https://doi.org/10.20965/jrm.2017.p0419

Paper:

Teaching Tasks to Multiple Small Robots by Classifying and Splitting a Human Example

Jorge David Figueroa Heredia*, **Jose Ildefonso U. Rubrico****, **Shouhei Shirafuji****, **and Jun Ota****

*Department of Precision Engineering, School of Engineering, The University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo 113-8656, Japan
E-mail: figueroa@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: {rubrico, shirafuji, ota}@race.u-tokyo.ac.jp
[Received June 22, 2016; accepted December 25, 2016]

In this study, we present a novel framework to address the problem of teaching manipulation tasks performed by a single human to a set of multiple small robots in a short period. First, we focused on classifying the manipulation style used during a human-performed task. An allocator process is proposed to determine the type and number of robots to be taught based on the capabilities of available robots. Then, according to the detected task requirements, robot behaviors are generated to create robot programs by splitting human demonstration data. Small robots were used to evaluate our approach in four defined tasks that were taught by a single human. Experiments demonstrated the efficiency of the method to classify and judge whether the division of a task is necessary or not. Moreover, robot programs were generated for manipulating selected objects either individually or in a cooperative manner.

Keywords: teaching multiple robots, human-robot interaction, cooperative manipulation

1. Introduction

1.1. Background

Over the past 20 years, many researchers have made efforts to solve problems by making use of multiple robots [1].

There is a wide variety of studied topics in this field demonstrating the benefits of using a multi-robot system. Examples of these studies include the cooperation between robots to transport objects by pushing [2, 3], carrying [4, 5], or even using tools such as ropes [6] or carts [7], to move objects. Reducing the time to complete tasks by increasing the area of coverage and the range of operation (e.g., motion planning [8, 9]) and overcoming the gap between the capabilities of humans and robots are some of the benefits evaluated and discussed in studies where a multi-robot system has been used.

To execute any task, robots need to get information on how to perform the task [10, 11], which may be achieved through computer algorithms. In cases where robots are required to perform a fixed number of tasks repeatedly, programming the tasks in advance is the approach to follow, as in the case of the studies mentioned previously.

Transferring information to robots through a human example is an effective method, as human experience is used. However, teaching robots is difficult, particularly when multiple small robots are involved.

The following are some of the challenges faced by the robotics community in multi-robot teaching:

- The traditional method of teaching multiple robots one by one is time consuming.
  - How to teach multiple robots quickly and efficiently?
- One small robot may not be able to replicate a task taught by a human.
  - When to divide and how to decide the type and the number of robots to be used to perform a task?

Teaching robots has been a sought-after research field in recent years. An important approach in this field, which forms the focus of this study, is one that uses humans to perform the teaching, which is also called learning from demonstration [12].

This approach began to be implemented with satisfactory results in systems involving a single robot learner. There are methods that allow full-body imitation for robots [13, 14], teaching through gestures [15, 16], studies that focus on tasks to assist humans in homes or offices [17, 18], etc.

It is indispensable to take advantage of multiple robots in tasks that require cooperation and a combination of abilities (e.g., [19]). Exploring approaches that allow a single human to teach/transmit information quickly to multiple small robots by example is a field in which it is necessary to place more effort.

A multi-robot system using information obtained during the demonstration of a task by a human was proposed in the study of Maeda, Ishido, Kikuchi, and Arai [20]. In their study, a view-based system received information during the transportation of an object by a human from point A to point B. The data was then analyzed, and in the case when the range of operation of one robot could not complete the transportation, a second robot with the same characteristics was responsible for continuing the transportation of the object to point B. However, they did not consider the simultaneous manipulation of the object or parameters such as force limitations or decisions on the type of robot to use.

Chernova and Veloso [21] proposed a study in which a single human taught individual rules to multiple robots through a GUI, at the same time. Each robot performed the specific assignment based on policies transmitted by a human; their teaching process consisted of several task demonstrations, including variations thereof. As a result, robots could properly decide which action to take during the execution of a common task. However, the robots used were selected in advance, and teaching tasks involving the simultaneous physical manipulation of objects from robots was not considered.

In this study, we focused on fixed tasks rather than tasks with variations. The reason is that we focused on how to categorize tasks taught by a human, such that it would be possible to determine the type and the number of robots required to be taught to perform manipulation tasks.

To our knowledge, previous studies have not yet presented a system that considers human example data to teach multiple small robots how to perform manipulation tasks. In particular, this study proposes an approach that considers both the content of the task and the capabilities of the available robots to decide how to divide the task into multiple robots.

1.2. Objective of this Study

The research question to address in this study is how a single human can quickly teach single tasks to multiple robots. The present study was conducted with the objective to allow a single human teach multiple robots how to perform fixed manipulation tasks.

The challenges addressed in this study pertain to whether a task taught by a human can be categorized in order to decide on the type and the number of robots to be used, and how we can transfer human actions into robotic actions while coordinating the cooperation of multiple robots.

To provide a solution to the above challenges, the following approaches were adopted:

- Allocator: Based on the task characteristics and capabilities of the available robots, we first classified the task according to the manipulation style used. Then, we used heuristic rules to decide on the number and the type of robots to be taught.

- Mapping: This involves teaching robots how to perform human actions. Robot programs will be made up of a sequence of basic robot behaviors controlled by events, while considering the timing for the cooperation between robots. The basic behaviors are obtained through splitting the procedure of human task demonstration.

The design of the allocator with the task classification and that of the mapping with the splitting procedures are the main contributions of this study. The proposed system was verified with tasks involving the handling of objects that can be handled by a human, such as briefcases, scales, and folding chairs.

2. Problem Statement

A human can perform complex movements to execute manipulation tasks; owing to its physical limitations, one small robot may not be able to perform the same task. The use of multiple robots may help overcome the existing gap between the physical limitations of a single human and those of small robots.

By teaching, humans should be able to use their experience to transmit information on how to perform tasks to multiple robots. The arising problem is that it is very difficult to transmit human actions to multiple robots [22], particularly if we wish to quickly transmit our experience to robots that have lower capabilities compared to humans.

During the teaching process, a manipulation task may be divided among multiple robots, which should then cooperate with each other efficiently to execute a task performed by a single human.

Addressing this problem is important because the time and effort to transmit task information to multiple robots would be reduced; this would allow humans to use the same set of robots for a diversity of tasks.

In this study, we propose an approach in which a human can teach simultaneous manipulation tasks to multiple robots by using a novel approach involving human-robot interaction (HRI [23]).

- Our goal is to quickly generate programs for multiple robots, using as input data the motions and the force generated through a teaching tool used by a human during the task demonstration.

In the proposed study, we considered the following assumptions:

- There are no obstacles between the objects and the robots.

- The position of the robots with reference to the object is known.

- Robots have lower capabilities compared to humans, and they have at least one manipulation tool.

- The objects used have no separable parts.

- The capabilities of the robots to be used are introduced to the system in advance.

Table 1. Task classification: manipulation styles considered.

Number of Hands	One	One	One	Two	Two	Two
Manipulation Style	Lift / Pull	Flip and non-slip	Flip and slip	Lift / Pull	Flip and non-slip	Flip and slip

To generate programs for multiple small robots to perform a particular manipulation task, we propose to first classify the type of task that is intended to be taught; based on this, the type and the number of robots to be taught can be determined.

Classification of tasks: We started from the assumption that the tasks performed by humans involve the manipulation of objects using the hands. In this study, we investigated the manipulation style as a key factor for classifying the task. In particular, we focused on manipulations involving lifting, pulling, and flipping objects because many object-manipulation tasks include these basic actions; most of the tools used by the robots to manipulate objects can perform these actions as well.

In addition to considering whether the task was performed by a human using one or two hands, we also considered the possibility of the object slipping owing to floor conditions during a flipping manipulation. The style of manipulation will be categorized into one of the six types listed in **Table 1**.

Type and number of robots: Considering the manipulation style detected, information related to the physical relationship between the object and the environment was obtained. Thus, the minimum requirements to determine the type of robots could be determined. Among the available robots, a single robot may not be able to replicate certain physical requirements, such as force or motion trajectory. In such cases, the division of the task and its distribution to multiple robots are essential for the completion of the manipulation task.

3. Proposed Method

3.1. System Overview

The proposed teaching method is composed of two main steps: the allocation process and the transferring of human actions to robotic actions. **Fig. 1** shows an overview of the system.

In the allocation step, we aim to classify the type of task to be taught, and based on that, to decide the type and number of robots to be used.

First, we extracted information during the demonstration of the task by a human. We have chosen to design and use a tool from which we can obtain data directly from the hands.

