Human-Inspired Robot Task Teaching and Learning

by

Xianghai Wu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2009

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.
I understand that my thesis may be made electronically available to the public.

Abstract

Current methods of robot task teaching and learning have several limitations: highly-trained personnel are usually required to teach robots specific tasks; service-robot systems are limited in learning different types of tasks utilizing the same system; and the teacher's expertise in the task is not well exploited. A human-inspired robot-task teaching and learning method is developed in this research with the aim of allowing general users to teach different object-manipulation tasks to a service robot, which will be able to adapt its learned tasks to new task setups.

The proposed method was developed to be interactive and intuitive to the user. In a closed loop with the robot, the user can intuitively teach the tasks, track the learning states of the robot, direct the robot attention to perceive task-related key state changes, and give timely feedback when the robot is practicing the task, while the robot can reveal its learning progress and refine its knowledge based on the user's feedback.

The human-inspired method consists of six teaching and learning stages: 1) checking and teaching the needed background knowledge of the robot; 2) introduction of the overall task to be taught to the robot: the hierarchical task structure, and the involved objects and robot hand actions; 3) teaching the task step by step, and directing the robot to perceive important state changes; 4) demonstration of the task in whole, and offering vocal subtask-segmentation cues in subtask transitions; 5) robot learning of the taught task using a flexible vote-based algorithm to segment the demonstrated task trajectories, a probabilistic optimization process to assign obtained task trajectory episodes (segments) to the introduced subtasks, and generalization of the taught task trajectories in different reference frames; and 6) robot practicing of the learned task and refinement of its task knowledge according to the teacher's timely feedback, where the adaptation of the learned task to new task setups is achieved by blending the task trajectories generated from pertinent frames.

An agent-based architecture was designed and developed to implement this robot-task teaching and learning method. This system has an interactive human-robot teaching interface subsystem, which is composed of: a) a three-camera stereo vision system to track user hand motion; b) a stereo-camera vision system mounted on the robot end-effector to allow the robot to explore its workspace and identify objects of interest; and c) a speech recognition and text-to-speech system, utilized for the main human-robot interaction.

A user study involving ten human subjects was performed using two tasks to evaluate the system based on time spent by the subjects on each teaching stage, efficiency measures of the robot's understanding of users' vocal requests, responses, and feedback, and their subjective evaluations. Another set of experiments was done to analyze the ability of the robot to adapt its previously learned tasks to new task setups using measures such as object, target and robot starting-point poses; alignments of objects on targets; and actual robot grasp and release poses relative to the related objects and targets. The results indicate that the system enabled the subjects to naturally and effectively teach the tasks to the robot and give timely feedback on the robot's practice performance. The robot was able to learn the tasks as expected and adapt its learned tasks to new task setups. The robot properly refined its task knowledge

based on the teacher's feedback and successfully applied the refined task knowledge in subsequent task practices. The robot was able to adapt its learned tasks to new task setups that were considerably different from those in the demonstration. The alignments of objects on the target were quite close to those taught, and the executed grasping and releasing poses of the robot relative to objects and targets were almost identical to the taught poses. The robot-task learning ability was affected by limitations of the vision-based human-robot teleoperation interface used in hand-to-hand teaching and the robot's capacity to sense its workspace. Future work will investigate robot learning of a variety of different tasks and the use of more robot in-built primitive skills.

Acknowledgements

First, I would like to thank my mentor and supervisor Jonathan Kofman for his kindness and patience, his tireless and constructive guidance, encouragement and help throughout my Ph.D. program. Dr. Kofman assisted me carefully and thoughtfully to choose courses that were best to enrich and expand my knowledge so that I would be well prepared for my later research. Dr. Kofman gave me tremendous help and guidance in choosing a topic for my PhD research through our cordial and ideal sharing, exchange, exploration, and discussion. In the processes of the design and development of the human-inspired robot-task teaching and learning method and system, I have also received a lot of valuable advice and support from him. In addition, Dr. Kofman scrupulously reviewed this thesis and my other publications and provided editorial comments and corrections. From him, I have learned and strengthened my attitude and methodology for conducting effective and fruitful research.

My genuine thanks also go to Chris Liu, who helped me calibrate the stereo cameras mounted on the robot end-effector for the robot to explore its workspace and on the walls to track the operator's hand motion, and for revisions to the marker-based hand-motion tracking system.

I also want to thank Kiatchai Borribanbunpotkat for his great help on the design of the apparatus used to mount the stereo cameras to the robot end-effector.

In addition, I give my sincere thanks to the human participants involved in the user study of the robot-task teaching and learning system for their precious time, effort and comments.

The Natural Sciences and Engineering Research Council of Canada (NSERC) as well as the Ontario Ministry of Training - Colleges and Universities are also thanked for their financial support through graduate scholarships.

