Paper:

Refining Two Robots Task Execution Through Tuning Behavior Trajectory and Balancing the Communication

Jorge David Figueroa Heredia*, **Shouhei Shirafuji****, **Hamdi M. Sahloul***,
Jose Ildefonso U. Rubrico**, **Taiki Ogata****, **Tatsunori Hara****, **and Jun Ota****

*Department of Precision Engineering, School of Engineering, The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan
E-mail: {figueroa, sahloul}@race.u-tokyo.ac.jp
**Research into Artifacts, Center for Engineering (RACE), The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan
E-mail: {shirafuji, rubrico, ogata, hara_tatsu, ota}@race.u-tokyo.ac.jp
[Received September 4, 2017; accepted April 3, 2018]

A method for modifying robot behaviors is introduced to improve robot performance during the execution of object manipulation tasks. The purpose of this method is to minimize the execution time of tasks and prevent collision with obstacles, including objects to be manipulated and the robot itself, by considering two approaches. The first is to use the potential that robots can provide, considering that the programs are based on events that are subject to the response of sensors. The second is to determine the maximum rate at which commands can be sent, without affecting the responses from the sensors, and, based on that, to accelerate or decelerate the execution of the task. The proposed method focuses on the refinement of two approaches: (a) modifying the trajectory of some behaviors, so that they are not executed step by step, but are executed in parallel, and (b) increasing the rate of sending robotic commands. To validate the proposed method, four real-world tasks are presented, including the flipping of a briefcase, the flipping of a weighing scale, the lifting of a weighing scale, and the opening of a folding chair, performed by a set of small robots. The reduction in execution time of the tasks varied between 54.2% and 73.6%; the implications of the improvement are discussed based on experimental results.

Keywords: cooperative manipulation, path refinement, human-robot interaction

1. Introduction

1.1. Background

Incorporating robots to conduct tasks generally performed by humans in environments such as homes or offices is a research area garnering widespread interest [1]. Although several studies deal with the motion-planning problem of mobile manipulator robots with maximum load-carrying capacity [2–5], it is difficult to handle heavy or large objects with mobile robots in some cases. Owing to the physical limitations of mobile robots compared with humans, the use of multiple robots becomes very important to overcome these limitations [6–8]. To expand the number of tasks that robots can perform, the transmission of information to robots on behaviors needed to complete tasks is required [9].

With robotic commands, robots execute different behaviors that enable them to perform tasks. These behaviors initially assigned to robots may require modifications to increase performance efficiency for reasons such as inefficient communication during behavior execution because of a lack of knowledge about the limitations of the existing architecture [10, 11] or to make more efficient trajectories to follow [12].

Modifying the behavior of robots has been studied with different purposes. Mericli et al. [13] proposed an approach using complementary corrective human feedback. After having observed the performance of robots, humans can correct an existing hand-coded algorithm through commands; the robot's performance was noted to be improved in the evaluation experiments.

In the trajectory planning of robots, initial solutions are modified to obtain solutions that are more efficient. Popular algorithms used include the construction of rapidly exploring random trees (RRTs), where the best solution is selected according to cost functionality based on criteria described in the study by LaValle et al. [14]. The algorithm called probabilistic roadmaps (PRMs) is also used, which constructs a roadmap graph by uniformly sampling configurations [15]. Predictive roadmaps are used for robots as the trajectory to follow, as Burns used in his study [16].

Probabilistic methods, such as RRT, PRM, or the combination of those, rapidly exploring roadmaps (RRMs) [17], include a refining and modifying process at the end of the algorithms.

Regarding the trajectories of robots, in general, they depend on constraints of several values such as position, velocity, acceleration, and jerk [18]. In addition, based on parameters obtained usually by sensors, several studies have been proposed with offline [19] and online [20, 21] manners for methods to generate trajectories that can be refined if circumstances so require.

Approaches for modifying the trajectory of a single robot based on the elastic band have been proposed, with good results, both offline [22] and online [23]. In the case of multiple robots, centralized [24] and decentralized [25, 26] approaches have been proposed to control and modify trajectories. Most studies in the literature use a two-phase decoupled approach, as in the study by Chiddarwar et al. [27], which introduced an approach based on a path modification sequence and the use of the incremental A* algorithm [28]; simulation results showed improvement in execution time.

For balancing the communication between robots based on the existing architecture for robot manipulation tasks, no studies have been conducted to the authors' knowledge.

1.2. Objective of this Study

The objective of the study is to refine task execution in terms of total time to complete the task by robots.

We focus on manipulation tasks executed by as many as two robots. The challenge addressed in this study is as follows:

- How can a precreated program for the two robots be modified, without compromising the successful execution of the task?

To address the above challenge, we solve the problem in the following two ways:

- Modify the trajectories generated by the behaviors of the robots by parallelizing their execution.
- Modify the communication rate of the robots based on finding the appropriate balance according to the existing architecture.

2. Problem Statement

2.1. Assumption

In this research, we considered the following assumptions:

- The robots have an arm section, in which manipulation is conducted, and a mobility section, in which gross motion is generated.
- For the working environment of the robots, there are no obstacles preventing a change of trajectory.

2.2. Input Specifications

We assume that the initial motion of two robots is as follows:

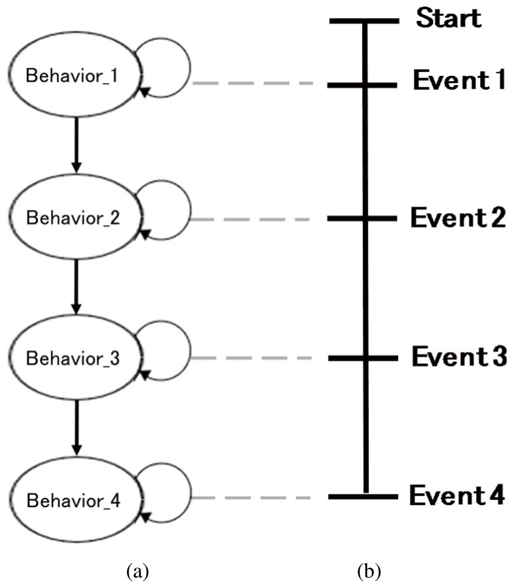


Fig. 1. Example of a state machine used as input: (a) behaviors governed by events and (b) time that each event took place.

Table 1. Categories of system architecture.

System Architecture	Category I	Category II	Category III
Specifications S: sensors A: actuators C: CPUs	each S and each A has its separate C	some S share C between them some A share C between them	some S share C with some A

Robot Programs:

They are structured as state machines given to two robots to execute the task successfully, with proper behavior of the robots and proper communication between them. These programs can be obtained by using a teaching process, as shown in [9].

