to control the robots so that they do not exist in the prohibited area at the same time. By making the range of the prohibited area variable with the variable d, it is possible to respond to various situations. $d = 1$ and the state of the warehouse when one trouble occurs is shown in Fig. 8 (left). The prohibited area around the human is in state 37, blocking the robot's path to the task, so robot 1 is prohibited from making transition 15 from its arrival in state 27 to the task from the third transition to seven transitions later, and continues to wait. In the eighth transition, when the human moves from state 47 to state 46 in transition 37, state 47 is excluded from the prohibited area, and robot 1 is allowed to make the transition 15 toward the task. The state of the warehouse when the range of the prohibited area is changed to $d = 2$ is shown in Fig. 8 (right). In the eighth transition, when the human moves from state 47 to state 46 in transition 37, state 27 is removed from the prohibited area, and robot 1 is allowed to make the transition to task 15. In the eighth transition, the human moves from state 47 to state 46 in transition 37, which removes state 27 from the prohibited area and allows robot 1 to make the transition to task 15.

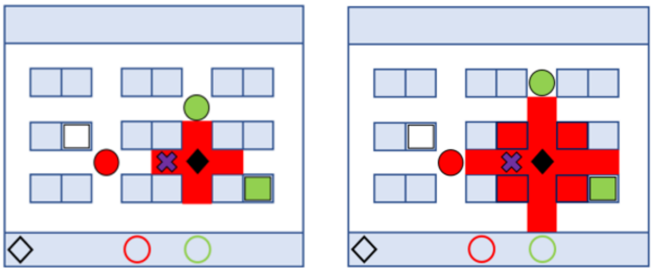

Fig. 8. Adopting modelling 2 (Change the scope of the prohibited area)

Furthermore, the system is managed in a queue structure in order to respond to multiple problems in the order in which they occur. By introducing online supervisor control, recalculation is automatically performed when necessary. The state of the warehouse when there are unprocessed troubles during trouble handling is shown in Fig. 9 (left). After the human handles the trouble, it recalculates the shortest path and goes to the second trouble. The state of the warehouse when a new trouble occurs when no trouble is assigned is shown in Fig. 9 (right). The person who was on his way to the destination calculates a new shortest path and heads for the second trouble.

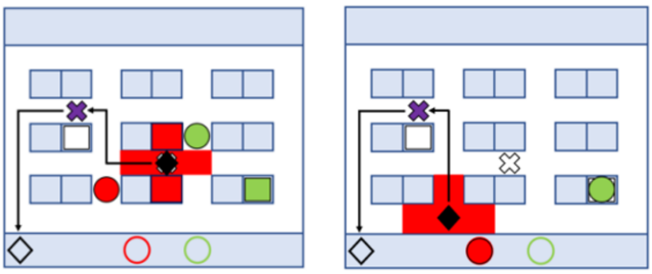

Fig. 9. Adopting modelling 2 (Online Supervisor Control)

B. Comparative experiments of Modelling 1 and 2

In order to compare the efficiency aspects of Modelling 1 and Modelling 2, we conducted a comparison experiment. In the experiment, all the locations were randomized, and the time was randomized only for troubles, and the total number of transitions to complete the task was measured for 10 times. The results of the measurements for the four patterns of no

[1]The *mutex* function is a function that performs exclusion control.

trouble, modelling 1, and modelling 2 ($d = 1, 2$) are shown in TABLE. II. Based on the results, we tested the difference of the population means between Modelling 1 and Modelling 2 ($d = 1$), and between Modelling 1 and Modelling 2 ($d = 2$), and found that Modelling 2 is superior to Modelling 1 in terms of efficiency.

Total number of transitions	No trouble	Model1	Model2 d=1	Model2 d=2
Average	25.1	31.8	27.1	28.2
Sum of squares	6353	10294	7429	8014

TABLE II
COMPARISON EXPERIMENT (UNIT: TRANSITION)

C. *Collaboration between humans and robots*

Collaborative work can take advantage of both the robot's and the human's strengths. The definition of human-robot collaboration is such that a task is only completed when the robot is simultaneously in the task position and the human is in the position next to the task (east or west). The reason why we chose the position for human cooperation to be next to the task (east or west) is that the robot passes from the top (north) to the bottom (south) of the shelf when processing the task, and if the position for human cooperation is placed above (north) or below (south) the special task, the robot's actions before and after processing the task will be limited by the presence of humans. The reason for this is that if the robots are placed above (north) or below (south) the special task, their actions before and after the task will be limited by the presence of humans. Two tasks in the warehouse that require the collaboration of a human and a robot.