Therefore, we opted for electronic gloves, also known as data gloves [24].

Using the data collected, the teaching process is guided through a set of graphical user interfaces (GUIs). The GUIs allow establishing a human-robot interaction, which enabled the system to obtain additional information from a human to classify the task. Finally, by using the collected data and a set of heuristic rules, the type and number of robots to be used was decided.

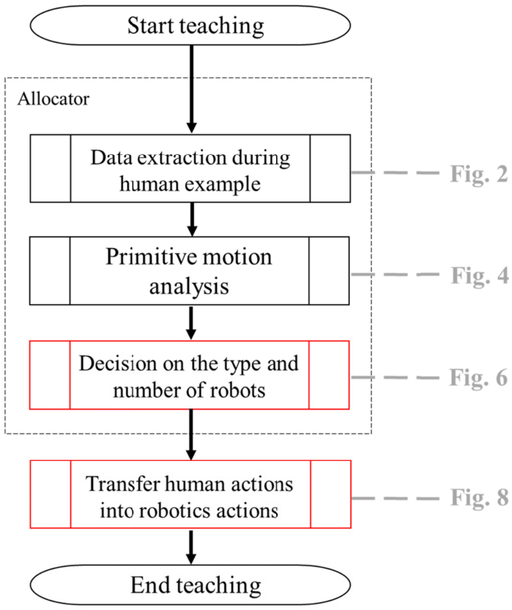


Fig. 1. Overview of the system: steps in the process of teaching.

In next step, the actions performed by a human and assigned to selected robots will be mapped to commands for robots. The process of performing human actions through robot behaviors is carried out using a GUI; once this process is completed, the robot program is generated.

3.2. Teaching Process

3.2.1. Allocation

The data we used for our analysis are based on the applied force as well as on the orientation and motion of the hands during the manipulation of objects. The subprocesses developed are described below.

Data Extraction During the Human Example

The developed prototype data gloves were integrated with force sensors [a], an inertial measurement unit (IMU) sensor [b], a Bluetooth sensor [c], and a microcontroller [d]. The sensor data were handled by the microcontroller and were then sent to the system via Bluetooth. Along with the sensor data, a video of the human performing the task was recorded as the teaching data.

The stored data contained several force points, global and local acceleration data of hand motions, as well as the orientation of the hands with respect to the world. The algorithm used for the data extraction is shown in **Fig. 2**.

In the first step of the algorithm, a human wears the data gloves and the sensors to start generating information, which is then processed and sent to the framework.

Using the data from the IMU sensor, the orientation and motion trajectory of the hands are calculated [25].

To start or stop the extraction of the teaching data, namely the video and sensor data, as well as to visually

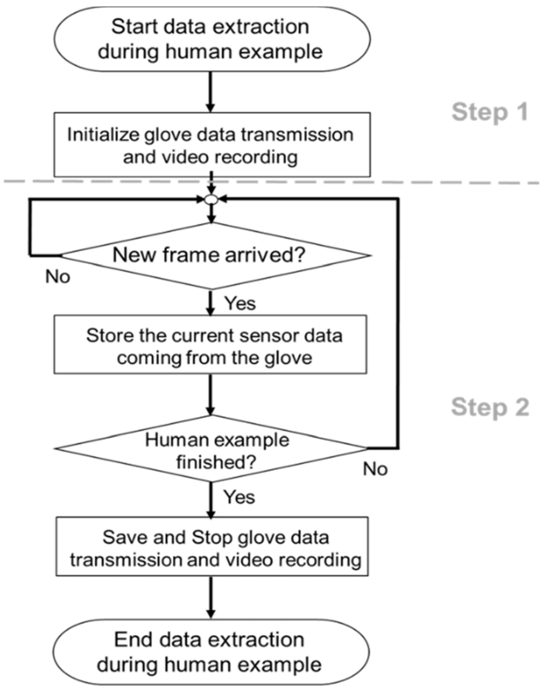


Fig. 2. Data extraction during a human example: flowchart of the teaching-data storage process.

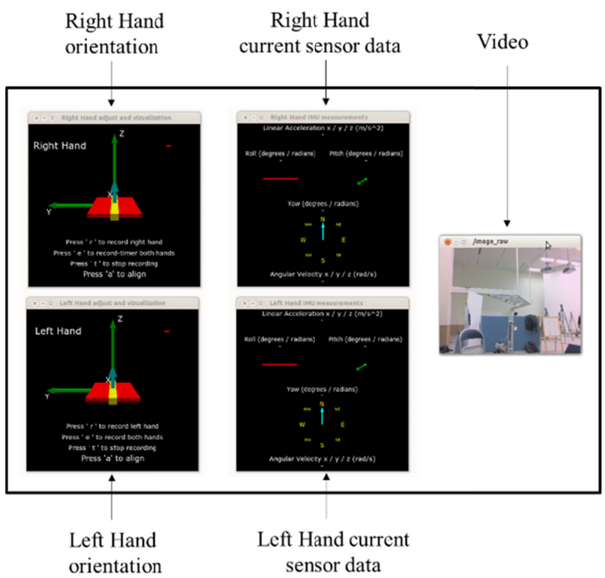


Fig. 3. GUI to support the human during data extraction: displays the orientation of the hands, and the data generated by each sensor.

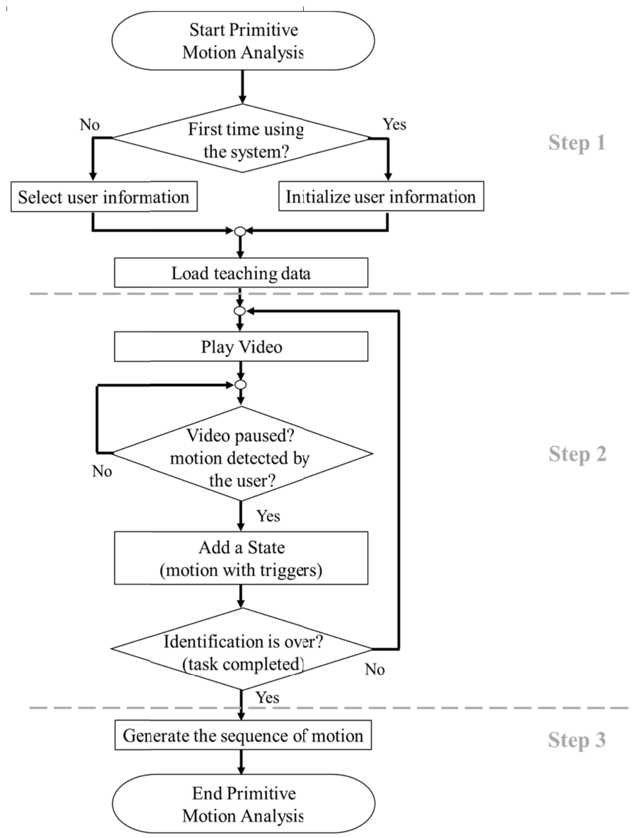


Fig. 4. Flowchart of the sub-process primitive motion analysis.

support the human, we included a GUI (based on [e]), which is shown in **Fig. 3**.

In step two, the human example demonstration as well as the storage of the received sensor data occur. To match the stored sensor data with the video, the recording frequency and the data generated by the gloves must be synchronized. Each time the system has a new frame, the latest sensor data corresponding to that time are stored as part of the teaching data, together with the timestamp of the video frame. This process is looped until the task demonstration is completed.

Primitive Motion Analysis

To start using the data generated and stored throughout the human example demonstration, we developed a GUI that guides the human to identify and define the sequence of primitive motions for each hand. The logic used is presented in the flowchart in **Fig. 4**.

The number of motions generated during the demonstration of the human example may be large, and many of these motions may not be essential to perform the task; this assumption is of significance to small robots, which do not have the same capabilities as humans.

Our intention is to allow humans to identify the important motions generated by the hands within the teaching data.

In the first step, the user information is fed as input, and the teaching data is loaded.

In the second step, the user begins teaching by playing back the video included in the teaching data, while at the same time, the matched sensor data are displayed to the user next to the video as illustrated in **Fig. 3**.

