國立交通大學

資訊科學與工程研究所

博 士 論 文

基於人類示範之意圖推論於工具操作任務之應用

Intention Deduction by Demonstration for Tool-Handling Tasks

研 究 生：陳豪宇

指導教授：傅心家 教授
楊谷洋 教授

中 華 民 國 一 百 零 一 年 六 月

基於人類示範之意圖推論於工具操作任務之應用

Intention Deduction by Demonstration for Tool-Handling Tasks

研 究 生 ： 陳豪宇　　Student: Hoa-Yu Chan

指導教授 ： 傅心家 教授　　Advisor: Prof. Hsin-Chia Fu

楊谷洋 教授　　Prof. Kuu-young Young

國 立 交 通 大 學

資訊科學與工程研究所

博 士 論 文

A Dissertation

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

June 2012

Hsinchu, Taiwan, Republic of China

中華民國一百零一年六月

基於人類示範之意圖推論於工具操作任務之應用

研　究　生：陳豪宇　　指導教授：傅心家 教授

楊谷洋 教授

國立交通大學資訊科學與工程研究所

摘　　要

在不久的將來，可以預期越來越多的機器人工作環境會從工廠移至家庭環境，如果機器人在面對各種家庭任務時均需有對應的程式是不切實際的。 因此，有學者提出讓機器人從示範中學習的概念，可以減少使用者分析與程式的負擔。 然而許多遵循這概念的方法卻需要限制使用者動作或任務計畫進而得以推論使用者的意圖。 爲了避免這些限制，我們提出一個新的方法讓機器人能從工具操作任務執行的軌跡中推論出示範者的意圖。 在家庭環境中，工具操作任務是常見的任務，但是分析示範者的意圖卻不容易。 我們的方法乃基於交叉驗證的概念，定位出符合精細技巧操作的軌跡片段，並且利用動態規劃尋找最有可能的意圖。 我們提出的方法不需要事先定義可操作的動作或限制動作的速度，並且在示範的過程中允許變換動作順序，也可加入多餘的動作。 在實驗中，我們提出的方法以三種任務進行測試，分別是倒水、泡咖啡及塗果醬任務，在示範中改變任務物件的位置和數目以測試其影響。 更進一步，我們分析任務中各參數的影響來研究方法的適用性。 實驗結果顯示我們的方法不但對於這三種任務可以推論使用者的意圖，而且可以讓使用者在沒有限制的情況下以較自然且有效率的方式示範動作。

Intention Deduction by Demonstration for Tool-Handling Tasks

Student: Hoa-Yu Chan Advisor: Prof. Hsin-Chia Fu
Prof. Kuu-young Young
Institute of Computer Science and Engineering
National Chiao Tung University

Abstract

In the near future, more robots come to the home-like environment, the programming for task execution becomes very demanding, if not infeasible. The concept of learning from demonstration is thus introduced, which may remove the load of detailed analysis and programming from the user. However, many methods which follow the concept of learning from demonstration limit the motions of the user or task plan to deduce the intentions of the user. To avoid these limitations, in this dissertation, we propose a novel approach for the robot to deduce the intention of the demonstrator from the trajectories during task execution. We focus on the tool-handling task, which is common in the home environment, but complicated for analysis. We apply the concept of cross-validation to locate the portions of the trajectory that corresponds to delicate and skillful maneuvering, and apply an algorithm based on dynamic programming to search for the most probable intention. The proposed approach does not predefine motions or put constraints on motion speed, while allowing the event order to be altered and the presence of redundant operations during demonstration. In experiments, we apply the proposed approach for three different kinds of tasks: pouring, coffee-making, and fruit jam, with the number of objects and their locations varied during demonstrations. To further investigate its scalability and generality, we also perform intensive analysis on the parameters involved in the tasks. The results show that our approach can not only deduce the intentions of user in the three kinds of tasks but also let the demonstrations be executed in a natural and effective manner without the limitations.

Acknowledgment

首先感謝我的指導教授—楊谷洋教授與傅心家教授，在我攻讀學位長達十年的期間給予指導、支持與協助。亦感謝口試委員林正中教授、單智君教授、蔡清池教授、蘇木春教授、許秋婷教授、林銘瑤博士於口試時給予的寶貴意見，使本論文可以更加完整。另外謝謝同學一元、木政、修任、一哲與「人與機器實驗室」的同窗在研究與實驗上的幫忙，使本論文的研究可以完成。最後要感謝我的家人—我的父母、外公、外婆、妹妹，以及寵物貓咪、烏龜與小狗乖乖。

Table of contents

List of Tables

List of Figures

Nomenclature

D	delicate motion
E	the difference between generated motion and validating motion
G	generated motion
i	an index of training motions or generated motions
j, k	an index of delicate motions or move motions
M	move motion
MI	an index linking to a set of delicate motions
Q	a motion consists of delicate motions and move motions
T	training motion
V	validating motion

Chapter 1 Introduction

1.1 Background

Due to the progress in service robot, more robots are entering the home or office environment. It can be expected that many challenging problems shall emerge when they deal with these highly uncertain and varying environments, such as path planning and manipulation [1,2]. One issue of interest is how to teach the robot to perform daily works effectively. To relieve the human operator from detailed task analysis and program coding, researchers have proposed letting the robot learn how to execute the task from observing human demonstration by itself [3]. However, the motions of human demonstration are difficult to be analyzed because the motions related to home or office environment may not be able to be predefined for robot learning. Besides, in the trajectory level, it is difficult that human repeats a motion in exactly the same speed and trajectory during multiple demonstrations, and the operated objects may be not fixed in the environment. Moreover, in the task level, some tasks can be accomplished with a varying operating order or redundant motions. These pose challenges to learning from demonstration.

Many researchers have proposed approaches to tackling these challenges [4–7]. Among them, Calinon proposed an approach using Gaussian Mixture Regression (GMR) and Lagrange optimization to extract the unchanged motions from the multiple demonstrations [8,9]. The proposed approach demands the order of the operating motions to be the same during demonstrations. Dautenhahn and Nehaniv proposed an approach for the robot to learn from human demonstration by imitation [10], referred to as the correspondence problem, and later the team developed a system that can learn 2D arranging tasks [11,12]. Dillmann proposed a hierarchical structure for the robot to deal with complex tasks while the motion order can be changed [13], and later they went on analyzing human motion features for high-level tasks [14]. With both symbolic and trajectory levels of skill representation, Ogawara proposed a method that determines the essential motions from the possi-

ble motions [15]. However, these proposed methods may need to pre-define human motions, limit the human operator to follow some motion type or speed, or limit operating order. The robot may, thus, not be able to learn the task automatically or the human operator not demonstrate the task naturally and efficiently. In contrast, babies of 7 $\sim$ 8 month old can learn motions by regular pattern, while it is still unclear of the learning process [16]. Therefore, it motivates us to propose a method which can learn motion by demonstration without these shortcomings.

1.2 Contributions of the Dissertation

In this dissertation, we propose an approach for the robot to learn the human intention from her/his demonstration. To allow the human operator for more natural and efficient manipulation during demonstration, the proposed approach (a) does not need to pre-define motions, (b) does not constrain the operator to perform the task with certain motion speed or motion type, (c) allows the order of the events to be altered, and (d) allows some redundant operations.

For the motions of human manipulations, Dillmann classify them into three different types by their goals: transport operations for moving objects, device handling for changing the internal states of the objects, and tool handling for using tool to interact with objects. We focus on the tool-handling task, which is common in daily life [14]. The motions of this kind of task can be performed continuously without stopping, because the tool can be operated for multiple objects sequentially without leaving the hand, and it is complicated for analysis. Without predefined motions, the method of cross-validation is suitable to decide the unknown parameters of a learning system. Based on the concept of cross-validation, but with some modification, we propose an approach to identifying the portions of the trajectory corresponding to the delicate and dexterous maneuver of the demonstrator, referred to as motion features. These motion features, in some sense, exhibits the human skill in executing a certain task. The challenge is how to find the correct intentions, among all possible ones, that lead to the demonstrated trajectories. To tackle the complexity,

we apply the method of dynamic programming for the search. For demonstration, experiments based on three different kinds of tasks, the pouring, coffee-making, and fruit jam tasks, are performed. During the experiments, the locations of the operated objects and operating sequence may vary, and the motion features derived from the demonstrated trajectories are used for task execution under different experimental settings. To further demonstrate the scalability and generality of the proposed approach, we perform intensive analysis on the parameters involved in the tasks, such as numbers of objects and demonstrations, among others.

1.3 Organization of the Dissertation

This dissertation is divided into six chapters. In Chapter 1, the background information of robot learning from demonstration and the contributions are introduced to explain why we proposed such a new approach for intention deduction. In Chapter 2, previous approaches are introduced and discussed. In Chapter 3, we describe the proposed approach, which is based on the concept of cross-validation to deduce the intention from demonstrations In Chapter 4, the experimental results are reported for performance evaluation, and the discussion on the proposed approach is given. Finally, in Chapter 5, the conclusions are presented and suggestions are stated for future research.

Chapter 2 Preliminaries and Surveys

Many researchers have proposed letting the robot learn how to execute the tool-handling task from observing human demonstration by itself [4–7]. The tool-handling task is of the focus in this dissertation, which involves the interactions between tools and objects [17]. The resultant trajectory from task execution can be mainly divided into two types of motions: delicate motion (D) for delicate maneuver and move motion (M) between the delicate motions [15]. The delicate motion is much of the interest, since it serves to achieve the goal; by contrast, the move motion is not very critical. Meanwhile the delicate and move motions are executed alternately. As some delicate motions are noncontact motions, which do not contact the operated objects, such as those in pouring motions, the operated objects may not be determined directly. Besides, some tasks may be able to be accomplished when the order of executing the delicate motions is changed or redundant delicate motions are added during demonstration. Therefore, in order to tackle the uncertainty in demonstration, multiple demonstrations are collected. The robot may need to recognize the delicate motions and analyze the ordering of the delicate motions from the multiple demonstrations. Among the current approaches, they can basically be classified into three types based on either using pre-known human motions or certain assumptions [4].

2.1 Motion Recognition Type

The approaches of the first type are based on pattern recognition methods to classify motions before analyzing the ordering of the delicate motions [18–27]. In this type, human needs to predefine a set of operating motions before the robot observes the human demonstrations. Human collects all possible operating motions, labels the collected motions into different classes of motions, and uses them as training data. For example, in [18], the researchers define a set of human motion data, which consists of Power Grasp, Precision Grasp, Pour, Hand-over, Release, OK-

sign, Garbage, Start, and End, to classify human motions. Usually, Hidden Markov Model(HMM) is used to learn and recognize the operating motions because HMM can process stochastic human motion data in space and time. After motion recognition, in order to analyze the relation between operating motions, the relation of operating motions may be transformed into a symbolic sequence problem such as longest common subsequence problem (LCS) to find the common relation in the multiple demonstrations. Finally, the recognized motions are used to generate an operating trajectory based on the motion relations for robot execution. The concept of this type is not complicated, and the system can be implemented easily. However, it may not be able to handle all possible motions of daily life because the operating motions need to be collected before recognition.

2.2 Motion Matching Type

The approaches of second type segment human motions and matches the common motions without predefining motions [14,15,28–35].First, the motions of the demonstrations are segmented based on some motion features such as the difference of operating speed [34,36] or the cyclic motion [37] without predefining motions. The approaches search for the common motions of all demonstrations by motion matching and they analyze the essential motions and the ordering of the motions. In [15], the problem of searching the common motions between the demonstrations is transformed to multiple sequence alignment problem to handle redundant motions. In [14], the researchers measure the similarities between all possible permutations of the motions of the subtasks to analyze the hierarchical structure of the common motions of the task. Finally, the common motions are searched to be the essential motions of the task and used to generate a new operating trajectory for execution. These approaches do not need to predefine possible motions, so this type can handle many known or unknown tasks of daily life. However, the motion hints such as the difference of operating speed or the cyclic motion, which are used to segment, limit human motions because only the hint-following motions can be segmented correctly.

2.3 Motion Synchronization Type

The approaches of the third type find common portions from synchronized operating trajectories without motion segmentation [8,9,38–55]. First, the operating trajectories of the multiple demonstrations are synchronized as signals to avoid segmentation. Usually, the synchronization methods such as Dynamic Time Warpping (DTW) [56] or Continuous Profile Models (CPM) [57,58] are used to synchronize multiple signals. After synchronization, the difference of each portion of the synchronized trajectories are measured to find common operating motions from multiple demonstrations. For example, the synchronized trajectories can be transformed to the probability distribution by using Gaussian Mixture Models (GMM) to measure the expected operating trajectory [9]. Finally, the common operating trajectory is used to generate a new operating trajectory for the robot execution in a new environment. The approaches of this type can learn tasks without predefining motions or limiting human to follow some motion hints. However, these synchronization methods cannot handle permuting motions, so the ordering of the motions must be the same during multiple demonstrations. The order of operating motions is limited, and human may not be able to demonstrate task naturally and efficiently.

Chapter 3 Proposed Approach

In this chapter, we describe the proposed approach which takes the intention deduction problem to be that of locating the delicate motion from the demonstrated trajectory. In Sec. 2.1, we describe the process of intention deduction. In Sec. 2.2, we explain similar function, which is used in intention deduction. And, in Sec. 2.3, we design a series of experiments for investigating its extensibility and robustness.

3.1 Intention Deduction

In this section, first, we introduce what the intention is in the tool-handling task, and then formulate it. Then, we use the concept of validation to evaluate the motion index candidates. Finally, we use dynamic programming to search the optimal motion index from all possible candidates.

3.1.1 Intention in Tool-handling Task

Fig. 3.1 shows the conceptual diagram of the proposed approach for intention deduction from demonstration. In Fig. 3.1, the robot first observes a series of human demonstrations and records the corresponding trajectories and environmental states. From these recorded motion data, the robot searches for the possible intentions that lead to the delicate motions. The derived intentions can then be used to generate new trajectories that respond to new environmental states. Let us take the pouring task shown in Fig. 3.2 as an example. In Fig. 3.2(a), three vessels A, B, and C are arbitrarily located on the table. And, in Figs. 3.2(b)-(c), the operator pours the content from vessel A to vessels B and C, respectively, and then places vessel A back on the table. During the demonstrations, the initial locations of the vessels may vary, and so does the pouring sequence. From the recorded trajectories and corresponding locations of the vessels (environmental states), the proposed approach identifies the intention of the operator, i.e., the portions of the trajectory that cor-

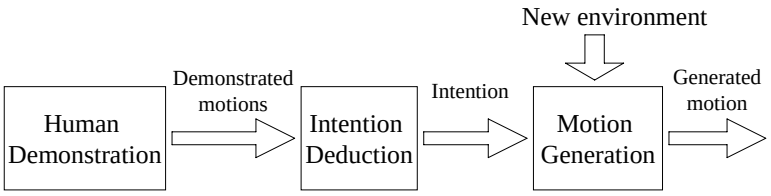


Figure 3.1: Conceptual diagram of the proposed approach.