Synchronized performance of tasks is based on the execution of robotic behaviors governed by events that are activated using sensor data [29]. The timing between robots is maintained according to communication between robots with respect to the current state of behaviors. A basic example of a state machine expected as input is shown in **Fig. 1**.

System Architecture:

For the system architecture of the robot systems, the main CPU requires the data generated by sensors mounted on the robot, and, based on the calculation using these data, robotic commands are generated and sent to the actuators of robots to perform specific behaviors.

Sensors and actuators use small CPUs in which the required drivers are installed, and raw information is processed to be converted into a format specified for the communication. The architecture in the connection between CPUs, sensors, and actuators on the robots comprises three types of category, as shown in **Table 1**. The

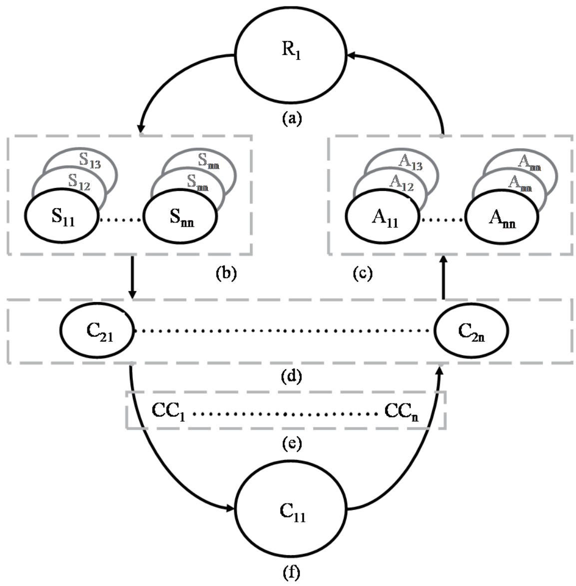


Fig. 2. General concept of system architecture used as input: Robot (a) has sensors to perceive the environment (b) and actuators to move its hardware (c). Sensors send raw data to sub-CPUs (d), and, in the same logic, actuators (c) received robotic commands from sub-CPUs (d). Information between sub-CPUs and the main CPU (f) flows through channels of communication (e).

difference in architecture lies basically in the communication channel, which is related to the number of CPUs used; for example, if each type of sensor has its own CPU, the communication channel is independent, but, if a different type of sensor shares the CPU, the data sent through the communication channel are also shared. We give detailed explanations of each category in Subsection 4.1.

We define the nomenclature used in **Table 1** to describe the architectures mentioned below, followed by a general description of the architecture shown in **Fig. 2**.

$R_\#$ = **R**obot;
$_\#$ = number of robots
S_{ij} = **S**ensor;
$_i$ = kind of sensors, $_j$ = number of sensors
A_{ij} = **A**ctuator;
$_i$ = kind of actuators, $_j$ = number of actuators
C_{ij} = **C**PU;
$_i$ = 1: main CPU, 2: sub-CPU, $_j$ = number of CPUs
$CC_\#$ = **C**ommunication **C**hannel;
$_\#$ = type of communication used by the architecture: categories shown in **Table 1**.

3. Proposed Method

3.1. System Overview

The proposed refinement method focuses on improving the total execution time of a task performed by robots, which includes two main points. **Fig. 3** shows an overview of the proposed system.

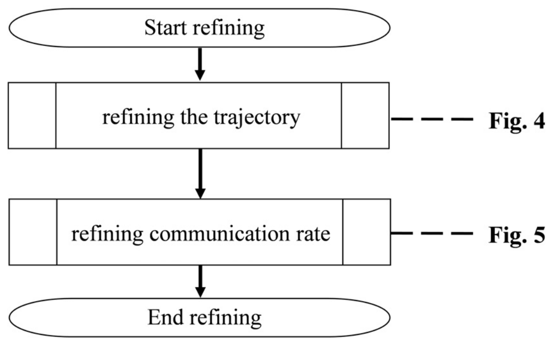


Fig. 3. Overview of refinement steps in proposed approach.

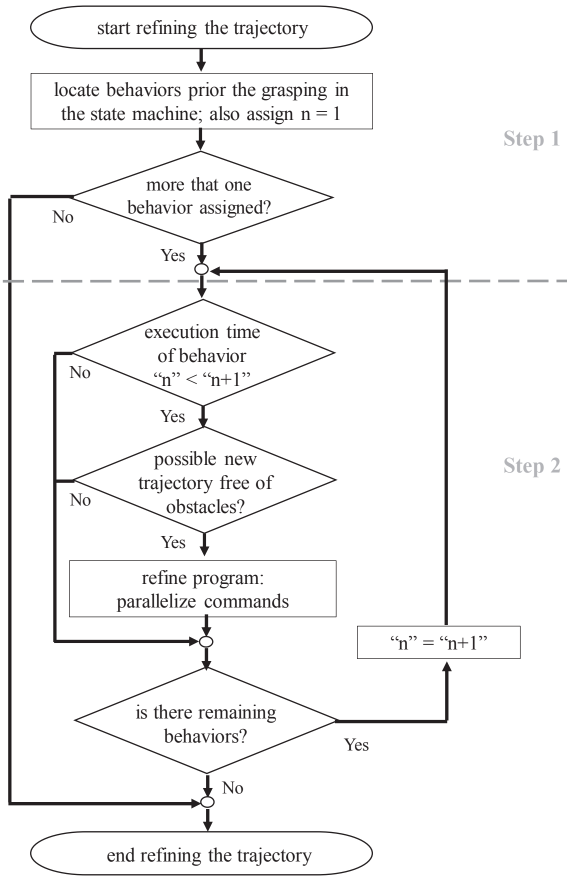


Fig. 4. Flowchart of process to refine the trajectory.

1. Refining the trajectory: Changing the original trajectories generated from the behaviors with the teaching process. See Subsection 3.2 and **Fig. 4** for details.

2. Refining the communication rate: Tuning the frequency of communication rate from the sensors to the main CPU and that from the main CPU to the actuators, as well as adjusting the movement speed of the robot base. See Subsection 3.3 and **Fig. 5** for details.