The moment the user visually identifies an action he/she deems important for the execution of the task; the user can pause the video and assign a name to store that particular primitive motion as a state. The information to be stored into a state is the data pertaining to sensor values: the applied force, a range of values pertaining to

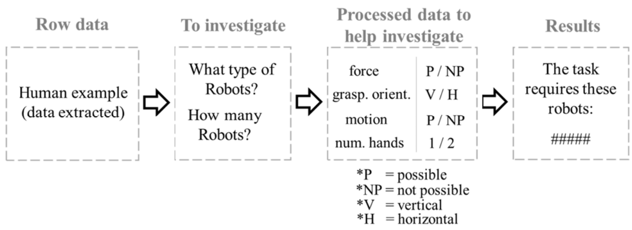


Fig. 5. Logic to decide the type and number of robots: extracted data is processed and used to decide on the robots to be taught.

hand orientation, and the global and local acceleration of the selected hand.

For a state to be valid, it is assumed that it contains information indicating that a force has been applied, meaning that the hand was either manipulating the object or making contact with the environment. A state may contain information on both hands or it can be assigned separately for each hand.

The user continues adding states (primitive motions with their triggers) until the human example included in the teaching data is completely analyzed.

Finally, in step three, a state list, which contains a sequence of the right-hand and the left-hand primitive motions detected during the human example demonstration, is generated as the output of the sub-process.

Decision on the Type and Number of Robots

The last sub-process in the allocator step helps decide the type and number of robots to be used. To accomplish this, a logic (**Fig. 5**) that uses the data extracted during a human example demonstration is described in this section.

A process involving task categorization and the use of heuristic rules is carried out. In our approach, we considered that the task fitted into one of the six styles of object manipulation, as described in Section 2.1.

To classify the manipulation style, additional information is obtained through user interaction with the system using the GUI. No more than two questions are asked to the human, and the answers are combined with the states detected in the primitive motion analysis; thus, the manipulation style is categorized.

The logic to decide the type and the number of robots is shown in **Fig. 6**. In step one, the state list previously created is loaded and displayed to the user via the GUI.

In step two, the manipulation style is determined. First, information is collected via the following questions to the user:

1) *Did you flip the object?*

2) *Did the object slip?*

Thus, the system will know if the object was either lifted/pulled (M. Style A), flipped and slipped (M. Style B), or flipped and not slipped (M. Style C).

We designed our algorithm with such a logic that in order to answer the questions, the human has to involve the states detected in the previous step, which contains the sensor data. Thus, based on the manipulation style de-

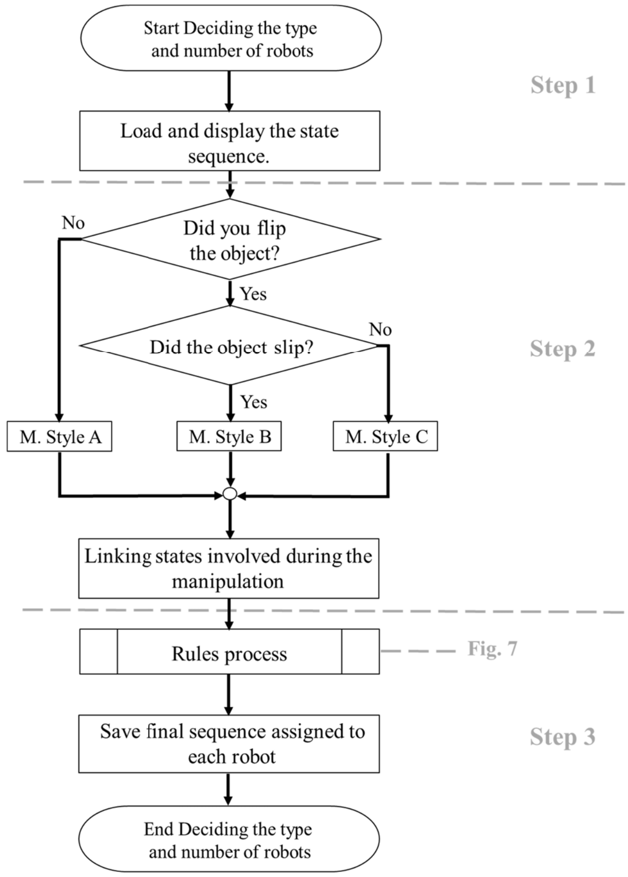


Fig. 6. Flowchart of process to decide the type and the number of robots.

tected and the information regarding the number of hands used as extracted from the states, the task is categorized.

In step three, a set of heuristic rules is applied by the system to decide on the robots to be used. The data obtained in the previous steps and the information regarding the hardware capabilities of the available robots is used by the framework to apply the rules, as shown in **Fig. 7**.

In our algorithm, the logic of the rules changes slightly depending on whether the task was categorized within the group where one hand was used for manipulating the object or within the group where two hands were used.

From the set of available robots, the system holds the information regarding the characteristics of the hardware mounted on the robots, particularly of the tools that robots use to manipulate objects. Such information includes the gripper type, the lift dimensions, the maximum grasping force, the maximum weight to lift, the degrees of freedom, and the mobility of the robot base.

From the sensor data stored in the teaching data, the system holds the information regarding the applied force, orientation of the hands, and motion trajectory of the hands.

From the categorization process, the system knows the manipulation style; in the case of the flip-and-slip manipulation, support will be required. During the categorization process, a relationship between the states involved

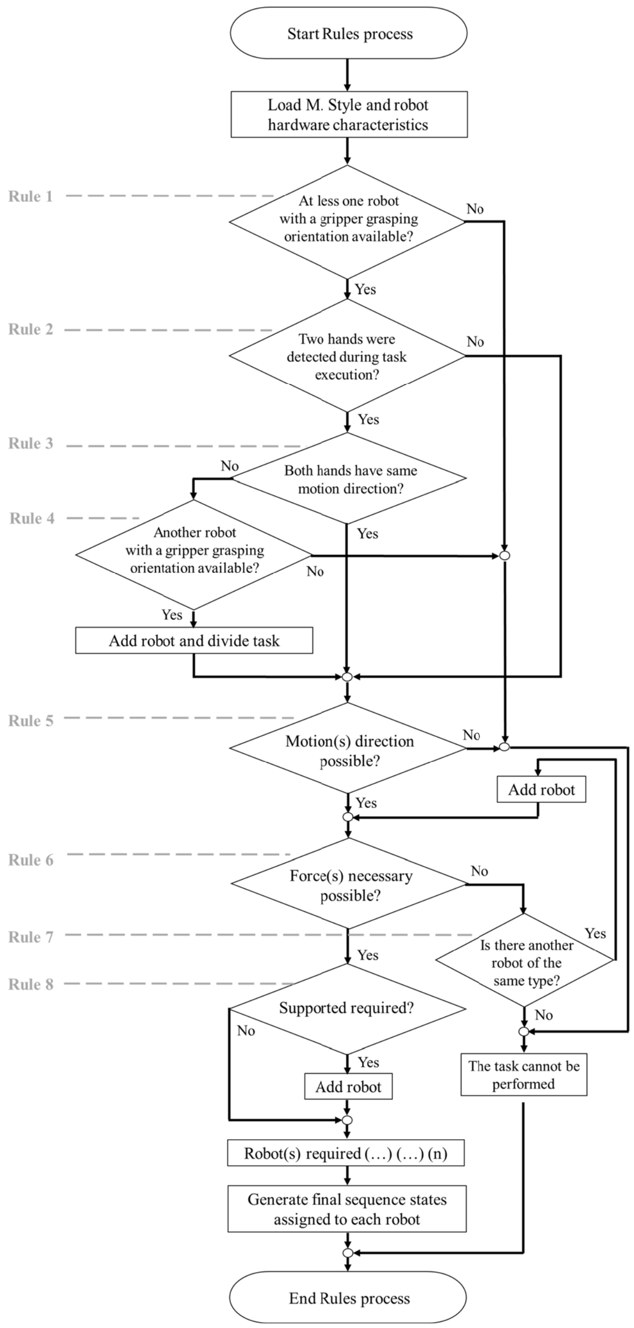


Fig. 7. Flowchart of the process to apply the heuristic rules. Sensor data information, robot hardware information and manipulation style are considered.

directly in the manipulation of the object was also established; thus, it was possible for the system to analyze the sensor data from a particular timestamp according to the rule requirements. Thus, the system was able to decide whether the task could be executed or not, and additionally, to decide if it could be performed by one robot or if it would be necessary to divide the task among multiple robots.

In the rules process, the system first loads the information corresponding to the detected manipulation style and the hardware capabilities of the robots. Then, the system starts applying the rules.

In the case when one hand is detected, the system checks if among the available robots there is one with the gripper type detected (rule one). Then, it checks if the candidate robot can replicate the motion direction (rule five). The system verifies if the force can be applied by the robot (rule six). If not, the system tries adding another robot with the same characteristics (rule seven). If the system could not find the ideal robots after the search, then the system announces to the user that the task cannot be executed.