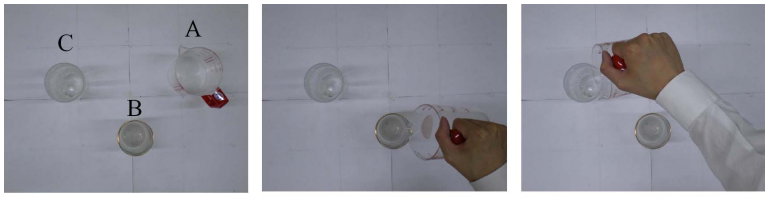


Figure 3.2: A pouring task: (a) the setting of the vessels, (b) pouring vessel A to vessel B, and (c) pouring vessel A to vessel C.

respond to the two pouring actions (delicate motions). With the derived intention, the robot is then able to execute the pouring task with the vessels located at various locations and possibly altered pouring sequences.

Before the discussion on the process of intention deduction, we first describe how the motion can be generated under new environmental states when the human intention has already been derived. We start with the representation of the intention I. Assume that there are N delicate motions and S objects involved in a demonstrated task. Because the intention is closely related to the delicate motions of the maneuver, I is formulated as a set of delicate motions, $D_n(t)$, associated with the corresponding objects Obj_s:

$$I = \{D_1(t), D_2(t), ..., D_N(t); Obj_1, Obj_2, ..., Obj_S\} \tag{3.1}$$

where $D_n(t)$ stands for the part of the demonstrated trajectory for delicate motion n and Obj_s the position and orientation of an object s. Note that, because an object may correspond to one, several, or no delicate motion, the number of delicate motions may not be equal to that of the objects. We then introduce the motion index (MI), which serves as an index linking to I. MI is formulated as an ordered set of the time-point pairs, $d_j = \{n_j, l_j, s_j\}$, which provides the starting time n_j,

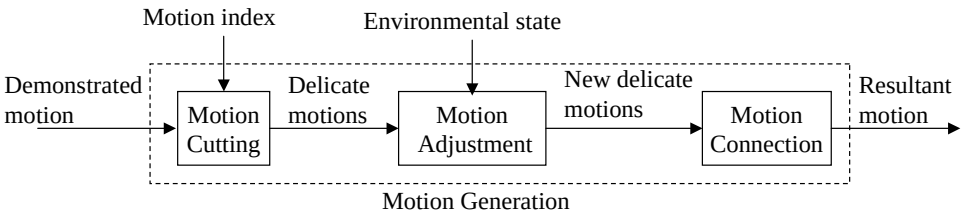


Figure 3.3: Process for motion generation.

end time l_j, and number of the operated object s_j for each of delicate motions D:

$$MI = \{d_1, d_2, .., d_N\} \tag{3.2}$$

where MI represents I. Fig. 3.3 shows the process for motion generation. According to MI, the motion cutting module locates the delicate motions D_j from the demonstrated motion in order. To respond to the new environmental state, the motion adjustment module moves these D_j to match the new locations of the objects and become D_{G_j}. Finally, the motion connection module uses the move motion M_{G_j} to smoothly connect every two D_{G_j}. As its accuracy is not that critical, M_{G_j} is generated using the cubic polynomial. With both D_{G_j} and M_{G_j}, we now have a feasible trajectory Q_G corresponding to the new environmental state:

$$Q_G = \{M_{G_1}, D_{G_1}, M_{G_2}, D_{G_2}, ..., D_{G_N}, M_{G_{N+1}}\} \tag{3.3}$$

Fig. 3.4(a) shows an example for motion cutting based on the pouring task shown in Fig. 3.2, and Fig. 3.4(b) that of motion generation. In Fig. 3.4(a), the demonstrated trajectory during task execution is projected on the X-Y plane, where the yellow and green rectangles indicate the locations of vessels B and C. The yellow and green trajectories are the delicate motions determined by the motion cutting module according to the given MI. In Fig. 3.4(b), the yellow and green rectangles indicate the locations of vessels B and C in the new environmental state. In responding to these new locations of vessels B and C, the delicate motions identified in Fig. 3.4(a) are transformed to be the yellow and green trajectories by the motion adjustment module. Finally, the three move motions, as the red trajectories, are utilized to smoothly connect the two delicate motions.

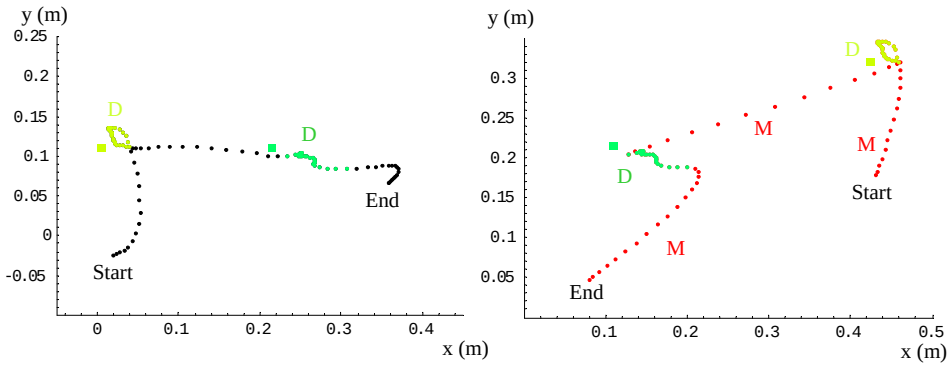


Figure 3.4: Examples for (a) motion cutting and (b) motion generation based on the pouring task shown in Fig. 3.2.

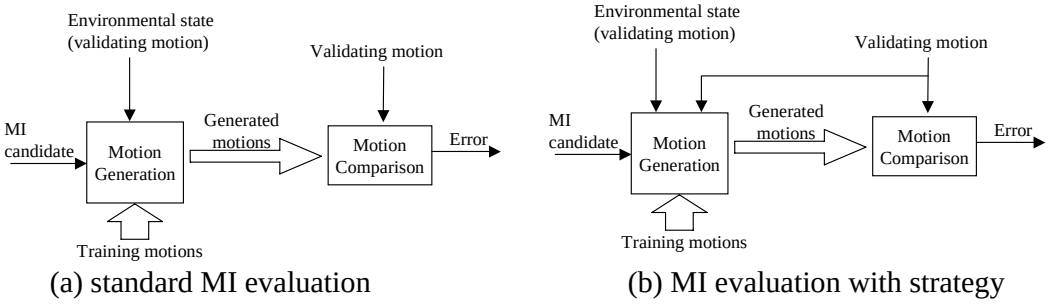


Figure 3.5: Process for MI evaluation: (a) standard MI evaluation and (b) MI evaluation with strategy.

3.1.2 Derivation of Optimal Motion Index

From the motion generation process discussed above, we can take the intention deduction process as that of finding proper motion index MI. To find the optimal MI among all MI candidates, we introduce first the process for MI evaluation, shown in Fig. 3.5(a). This process evaluates the fitness of the MI candidates derived from the demonstrated motion, based on a reasoning that proper MI should lead to a generated motion very similar to the human demonstrated motion, which includes all the delicate motions. In Fig. 3.5(a), from the demonstrated motions, we select one demonstrated motion as the validating motion and the rest as the training motions. We will discuss the selection of validating and training motions later. For an MI candidate derived from the validating motion, the motion generation module, described above, generates motions based on the training motions and the environmental state corresponding to the validating motion; the generated motions, with their lengths set to be equal to that of the validating motion, are then compared with the validating motion via the motion comparison module, yielding the differences between them (marked as errors). Because the operator may perform the demonstrations in different speeds and possibly with different orders for the events involved, the corresponding delicate motions are likely to be with various sampling rates, or to appear in different portions of the demonstrated trajectories. To tackle this, our strategy is to let each of the delicate motions of the validating motion be compared with every portion of the training motion, accompanied by altering sampling rates, showing in Fig. 3.5(b). Through this comparison process, the generated motion, whose delicate motions lead to the minimum difference when compared with those of the validating motion, is determined as the output and sent to the motion comparison module for the following comparison. As a high search complexity is expected, we come up with an approach analogous to that of dynamic time warping (DTW) in execution [56]. Details of this strategy will be explained in next section.

We go on with the process for MI generation, shown in Fig. 3.6. In Fig. 3.6, among all the demonstrated motions, one demonstrated motion is first selected as

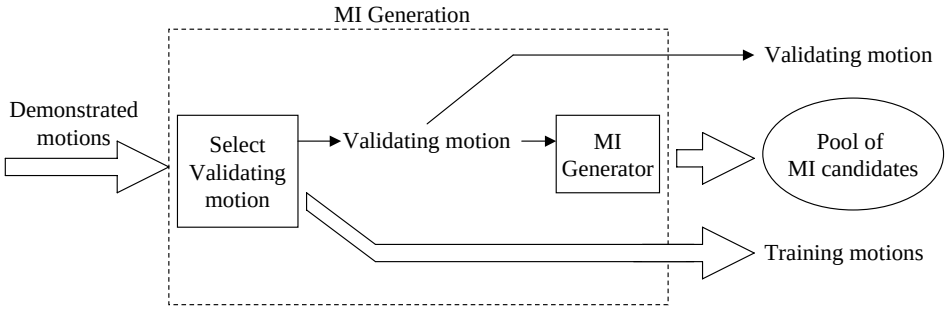


Figure 3.6: Process for MI generation.

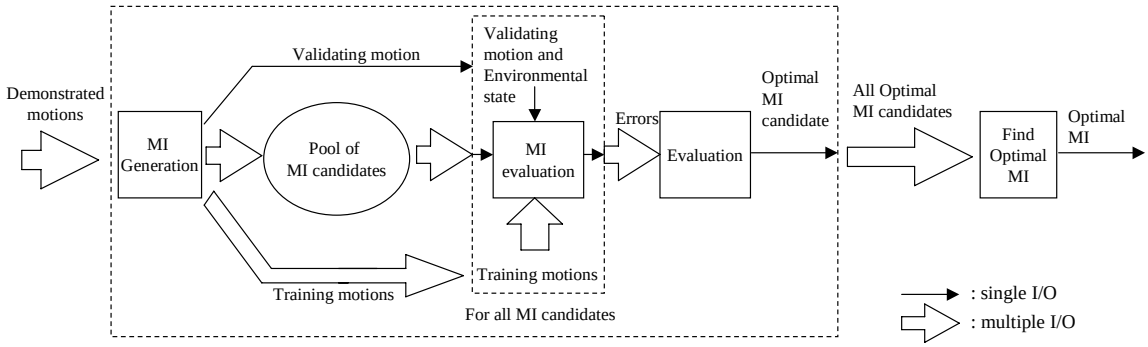


Figure 3.7: Process for optimal MI derivation.

the validating motion, denoted as Q_V, and the rest as the training motions, Q_T, for each sequence of the process. The process will be repeated until each of the demonstrated motions serves as the validating motion once. In next step, the MI generator will locate all possible MI candidates from Q_V. Because the proposed approach does not constrain the human operator to perform the task with certain motion speed or motion type, and also allows the order of the events to be altered during demonstration, there is in fact not a priori knowledge for the selection of MI. The criterion for MI generation is thus to let MI candidate correspond to every portion of Q_V with a duration longer than 0.3 second, as human cannot cognize an event until it happens 0.3 second later [59]. It can be expected that there will be a huge number of MI candidates. That is why we employ the method of dynamic programming for the search of the optimal MI.

With the MI evaluation process in Fig. 3.5(b) and MI generation process in Fig. 3.6, Fig. 3.7 shows the entire process for optimal MI derivation. For the outer dotted block in Fig. 3.7, the inputs are the demonstrated motions and each

of them serves as the validating motion once. Via the MI generation process, MI candidates along with the validating and training motions are sent into the MI evaluation process to determine which MI candidate leads to the minimum error, identified as an optimal MI candidate. As each validating motion corresponds to one optimal MI candidate, the outputs of the outer dotted block are the optimal MI candidates for each of them. Finally, the optimal MI is determined to be the one with the minimum error among all optimal MI candidates.

3.1.3 Implementation of Intention Deduction

For mathematical formulation of this optimal MI derivation process, we start with the description of MI for a given validating motion Q_V, denoted as MI_V:

$$MI_V = \{d_{V_1}, d_{V_2}, .., d_{V_N}\} \tag{3.4}$$

with

$$d_{V_j} = \{n_{V_j}, l_{V_j}, s_{V_j}\} \tag{3.5}$$

where d_{V_j} indexes the delicate motion D_{V_j} with n_{V_j}, l_{V_j}, and s_{V_j} the starting time, end time, and number of the operated object. According to MI_V, Q_V can then be expressed as the combination of a series of delicate and move motions:

$$Q_V = \{M_{V_1}, D_{V_1}, M_{V_2}, D_{V_2}, ..., D_{V_N}, M_{V_{N+1}}\} \tag{3.6}$$

On the other hand, with the same MI_V, the generated motion Q_G^i for each training motion Q_T^i can be formulated as

$$Q_G^i = \{M_{G_1}^i, D_{G_1}^i, M_{G_2}^i, D_{G_2}^i, ..., D_{G_N}^i, M_{G_{N+1}}^i\} \tag{3.7}$$

where $D_{G_j}^i$ and $M_{G_j}^i$ are its delicate and move motion, respectively. $D_{G_j}^i$ can be determined via the MI evaluation process above, of which the minimization between $D_{G_j}^i$ and D_{V_j} is dealt with a DTW-like method:

$$D_{G_j}^i = similar(Q_T^i, D_{V_j}) \tag{3.8}$$

In the *similar* function, the training motion Q_T^i is transformed to match the environment of the validating motion according to the possible operated object s_{V_j},

and generated delicate motion $D^i_{G_j}$ is searched from the transformed motion to be similar to D_{V_j} as close as possible. Therefore, the search can use a DTW-like method to minimize the difference between $D^i_{G_j}$ and D_{V_j} [56]. Details of this method will be explained in next section.