Create an automaton (Fig. 10) that satisfies the condition that the common event of a transition to perform a joint task must occur only once as a control request. The state in which no special task has been processed is represented by state 0, the common event by transition 109, and the state in which a special task has been processed by state 1.

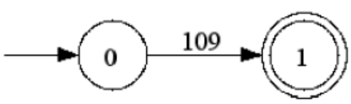

Fig. 10. Control Request

When two or more tasks occur, a different common event must be set for each task. The transitions of common events for task $k \in \{0, 1, 2, \ldots, l\}$ are defined by $k \times 10 + 109$. The proximity of humans and robots is unavoidable when performing collaborative tasks. In order to minimize the relaxation of the prohibited area, we excluded the prohibited area around the human only when they are collaborating. Synchronous synthesis of the control target and the control request is used to calculate the supervisor such that the control target satisfies the control request. Online supervisor control was introduced to automatically recalculate the supervisor when necessary. The state of the warehouse when there are unprocessed tasks during task processing is shown in Fig. 11 (left). After collaborating on a task, the system recalculates the shortest path for humans and goes to the second task. The state of the warehouse when a new task occurs when no task is assigned is shown in Fig. 11 (right). The human who was on its way to the destination calculates a new shortest path and heads for the second task.

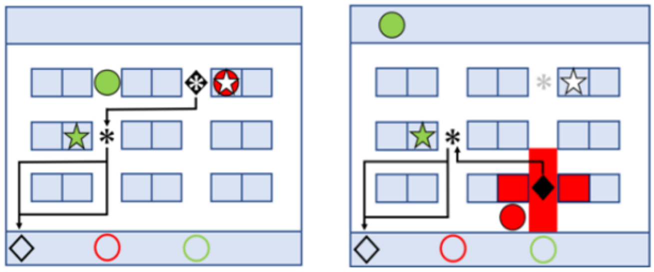

Fig. 11. Collaboration between humans and robots

D. *Scale up of experiments*

In order to assume a more realistic form, the scale of the experiment was increased. Specifically, an increase in the number of agents is considered. In the TCT used so far, the maximum number of agents is 3, which makes it impossible to expand the scale of the warehouse. Here, we introduced the Semi Model Free supervisory control algorithm, a discrete control algorithm that takes efficiency and speed into account. This is expected to further increase the number of agents.

In order to compare the computation time according to the number of agents, experiments were conducted to measure the computation time at various numbers of agents. In the experiments, the case of a trouble in a logistics warehouse is assumedIV-A. Specifically, the computation time was measured 10 times for each of 6 12 agents and the average value was calculated. In addition, measurements were taken in each of the five cases where the breakdown of agents was 1 5 humans. A graph was then created in each of the cases with 1 5 humans 12 From the graphs, it can be seen that regardless of the agent breakdown, the computation time increases as the number of agents increases.

V. CONCLUSION

A. *Summary*

In this study, we considered human-in-the-loop in a warehouse automation system using a discrete event system. By introducing online control, it is possible to respond flexibly to unpredictable troubles. By using a queue structure, it is possible to deal with multiple troubles that occur. By comparing the two proposed methods, we clarified the superiority of the two methods in terms of efficiency. In addition, we realized

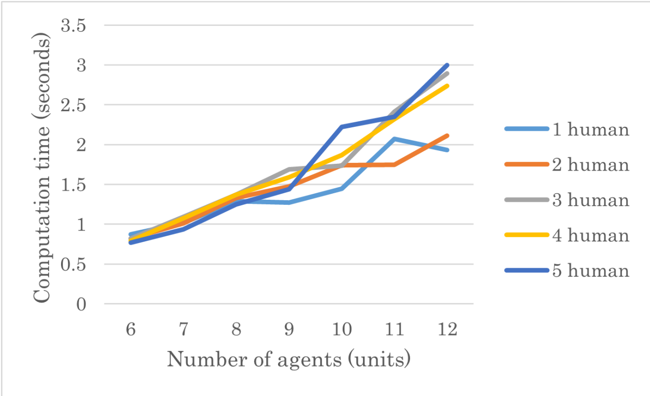

Fig. 12. Computation time with Semi Model Free.

the improvement of work efficiency by utilizing the characteristics of human and robot in collaborative work.Finally, to increase the scale of the experiment, the Semi Model Free supervisor control algorithm was introduced to increase the number of agents.