In the case when two hands are detected, the rules are similar to the previous case, with the difference being that the system seeks to determine whether a robot can replicate what was performed with both hands (rule three). If it is not possible, the system tries to divide the task by finding another robot (rule four). If this is not possible, the task is declared not executable.

Finally, the system checks if the assistance of a robot is necessary owing to the slipping of the object reported by the human during the classification process (rule eight). The type and the number of robots are determined, and the states are assigned to each robot. The sequence of the states with a proper timing (based on the timestamp linked to the states) is generated, and the rules process ends.

The final output from the allocator step is a file containing the sequence of the states assigned to each robot.

3.2.2. Transfer of Human Actions into Robotic Actions

To teach the robots how to perform primitive motions, using the output from previous step, the system knows the robots and the states (primitive motions) they need to perform by splitting the teaching data into several units that correspond to the states. To finally create the program for the robots, it is necessary to teach the robots how to perform the current states by employing behaviors they can execute.

Mapping Commands to the Robot

In this step, primitive motions detected during the human example demonstration are mapped into robotics commands. Therefore, a GUI was created, and the applied logic is shown in **Fig. 8**.

In step one, the user selects a robot among those appointed for the task; consequently, the GUI displays the states assigned to that robot, along with the basic behaviors that the robot can perform (e.g., close/open gripper, move forward, etc.).

In step two, the mapping process is carried out. A state is selected, and the user begins adding behaviors with their respective events, which are used to indicate when to stop the behavior. The events are linked to the sensors mounted on the robot; this enables monitoring aspects such as distances, contact points, force measurements, etc. Added behaviors are expected to be executed in the order they were introduced; thus, priorities among the behaviors that constitute a state are included.

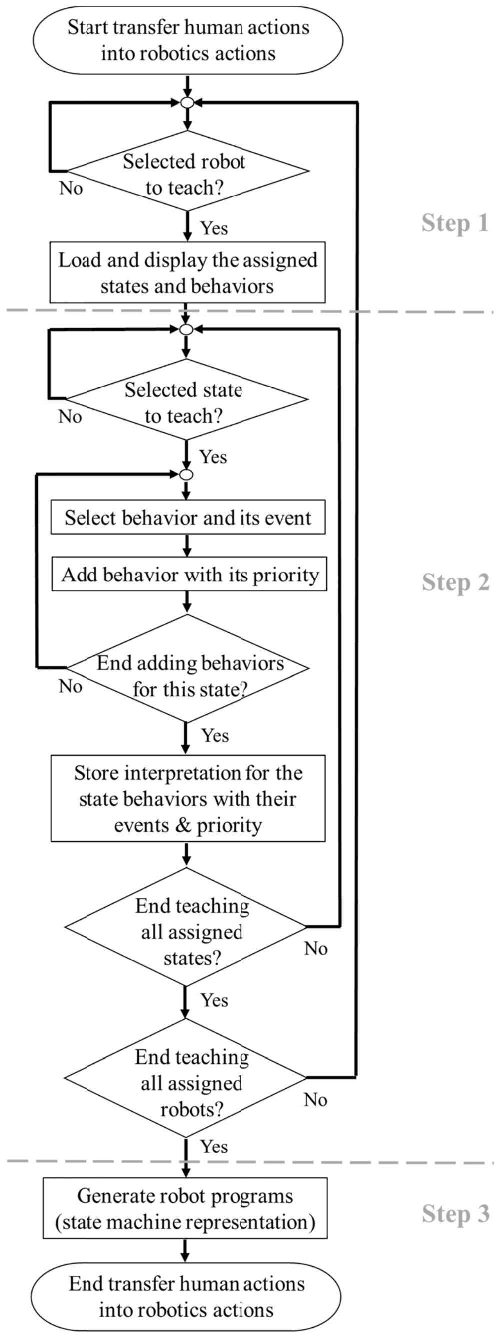


Fig. 8. Flowchart for the transfer of human actions into a robotics action process: mapping human actions into robotics actions.

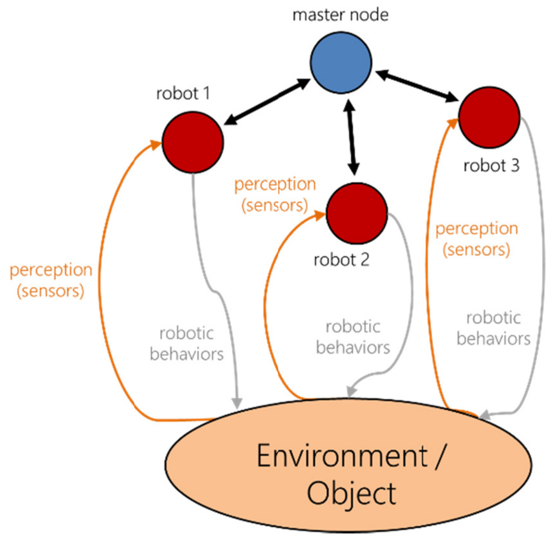


Fig. 9. Centralized system: reporting of the perception from environment and execution of robotic behaviors.

The user keeps repeating this procedure for all the states assigned to each of the robots.

Finally, in step three, the robot programs are created. The generated programs are represented by using state machines.

Robot Programs

The generated programs for robots already contain the timing in which the robotic actions have to be implemented to perform the taught task. For enabling robots to use the programs in a proper manner, the final step in our method is the implementation of a reliable communication protocol.

To maintain synchronization among robots, as well as a constant communication regarding the status while basic behaviors are being executed, a centralized system was created: a master node is used to manage the information coming from each robot assigned to the task via a wireless network.

The master node is responsible for authorizing the start of the execution for each of the states that make up the robot programs. To execute each of the behaviors that compose a state, the robot is constantly monitoring and reporting the status of its hardware via the response of its sensors. More specifically, checking values are assigned to the events.

Thus, we can describe each robot to be a primary node that is interconnected through the master node to all other primary nodes. To each of these primary nodes, several basic nodes are connected. Basic nodes represent the robot sensors and motor controllers, which consistently report their hardware changes and the current condition according to the contact with either the environment or the object, as shown in **Fig. 9**.

4. Experiments

In this section, we present the experiments and results to demonstrate the validity and applicability of the proposed framework. Three small robots were used for the evaluation of our approach in four defined tasks that were demonstrated by a single human.

We evaluated the time to generate the robot programs, the performance of the steps that integrate the teaching process, and the playback performance when the robots used the generated programs. At the end of the section, the discussion of the results is presented, followed by the conclusions of this study.

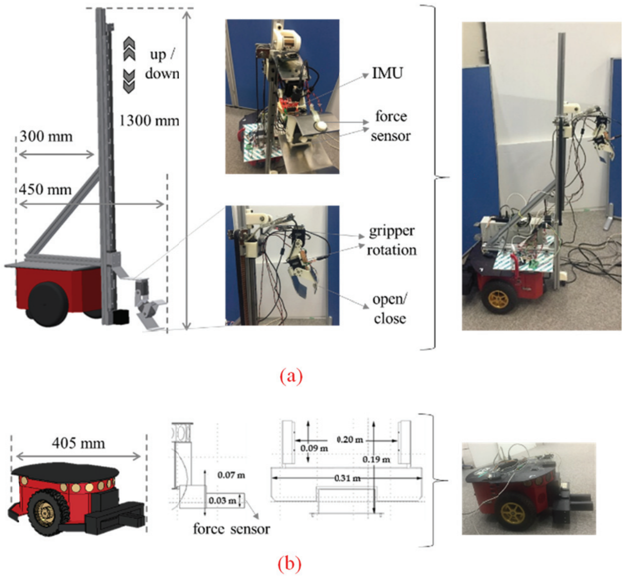


Fig. 10. (a) Lift robot: Pioneer 3 mobile robot, lift-gripper, dimensions, and sensors. (b) Parallel gripper robot: Pioneer 3 mobile robot, parallel gripper, dimensions, and sensors.

Table 2. Robots used: gripper characteristics.

Robots	Degree of Freedom (DoF)	Grasping Force	Max. weight to lift	Gripper-type (grasping orientation)
Lift Robot	3	up to 5kg (force)	3kg *lifting an object in a flat position	Vertical
Parallel Robot	2	up to 2kg (force)	2kg	Horizontal

4.1. Experimental Setup

4.1.1. Hardware Implementation

We used the Pioneer 3 mobile robot [f]; three small robots were available, and their hardware capabilities were used as input to our framework in advance. Each robot was fitted with a manipulation tool; we used two types of tools, which allowed the robots to have different capabilities. To monitor and control the status of the hardware in each robot, various sensors, such as force sensors [a] and IMU sensors [b], were utilized on the robots. **Fig. 10** shows the robots alongside their characteristics.