After the generated delicate motions are generated, $M^i_{G_j}$ determined by function M_G, which utilizes the cubic polynomial to smoothly connect the two delicate motions, $D^i_{G_{j-1}}$ and $D^i_{G_j}$:

$$M^i_{G_j} = M_G(D^i_{G_{j-1}}, D^i_{G_j}) \tag{3.9}$$

To determine the optimal motion index MI^*_V, Q_V will be compared with all Q_G generated according to every MI_V. Because we are looking for an MI_V that may induce all the necessary delicate motions, MI^*_V should not induce too much deviation between the delicate motions for Q_V and Q_G, and consequently between the move motions for them. By taking E_{max} as the maximum difference between the delicate and move motions for Q_V and those Q_G generated for all the training motions corresponding to some MI_V, we determine MI^*_V, among all MI_V, to be the one that leads to the smallest E_{max}:

$$MI^*_V = \underset{MI_V}{arg\min}\, E_{max} \tag{3.10}$$

with

$$E_{max} = \sum_{j=1}^{N} E_D(D_{V_j}) + \sum_{j=1}^{N+1} E_M(D_{V_{j-1}}, D_{V_j}) \tag{3.11}$$

where

$$E_D(D_V) = \max_i ||D_V - D^i_G||^2 \tag{3.12}$$

$$E_M(D_{V_a}, D_{V_b}) = \max_i ||M_V(D_{V_a}, D_{V_b}) - M_G(D^i_{G_a}, D^i_{G_b})||^2 \tag{3.13}$$

Here, E_D computes the difference between the respective delicate motions for Q_V and those Q_G, and E_M that for the move motions, with M_V as a function which outputs the move motion part between two delicate motions of the validating motion, D_{V_a} and D_{V_b}. Because each demonstrated motion serves as the validating motion once, the final optimal motion index MI^{**} for all demonstrated motions will be further chosen as that MI^*, among those for each Q_V, with the smallest corresponding

E_{max}, demoted as E^*. As the length L_V for each Q_V may not be the same, E^* needs to be normalized before the comparison. MI^{**} is then formulated as

$$MI^{**} = arg \min_{MI_V^*} E^*/L_V \tag{3.14}$$

The search for MI^{**} is of high complexity, as exhibited in Eqs. (3.10)-(3.14) above. As an attempt to enhance search efficiency, we employ the method of dynamic programming [60] and let the computation of E^* in Eq. (3.14) be expressed into a recursive formulation:

$$E^* = \min_{d_{V_k}} E_R(D_{V_k}) + E_M(D_{V_k}, D_{V_{N+1}}) \tag{3.15}$$

with

$$E_R(D_{V_k}) = \min_{d_{V_{k-1}}} (E_R(D_{V_{k-1}}) + E_M(D_{V_{k-1}}, D_{V_k})) + E_D(D_{V_k}) \tag{3.16}$$

where $E_R(D_{V_k})$ stands for the minimum difference between the motions from the first move motion to a given delicate motion; d_{V_k} and $d_{V_{k-1}}$, described in Eq. (3.5), index the delicate motions D_{V_k} and $D_{V_{k-1}}$; and $1 \leq k \leq N$. Because the number of delicate motions is not known in advance, N and k are not specific numbers. Also note that, the first move motion is generated between D_{V_0} and D_{V_1}, and the last one between D_{V_N} and $D_{V_{N+1}}$, with D_{V_0} and $D_{V_{N+1}}$ taken as the first and last point of the trajectory, respectively. In Eq. (3.15), E^* is derived as the minimum one for all $E_R(D_{V_k})$ with $E_R(D_{V_k})$ computed recursively via Eq. (3.16). With Eqs. (3.15) and (3.16), dynamic programming can take advantage of the table generated for $E_R(D_{V_k})$ to simplify the computation in deriving E^*.

Based on the discussions above, the algorithm for intention derivation algorithm is formulated in Algorithm 1. Time complexity for this optimal MI derivation process is related to the number (R) and length (L_V) of the demonstrated motions and the number (S) of objects involved in the task. Here, the lengths of the demonstrated motions are assumed to be close. In Eqs. (3.15) and (3.16), the generation of the table for $E_R(D_{V_k})$ takes up most of the time consumed. The table has $O({L_V}^2 \cdot S)$ elements, and each element deals with the complexity of the order of $O(R \cdot {L_V}^3 \cdot S)$. During the entire process, the table needs to be generated R times. The final time complexity is thus computed to be in the order of $O(R^2 \cdot {L_V}^5 \cdot S^2)$.

The divide-and-conquer method [60] may also be an alternate to solve Eq. (3.14). However, because our proposed approach takes every portion of the trajectory of the validation demonstration as the candidate for a possible delicate motion, it is not that straightforward to divide the trajectory properly. Consequently, the search for the optimal solution may demand a large number of divisions, leading to high computational load.

Algorithm 1 Find the intention of the task through R times of demonstrations

Input: the demonstrated trajectories Q_i $(1 \leq i \leq R)$ for the R times of demonstrations

Output: the optimal MI^{**}

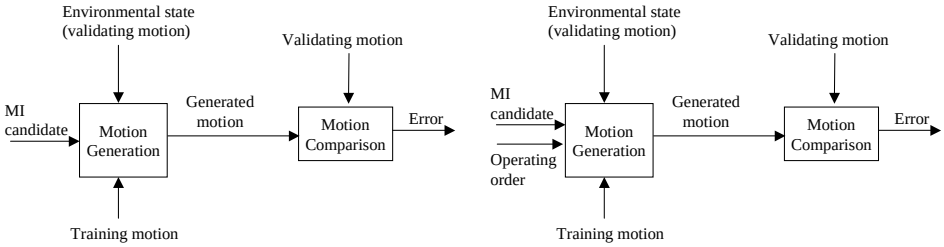
1: **for** $i = 1$ to n **do**
2: Select Q_i among the R recorded trajectories as the validating motion Q_V and the rest as the training demonstrations Q_T
3: Apply the method of dynamic programming, based on Eq. (3.10), to determine the optimal MI^* for Q_V
4: **end for**
5: Utilize Eq. (3.14) to determine the optimal MI^{**} for the demonstrator among those MI^* for the R validating motions
6: **return** MI^{**}

3.2 Similar Function

In this section, the reason why the similar function is used is first explained, and the implementation of the similar function, which is based on the dynamic time wrapping method, is then discussed.

3.2.1 Reasoning for Similar Function

Before we discuss the implementation of the similar function, the reason why we use the similar function in the validation is explained in the following. In the process of the intention deduction, it is important to decide the validating data and the



(a) with M.I. (b) with M.I. and operating order

Figure 3.8: The validation of M.I.: (a) with M.I. and (b) with M.I. and operating order.

training data because the fitness of each possible M.I. candidate is evaluated by the validation. In Fig. 3.8(a), the fitness of each M.I. candidate can be evaluated by comparing the difference between the generated motion and the validating motion in the motion comparison module, and the M.I. candidate which leads to the minimum error is the optimal M.I. candidate. However, the operating speed of the generated motion and the ordering of the delicate motions may be different from that of the validating motion because the motion generation module uses the M.I. candidate of the training motion to generate motion. The motion comparison module may need to use DTW to deal with the differences of motion speeds [56], but DTW cannot deal with the different ordering of delicate motions. Therefore, when DTW is used in the motion comparison module, we need to align the ordering of the delicate motions of the generated motion as that of the validating motion, as shown in Fig. 3.8(b). In Fig. 3.8(b), the motion speeds of the delicate motions are not included in the operating order because DTW can handle this.

With the concept of validation, the correct operating order and the optimal M.I. may be able to be searched simultaneously by validating all possible pairs of the operating order and the M.I. candidate when single training motion is inputted, as shown in Fig. 3.8(b). Moreover, the M.I. candidates of the training motion can be evaluated more accurately by inputting multiple validating motions with each operating order. However, the number of possible operating orders of each validating motions increases factorially with the number of the M.I.-assigned delicate motions. A lot of calculating time is needed to evaluate all operating order for each

M.I. candidate. In contrast, when multiple training motions and a single validating motion are inputted, the searching space is much larger because each training motion has an individual M.I. and an individual operating order. Therefore, we need a method to tackle this searching problem.

It is known that the optimal M.I. candidate leads to the minimum difference and the difference between two similar motions is smaller than two randomly chosen motions. Therefore, a shortcut method is proposed. The M.I. candidate of validating motion is inputted into the motion generation module without inputting the possible operating order candidate for the multiple training motions. The motion cutting module of the motion generation module is modified to use the similar function to minimize the difference between the delicate motion of the generated motion and the validating motion, as shown in Fig. 3.5(b). Note that this shortcut method may not be match the concept of validation.

We use the similar function to search for the part of the motion from each training motion as closely as possible to the M.I.-assigned delicate motion of the validating motion, so the difference between generated delicate motion and M.I.-assigned delicate motion can lead to minimum error. Besides, we use a DTW-like method to resample the delicate motion of generated motion in the calculation of the similar function, so the motion comparison module can calculate the difference without conducting DTW again.

3.2.2 Implementation of Similar Function

To calculate the similar function in Eq. (3.8), we use a DTW-like method, which uses the technique of dynamic programming to minimize the error between the delicate motions of the validating motion and the training motion. In the calculation of the similar function, first, the training motion Q_T^i is transformed to match the environment of the validating motion according to the M.I.-assigned operated object s_{V_j}. Then, the transformed motion is resampled to P_T^i for searching the minimum difference between the M.I.-assigned delicate motion of the validating motion and

some portion of the training motion. After that, the minimum error between the M.I.-assigned delicate motion of the validating motion and some portion of the training motion can be calculated as

$$E_{similar}{}^{*} = \min_{1 \le \varrho \le 2L_T^i - 1} H(\varrho, l_{V_j} - n_{V_j} + 1) \tag{3.17}$$

with

$$H(\varrho,\varsigma) = \begin{cases} \infty, & \text{if } \varrho \le 0, \\ ||P_T^i(\varrho) - D_{V_j}(\varsigma)||^2 + \min\limits_{\varrho-4 \le \iota \le \varrho-1} H(\iota, \varsigma - 1), & \text{if } \varrho \ge 1, \varsigma \ge 1, \\ 0, & \text{otherwise.} \end{cases} \tag{3.18}$$

where $E_{similar}{}^{*}$ is the minimum error between the delicate motions D_{V_j} and $D_{G_j}^i$, $H(\varrho,\varsigma)$ is a recursive function that outputs the minimum error between $D_{V_j}(1 \sim \varsigma)$ and $P_T^i(u \sim \varrho)$ (u is determined automatically during the process of minimization), and other symbols are explained in the following. The calculation of the similar function is different from DTW in two parts. First, in order to measure the difference independently without the influence of the time length of different generated motion, the generated motion is mapped and resampled to the time of the validating motion and the difference is measured based on the time of validating demonstration in the calculation of $H(\varrho,\varsigma)$. Because we want to search the minimum error under the dynamic speed whose range is from 1/2 to 2 times of the original speed of training motion, the number of the samples of P_T^i is $2L_T^i - 1$, which is the two times of the sampling rate of Q_T^i, and the four answers of the subproblems ($H(\varrho - 4, \varsigma - 1) \sim H(\varrho - 1, \varsigma - 1)$) are used to solve the problem $H(\varrho,\varsigma)$. Second, in order to check each possible start point of P_T^i, the number of the cases which $H(\varrho,\varsigma)$ are setted to zero is more than that using DTW because each sample points of P_T^i is a possible start point of the similar motion. Therefore, $H(\varrho,\varsigma)$ is a function that outputs the minimum error between $D_{V_j}(1 \sim \varsigma)$ and $P_T^i(u \sim \varrho)$, and the result of similar function can be obtained by tracking the selections in the calculation of the minimum error $E_{similar}{}^{*}$. Moreover, some elements of table $H(\varrho,\varsigma)$ can be shared for the different inputs of D_{V_j} to reduce the calculation time, so the time complexity of calculation of the similar function with the total possible inputs is $O(R \cdot {L_V}^3 \cdot S)$ when the range of the dynamic speed is fixed and the demonstration times of the validating motion and the training motions are assumed to be close.

Because the similar function is a shortcut method, which does not calculate the difference between the move motions of validating motion and generated motions, this method does not minimize the total difference between the validating motion and the generated motions. Although the penalty which is the expectative error between the move motions of the validating motion and the generated motions can be used in the calculation of the similar function, we do not use it in our experiment because the expectative error is difficult to be estimated precisely. Note that, in the experiment, it is observed that to create $E_R(D_V)$ table by using the similar function, many the same E_M values have been used. Therefore, if a cache is used to stand for the E_M, the calculation process can be reduced.

3.3 Experimental Design for Extensibility and Robustness

The proposed approach is developed for general tool-handling tasks, with the appealing features in (a) no need for pre-defined motions, (b) no constraints on motion speed or motion type, (c) allowance for event-order altering, and (d) allowance for redundant operations during demonstration. To further investigate its extensibility and robustness, we design a series of experiments based on three different kinds of tasks: the pouring, coffee-making, and fruit jam tasks.

In the pouring task, the operator is asked to hold a vessel and pour the content into other vessel(s), as described in the example shown in Fig. 3.2. The experiments are designed to evaluate the influence from the following factors:

- pouring order during execution;
- number of vessels to pour;
- number of demonstrations used for MI derivation.

We also analyze the time complexity during task execution, which is expected to match that predicted by the algorithm for intention derivation in Sec. 2.1. In the

coffee-making task, the operator uses a spoon to scoop coffee powder, sugar, or milk from the jars into the coffee cup and stir it. The number of jars are fixed for the experiments, but the operator can access the same jar(s) one or several several times. In addition to the performance on this coffee-making task, the effect of number of demonstrations on MI derivation and time complexity during task execution are also evaluated. To further explore its generality, in the fruit jam task, the operator picks up a knife from the table, scoops the fruit jam from the jar, spreads it on the toast in a zigzag motion, and places the knife back on the table. Meanwhile, we also analyze how the presence of the redundant motions affects system performance.