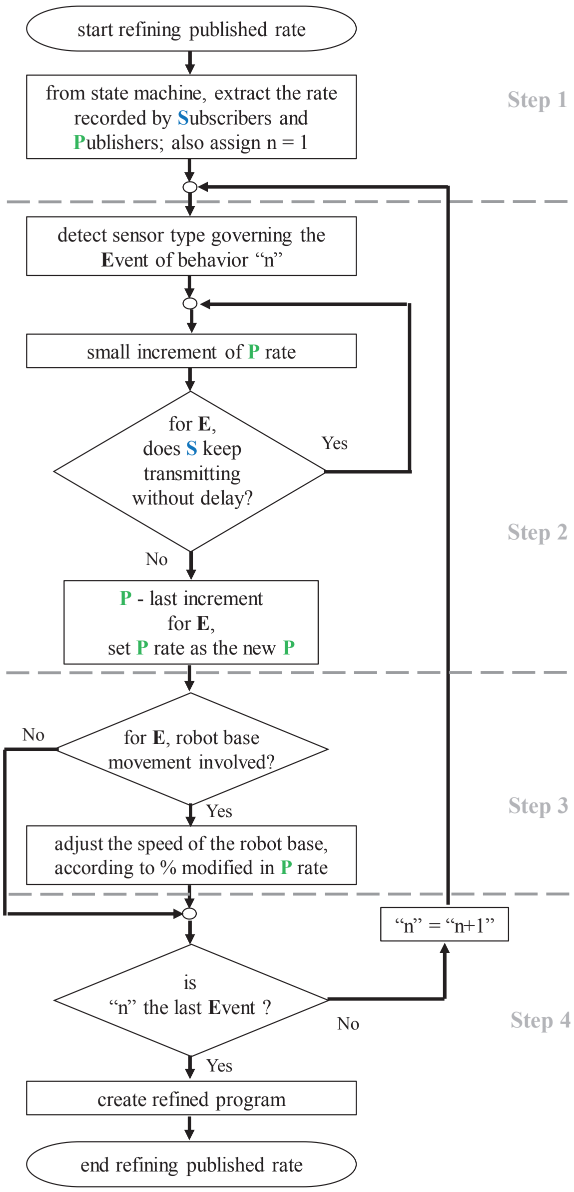


Fig. 5. Flowchart of process to refine the communication rate.

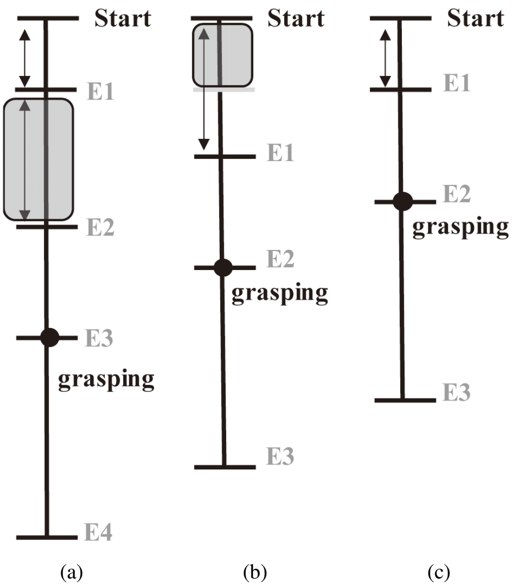


Fig. 6. Expectation after refining the trajectory: (a) store execution time in the state machine, (b) parallelize behaviors executed before the manipulation of the object, and (c) adjust the speed of the robot base.

3.2. Refining the Trajectory

In this process, we focus on modifying the robots' trajectories. We consider that adequate parameter tuning such as robot-object distance and execution time for each behavior is difficult to realize by a human and, therefore, difficult to consider in the state machines originally created.

We considered the situation that the trajectories generated by the state machine are executed sequentially. The machine waits for its assigned event to occur to continue with the next behavior to realize. We propose to modify the trajectory by checking whether we can integrate individual behaviors, such that the behaviors can be executed in parallel with respect to (a) the motion of other sections of the same robot, such as movement of the arm section and the mobile section, and (b) the motions of two robots. **Fig. 4** shows the detailed process.

In Step 1, the behaviors prior to the manipulation of the object are located from the state machine. In the case where more than one behavior are detected, two conditions are evaluated with the intent to parallelize the execution of the basic behaviors.

In Step 2, the first condition is to evaluate whether the present behavior "n" requires a shorter execution time than the subsequent behavior "$n+1$." If the condition is true, the algorithm continues with the condition that seeks to know whether there is no obstacle. If both conditions are true, it proceeds to parallelize the behaviors by modifying the input state machine, and the modified behaviors are executed one time to detect possible failures, such as collisions.

If the trajectories are changed owing to the parallelization of behaviors, we also need to consider the speed adjustment for the robot to improve the performance.

We adjust the speed of the robot base to reach the executing time of the behavior that does not involve robot base movement, as shown in **Fig. 6**.

3.3. Refining the Communication Rate

We propose a method for refining the communication rate based on the existing architecture and hardware capabilities. Both factors are critical to determine an efficient frequency in the flow of information through the channels of communication between CPUs.

Overall, the programs created for the robots depend on the perception of the surrounding environment to activate behaviors. This perception is done by using several sensors, including visual sensors such as cameras, orientation sensors such as inertial measurement units (IMUs), and touch or force sensors. With the information received, the execution command is decided and sent to the actuators on the robots.

In the proposed algorithm, "subscribers" and "publishers" are created. Subscribers receive sensor data every time that the sensors send new information, and, as a counterpart, publishers will send the robotic commands to the actuators placed on the robots.

From the input information, the algorithm can extract the rate at which subscribers were receiving data from the sensors as well as the rate at which the publishers were sending the commands to the robots.

In this study, it is assumed that we do not have access to changing the system architecture. Therefore, we focus strictly on the rate at which the main CPU has direct control, which is through publishers. The maximum rate at which robotic commands sent to the CPUs controlling the actuators can be calculated according to the event assigned for each behavior. Events are triggered by changes reported by the sensors, and each event may depend on specific sensors. The maximum rate at which publishers can send commands will be governed by the rate at which the sensors do not have delays in reporting values.

In this refining process, we aim to tune the rate of sending robotic commands, such that the task execution time is reduced. However, because of variables, such as the type of architecture, CPUs used, and program delays, the rate of data transmission from subscribers to the main CPU may be affected differently for each behavior, and the rate for sending robotic commands from the maim CPU to the actuators is modified.

Based on this concept, we propose a process to obtain the efficient rate for publishers, as shown in **Fig. 5**.

In Step 1, the rate recorded for each of the sensors by subscriber "**S**" is extracted, along with the current rate at which commands were sent by the publisher "**P**."

In Step 2, the sensor used as a trigger for the event "*n*" (Behavior 1) is detected, and a test for finding the efficient rate for the publisher begins. With increasing the frequency of **P**, we check whether the data from **S** arrive without delay (less than a certain small threshold value); if there is no delay, the increments to the rate **P** continue until the time that **S** reports a delay (larger than the threshold value). Next, the **P** rate value, before the last increment, is assigned.