To differentiate the robots in this section, we will refer to them hereafter as the lift robot (there are two robots carrying this tool, **Fig. 10(a)**) and the parallel robot (there is one robot carrying this tool, **Fig. 10(b)**). Their characteristics are presented in **Table 2**.

During the experiment, the tasks performed by a human were recorded with a webcam at a frequency of 30 fps, while sensor data from the prototype data glove were sent through the Bluetooth device every 35 ms. **Fig. 11** shows the prototype data gloves created and used for the experiments.

The data gloves were designed such that the movement

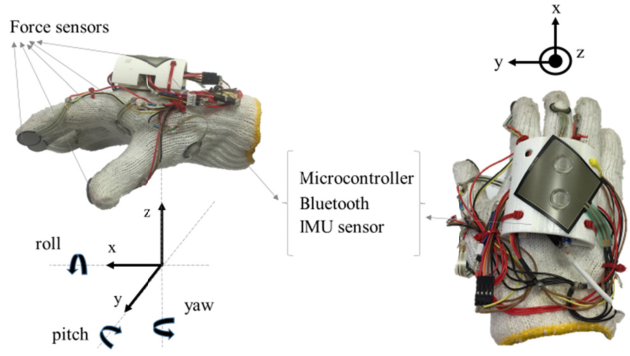


Fig. 11. Prototype data glove, sensor integration: 7 force sensors to detect contact with environment and measure grasping force, an IMU sensor to track hand orientation and motion direction, a microcontroller to process data, a Bluetooth to send the data.

of the fingers was restricted to force the human to understand the limitations of small robots slightly better and thus generate data that are more appropriate for teaching robots. The measurement error of the glove is approximately 30% at the most. Because the glove is utilized only to detect the hand movement to the left, the right, up, and down, this error is affordable.

4.1.2. Evaluated Tasks

Four fixed tasks were performed by a single human; we selected three real-world objects to manipulate, namely a folding chair, a plastic briefcase, and a weighing scale.

The characteristics of the objects and the goal for each task are shown in **Fig. 12**.

4.1.3. Experimental Details

In the experiments, we placed the object on the floor for tasks one, two, and four, whereas for task three, the object was placed on a table. This setting applies to both the human example and the small robots during the task execution. Object detection or path planning is not considered in our study, therefore, robots were placed in an initial position.

The tasks were selected with the intention of showing the different alternatives in which our approach behaves when generating programs for robots.

4.2. Experimental Results

For each of the tasks, a single human performed the demonstration, and the data generated through the sensors were used to complete the process of teaching.

For each of the four tasks, the total time to complete the teaching was recorded; the obtained results are shown in **Fig. 13**. The time taken to complete the tasks by the robots during the execution process is shown in **Fig. 14**.

The time required to generate robot programs was continuously recorded, except for the human example process. The time required to complete the allocation process, which involved the categorization of the task and the

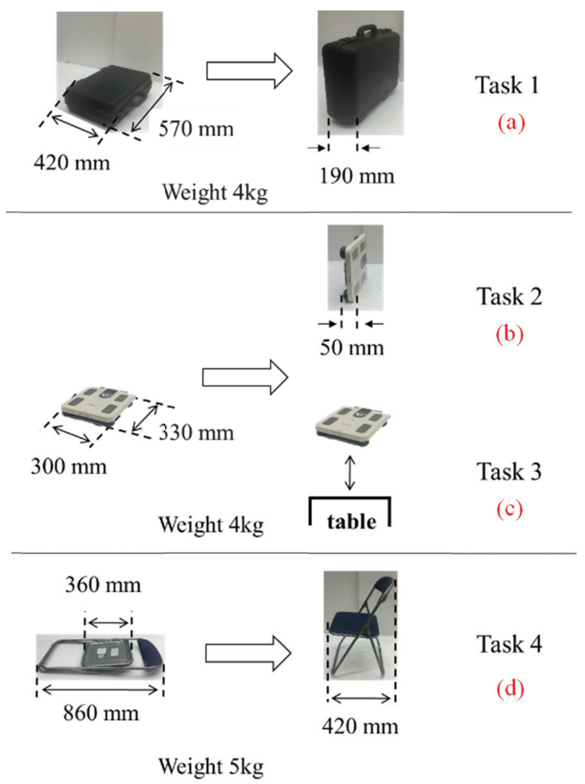


Fig. 12. Description of assessment task for each object. (a) Task 1: flip the briefcase lying on the floor, (b) Task 2: flip the scale lying on the floor, (c) Task 3: lift the scale lying on a table, and (d) open the folding chair lying on the floor.

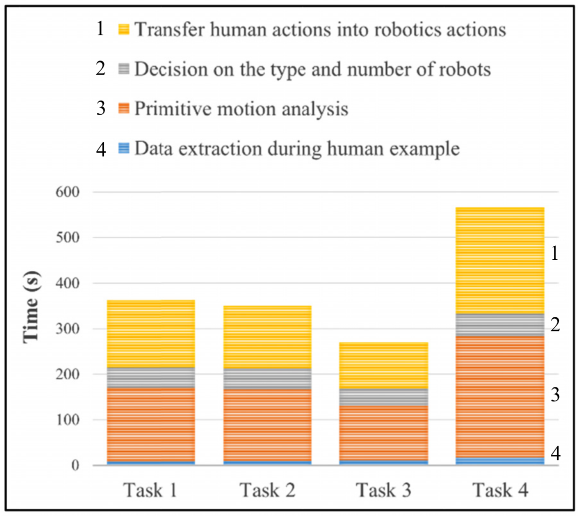


Fig. 13. Total time required during task execution by robots.

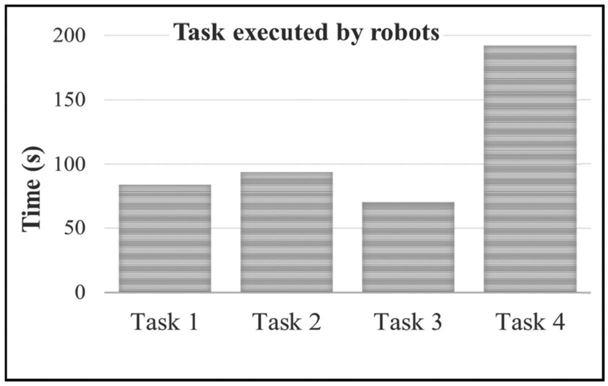


Fig. 14. IMU sensor data related to the orientation of hands during the execution of the tasks. This information is used to detect the type of gripper the robot should be mounted with.

decision on the type and the number of robots, was mostly allotted to the primitive motion analysis (states detection from human example). On the other hand, the data generation through the human example and the application of rules to make the decisions makes up for less of the total time.

The time required teaching how to perform human actions as robotic actions is related to the complexity of the task; in other words, it is related to the number of states to teach, rather than to the number of robots to be used.

The teaching process results are shown below; we selected task four (the most complex task evaluated) to describe the results systematically in detail, whereas the results from the other three tasks are discussed in the next section.

First, the data extraction for the task was conducted during the human example (sensor and video data). While using the data gloves, the user manipulated a folding chair that was lying on the floor. To open it, the user grabbed the chair, flipped it, and pulled it.

Using the extracted data, the primitive motion analysis was performed. By using the stored video, the states containing sensor data information for both hands were detected and generated; the names for the sensor data information are listed in **Table 3**.

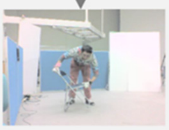
By using the generated states list, the type and number of robots were decided. First, the classification of the task was obtained. The user responses to the questions asked by the system using the GUI allowed us to detect that the style of manipulation for task four includes the flip and the non-slip with pull, using two hands.

Then, the system was ready to decide on the robots to be taught by applying the heuristic rules. The information used by the system was that generated by the sensors during the human example demonstration; more specifically, it is the sensor information that was linked to the states (motion primitives) during the classification task. Thus, the system focused on reviewing data at precise moments in the stored data time sequence.

For this evaluation, the force required to complete each of the defined tasks, together with the limits of the two types of robots, is shown in **Fig. 15**. Moreover, to detect the type of gripper required for the manipulation of the object, the data related to the orientation of the hands were determined from the IMU sensor data shown in **Fig. 16**.

Finally, the acceleration data from the IMU sensor were used along with orientation data to determine if the robots could replicate the motion trajectory; the data used are

Table 3. Primitive motion analysis using the data extracted during the human example demonstration: states generated for both hands.