To simulate that the robot learns a task in a home environment, we assume that the robot uses a vision system to identify objects by comparing the difference between background and foreground in the multiple demonstrations for obtaining the positions of the objects. Moreover, the handled tool whose position is varied during the demonstration can also be identified. We suggest that the operated objects should change the positions in the multiple demonstrations, but some operated objects which are heavy or fixed may not be moved in the task. These objects cannot be identified by the vision system. Fortunately, no matter how many operated objects hidden in the background, these objects can be seen as a special object - background object whose position is not changed during demonstrations just like the origin. In contrast, although we suggest that the positions of the operated objects should be changed during the multiple demonstrations, we do not require that the positions of the unoperated objects cannot be changed. For example, there are three cups in the pouring task of two cups, it is allowed that the position of the unoperated cup is changed carelessly for some reason during demonstrations. Therefore, even some identified objects which are not operated in the demonstrations are inputted into the intention deduction module, our learning method can still handle it.

Chapter 4 Experiments

In this chapter, first, the system implementation is introduced. Then, the results of experiments of the pouring, coffee-making, and fruit jam tasks are reported, and its robustness is evaluated. Finally, we discuss how the proposed approach performs when compared with previous ones, and an application scenario is proposed for breakfast preparation to demonstrate the practicality of this approach in daily lives.

4.1 System Implementation

The proposed system is implemented for experiments, as shown in Fig. 4.1. In Fig. 4.1, the task environment consists of the objects, including the tool, the operated objects, and the background objects. The human operator can see the task environment and operate the objects. The positions and orientations of the objects are measured by the tracking system Polhemus FASTRAK. This tracking system Polhemus FASTRAK consists of a system electronics unit, receivers, a power supply, which are shown in Fig. 4.2, and a long range transmitter, which are shown in Fig. 4.3. The update rate of the positions (XYZ) and orientations (ZYX Euler angles) of receivers is 30Hz, and the specifications of this tracking system Polhemus FASTRAK is described in Table 4.1. After the human operator demonstrates a task multiple times, these operating trajectories are recorded as the 7-dimensional sequences, which consist of positions and orientations (in the form of quaternion) in 3 and 4 dimensions, respectively. The data for the position is normalized by its standard deviation to balance the effects of the errors of due positions and orientations. These trajectories with the positions and orientations of all possible operated objects are then inputted into our learning method to deduce the intention. The executable operating trajectory for the robot manipulation is generated as 7-dimensional sequences (positions and orientations) according to the positions and orientations of the operated objects of the robot-faced environment. In the experiment, the core of our learning method is programmed by using C language,

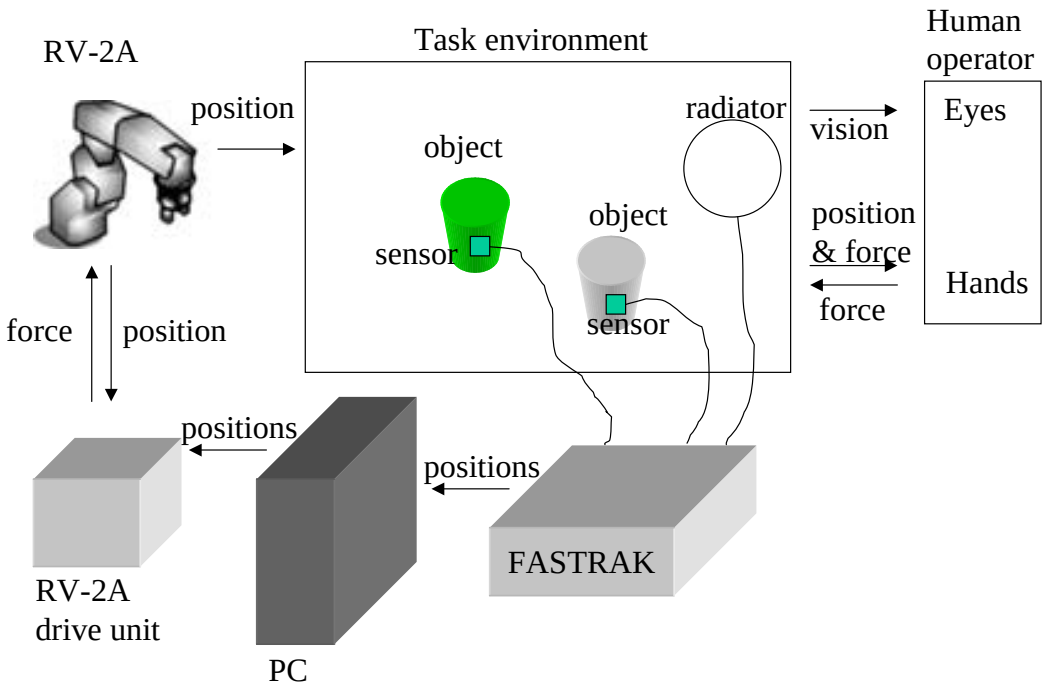


Figure 4.1: System implementation.

and the program is executed in the computer with CPU of Intel E6300, running at 1.86GHz with 3.62Gbyte RAM. Although this CPU has two cores, the program only uses one core to measure the time of the calculation. The robot manipulator Mitsubishi RV-2A is position-controlled, shown in Fig. 4.4, with its specifications listed in Table 4.2. The Denavit-Hartenberg parameters of this robot manipulator are specified in Table 4.3 [61], and the transformation matrix of each joint is defined as

$$A_i^{i-1}(\theta_i) = \begin{vmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{4.1}$$

This robot manipulator can accept the position command from internet by Ethernet interface card, and each position command can be operated at the operation control time 7.1 ms (≈141Hz). Therefore, the generated robot trajectory is resampled at 141Hz to perform robot manipulator, and the position command (J1∼J6) of each sampling point is solved by inverse kinematics [61]. In order to avoid damaging the devices, in the experiments, all vessels are demonstrated with empty content.

Figure 4.2: Tracking system Polhemus FASTRAK.

Figure 4.3: Long range transmitter.

Figure 4.4: Mitsubishi RV-2A type six-axis robot arm.

Table 4.1: Specifications of the tracking system Polhemus FASTRAK.

Latency		4 milliseconds
Interface		RS-232 with selectable baud rates up to 115.2K baud
Static Accuracy	position	0.03” RMS
	orientation	0.15 degrees RMS
Resolution	position	0.0002 inches
	orientation	0.025 degrees
Range		30 feet

Table 4.2: Specifications of the Mitsubishi RV-2A.

Degrees of freedom		6
Maximum load capacity (rating)		2Kg
Maximum reach radius		621mm
Working area	J1	320° (-160 to +160)
	J2	180° (-45 to +135)
	J3	120° (+50 to +170)
	J4	320° (-160 to +160)
	J5	240° (-120 to +120)
	J6	400° (-200 to +200)
Maximum speed (degree/s)	J1	150
	J2	150
	J3	180
	J4	240
	J5	180
	J6	330
Repeat position accuracy		±0.04mm

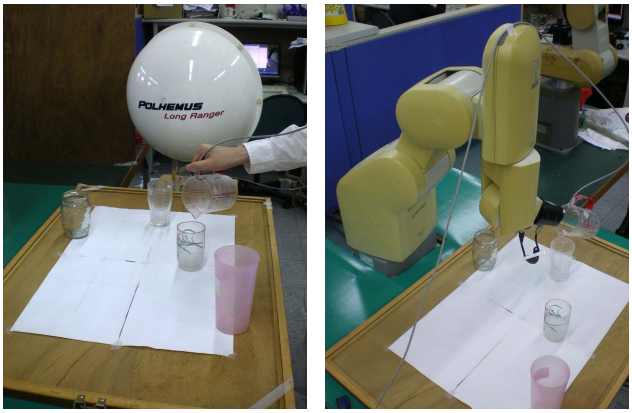
Table 4.3: Denavit-Hartenberg parameters of the Mitsubishi RV-2A.

Link	$a_i(m)$	$\alpha_i(rad.)$	$d_i(m)$	$\theta_i(rad.)$
1	0.1	$-\pi/2$	0.35	θ_1
2	0.25	0	0	$\theta_2 - \pi/2$
3	0.13	$-\pi/2$	0	$\theta_3 - \pi/2$
4	0	$\pi/2$	0.25	θ_4
5	0	$-\pi/2$	0	θ_5
6	0	0	0.24	θ_6

4.2 Pouring Task

We applied the proposed approach for the pouring task shown in Fig. 4.5 first. The experiment is divided into two stages: (a) human demonstration and (b) robot execution. Fig. 4.5(a) shows the experimental setup for human demonstration, which includes the human operator and the Polhemus FASTRAK tracking system. There are five vessels placed randomly on the table. The human operator held a vessel (vessel A) and poured the content into the other vessels (vessels B, C, D, and E) on the table. The Polhemus FASTRAK tracking system, with a sampling rate of 30 Hz for each of the sensors, was used to measure and record the demonstrated trajectories and positions of the objects. These trajectories were recorded as the 7-dimensional sequences, which consist of positions and orientations (in the form of quaternion) in 3 and 4 dimensions, respectively, with the position normalized by its standard deviation. From these recorded trajectories, we applied the intention deduction algorithm, discussed in Sec. 2.1, to derive the intention of the operator from all possible intentions. We then moved on to the second stage of the experiment, and let the Mitsubishi RV-2A 6-DOF robot manipulator follow the derived intention to execute the pouring task under new environmental states, as shown in Fig. 4.5(b). There are two changeable parameters of the pouring tasks in the experiment, which are the poured vessels ({B,C}, {B,C,D}, and {B,C,D,E}) and the pouring order (arbitrary order or same order), so there are 6 different settings of the pouring tasks. The human operator demonstrated 18 times in each setting of the pouring tasks, and the total number of demonstrations is 108.

To test the results of our method, for each pouring task (6 pouring tasks), the 8 of the demonstrations are randomly selected to be training data to be inputted into the learning method, and the rest demonstrations whose operating orders are equal to the operating orders of generated trajectories are selected to be testing data. In each test, the positions of vessel B$\sim$ E and origin are inputted to test the effect of the unoperated objects and the background object because the robot does not know which object is operated. These processes are executed 5 times, and the average errors, which describe the difference between the generated trajectories and



(a) Human demonstration (b) Robot execution

Figure 4.5: Experimental setups for the pouring task: (a) human demonstration and (b) robot execution.

the trajectories of the testing data, are calculated by DTW. Besides, we then moved on to the second stage of the experiment, and let the Mitsubishi RV-2A 6-DOF robot manipulator follow the generated trajectories to execute the pouring tasks under a new environmental state, as shown in Fig. 4.5(b).

Fig. 4.6 shows the derived intentions for each of the eight demonstrations of the pouring task in one test. Because the tilt-angle changes are the clear features of pouring motions, the time series graphs which describe the tilt-angle of the trajectories are used to illustrate the results. In Fig. 4.6, delicate motions related to vessels B, C, D, and E were identified from the trajectory of vessel A, marked by the yellow, green, blue, and purple blocks, respectively. It was observed that most of the delicate motions were located at those portions with evident tilt-angle changes, implicating the pouring action. The derived intention for demonstration 3 was determined to be optimal among all.

Fig. 4.7 shows one of the generated trajectories and one of the trajectories of the testing data. In Fig. 4.7, the black line is the trajectory of the testing data, and the color line is the generated trajectory, which consists of the red lines (move motions) and the other color lines (delicate motions for operated objects). In XY-plane subfigure of Fig. 4.7, the color rectangles (yellow,green,blue, and purple) describe the positions of the operated object (B,C,D, and E). In the XY-plane

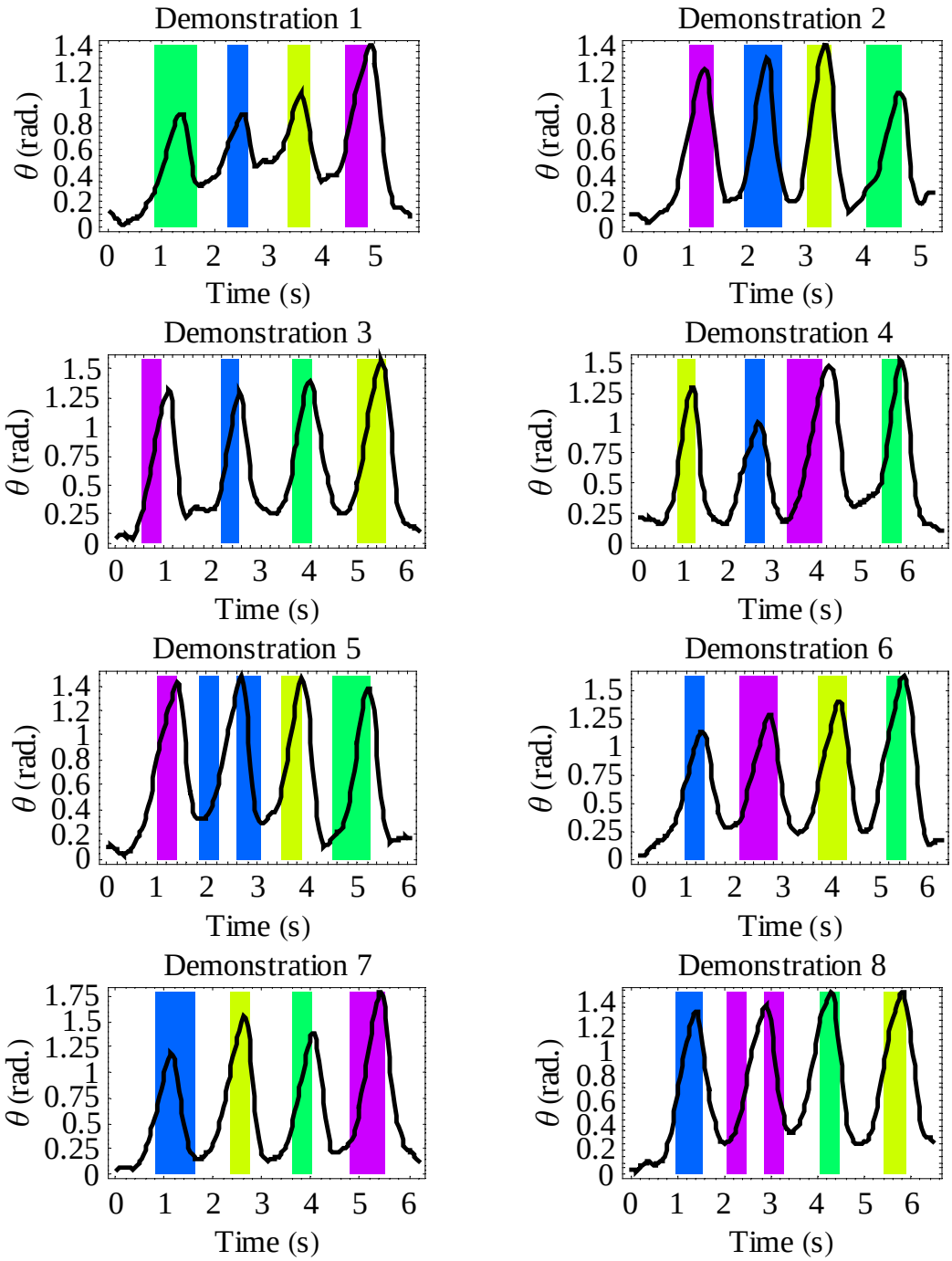


Figure 4.6: The derived intentions for the eight demonstrations of the pouring task.