In Step 3, if the behavior involves movement of the robot base, the speed of the robot base is adjusted according to the percentage rate at which the **P** rate was modified.

Finally, in Step 4, we check whether the behavior analyzed at present is the last or not. If not, the algorithm returns to Step 1 to analyze the next behavior. Otherwise, the refinement process is finished.

4. Experiments

In this section, we present the experiments and results to demonstrate the validity and applicability of the proposed refining method.

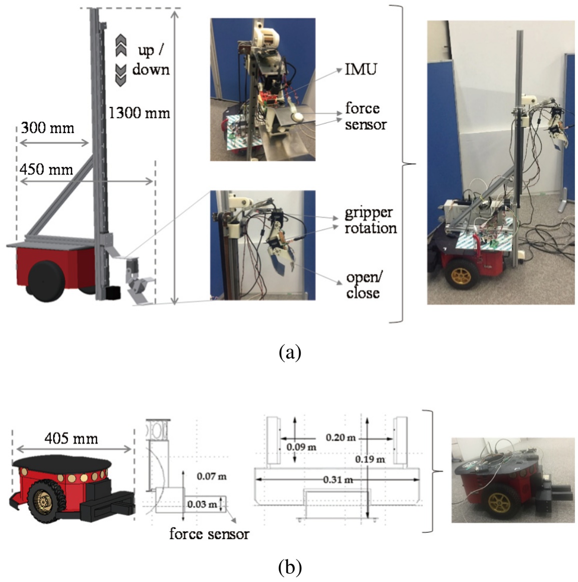


Fig. 7. Dimensions and sensors of the mobile robots used in the experiments: (a) lift gripper and (b) parallel gripper.

Two small robots are used to perform the tasks, equipped with various sensors. We evaluate the improvement in the execution time after the refining process.

4.1. Experimental Setup

4.1.1. Hardware Implementation

As the experimental apparatus, the Pioneer 3 mobile robot was used. Each of the two robots had a tool to manipulate objects, and sensors such as force sensors and IMU sensors were used, the data obtained by which was processed by a CPU. **Fig. 7** shows the robots alongside their characteristics.

We describe three categories in **Table 1**, explained in Section 2.1 and given in detail below:

Category I:

Robots use different sensors to perceive real worlds. Each sensor collects the raw data and sends them to respective sub-CPUs. Following the requirements in the state machine, robotic commands are sent from sub-CPUs to each actuator. **Fig. 8(a)** shows the described architecture.

Category II:

Several different sensors use the same sub-CPU, which means that several sensors share the sub-CPUs. Similarly, several actuators share the sub-CPUs. **Fig. 8(b)** shows the described architecture.

Category III:

All the sensors and actuators use the same sub-CPU. **Fig. 8(c)** shows the described architecture.

4.1.2. Evaluated Tasks

Four kinds of task are evaluated with three different objects. **Fig. 9** shows the initial and goal configuration of the objects in each task.

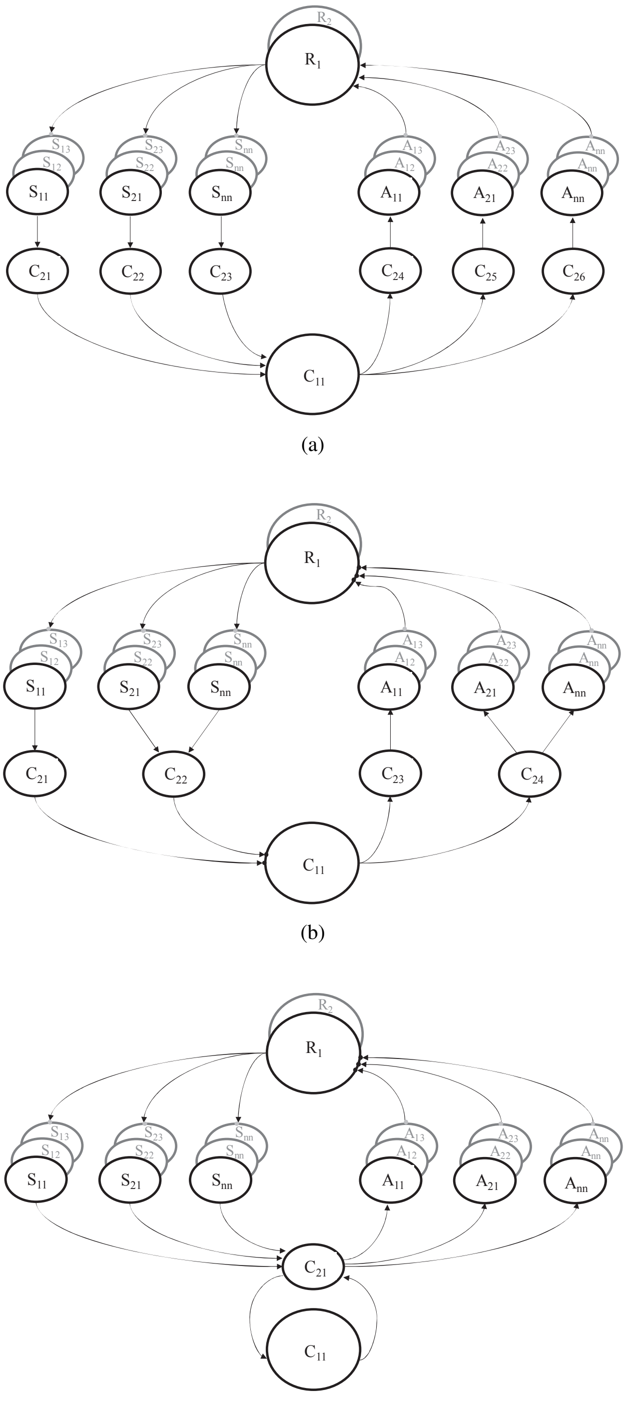


Fig. 8. Categories of system architecture: (a) Category I, (b) Category II, and (c) Category III.

4.1.3. Experimental Details

For each of the tasks, robots were located at a predefined distance from the objects. Each experiment was performed once.

Because the processes for evaluating the four tasks are similar, hereafter we use Task 4 as a reference task, the most complex one, to describe details of the experiments. **Fig. 10** shows the initial setting in the experiment.