Primitive motion analysis		Human example (stored video data)
States Right Hand	States Left Hand	
searching horizontal	Idle	
found	Keep State	
grasp	Keep State	
hold vertical	Keep State	
Keep State	searching vertical	
Keep State	found	
Keep State	grasp	
pull_out	pull_out	

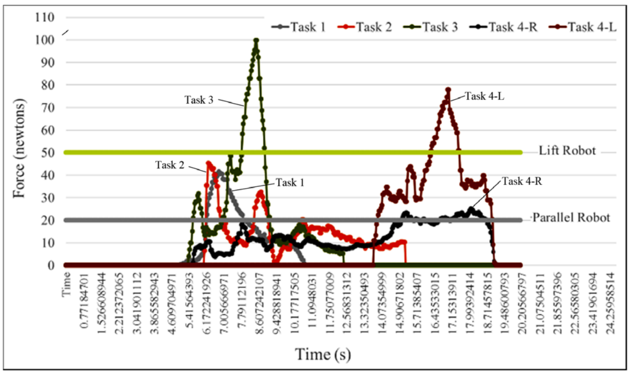


Fig. 15. Generated force data used during the human example for each of the four tasks along with the limits of the robots we are using.

shown in **Fig. 17**.

Below, the judging process of how the type and the number of robots were decided for task number four is explained, and it is followed by the robot program created through the process of mapping human actions into robotic actions.

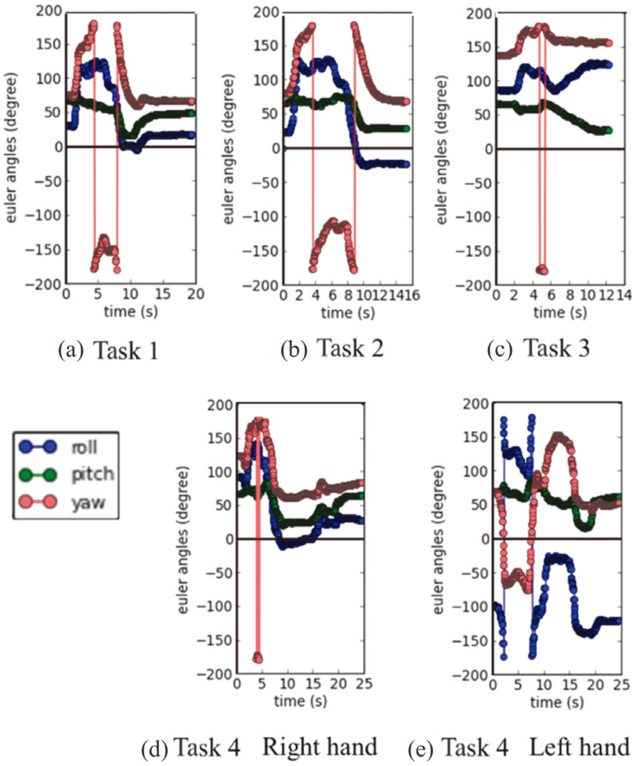


Fig. 16. IMU sensor data related to the orientation of hands during the execution of the tasks. This information is used to detect the type of gripper the robot should be mounted with.

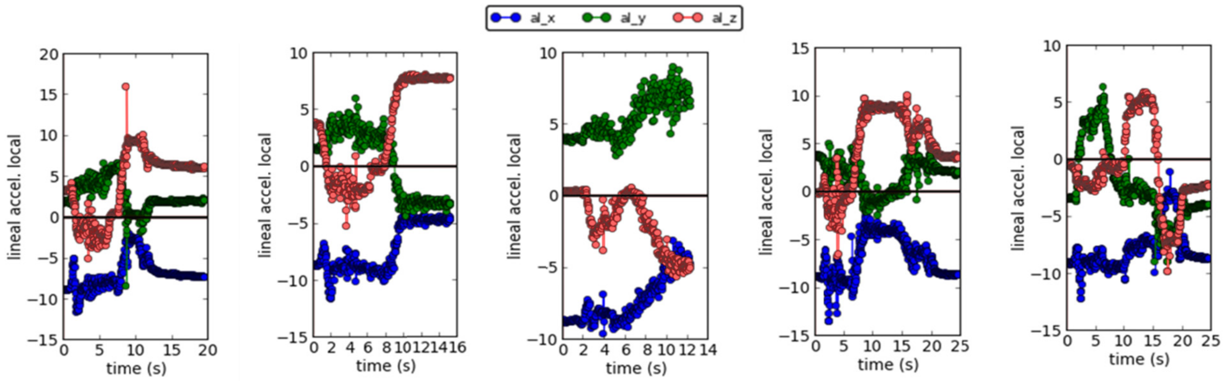
For task four, the decision was made as follows: the task involved the use of two hands; the states that were indicated during the task classification process and those used while the object was manipulated were conducted during the timeframe of 5 ∼ 20 s for the right hand and 13 ∼ 20 s for the left hand. Our framework determined that the grasping orientation was in the vertical direction for both hands (**Figs. 16(d)** and **(e)**). Moreover, the required force was no greater than 5 kg for the right hand and close to 7 kg for the left hand.

By checking the generated data regarding the orientation and the acceleration, it was found that the gripper type remains vertical for both hands; however, the motion of the hands when the “pull out” state occurred (15 ∼ 17 s) was in opposite directions (**Figs. 17(d)** and **(e)**). Therefore, it was decided to divide the task and use two lift-type robots.

In the state sequence generated for task four, one lift robot was assigned to execute the states generated by the right hand, and the other lift robot was assigned to execute the states generated by the left hand. The sequence to maintain synchronization was also generated.

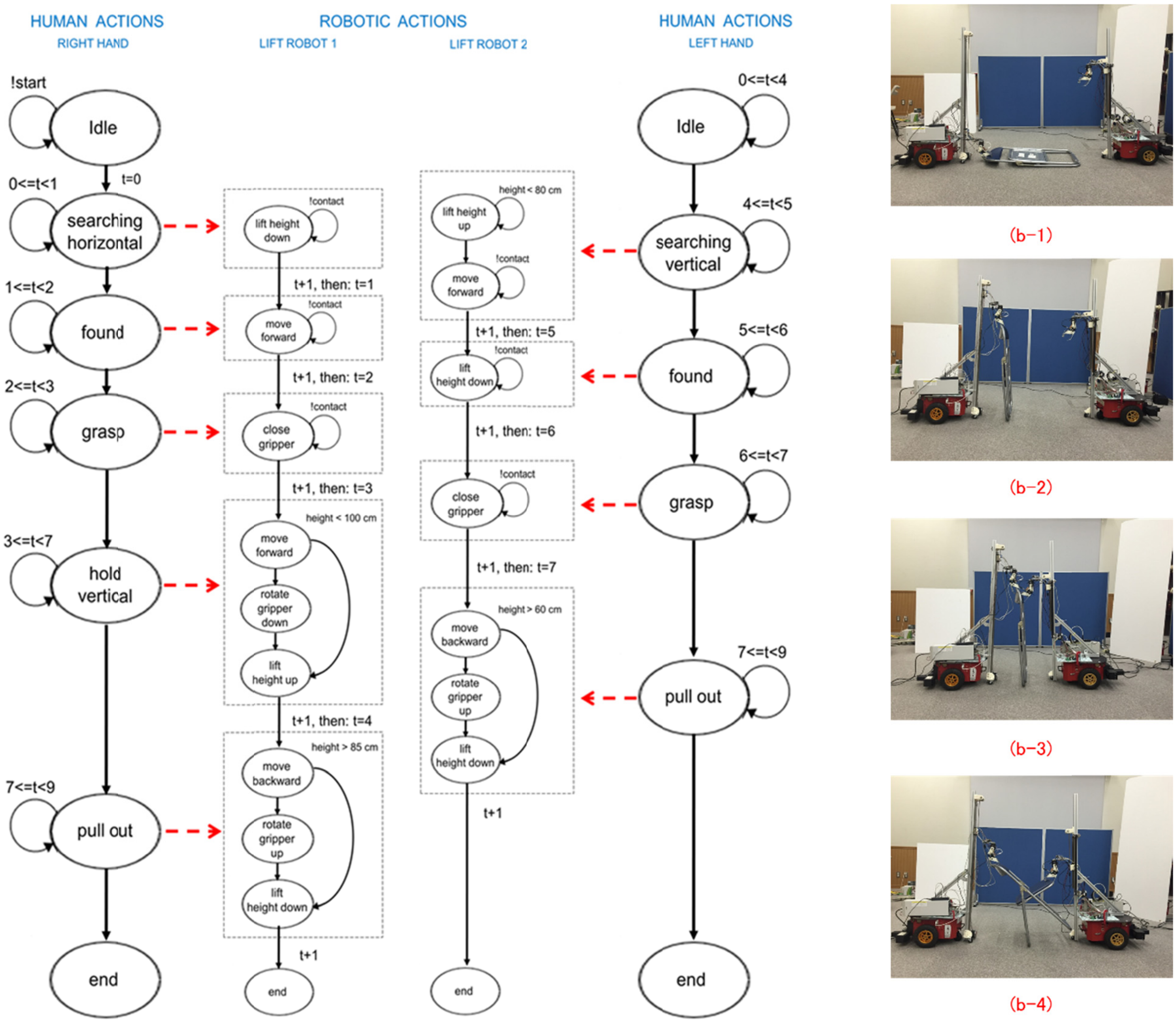
As the last step of the evaluation, the mapping process for robotic commands using basic robotic behaviors for each state, which was stored in the sequence generated during the allocator process, was performed. Thus, the program for the robots to complete the task was generated and presented as a state machine, as shown in **Fig. 18(a)**.