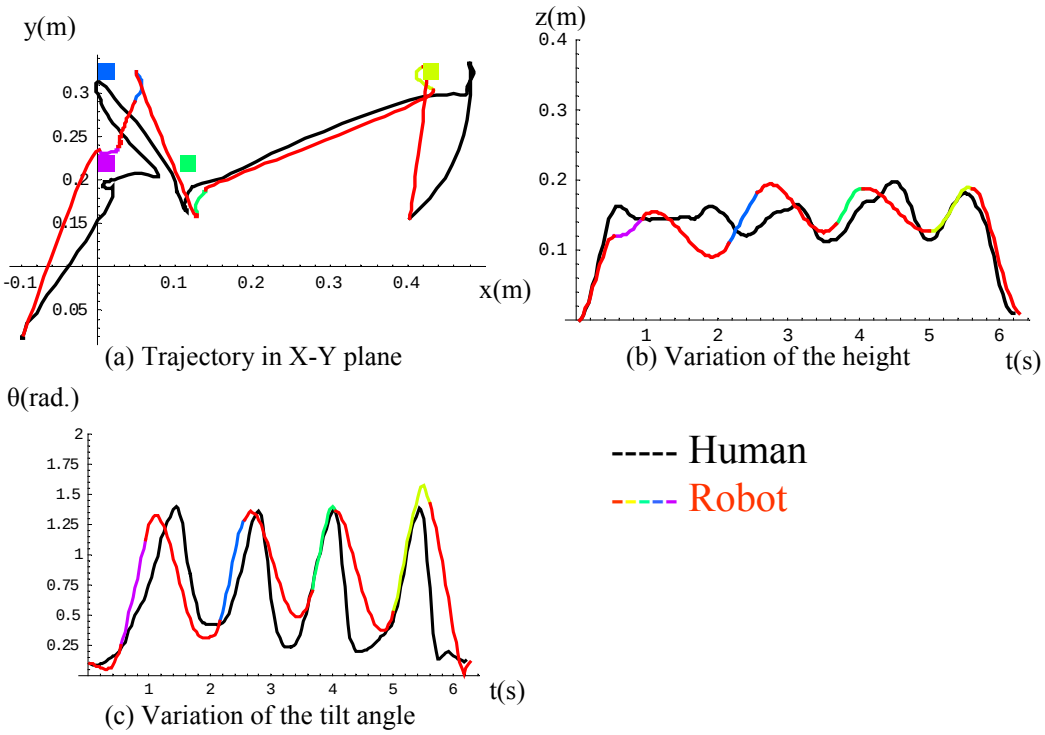


Figure 4.7: Experimental results for the pouring 4 cups task executed by both the human operator and robot manipulator under new environmental states: (a) trajectory in X-Y plane, (b) variation of the height, and (c) variation of the tilt angle of vessel A.

subfigure and the height subfigure and the tilt angle subfigure of Figure 4.7, it was observed that the most of the delicate motions were located at those positions near the operated objects and those portions with minimum height and those portions with maximum tilt angle because these features are the features of the pouring motions. This generated trajectory and human trajectory exhibited certain degree of similarity, but not exactly the same. However, the generated trajectory can be used to manipulate the robot to accomplish the pouring tasks. Moreover, the generated trajectories of the other tasks show the correct delicate motions and match the human operations.

Table 4.4: Average position error between the trajectories of human operator and the generated trajectories in the pouring tasks.

Pouring task	Order	Error (m)
2 cups	Same	0.052
	Arbitrary	0.051
3 cups	Same	0.049
	Arbitrary	0.049
4 cups	Same	0.052
	Arbitrary	0.045

Table 4.4 shows average position errors between the trajectories of the human operator and the generated trajectories for the six combinations of the pouring task, and these error are calculated by DTW on 3D positions. First, it was observed that the orders of the operations do not have a great effect upon the results of our method. Second, it was observed that the number of the operated objects does not have a great effect upon the results of our method.

4.2.1 Robustness in Pouring Tasks

Figure 4.8 shows the relation between the errors and the number of demonstrations (2 to 8) in the pouring 2˜4 cups task, as shown in Table 4.4. First, it was observed that our method may need 4˜5 demonstrations to learn the goals of the 2˜4-cup

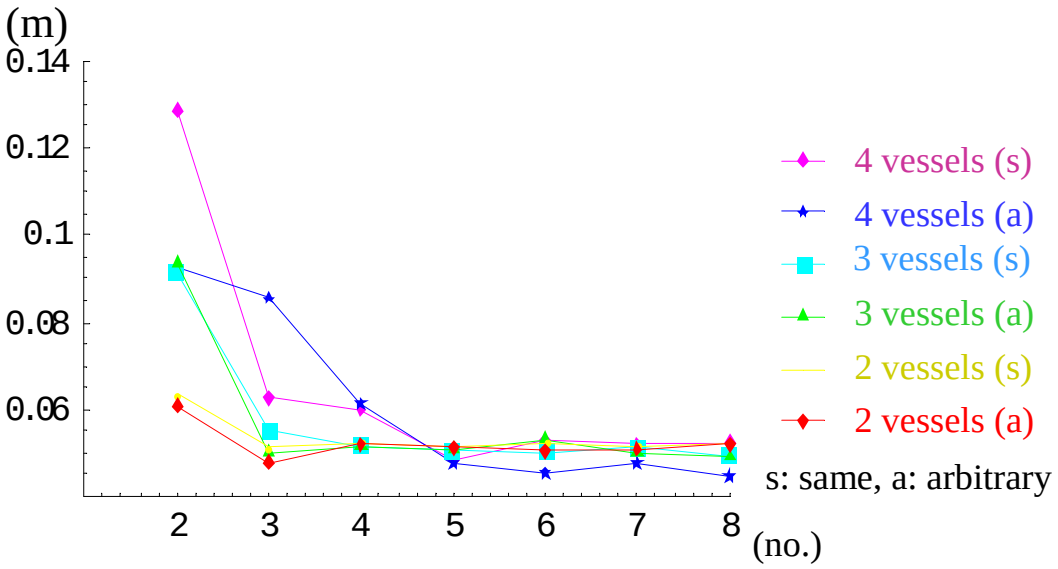


Figure 4.8: The errors of the generated trajectories with the different numbers of the training demonstrations in the pouring 2˜4 cups task.

pouring tasks. Second, it was observed that the number of demonstrations which is needed to decrease the errors of the 4-cup pouring task is more than those for the 2˜3-cup pouring tasks. It implies that the difficulty of learning task increases with the number of the poured cups. Third, it was observed that many errors of the arbitrary order pouring tasks are smaller than that of the same order pouring tasks. It implies that human operator may introduce more unusual operations in the same order pouring tasks because human operator needs to pay more attentions to unnatural operating orders during demonstrations. As for the time of the calculation of the pouring tasks, Fig. 4.9 shows that it increases approximately squarely with the the number of the demonstrations. This matches the expectation of the time complexity.

We further evaluate how the presence of the redundant operations during demonstration may affect system performance. The analysis was based on the 2-vessel pouring task. Among a group of 2-vessel demonstrations, we gradually added in some demonstrations involving three or four vessels, taken as the introduction of the redundant operations. With this, we attempted to find out whether the proposed approach could still successfully recognize that the demonstrations were intended for the 2-vessel pouring task, even some proportion of the demonstrations

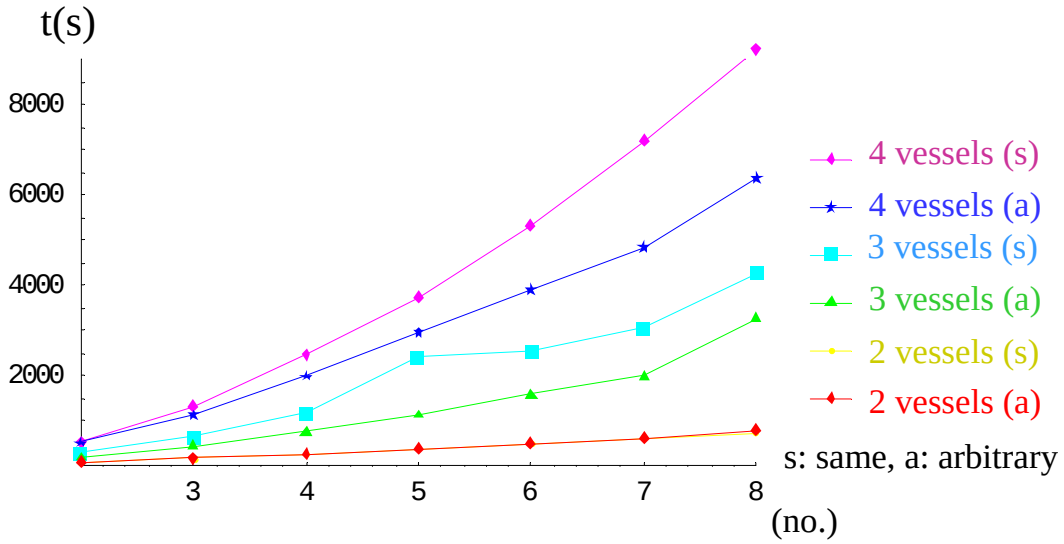


Figure 4.9: The calculating time with the different numbers of demonstrations in the six pouring tasks.

involving redundant operations. Table 4.5 lists the number of success out of 10 tests where the number of demonstrations involving 3 or 4 vessels increased from 1 to 4 out of a total 8 demonstrations. Because the redundant operations are not common motions which can not be searched in each training motion in Eq. (3.8), larger percent of the redundant operations usually led to larger E_{max} in deriving the optimal MI, as shown in Eq. (3.14). When the demonstrations involving redundant operations consisted of half of the total demonstrations, the proposed approach can still reach a high success rate at 80%.

Table 4.5: Number of success in the presence of redundant operations

2 vessels	3 vessels	4 vessels	number of success
7	1	0	9
6	1	1	9
5	2	1	8
4	2	2	8

About the operating order of the tasks, in the experiment, it is observed that the path length and demonstration time of the tasks whose operating orders are arbitrary are smaller than that of the task whose operating orders are same, as

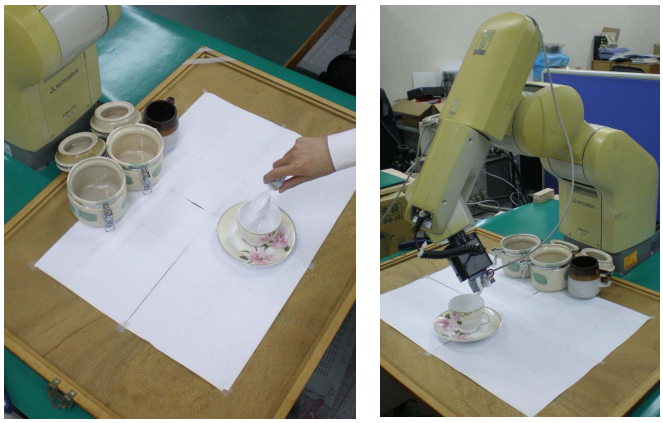
shown in Table 4.6. Therefore, the arbitrary order of the operations can decrease the path length and the demonstration time of the operations. Besides, it can decrease the probability of the mistakes of the operations, the pause motions, and the redundant operations because the operator does not need to care about the order of the operations when he/she demonstrates a complex task. From Fig. 4.9 and Table 4.6, it clear that the calculation time increases with the demonstration time. It means that the arbitrary operating orders can also decrease the calculation time in the pouring tasks. Therefore, in order to decrease the calculation time of our method, the operator should accomplish each demonstration as fast as possible and do not care about the operating order in each demonstration.

Table 4.6: The average path length and demonstration time in the pouring tasks.

		path length (m)	demonstration time (s)
Pouring 2 cups	Same order	1.017	3.450
	Arbitrary order	0.973	3.452
Pouring 3 cups	Same order	1.436	4.900
	Arbitrary order	1.320	4.569
Pouring 4 cups	Same order	1.776	6.396
	Arbitrary order	1.523	5.856

4.3 Coffee-making Task

In the coffee-making tasks, a spoon (spoon A) and a vessel (vessel B) are placed randomly on an area of 0.21x0.21 $meter^2$ on the table in each demonstration, and three vessels (vessel C, D, and E) are placed on the assigned positions where are not changed during the experiment, as shown in Fig. 4.10. Spoon A is always held as a tool and operated to do the scooping motions and stirring motions on the vessels. Two different coffee-making tasks are designed to test the effects of the repeated operations. The motions of the first coffee-making task are using spoon A to scoop the content of vessel C into vessel B, to scoop the content of vessel D into vessel B, to scoop the content of vessel E into vessel B, and to stir the content of vessel B. The



(a) Human demonstration (b) Robot execution

Figure 4.10: Experimental setups for the coffee-making task: (a) human demonstration and (b) robot execution.

motions of the second coffee-making task are using spoon A to scoop the content of vessel C into vessel B, to scoop the content of vessel C into vessel B, to scoop the content of vessel D into vessel B, and to stir the content of vessel B. Both coffee-making tasks are demonstrated 22 times, and the total number of demonstrations is 44.

In the coffee-making task, the Polhemus FASTRAK tracking system, with a sampling rate of 30 Hz for the sensors attached on the spoon A, were used to measure the demonstrated trajectories. These trajectories were recorded as the 7-dimensional sequences, which consist of positions and orientations (in the form of quaternion) in 3 and 4 dimensions, respectively, with the position normalized by its standard deviation.

In the coffee-making tasks, vessel C, D, and E never change positions in the training data. In order to recognize the operated objects in the background, a background object is created to replace all background objects, and the position of the background object is the origin. Therefore, in the coffee-making tasks, only two operated objects are inputted, which are the vessel B and the background object.