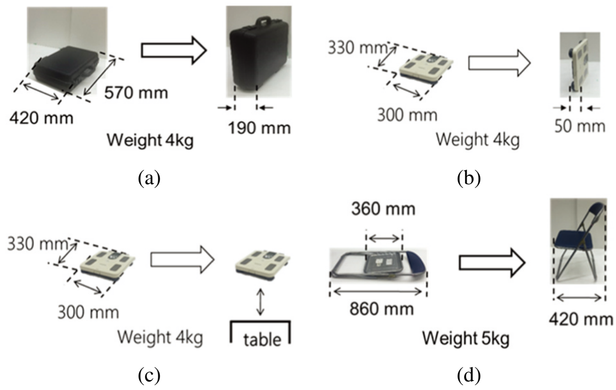


Fig. 9. Description of initial and goal configuration of the objects in the tasks. (a) Task 1: Flip the briefcase lying on the floor, (b) Task 2: Flip the weighing scale lying on the floor, (c) Task 3: Lift the weighing scale lying on a table, (d) Task 4: Open the folding chair lying on the floor.

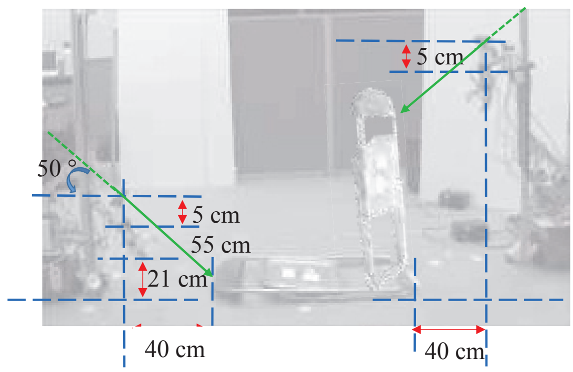


Fig. 10. Initial setting for Task 4.

4.2. Experimental Results

The refining process results and details for each of the steps inside the process regarding Task 4 are described below, and the results of the other three tasks are discussed in Subsection 4.3.

First, for task realization, the robots execute input state machines. **Fig. 11** shows these results. The state machine and the data obtained from the state such as the total time to execute the task and the execution time for each behavior, are shown.

Sensor values were used as triggers to transit from one behavior to another. The sensor configuration was in Category II, meaning that they shared the same CPU; the communication channel transmitted data to the main CPU at a rate of 200 Hz, while the rate detected for sending robotic commands was 10 Hz. The initial robot speed was set as 1.05×10^{-2} m/s.

With the acquired data, the first step of the refining process involved modifying the trajectory for behaviors. The modification of the trajectory was carried out by executing behaviors in parallel instead of step by step. This rule was applied to behaviors performed before the robots began to manipulate the object. In the case of Task 4, modification

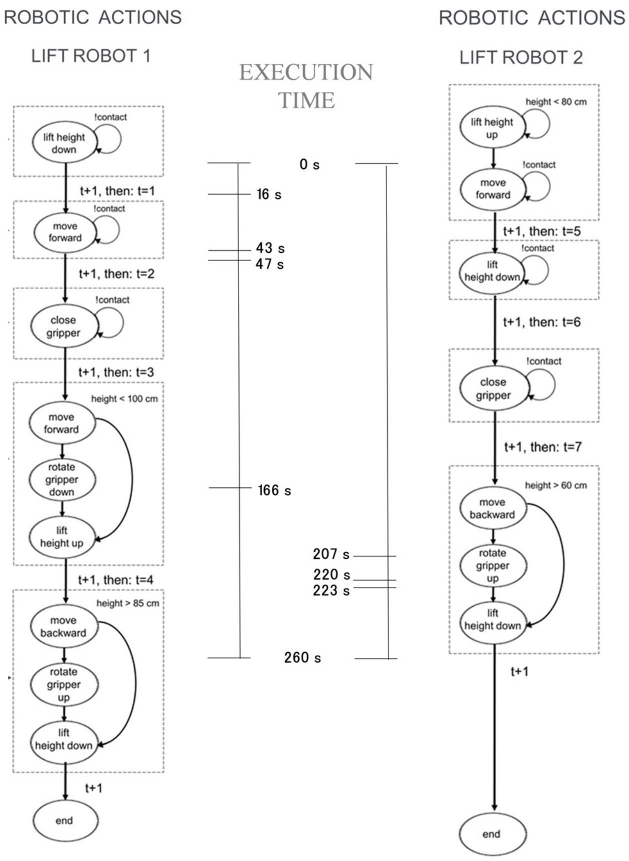


Fig. 11. State machines and initial execution time for Task 4.

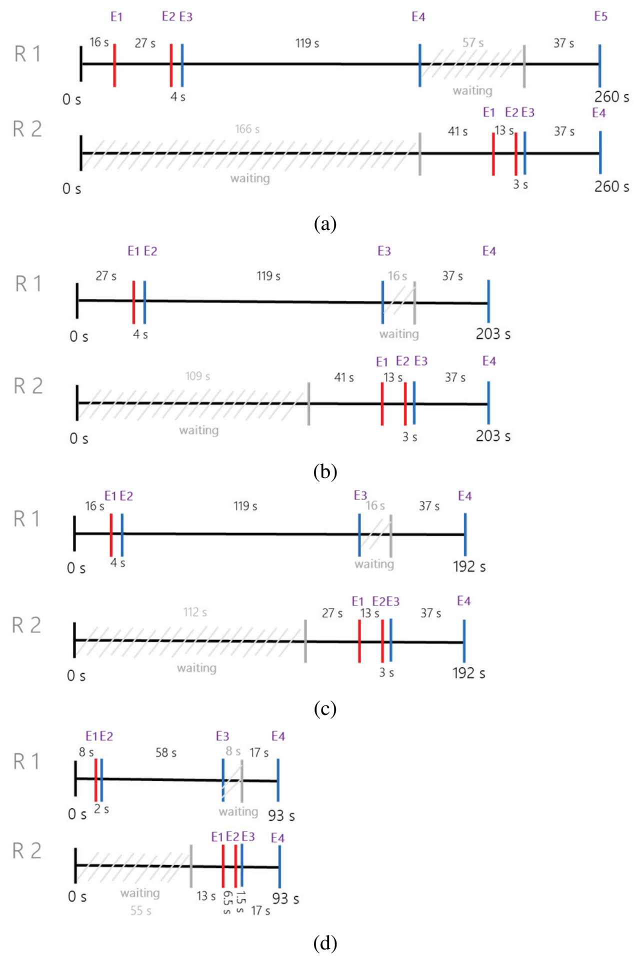


Fig. 12. Refinement process of the trajectories in Task 4: (a) original trajectories, (b) trajectory modification, (c) speed adjustment, and (d) refinement of communication rate.

was only applied to Robot 1 that had two behaviors assigned prior to pull-out behavior, unlike Robot 2, which has only one behavior prior to the pull-out behavior.