B. Future Challenges

Future prospects include extending the size of the warehouse itself in order to further increase the scale of the experiment. In addition, we will conduct a detailed analysis of the characteristics of humans and robots in order to pursue the new proposed method. Specifically, we aim to propose a new method that takes into account delays in turning movements, different sizes and speeds of the robot itself, indeterminate human delays, and noisy rational models[18].

References

[1] T.Tsukada, "Rapidly growing use of online shopping and e-money - results from the household consumption survey 2018 -." https://www.stat.go.jp/info/today/141.html, 2019.

[2] H.Ishihara, "New coronavirus infections change online shopping - results of a household consumption survey," https://www.stat.go.jp/info/today/162.html, 2020.

[3] R. Kamoshida and Y. Kazama, "Acquisition of automated guided vehicle route planning policy using deep reinforcement learning," in *2017 6th IEEE International Conference on Advanced Logistics and Transport (ICALT)*. IEEE, 2017, pp. 1–6.

[4] X. Liao, Y. Wang, Y. Xuan, and D. Wu, "Agv path planning model based on reinforcement learning," in *2020 Chinese Automation Congress (CAC)*. IEEE, 2020, pp. 6722–6726.

[5] L. M. Tinelli, K. C. T. Vivaldini, and M. Becker, "Intelligent warehouse product position optimization by applying a multi-criteria tool," in *International Workshop on Robotics in Smart Manufacturing*. Springer, 2013, pp. 137–145.

[6] Y. Ito and S. Kato, "An apriori-based approach to product placement in order picking," in *2016 IEEE International Conference on Agents (ICA)*. IEEE, 2016, pp. 114–115.

[7] W. M. Wonham and K. Cai, "Supervisory control of discrete-event systems," Springer, 2019.

[8] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[9] P. J. G. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control and Optimization*, vol. 25, no. 1, pp. 206–230, 1987.

[10] K. Cai, "Warehouse automation by logistic robotic networks – a cyber-physical control approach," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, pp. 693–704, 2020.

[11] Y. Tatsumoto, M. Shiraishi, and K. Cai, "Application of supervisory control theory with warehouse automation case study," *Systems/Control/Information*, vol. 62, no. 6, pp. 203–208, 2018.

[12] Y. Tatsumoto, M. Shiraishi, K. Cai, and Z. Lin, "Application of online supervisory control of discrete-event systems to multi-robot warehouse automation," *Control Engineering Practice*, vol. 81, pp. 97–104, December 2018.

[13] M. Kasahara and K. Cai, "Online multi-agent supervisory control for warehouse automation: heterogeneous payloads," *the 29th Mediterranean Conference on Control and Automation, Bari, Italy*, Jun 2021.

[14] ——, "Online multi-agent supervisory control for warehouse automation: prioritized tasks," *the 17th International Conference on Automation Science and Engineering, Lyon, France*, Aug 2021.

[15] W. Li, D. Sadigh, S. S. Sastry, and S. A. Seshia, "Synthesis for human-in-the-loop control systems," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2014, pp. 470–484.

[16] C. Cimini, F. Pirola, R. Pinto, and S. Cavalieri, "A human-in-the-loop manufacturing control architecture for the next generation of production systems," *Journal of manufacturing systems*, vol. 54, pp. 258–271, 2020.

[17] K. Savla, T. Temple, and E. Frazzoli, "Human-in-the-loop vehicle routing policies for dynamic environments," in *2008 47th IEEE Conference on Decision and Control*. IEEE, 2008, pp. 1145–1150.

[18] M. Kwon, E. Biyik, A. Talati, K. Bhasin, D. P. Losey, and D. Sadigh, "When humans aren't optimal: Robots that collaborate with risk-aware humans," in *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2020, pp. 43–52.