For each coffee-making task (2 coffee-making tasks), the 8 of the demonstrations are randomly selected to be training data to be inputted into the learning method, and the rest demonstrations are selected to be testing data. These processes are

executed 5 times, and the average errors, which describe the difference between the generated trajectories and the trajectories of the testing data, are calculated by DTW. Then, we let the Mitsubishi RV-2A 6-DOF robot manipulator follow the generated trajectories to execute the coffee-making tasks under new environmental states, as shown in Fig. 4.10(b).

Figs. 4.11 and 4.12 show the derived intentions for each of the eight demonstrations of the two coffee-making tasks. Because the height changes are the clear features of scooping motions, the time series graphs which describe the height of the trajectories are used to illustrate the results. Because the locations of the three vessels (vessels C, D, and E) were not changed during the experiment, they were viewed as one object fixed on the origin. The two colors blocks mark the delicate motions (green for vessels C, D, and E, and yellow for vessel B). The derived intention for demonstration 2 (in coffee-making task 1) and 1 (in coffee-making task 2) were determined to be the optimal among all.

Figs. 4.13 and 4.14 show the generated trajectories and the trajectories of the testing data in the two coffee-making tasks. In the subfigures of Figs. 4.13 and 4.14, the black line is the trajectory of the testing data, and the color line is the generated trajectory, which consists of the red lines (move motions) and other color lines (delicate motions for operated objects). In each XY-plane subfigure, the color rectangles (yellow,green,blue, and purple) indicate the positions of the operated object (B,C,D, and E). Because there are only two operated objects, which are the vessel B and the background object, in each subfigure of the coffee-making tasks, the delicate motions which scoop the content of vessel C, D, and E are green lines, which operate only on the background object, but the operated positions of the background object are different. Although the background objects of both two coffee-making tasks are operated, the difference of the operated objects of the generated trajectories are observed clearly in each coffee-making tasks. These generated trajectories and human trajectories exhibited certain degree of similarity, but not exactly the same. However, the generated trajectories can be used to manipulate the robot to accomplish the coffee-making tasks.

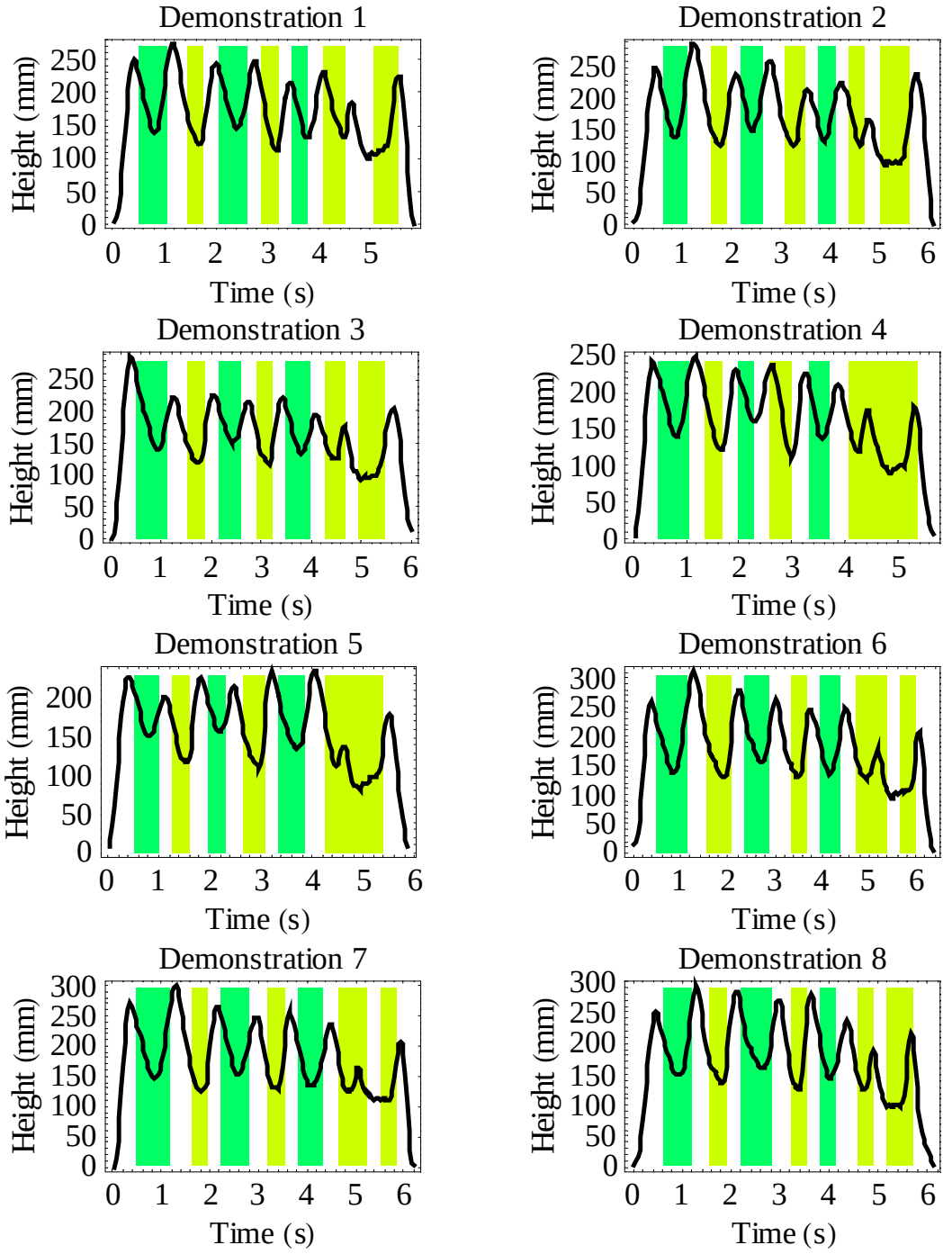


Figure 4.11: The derived intentions for the eight demonstrations of the coffee-making task 1.

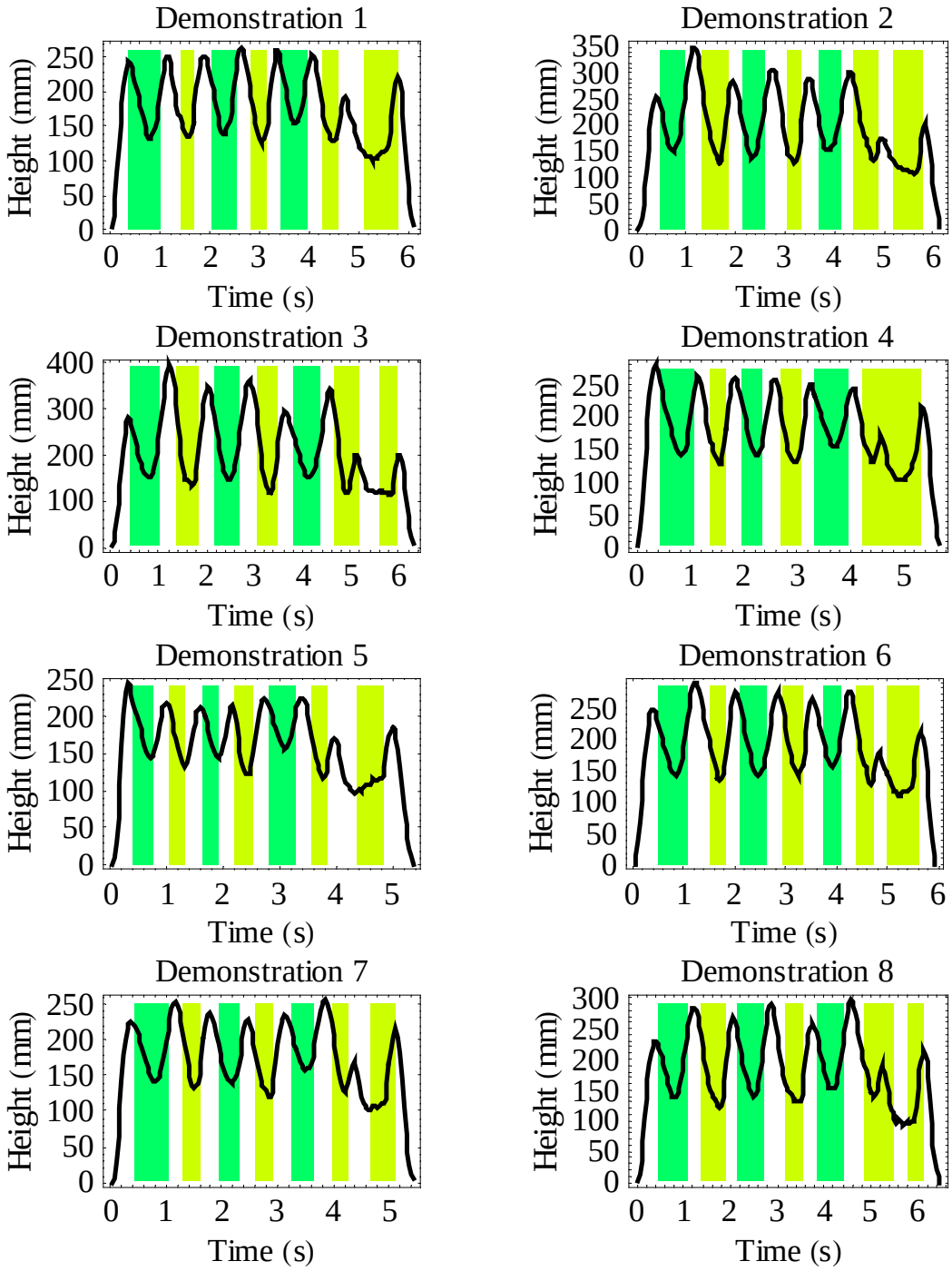


Figure 4.12: The derived intentions for the eight demonstrations of the coffee-making task 2.

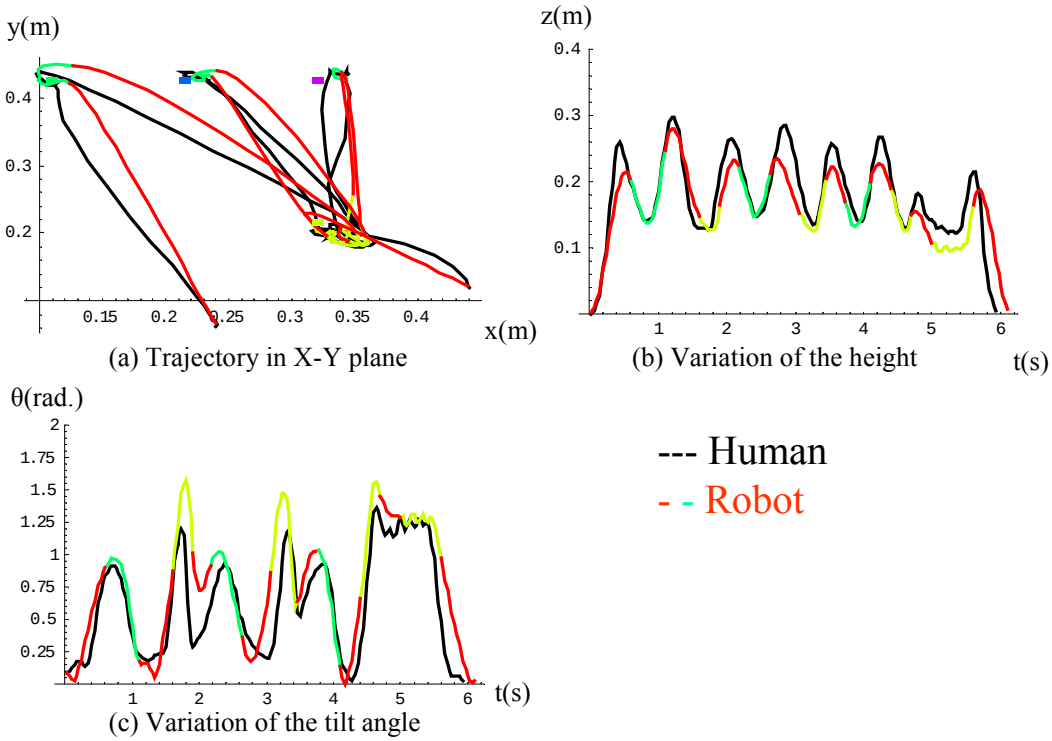


Figure 4.13: Experimental results for the coffee-making task 1 executed by both the human operator and robot manipulator under new environmental states: (a) trajectory in X-Y plane, (b) variation of the height, and (c) variation of the tilt angle of spoon A.

Table 4.7: Average position error between the trajectories of human operator and the generated trajectories in the coffee-making tasks.

Task	order	Error (m)
Coffee-making	Pattern 1	0.030
	Pattern 2	0.027

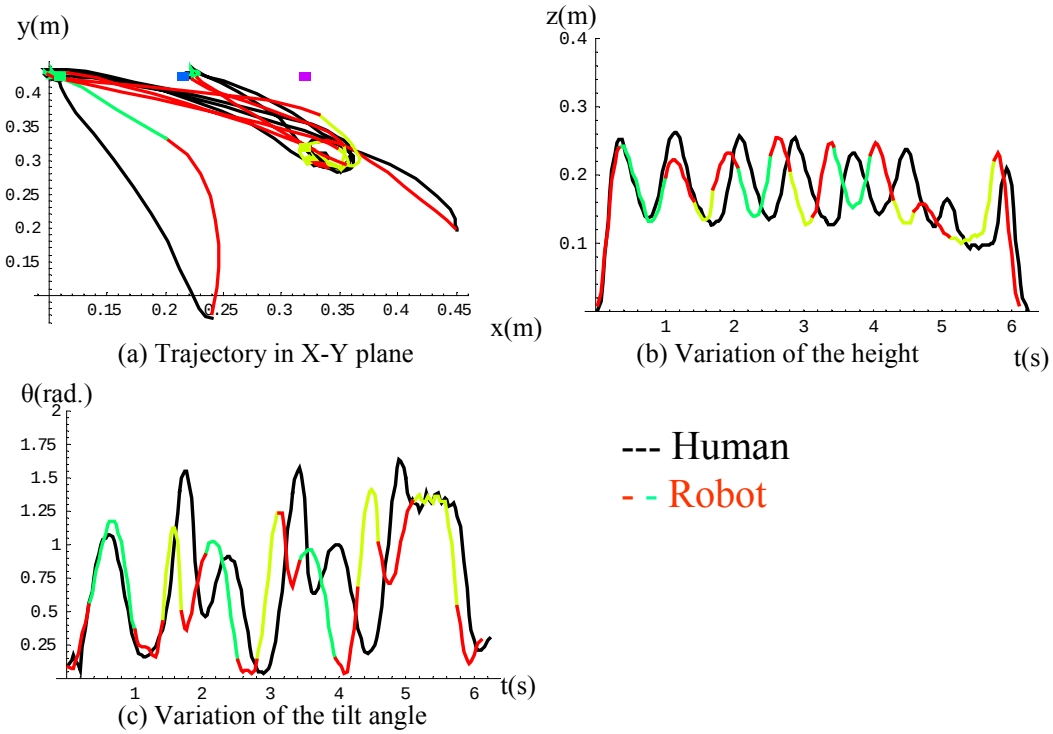


Figure 4.14: Experimental results for the coffee-making task 2 executed by both the human operator and robot manipulator under new environmental states: (a) trajectory in X-Y plane, (b) variation of the height, and (c) variation of the tilt angle of spoon A.