By changing the trajectory and by adjusting the speed of the robot base, the robots can perform the task more efficiently.

Subsequently, the refining step, in which the maximum communication rate for sending robotic commands according to the current system architecture was determined, and a new adjustment for the speed was applied, was carried out. The appropriate communication rate was found to be 23 Hz, and it was applied to both robots.

With the communication rate determined, as well as the robot speed adjusted, the task was executed again, and the result was an improvement in the execution time of 64.2% compared with the time recorded when using the input state machine.

More concrete results are shown in **Fig. 12**. Here, the task time is 260 s for the original trajectory, shown in **Fig. 12(a)**. With the modification in trajectory, the time became 203 s (parallelization of downward movement of the arm section in Robot 1, shown as R1, and forward movement of the mobile section in Robot 1) as shown in **Fig. 12(b)**. It became 192 s with the speed adjustment (the quicker forward movement of the mobile section in Robot 1), as shown in **Fig. 12(c)**. Last, the total time was reduced to 93 s with the refinement of communication rate (more-frequent sensor input from the sensors to the CPU and less-frequent output to the actuator from the CPU) – see **Fig. 12(d)**. As for other tasks, see [a].

4.3. Discussion

In this section, the results of the above described evaluation are discussed. By modifying the trajectory of the robots and by adjusting the rate at which the robotic commands were sent, it was possible to refine the programs for each of the four tasks.

In the first step of the refinement, although the beginning of behaviors was adjusted to be the same after the parallelization process, the assigned events were maintained to indicate the completion of the behavior, with the exception that the event of the absorbed behavior no longer determined whether the robot continued with the next behavior or not, as shown in **Fig. 13**. Furthermore, the speed of the robots was adjusted to improve the execution time.

In the second step of refinement, a balance for the communication rate according to the system architecture was found in a way that the main CPU could send robotic commands to the robots without delaying the arrival of data

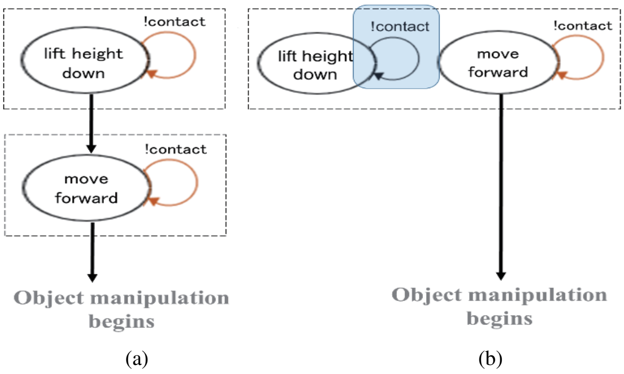


Fig. 13. Behaviors being executed in parallel. Behaviors assigned to Robot 1 before manipulation in Task 4, (a) input state machine and (b) trajectory-modified state machine.

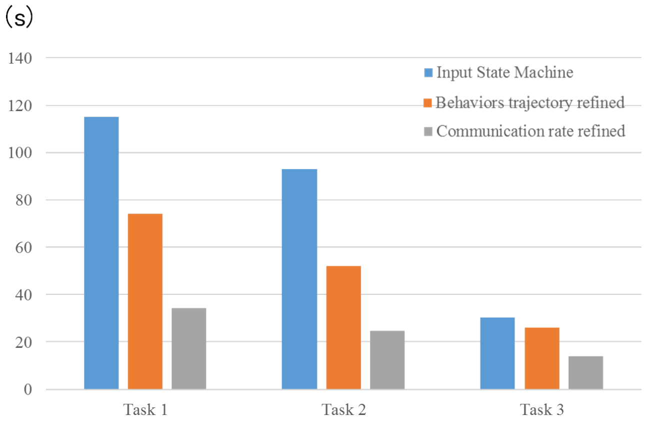


Fig. 14. Execution time of the robots before and after applying refining process for Tasks 1, 2, and 3.

from the sensors.

The results shown in **Fig. 12** indicate that the execution time for Task 4 was reduced by 64.2%, without any accident caused by this refinement due to delays in data sent by the robot through subscribers.

Similarly, the proposed refinement method was applied for the first three tasks evaluated, and the results concerning the task execution time achieved after each phase are shown in **Fig. 14**. Through the proposed method, the reduction in execution times of Tasks 1 to 3 was 70.3%, 73.6%, and 54.2%, respectively. These values are significant, and the validity of the proposed method was demonstrated by the four experiments.

In this study, we focused on tasks that were executed by making use of as many as two robots for manipulating objects. However, the presented approach theoretically can be extended to systems including more robots.

5. Conclusion and Future Work

We proposed a refinement method for improving efficiency in terms of task execution time for two robots that use programs based on state machines to perform a task. We focused on two points in the refining process of the behaviors to be executed by the robots: modification of the trajectory and balancing the rate of communication for sending robotic commands.

In the evaluations, four test tasks, including flipping a briefcase, flipping a weighing scale, lifting a weighing scale, and opening a folding chair, were used. The results demonstrated an improvement in the task execution time of between 54.2% and 73.6%.

The application areas of the proposed algorithm are rather diverse. The method is applicable to motion planning/generation problems of loosely coupled cooperative robots in industrial and service fields [30]. Further, it can generate smooth movement of robots with reduction of motion and communication loss. The authors believe that the proposed method pertains to Level 6 (system/subsystem model or prototype demonstration in a relevant environment) in the Technology Readiness Level (TRL) [31].

Our future work will involve improving the robustness of the system by predicting the occurrence of the unexpected events, such that the system can react to them, including sensor failures. The extension to multiple robot systems with more than two robots is also an aspect to be addressed in the future.

Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Number 26120006.