During the mapping process, the robots were taught



(a) Task 1 (b) Task 2 (c) Task 3 (d) Task 4 Right hand (e) Task 4 Left hand

Fig. 17. IMU sensor data related to the acceleration of hands during the execution of the tasks. This information is used to detect if the motion trajectory is achievable by candidate robots.



(a) (b)

Fig. 18. Robot programs for Task 4: (a) State machine with the states, the state behaviors assigned to each robot, as well as the timing to maintain cooperation. (b) Robots performing task four by using the generated program. (b-1), (b-2), (b-3) and (b-4) correspond to time of 47, 166, 223 and 260 s.

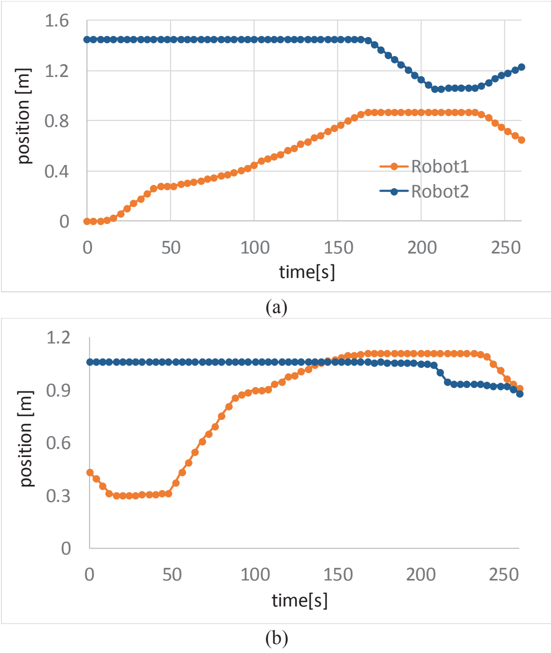


Fig. 19. Generated trajectories of two robots that conduct Task 4: (a) horizontal direction (b) vertical direction.

basic behaviors along with their events, which indicate to robots until when to continue with a particular behavior. For the experiments, we included two events: contact (the robot gripper made contact even with the environment or with the object) and value (the gripper reaches certain value).

The event can be used for a single robotic behavior or a set of them; both cases were used in the experiments presented in this section.

The last part of the evaluation was to perform the tasks by using the generated programs; the selected robots performing task four are shown in **Fig. 18(b)**. The experiments were conducted 10 times for each task. The positions of the two robots for a certain experiment are shown in **Fig. 19**. Here, the positions are expressed in a certain coordinate, and the positions of the robots are defined as those of specific points in the robots (representative points). **Fig. 19(a)** is the graph occurring from the horizontal direction and **Fig. 19(b)** is that occurring from the vertical direction. From these results, we can observe that the robots moved safely by using the generated program. Here, the success ratio of task realization for Tasks 1, 2, and 3 is 80%, and that for Task 4 is 40%. Most failures are due to hardware problems, such as problems with the stock of the gripper.

4.3. Discussion

This section discusses the results of the above-described evaluation. Using the multiple-robot proposed framework to teach the four described tasks, we were able to evaluate the six manipulation styles considered in this study, as listed in **Table 4**.

The total time spent since the human performed the demonstration of each task until the robot programs were generated is shown in **Fig. 13**. We can observe that by including the human in the loop during the process of teaching, the transmission of information to multiple robots was performed in less than 570 s.

Different information was used to determine the number and type of robots to be used for each task, such as questions to the human for identifying the type and characteristics of the manipulation, sensor data linked to states generated during the primitive motion analysis process, and information regarding the hardware of the used robots.

For each task, the above-mentioned information was analyzed via heuristic rules, and thus, the type and the number of robots were selected. From the information, we sought to detect the grasping force, the orientation of the hand when grasping the object (vertical or horizontal), and the direction of motion while the object was being manipulated.

Similar to task four, the details obtained after applying the rules to judge and decide on the type and the number of robots to be used in each task are summarized in **Table 5**.

The states sequence generated for task one did not require the cooperation of multiple robots. Thus, one lift robot was chosen for the execution of one state at a time, until it completed all the states assigned to the task.

The states sequence generated for task two required co-

Table 4. Manipulation styles considered within the evaluated tasks.

Number of Hands	One	One	One	Two	Two	Two
Manipulation Style	Lift	Flip and non-slip	Flip and slip	Lift	Flip and non-slip	Flip and slip
Evaluation	Task 3	Task 1	Task 2	Almost same with Task 3	Task 4	Involving Task 2 and Task 4

Table 5. Allocator process results: type and number of robots decided.

Information Accessed	Task 1	Task 2	Task 3	Task 4
hands used	one hand	one hand	one hand	two hands
type of grasping	vertical	vertical	vertical	vertical (both hands)
force required	no greater than 5Kg-force	no greater than 5Kg-force	greater than 5Kg-force	no greater than 5Kg-force (right hand) greater than 5Kg-force (left hand)
motion trajectory	possible	possible	possible	not possible (divided into two robots)
supported required	no	yes (object slipped)	no	no
decision results	one lift robot	one lift robot one parallel robot (support required)	two lift robots (force not enough)	two lift robots (each hand assigned to one robot)

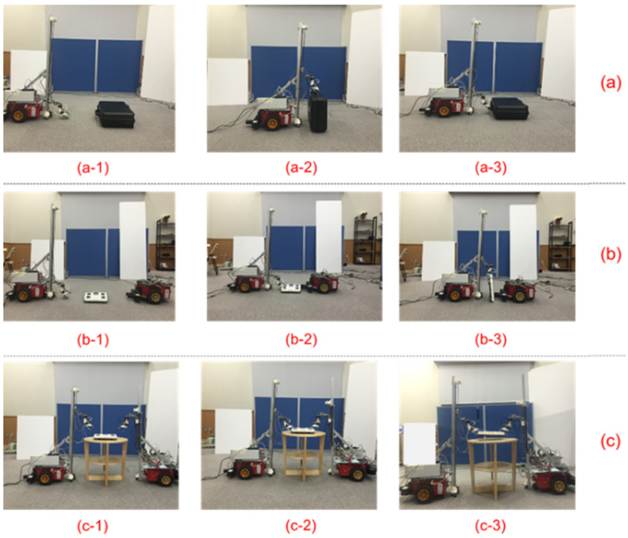


Fig. 20. Tasks executed by robots: (a) task one, flip a briefcase, (b) task two, flip a weighing scale with support, (c) task three, cooperating to lift a weighing scale.

operation between the two robots assigned; the main robot (lift robot) had to rest at the beginning of the task until the parallel robot came to provide support for prevent slippage of the object. After that, the lift robot continuing the remaining states until it completed the task.

The state sequence for task three was generated such that the robots had to cooperate in parallel; both robots performed the same states at the same time and thus were able to apply the required force.

In the case of task four, which was previously discussed, we can add that as the two assigned robots were commanded to perform the action (state: pull out) simultaneously, it was considered by the system that an average force had to be applied by each robot.

After the type and number of robots had been decided upon, the sequence for the execution of the task was generated; this included the timing for cooperation, if more than one robot was used. Similar to task four, for the first three tasks, the robots were taught to perform the states detected during the human example demonstration through basic robotic behaviors. They were assigned to each state, the events thereof, and the timing in programs for each task was used by the robots to successfully execute the taught tasks, as shown in **Fig. 20**.