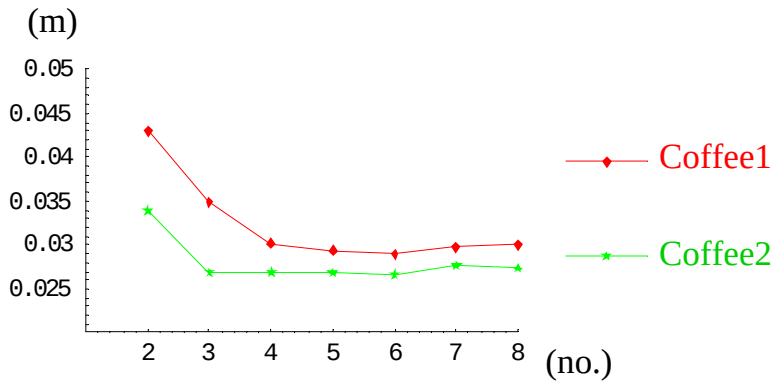


Figure 4.15: The errors of the generated trajectories with the different numbers of the training demonstrations in the coffee-making tasks.

Table 4.7 shows the average errors between the trajectories of the testing data and the generated trajectories for the two coffee-making tasks, and these errors are calculated by DTW on 3D positions. In Table 4.7, first, it was observed that the different operations of two coffee-making tasks do not have a great effect upon the results of our method. Second, because the working space of the coffee-making tasks is smaller than that of the pouring tasks, it may cause that the error of the coffee-making tasks is smaller than that of the pouring tasks.

4.3.1 Robustness in Coffee-making Tasks

Figure 4.15 shows the relation between the errors and the number of demonstrations (2 to 8) in the two coffee-making tasks, as shown in Table 4.7. First, it was observed that our method may needs 4 demonstrations to learn the goals of the coffee-making tasks. Second, it was observed that the errors of the first coffee-making task is larger than that of the second, and it implicates that learning the first coffee-making task is more difficult than learning the second one.

Because the coffee-making tasks has only two objects (vessel B and background object) and the demonstration time of each demonstration is almost equal, we compare the time complexity with the time of the calculation by changing the number of the demonstrations. Fig. 4.16 shows that the time of the calculation of the coffee-making tasks increases approximately squarely with the the number of the

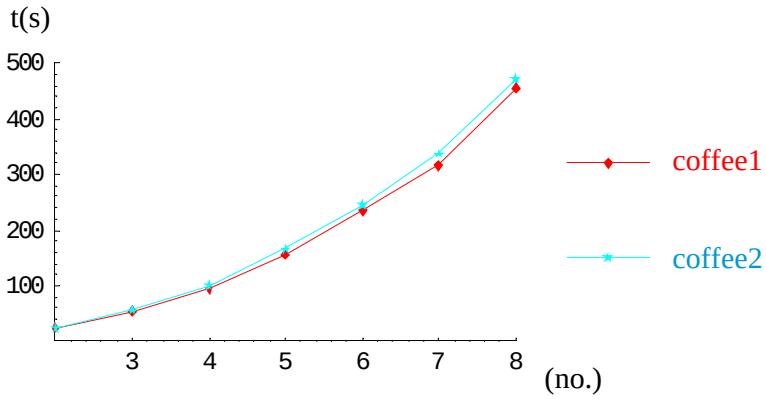
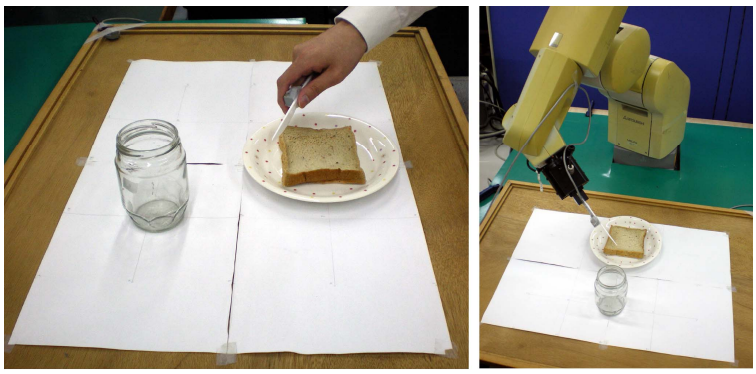


Figure 4.16: The calculating time with the different numbers of demonstrations in the two coffee-making tasks.



(a) Human demonstration (b) Robot execution

Figure 4.17: Experimental setups for the fruit jam task: (a) human demonstration and (b) robot execution.

demonstrations. This matches the expectation of the time complexity. Fig. 4.16 also shows that the difference between the calculation time of two coffee-making tasks is not evident. This is because the number of the operated objects of two coffee-making tasks are the same and the demonstration time of each demonstration is almost equal. Therefore, although learning the first coffee-making task is more difficult than learning the second one, the difficulties of the tasks do not influence the calculation time evidently.

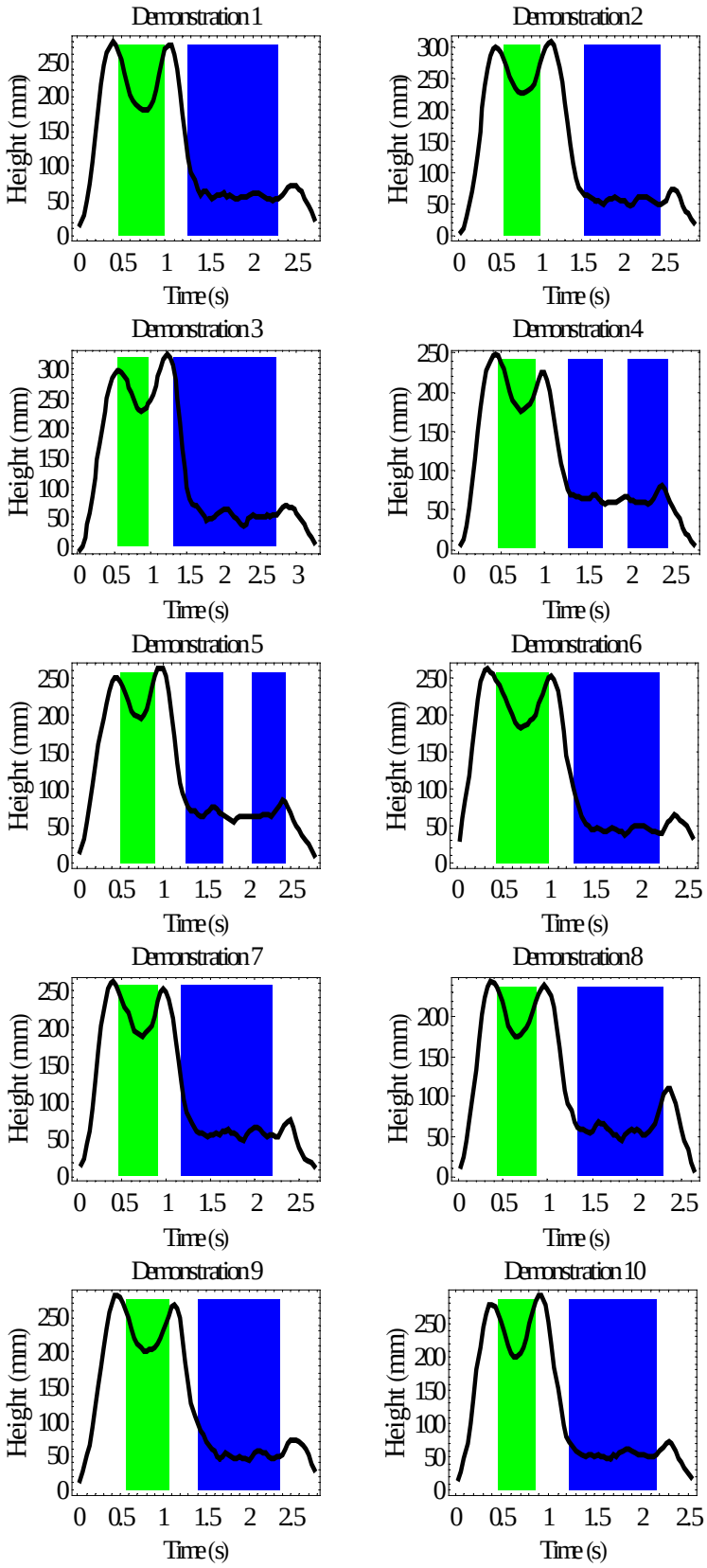


Figure 4.18: The derived intentions for the ten demonstrations of the fruit jam task.

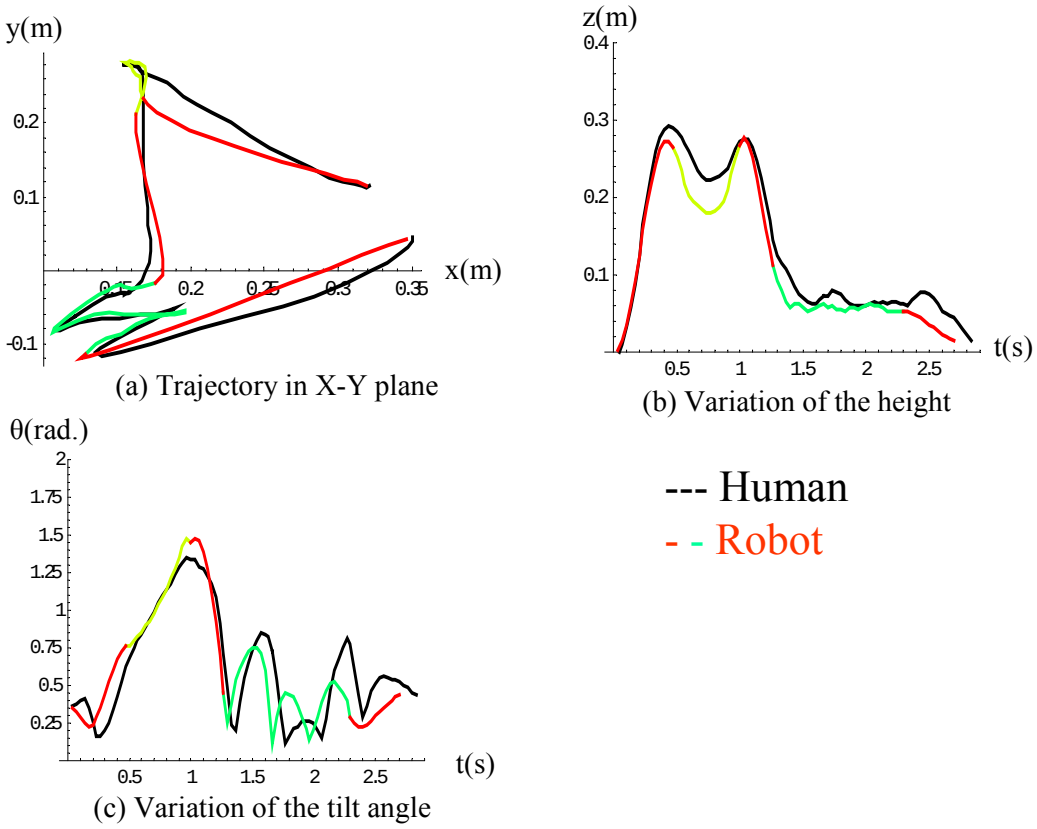


Figure 4.19: Experimental results for the fruit jam task executed by both the human operator and robot manipulator under new environmental states: (a) trajectory in X-Y plane, (b) variation of the height, and (c) variation of the tilt angle of knife A.

4.4 Fruit Jam Task

To further explore its generality, we also apply the proposed approach for a fruit jam task, in which the robot learns how to spread fruit jam on toast. Fig. 4.17 shows the experimental setup, which was also divided into the stages of (a) human demonstration and (b) robot execution. In Fig. 4.17 (a), the human operator picked up a knife from the table, scooped the fruit jam from the jar, spread it on the toast in a zigzag motion, and then placed knife back on the table. A total of ten demonstrations had been performed, with the locations of the knife, jar, and toast varied during each demonstration. The intention deduction algorithm was then applied to derive the intention of the operator from all possible intentions. Fig. 4.18 shows the derived intentions for each of the ten demonstrations. In Fig. 4.18, delicate motions related to the jar and toast were identified from the trajectory of the knife, marked by the green and blue blocks, respectively. It was observed that these two delicate motions were located at the portions of the first local minimum and a flat region following, implicating the scooping and zigzag motions. The derived intention for demonstration 1 was determined to be optimal. We then let the robot manipulator utilize this intention to execute the fruit jam task, in which the jar and toast were placed in new locations. And, the human operator was also asked to execute the same task. Fig. 4.19 (a) shows the variations of the height of the knife during task execution, Fig. 4.19 (b) its trajectories in the X-Y plane, and Fig. 4.19 (c) the variations of its tilt angle. Similar to the phenomenon exhibited in the pouring and coffee-making tasks, the trajectories for the human operator and robot manipulator were not exactly the same, while the robot manipulator successfully accomplished the task.

4.5 Discussion

4.5.1 Comparison with Previous Approaches

It shall better demonstrate the capability of the proposed approach by comparing its performance with those of previous works. However, good benchmarks or methods are still in demand for this field [5]. According to our survey, the illustration in Calinon [9,38] provides a good example for demonstration. In [9], to illustrate the objects are operated correctly, the operating trajectory is projected into the X-Y plane and the positions of the objects are marked to show that the motions operate the objects correctly. To illustrate the object-operating motions, the time series graph which describes the probability distribution of the multiple trajectories is used to show that the motions operate the correct objects at the correct moment. With these merits, we thus illustrate our results in a similar way, which exhibits the delicate motions operate the objects correctly at the correct moment. Moreover, in order to evaluate the correctness of the generated trajectory, we compare the generated trajectory with that of the human demonstration. As the motion of the human demonstration may not be perfect, it can be used to distinguish the wrong operations.