References:

[1] K. Yamazaki, T. Tsubouchi, and M. Tomono, "Furniture Model Creation through Direct Teaching to a Mobile Robot," J. Robot. Mechatron., Vol.20, No.2, pp. 213-220, 2008.
[2] M. H. Korayem and H. Ghariblu, "Analysis of Wheeled Mobile Flexible Manipulator Dynamic Motions with Maximum Load Carrying Capacities," Robotics and Autonomous Systems, Vol.48, Nos.2-3, pp. 63-76.
[3] M. H. Korayem, A. Heidari, and A. Nikoobin, "Maximum Allowable Dynamic Load of Flexible Mobile Manipulators using Finite Element Approach," The Int. J. of Advanced Manufacturing Technology, Vol.36, No.9, pp. 1010-1021, 2008.
[4] M. H. Korayem, A. Nikoobin, and V. Azimirad, "Maximum Load Carrying Capacity of Mobile Manipulators: Optimal Control Approach," Robotica, Vol.27, No.1, pp. 147-159, 2009.
[5] M. H. Korayem, V. Azimirad, and A. Nikoobin, "Maximum Load-Carrying Capacity of Autonomous Mobile Manipulator in an Environment with Obstacle Considering Tip Over Stability," The Int. J. of Advanced Manufacturing Technology, Vol.46, No.5, pp. 811-829, 2010.
[6] L. E. Parker, "ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation," IEEE Trans. on Robotics and Automation, Vol.14, No.2, pp. 220-240, 1998.
[7] T. Sakuyama, J. D. Figueroa Heredia, T. Ogata, T. Hara, and J. Ota, "Object Transportation by Two Mobile Robots with Hand Carts," Int. Scholarly Research Notices, Vol.2014, pp. 1-15, Article ID 684235, 2014.
[8] Z. Liu, H. Kamogawa, and J. Ota, "Fast Grasping of Unknown Objects Through Automatic Determination of the Required Number of Mobile Robots," Advanced Robotics, Vol.27, No.6, pp. 445-458, 2013.
[9] J. D. Figueroa Heredia, J. Rubrico, S. Shirafuji, and J. Ota, "Teaching Tasks to Multiple Small Robots by Classifying and Splitting a Human Example," J. Robot. Mechatron., Vol.29, No.2, pp. 419-433, 2017.
[10] J. Hasegawa, K. Kurihara, and N. Nishiuchi, "Collision Free Path Planning Method for Mobile Robot," Proc. 2002 IEEE Int. Conf. on Systems, Man and Cybernetics, Vol.3, pp. 6-9, 2002.

[11] C. Mudgal and J. Vassileva, "An Influence Diagram Model for Multi-agent Negotiation," Proc. 4th IEEE Int. Conf. on MultiAgent Systems, pp. 451-452, 2000.

[12] W. Verdonck, "Experimental Robot Payload Identification with Application to Dynamic Trajectory Compensation," Ph.D. Thesis, Katholieke University, 2004.

[13] C. Mericli, M. Veloso, and H. L. Akin, "Task Refinement for Autonomous Robots using Complementary Corrective Human Feedback," Int. J. of Advanced Robotic Systems, Vol.8, No.2, pp. 68-79, 2011.

[14] S. LaValle and J. Kuffner, "Randomized Kinodynamic Planning," Int. J. of Robotics Research, Vol.20, No.5, pp. 378-400, 2001.

[15] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Trans. on Robotics and Automation, Vol.12, No.4, pp. 566-580, 1996.

[16] B. Burns, "Exploiting Structure: A Guided Approach to Sampling-Based Robot Motion Planning," Ph.D. Thesis, University of Massachusetts, 2007.

[17] R. Alterovitz, S. Patil, and A. Derbakova, "Rapidly-exploring Roadmaps: Weighing Exploration vs. Refinement in Optimal Motion Planning," Proc. IEEE Int. Conf. on Robotics and Automation, pp. 3706-3712, 2011.

[18] T. Kroger and F. M. Wahl, "Online Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events," IEEE Trans. on Robotics, Vol.26, No.1, pp. 94-111, 2010.

[19] W. Khalil and E. Dombre, "Modeling, Identification & Control of Robots," Kogan Page, 2004.

[20] O. Dahl and L. Nielsen, "Torque-limited Path Following by Online Trajectory Time Scaling," IEEE Trans. on Robotics and Automation, Vol.6, No.5, pp. 554-561, 1990.

[21] R. H. Castain and R. P. Paul, "An On-Line Dynamic Trajectory Generator," The Int. J. of Robotics Research, Vol.3, No.1, pp. 68-72, 1984.

[22] S. Quinlan and O. Khatib, "Elastic Bands: Connecting Path Planning and Control," Proc. IEEE Int. Conf. on Robotics and Automation, pp. 802-807, 1993.

[23] M. Khatib, H. Jaouni, R. Chatila, and J. P. Laumond, "Dynamic Path Modification for Car-Like Nonholonomic Mobile Robots," Proc. IEEE Int. Conf. on Robotics and Automation, pp. 2920-2925, 1997.

[24] J. F. Canny, "The Complexity of Robot Motion Planning," MIT Press, 1988.

[25] P. A. O'Donnell and T. Lozano-Perez, "Deadlock-free and Collision-free Coordination of two Robot Manipulators," Proc. IEEE Int. Conf. on Robotics and Automation, pp. 484-489, 1989.

[26] K. G. Kant and S. W. Zucker, "Towards Efficient Trajectory Planning: Path Velocity Decomposition," Int. J. of Robotics Research, Vol.5, No.3, pp. 72-89, 1986.

[27] S. S. Chiddarwar and N. R. Babu, "Conflict Free Coordinated Path Planning for Multiple Robots using a Dynamic Path Modification Sequence," Robotics and Autonomous Systems, Vol.59, Nos.7-8, pp. 508-518, 2011.

[28] S. Koenig and M. Likhachev, "Incremental A*," Advances in Neural Information Processing Systems, Vol.14, pp. 1539-1546, 2002.

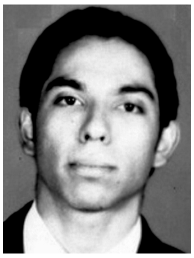
[29] P. Maes, "Situated Agents can have Goals," Robotics and Autonomous Systems, Vol.6, Nos.1-2, pp. 49-70, 1990.

[30] G. Demesure, M. Defoort, A. Bekrar, D. Trentesaux, and M. Djemaï, "Cooperation Mechanisms in Multi-agent Robotic Systems and Their Use in Distributed Manufacturing Control: Issues and literature review," Proc. 40th Annual Conf. of the IEEE Industrial Electronics Society (IECON 2014), pp. 2538-2543, 2014.

[31] J. Mankins, "Technology Readiness Levels – A White Paper," Advanced Concepts Office, Office of Space Access and Technology, National Aeronautics and Space Administration (NASA), 1995.

Supporting Online Materials:

[a] Appendix
http://otalab.race.u-tokyo.ac.jp/publication/appendix/
[Accessed April 13, 2018]



Name:
Jorge David Figueroa Heredia

Affiliation:
Department of Precision Engineering, School of Engineering, The University of Tokyo

Address:
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan

Brief Biographical History:
2005-2007 Hitachi Global Storage Technologies
2008- Training Program Electronics Engineering (JICA)
2011- Department of Precision Engineering, School of Engineering, The University of Tokyo

Main Works:
- T. Sakuyama, J. D. Figueroa Heredia, T. Ogata, T. Hara, and J. Ota, "Object transportation by two mobile robots with hand carts," Int. Scholarly Research Notices, 684235, pp. 1-15, 2014.