Based on the results, we can say that through adequate classification of tasks and the combination of several basic behaviors controlled by events with the splitting of human demonstration data, the proposed framework allows a human to easily teach tasks to multiple robots.

Here, the aim of the experiments was to verify the effectiveness of the proposed system, and we assumed the process was built by skilled human operators who can accomplish programming without making mistakes. Consideration of uncertainties or errors in the programming process by human operators is one of the future issues to be clarified.

5. Conclusion and Future Work

We have proposed a novel teaching approach for multiple robots. The framework of the system allows a single human to transmit information related to teaching robots of lower capabilities how to perform object manipulation tasks. Rough assumptions were considered in this study; however, it was demonstrated in the experiments that the proposed study successfully transferred information to multiple robots with two main contributions in comparison to existing methods.

The first contribution lies in the proposal to classify the tasks and determine the number and the type of robots to teach. The second contribution is that the method allows a human to teach multiple robots by adequately splitting the human demonstration data into several units.

The approach was evaluated by teaching four different defined tasks by a single human to a set of small robots. Based on the task result requirements, generation of robot programs was carried out either individually or simultaneously, within an average time of 390 s.

Our future work will involve the use of learning algorithms, which make use of taught tasks; thus, the system becomes smarter to the point where it can suggest actions, criteria to follow, behaviors to use, etc. Furthermore, we plan to conduct experiments on more complicated tasks, where force control or certain types of manipulation skills are necessary.

Acknowledgements

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

References:

[1] T. Arai, E. Pagello, and L. E. Parker, "Editorial: Advances in multi-robot systems," IEEE Trans. on Robotics and Automation, Vol.18, Issue 5, pp. 655-661, 2002.
[2] D. Rus, B. Donald, and J. Jennings, "Moving furniture with teams of autonomous robots," Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 235-242, 1995.
[3] D. Stilwell and J. Bay, "Toward the development of a material transport system using swarms of ant-like robots," IEEE Int. Conf. on Robotics and Automation, pp. 766-771, 1993.
[4] M. Fujii, W. Inamura, H. Murakami, K. Tanaka, and K. Kosuge, "Cooperative Control of Multiple Mobile Robots Transporting a Single Object with Loose Handling," IEEE Int. Conf. on Robotics and Biomimetics, pp. 816-822, 2007.
[5] 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, Issue 6 pp. 445-458, 2013.
[6] B. Donald, L. Gariepy, and D. Rus, "Distributed manipulation of multiple objects using ropes," IEEE Int. Conf. on Robotics and Automation, pp. 450-457, 2000.
[7] T. Sakuyama, J. D. Figueroa Heredia, T. Ogata, T. Hara, and J. Ota, "Object Transportation by Two Mobile Robots with Hand Carts," Hindawi Publishing Corporation Int. Scholarly Research Notices, Vol.2014, 2014.
[8] C. Ferrari, E. Pagello, J. Ota, and T. Arai, "Multirobot motion coordination in space and time," Robotics and Autonomous Systems, Vol.25, pp. 219-229, 1998.
[9] P. Svestka and M. H. Overmars, "Coordinated path planning for multiple robots," Robotics and Autonomous Systems, Vol.23, Issue 3, pp. 125-152, 1998.
[10] K. Yamazaki, T. Tsubouchi, and M. Tomono, "Furniture model creation through direct teaching to a mobile robot," J. of Robotics and Mechatronics, Vol.20, No.2, pp. 213-220, 2008.

[11] J. H. Figueroa Heredia, H. Sahloul, and J. Ota, "Teaching mobile robots using custom-made tools by a semi-direct method," J. of Robotics and Mechatronics, Vol.28, No.2, pp. 242-254, 2016.

[12] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," Robotics and Autonomous Systems, Vol.57, Issue 5, pp. 469-483, 2009.

[13] M. Riley, A. Ude, K. Wade, and C. G. Atkeson, "Enabling Real-time full-body imitation: natural way of transferring human movement to humanoids," Proc. IEEE Int. Conf. on Robotics and Automation, Vol.2, pp. 2368-2374, 2003.

[14] D. Matsui, T. Minato, K. F. MacDorman, and H. Ishiguro, "Generating natural motion in an android by mapping human motion," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3301-3308, 2005.

[15] S. Calinon and A. Billard, "Incremental Learning of Gestures by Imitation in a Humanoid Robot," ACM/IEEE Int. Conf. on Human-Robot Interaction, pp. 255-262, 2007.

[16] M. Ogino, H. Toichi, Y. Yoshikawa, and M. Asada, "Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping," Robotics and Autonomous Systems, Vol.54, Issue 5, pp. 414-418, 2006.

[17] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," Robotics and Autonomous Systems, Vol.47, Issues 2-3, pp. 109-116, 2004.

[18] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos, "Robot Learning Manipulation Action Plans by "Watching" Unconstrained Videos from the World Wide Web," Conf. on Artificial Intelligence (AAAI), pp. 3686-3692, 2015.

[19] R. A. Knepper, T. Layton, J. Romanishin, and D. Rus, "IkeaBot: An Autonomous Multi-Robot Coordinated Furniture Assembly System," IEEE Int. Conf. on Robotics and Automation, pp. 855-862, 2013.

[20] Y. Maeda, N. Ishido, H. Kikuchi, and T. Arai, "Teaching of grasp/graspless manipulation for industrial robots by human demonstration," IEEE/RSJ Int. Conf. Intelligent Robots and Systems, Vol.2, pp. 1523-1528, 2002.

[21] S. Chernova and M. Veloso, "Confidence-Based Multi-Robot Learning from Demonstration," Int. J. of Social Robotics, Vol.2, Issue 2, pp. 195-215, 2010.

[22] T. Inoue, "Future tasks of research in robotics," IEEE Int. Conf. on Robotics and Automation, Vol.1, 1995.

[23] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," Found Trends Human-Computer Interaction, Vol.1, Issue 3, pp. 203-275, 2007.

[24] D. J. Sturman and D. Zeltzer, "A survey of glove-based input," Computer Graphics and Applications IEEE, Vol.14, Issue 1, pp. 30-39, 1994.

[25] W. Premerlani and P. Bizard, "Direction cosine matrix imu: Theory," Tech. Rep., 2009.

Supporting Online Materials:

[a] FSR 402 (Force Sensing Resistor)
http://www.interlinkelectronics.com/FSR402.php
[Accessed June 7, 2016]

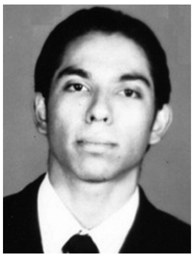
[b] 9 Degrees of Freedom – Razor IMU
https://www.sparkfun.com/products/10736
[Accessed June 7, 2016]

[c] Bluetooth Bee
http://www.seeedstudio.com/wiki/Bluetooth_Bee
[Accessed June 7, 2016]

[d] Microcontroller board Arduino
https://www.arduino.cc/en/Main/ArduinoBoardProMini
[Accessed June 7, 2016]

[e] Tang Tiong Yew, Kristof Robot, Paul Bouchier, Peter Bartz, razor_imu_9dof. package, http://wiki.ros.org/razor_imu_9dof
[Accessed June 7, 2016]

[f] Adept Mobile Robots LLC, USA
http://www.mobilerobots.com/ResearchRobots/PioneerP3DX.aspx
[Accessed June 7, 2016]



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- Joined Graduate Program, 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, 2014 (684235), pp. 1-15, 2014.

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

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, 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. of Automation Technology, 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:
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:
• "Designing Non-circular Pulleys to Realize Target Motion between Two Joints," IEEE/ASME Trans. on Mechatronics (in press).
• "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.
• "Detection and prevention of slip using sensors with different properties embedded in elastic artificial skin on the basis of previous experience," Robotics and Autonomous Systems, Vol.62, No.1, pp. 46-52, 2014.

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

Name:
Jun Ota

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

Address:
5-1-5 Kashiwanoha, Kashiwa, 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:
• Y. Huang, R. Chiba, T. Arai, T. Ueyamae, 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.
• Z. Huang, A. Nagata, M. Kanai-Pak, J. Maeda, Y. Kitajima, M. Nakamura, K. Aida, N. Kuwahara, T. Ogata, and J. Ota, "Self-help training system for nursing students to learn patient transfer skills," IEEE Trans. on Learning Technologies, Vol.7, No.4, pp. 319-332, 2014.
• H. Nakanishi, N. Shichijo, M. Sugi, T. Ogata, T. Hara, and J. Ota, "Modeling the process of animation production," Int. J. of Automation Technology, Vol.7, No.4, pp. 439-450, 2013.

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)