Figs. 4.7, 4.13, 4.14, and 4.19 show that our approach can generate robot trajectories that are similar to the human trajectories, which operate objects correctly. Figs. 4.6, 4.11, 4.12, and 4.18 show that the derived delicate motions correspond those of the human and operate the objects correctly. In Figs. 4.8 and 4.15, the number of demonstrations needed by the proposed approach to deduce the intention increases with the number of operated objects. However, Fig. 4.9 shows that the calculating time increases steeply with the number of operated objects and the demonstration time. This result can be expected from the time complexity $O(R^2 \cdot {L_V}^5 \cdot S^2)$, pointing out a major limitation of the proposed approach.

4.5.2 Application Scenario for Breakfast Preparation

To illustrate how to apply the proposed approach for tasks in daily lives, we take the task of breakfast preparation as an example, which consists of several sub-tasks. The breakfast includes one cup of coffee and one slice of toast with fruit jam. For analysis, each sub-task is distinguished by different tools. During preparation, a set of objects are put on the table first, including a cup, a spoon, a bottle of hot water, a coffee jar, a sugar jar, a milk bottle, a disk, a knife, a slice of toast, and a fruit jam jar. Because these objects may be placed on the table randomly, the positions of these objects are inputted into the scheme in each demonstration.

First set of sub-tasks are removing and opening the lid of each jar and the cover of each bottle. The lid or cover can be viewed as a handled tool and the jar or bottle as the operated object, so that these sub-tasks can be analyzed as the tool-handling tasks. After that, the preparation of coffee task can be divided into the sub-tasks of pouring hot water, adding coffee and sugar, and pouring milk, which can be analyzed as the demonstrated pouring and coffee-making tasks. Note that the weight sensors or vision may be needed to measure the height changes of the contents of the coffee jar and the sugar jar for learning the scooping motions in the coffee-making task. In turn, the sub-task of spreading the toast with fruit jam can be analyzed as that in the experiment. Similar to this breakfast preparation, various tasks in daily lives can be deal with alike.

Chapter 5 Conclusions

In this chapter, the conclusions are presented and suggestions are stated for future research.

5.1 Conclusions

In this dissertation, we have proposed a novel approach for intention deduction from demonstrated trajectories for tool-handling tasks. The proposed approach (a) does not need to pre-define motions, (b) does not constrain the operator to perform the task with certain motion speed or motion type, (c) allows the order of the events to be altered, and (d) allows some redundant operations. In realization, the concept of cross-validation and the algorithm based on dynamic programming have been employed to search for the optimal intention. We have performed a series of experiments and analyses to demonstrate its extensibility and robustness based on the pouring, coffee-making, and fruit jam tasks. As arbitrary order of the operations decreases path length and demonstration time, the demonstration can thus be executed in a natural and effective manner.

5.2 Future Research

First, the proposed approach would be applied for more various types of tasks related to home-like environments. However, the time complexity $O(R^2 \cdot {L_V}^5 \cdot S^2)$ means that this approach cannot handle very complex task with long demonstration time. Therefore, more efficient methods for intention deduction should be developed. Second, in our implemented system, the user still needs to prepare sensors for the operated objects and make an adjustment. To avoid preparing sensors for all possible operated objects, the implemented system should be improved with a better vision system, so the robot can observer the human demonstration in the background without interfering with the daily work. Third, in our approach, the move

motions are generated by the cubic polynomial to smoothly connect the delicate motions, but, in the real world, these move motions may lead the tool or the robot self to collide with other objects. Therefore, the move motions should be generated not only to smoothly connect the delicate motions but also to let the tool be moved safely without collision.

References

[1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms and Implementation*. MIT Press, 2005.

[2] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proceedings of the IEEE international Conference on Robotics and Automation*, 2009.

[3] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 799–822, 1994.

[4] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Survey: Robot programming by demonstration," in *Chapter 59, Handbook of Robotics*, 2008.

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

[6] M. Lopes, F. Melo, L. Montesano, and J. Santos-Victor, *Abstraction Levels for Robotic Imitation: Overview and Computational Approaches*. Springer, 2010.

[7] S. Schaal, "Is Imitation Learning the Route to Humanoid Robots?," *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.

[8] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 286–298, 2007.

[9] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[10] C. L. Nehaniv and K. Dautenhahn, *The correspondence problem*. MIT Press, 2002.

[11] A. Alissandrakis, C. L. Nehaniv, K. Dautenhahn, and J. Saunders, "Using jabberwocky to achieve corresponding effects: Imitating in context across multiple platforms," in *Workshop on the Social Mechanisms of Robot Programming by Demonstration, IEEE International Conference on Robotics and Automation*, 2005.

[12] A. Alissandrakis, C. L. Nehaniv, K. Dautenhahn, and J. Saunders, "Achieving corresponding effects on multiple robotic platforms: Imitating in context using different effect metrics," in *International Symposium on Imitation in Animals and Artifacts*, 2005.

[13] M. Pardowitz, R. Zöllner, and R. Dillmann, "Incremental learning of task sequences with information-theoretic metrics," in *European Robotics Symposium*, 2006.

[14] M. Pardowitz, S. Knoop, R. Dillmann, and R. D. Zollner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 2, pp. 322–332, 2007.

[15] K. Ogawara, J. Takamatsu, H. Kimura, and K. Ikeuchi, "Extraction of essential interactions through multiple observations of human demonstrations," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 4, pp. 667–675, 2003.

[16] D. A. Baldwin and J. A. Baird, "Discerning intentions in dynamic human action," *Trends in Cognitive Sciences*, vol. 5, no. 4, pp. 171–178, 2001.

[17] K. Ogawara, J. Takamatsu, H. Kimura, and K. Ikeuchi, "Extraction of fine motion through multiple observations of human demonstration by dp matching and combined template matching," in *Workshop on Robot and Human Interactive Communication, IEEE International Conference on*, 2001.

[18] K. Ogawara, S. Iba, T. Tanuki, H. Kimura, and K. Ikeuchi, "Recognition of human task by attention point analysis," in *IEEE International Conference on Intelligent Robot and Systems*, 2000.

[19] G. E. Hovland, P. Sikka, and B. J. Mccarragher, "Skill acquisition from human demonstration using a hidden markov model," in *Robotics and Automation, Proceedings of IEEE International Conference on*, 1996.

[20] S. Ekvall and D. Kragic, "Robot learning from demonstration: a task-level planning approach," *International Journal of Advanced Robotic Systems*, vol. 5, no. 3, pp. 223–234, 2008.

[21] B. Jansen and T. Belpaeme, "A computational model of intention reading in imitation," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 394–402, 2006.

[22] A. L. Thomaz and C. Breazeal, "Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance," in *Proceedings of the 21st national conference on Artificial intelligence*, 2006.

[23] J. Yang, Y. Xu, and C. S. Chen, "Human action learning via hidden Markov model," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 27, no. 1, pp. 34–44, 1997.

[24] L. M, N. Demmel, and M. Beetz, *Becoming Action-aware through Reasoning about Logged Plan Execution Traces*, 2010.

[25] S. K. Tso and K. P. Liu, "Hidden Markov model for intelligent extraction of robot trajectory command from demonstrated trajectories," in *Industrial Technology, Proceedings of The IEEE International Conference on*, 2002.

[26] W. Erlhagen, A. Mukovsky, E. Bicho, and H. Bekkering, "Development of action understanding in imitation tasks," in *Workshop on The Social Mechanisms of Robot Programming by Demonstration, Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[27] S. Ekvall, D. Aarno, and D. Kragic, "Online task recognition and real-time adaptive assistance for computer-aided machine control," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 1029–1033, 2006.

[28] S. R. Schmidt-rohr, L. Martin, J. Rainer, and R. Dillmann, "Programming by demonstration of probabilistic decision making on a multi-modal service robot," *Analysis*, pp. 784–789, 2010.

[29] J. Lieberman and C. Breazeal, "Improvements on action parsing and action interpolation for learning through demonstration," in *Proceedings of the 4th IEEE International Conference on Humanoid Robots*, 2004.

[30] A. Chella, H. Dindo, and I. Infantino, "A cognitive framework for learning by imitation," in *Workshop on The Social Mechanisms of Robot Programming by Demonstration, Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[31] J. Fritsch, N. Hofemann, and K. J. Rohlfing, "Detecting when to imitate in a social context with a human caregiver," in *Workshop on The Social Mechanisms of Robot Programming by Demonstration, Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[32] S. Ekvall and D. Kragic, "Learning task models from multiple human demonstrations," in *Robot and Human Interactive Communication, The 15th IEEE International Symposium on*, 2006.

[33] J. Lieberman, *Teaching a robot manipulation skills through demonstration*. Massachusetts Institute of Technology, Dept. of Mechanical Engineering, 2004.

[34] O. C. Jenkins and M. J. Mataric, "Deriving action and behavior primitives from human motion data," in *In International Conference on Intelligent Robots and Systems*, 2002.

[35] W. Takano, H. Imagawa, D. Kulic, and Y. Nakamura, "What do you expect from a robot that tells your future? the crystal ball.," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[36] M. J. Mataric, *Sensory-Motor Primitives as a Basis for Imitation: Linking Perception to Action and Biology to Robotics*. MIT Press, 2000.

[37] R. Chalodhorn, K. MacDorman, and M. Asada, "Automatic extraction of abstract actions from humanoid motion data," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2004.

[38] S. Calinon and A. Billard, "Statistical Learning by Imitation of Competing Constraints in Joint Space and Task Space," *Advanced Robotics*, vol. 23, no. 15, pp. 2059–2076, 2009.

[39] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Learning dynamical system modulation for constrained reaching tasks," in *Proceedings of the IEEE/RAS International Conference on Humanoid Robots*, 2006.

[40] S. Calinon and A. Billard, "A framework integrating statistical and social cues to teach a humanoid robot new skills," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2008.

[41] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, 2007.

[42] S. Calinon, F. Guenter, and A. Billard, "On learning the statistical representation of a task and generalizing it to various contexts," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2006.

[43] S. Calinon, F. Dhalluin, D. Caldwell, and A. Billard, "Handling of multiple constraints and motion alternatives in a robot programming by demonstration framework," in *Proceedings of IEEE International Conference on Humanoid Robots*, 2009.

[44] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 370–384, 2006.

[45] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1463–1467, 2008.

[46] S. Calinon and A. Billard, "Active teaching in robot programming by demonstration," in *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, 2007.

[47] F. Guenter, M. Hersch, S. Calinon, and A. Billard, "Reinforcement learning for imitating constrained reaching movements," *RSJ Advanced Robotics, Special Issue on Imitative Robots*, vol. 21, no. 13, pp. 1521–1544, 2007.

[48] P. Evrarard, E. Gribovskaya, S. Calinon, A. Billard, and A. Kheddar, "Teaching physical collaborative tasks: Object-lifting case study with a humanoid," in *Proceedings of IEEE International Conference on Humanoid Robots*, 2009.

[49] S. Calinon and A. Billard, "What is the teacher's role in robot programming by demonstration? - Toward benchmarks for improved learning," *Interaction Studies. Special Issue on Psychological Benchmarks in Human-Robot Interaction*, vol. 8, no. 3, pp. 441–464, 2007.

[50] S. Calinon, P. Evrard, E. Gribovskaya, A. Billard, and A. Kheddar, "Learning collaborative manipulation tasks by demonstration using a haptic interface," in *Proceedings of the International Conference on Advanced Robotics*, 2009.

[51] S. Calinon and A. Billard, "Recognition and reproduction of gestures using a probabilistic framework combinig PCA, ICA and HMM," in *International Conference on Machine Learning*, 2005.

[52] F. Guenter and A. Billard, "Using reinforcement learning to adapt an imitation task.," in *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[53] M. Kaiser and R. Dillmann, "Building elementary robot skills from human demonstration," in *In International Symposium on Intelligent Robotics Systems*, 1996.

[54] A. Coates, P. Abbeel, and A. Y. Ng, "Learning for control from multiple demonstrations," in *Proceedings of the 25th International Conference on Machine Learning*, 2008.

[55] T. Cederborg, M. Li, A. Baranes, and P.-Y. Oudeyer, "Incremental local online gaussian mixture regression for imitation learning of multiple tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[56] H. Sakoe and C. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[57] J. Listgarten, R. M. Neal, S. T. Roweis, and A. Emili, "Multiple alignment of continuous time series," in *in Advances in Neural Information Processing Systems*, 2005.

[58] J. Listgarten, R. M. Neal, and S. Cutler, "Bayesian detection of infrequent differences in sets of time series with shared structure," *Neural Information Processing System*, pp. 905–912, 2007.

[59] S. Sutton, M. Braren, J. Zublin, and E. John, "Evoked potential correlates of stimulus uncertainty," *Science*, vol. 3700, no. 150, pp. 1187–1188, 1965.

[60] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. MIT Press, 1990.

[61] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*. The McGraw-Hill Companies, Inc., 1996.

Curriculum Vitae

Name: Hoa-yu Chan (陳豪宇)

Education

B.S. degree from National Central University in Computer Science and Information Engineering in 2000

M.S. degree from National Chiao Tung University in Computer Science and Information Engineering in 2002

Research Interests

machine learning, imitation learning, and robot programming by demonstration.

Publications

Journal Paper

1. Hoa-yu Chan, Kuu-young Young, and Hsin-Chia Fu, "Intention deduction from demonstrated trajectory for tool-handling task", Journal of Chinese Institute of Engineer, Vol. 36, (Accepted).
2. Hoa-yu Chan, Kuu-young Young, and Hsin-Chia Fu, "Intention learning from human demonstration", Journal of Information Science and Engineering, Vol. 27, No. 3, May 2011, pp. 1123-1136, (Nominated for best paper award).

Conference Paper

3. Hoa-yu Chan, Kuu-young Young, and Hsin-Chia Fu, "Learning by demonstration for tool-handling task", IEEE International Conference on Instrumentation, Control and Information Technology (SICE), Oct. 2010.
4. Hoa-yu Chan, Kuu-young Young, Hsin-Chia Fu, and Chin-Teng Lin, "Motion feature extraction from human demonstration using dynamic programming", CACS International Automatic Control Conference, Nov. 2009.