Membership in Academic Societies:
- The Robotics Society of Japan (RSJ)
- Association for Computing Machinery (ACM)

Name:
Shouhei Shirafuji

Affiliation:
Research into Artifacts, Center for Engineering, The University of Tokyo

Address:
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan

Brief Biographical History:
2014-2015 Osaka University
2015- The University of Tokyo

Main Works:
- "Novel frictional-locking-mechanism for a flat belt: Theory, mechanism, and validation," Mechanism and Machine Theory, Vol.116, pp. 371-382, 2017.
- "Designing Non-circular Pulleys to Realize Target Motion between Two Joints," IEEE/ASME Trans. on Mechatronics, Vol.22, No.1, pp. 487-497, 2016.
- "Development of a tendon-driven robotic finger for an anthropomorphic robotic hand," The Int. J. of Robotics Research, Vol.33, No.5, pp. 677-693, 2014.

Membership in Academic Societies:
- The Institute of Electrical and Electronics Engineers (IEEE)
- The Robotic Society of Japan (RSJ)

Name:
Hamdi M. Sahloul

Affiliation:
Department of Precision Engineering, Graduate School of Engineering, The University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656 Japan

Brief Biographical History:
2010- Embedded systems development, Prodigy Systems, Yemen
2014- M.D. Program, The University of Tokyo
2016- Ph.D. Program, The University of Tokyo

Main Works:
• H. M. Sahloul, J. D. Figueroa Heredia, S. Shirafuji, and J. Ota, "Foreground segmentation with efficient selection from ICP outliers in 3D scene," Robotics and Biomimetics (ROBIO), 2015 IEEE Int. Conf. on, pp. 1371-1376, 2015.
• J. D. Figueroa Heredia, H. Sahloul, and J. Ota, "Teaching Mobile Robots Using Custom-Made Tools by a Semi-Direct Method," J. Robot. Mechatron., Vol.28, No.2, pp. 242-254, 2016.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Jose Ildefonso U. Rubrico

Affiliation:
Research Fellow, Research into Artifacts, Center for Engineering (RACE), The University of Tokyo

Address:
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan

Brief Biographical History:
1996-2003 Faculty Member, University of the Philippines Diliman
2003-2006 Ph.D. Engineering, The University of Tokyo
2007-2015 Design Engineer, Schlumberger K.K.
2015- Research Fellow, The University of Tokyo

Main Works:
• J. I. U. Rubrico, T. Higashi, H. Tamura, and J. Ota, "Online rescheduling of multiple picking agents for warehouse management," Robotics and Computer-Integrated Manufacturing, Vol.27, No.1, pp. 62-71, 2011.
• J. I. U. Rubrico, T. Higashi, H. Tamura, M. Nikaido, and J. Ota, "A fast Scheduler for Multiagent in a warehouse," Int. J. Automation Technol., Vol.3, No.2, pp. 165-173, 2009.
• J. I. U. Rubrico, J. Ota, T. Higashi, and H. Tamura, "Metaheuristic scheduling of multiple picking agents for warehouse management," Industrial Robot: An Int. J., Vol.35, No.1, pp. 58-68, 2008.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)

Name:
Taiki Ogata

Affiliation:
Department of Computer Science, Tokyo Institute of Technology

Address:
G3-820, 4259 Nagatsuta, Midori-ku, Yokohama 226-8502, Japan

Brief Biographical History:
2009-2011 Project Researcher, The University of Tokyo
2011-2018 Assistant Professor, The University of Tokyo
2018- Specially Appointed Associate Professor, Tokyo Institute of Technology

Main Works:
• "Automated Design of Image Recognition in Capturing Environment," IEEJ Trans. on Electrical and Electronic Engineering, Vol.12, pp. S49-S55, 2017.

Membership in Academic Societies:
• The Japan Society for Precision Engineers (JSPE)
• The Society of Instrument and Control Engineers (SICE)
• Society for Serviceology (SfS)

Name:
Tatsunori Hara

Affiliation:
Associate Professor, Research into Artifacts, Center for Engineering (RACE), The University of Tokyo

Address:
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan

Brief Biographical History:
2009- The University of Tokyo

Main Works:
• "Modeling the transition to a provider-customer relationship in servitization for expansion of customer activity cycles," CIRP Annals – Manufacturing Technology, Vol.65, No.1, pp. 173-176, ISSN 1660-2773, 2016.
• "Design-of-use and design-in-use by customers in differentiating value creation," CIRP Annals – Manufacturing Technology, Vol.62, No.1, pp. 103-106, ISSN 1660-2773, 2013.
• "A CAD system for service innovation: integrated representation of function, service activity, and product behaviour," J. of Engineering Design, Vol.20, No.4, pp. 367-388, 2009.

Membership in Academic Societies:
• The International Academy for Production Engineering (CIRP)
• Society for Serviceology (SfS)
• The Society for Tourism Informatics
• The Japan Society of Mechanical Engineering (JSME)

Name:
Jun Ota

Affiliation:
Professor, Research into Artifacts, Center for Engineering (RACE), The University of Tokyo

Address:
5-1-5 Kashiwanoha, Kashiwa-shi, Chiba 277-8568, Japan

Brief Biographical History:
1989-1991 Nippon Steel Corporation
1991- The University of Tokyo
1996-1997 Visiting Scholar, Stanford University

Main Works:
- Z. Huang, C. Lin, M. Kanai-Pak, J. Maeda, Y. Kitajima, M. Nakamura, N. Kuwahara, T. Ogata, and J. Ota, "Robot patient design to simulate various patients for transfer training," IEEE/ASME Trans. on Mechatronics, Vol.22, No.5, pp. 2079-2090, doi: 10.1109/TMECH.2017.2730848, 2017.
- P. Jiang, R. Chiba, K. Takakusaki, and J. Ota, "Generation of the human biped stance by a neural controller able to compensate neurological time delay," PLoS ONE, Vol.11, No.9, e0163212, doi: 10.1371/journal.pone.0163212, 2016.
- Y. Huang, R. Chiba, T. Arai, T. Ueyama, and J. Ota, "Robust multi-robot coordination in pick-and-place tasks based on part-dispatching rules," Robotics and Autonomous Systems, Vol.64, pp. 70-83, 2015.

Membership in Academic Societies:
- The Japan Society for Precision Engineers (JSPE)
- The Robotics Society of Japan (RSJ)
- The Society of Instrument and Control Engineers (SICE)
- The Japan Society of Mechanical Engineers (JSME)
- The Institute of Electrical and Electronics Engineers (IEEE)