Last but not least, I thank my family and my parents and parents-in-law for their enormous support and encouragement. Most of all, I thank my wife for her love and support.

Dedication

I dedicate this thesis to my parents. They have been constantly sacrificing themselves to raise and support me so much so that they could not have had enough food and money to buy clothes for themselves for more than one decade. They are always my pivotal foundation for my study and life. I owe too much to them. I love them.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

The need for versatile service robots to perform an increasing variety of tasks is growing for industries wishing to reduce labor costs, for general users wanting to be free of daily routines, and for the growing aging population that needs home-care assistance. Ideally, a service or personal robot would be adaptable to be able to continuously learn how to perform new tasks in new environments according to new arising needs of the user (Cherubini et al., 2007). A system that allows a robot to learn from a general user's teaching is presented in this thesis. Such a system may be the most practical solution to meet the above needs. The general goal of this research was to develop a robot teaching and learning system that could enable a general user to teach a service robot new tasks in new environments in an intuitive manner.

The emerging demand for versatile service or personal robots has led to much recent research on development of service robots (Thomaz and Breazeal, 2008; Asfour et al., 2008; Ekvall and Kragic, 2004; Hoffmann, 2004; Choi et al., 2000; Dillmann et al., 1999; Friedrich et al., 1996a, 1999). For service robots to be widely used, they need to satisfy the following basic premises, partly suggested by Lee et al. (2002), Asfour et al. (2008), and Thomaz and Breazeal (2008):

(1) The robot should be multi-functional, ideally autonomous, flexible and adaptive.

(2) The interaction between service robots and users should be natural, simple and intuitive, and the robot should be able to be instructed by general users in natural and intuitive ways.

(3) The robot should be able to be taught easily by general users, and gain experience and skills through its daily performance.

(4) The robot must be reliable, safe, and affordable.

Service robots have to be adaptive to a great extent because of the great variation in their working environments, tasks required and needs of users (Asfour et al., 2008). With the current level of technology, it seems impossible to build a fully autonomous robot that has the inherent capability to adapt to all uncertainty, and perform all new demanded tasks. However, more realistically, a service robot could be manufactured with sufficient artificial intelligence to be able to learn information about new environments and tasks from its experience and from users' teaching. Users should be able to teach a robot when most convenient for themselves, a concept called active teaching, or teach at the request of the robot, referred to as passive teaching (Katagami and Yamada, 2003).

For a service robot to perform an object-manipulation task successfully, the robot needs to have knowledge of at least the following:

(1) Task-related working environment. This is the world model or map of the working environment. It should include information about occupied spaces, and locations and orientations of obstacles and objects of interest.

(2) Information about objects involved in the task. This includes the objects' shape, color, texture, relationships with its surrounding environment, functional roles, and related basic object-manipulation skills.

(3) Task structure or plan. This includes the goals of the task and subtasks or steps, the relations between these steps, the actions involved in each subtask, the related pre-conditions that must be satisfied before starting the subtask, the until-conditions that must be reached before terminating the subtask, and the involved subtask trajectories.

A general user should be given the ability to teach or instruct all three aspects of knowledge to the robot because all the information can be different from place to place, from task to task and from user to user, and the information can also change with time. This research mainly focused on teaching the robot the task plan. In addition, limited task-relevant information about objects was taught and learned as necessary for the task learning and execution. The teaching of the working environment is not explicitly addressed in this thesis. Instead, the robot learning of locations and orientations of objects of interest was achieved using simple autonomous vision-based object detection methods. Finding effective solutions for robot teaching by general users is a crucial step for service robots to be widely commercialized (Lockerd and Breazeal, 2004).

1.2 Problems in Service-Robot Task Teaching and Learning

In the domain of service-robot teaching and learning, the ultimate goal is to let people unknowledgeable in robotics have the capability to teach a service robot different tasks at different times and in different environments (Cherubini et al., 2007). However, current methods of robot-task teaching and learning in the literature need much further development to reach this goal. There are three main problems of these methods: there is no checking or teaching of background knowledge (Dillmann, 2004), the task expertise of the teacher is not well exploited (Billard et al. 2008; Argall et al., 2007), and it is difficult for general users to teach robots tasks effectively and intuitively (Weiss et al., 2009).

1.2.1 Background Knowledge Mainly Pre-programmed

Background knowledge needed for the robot to learn a task includes information about the objects involved, the working environment, and basic object-manipulation skills. This knowledge is vital to the success of service-robot teaching and learning, just as students are required to have the necessary prerequisite knowledge in order to register for a new course. In the literature, the related background knowledge for a task is generally pre-programmed into the learning systems (Dillmann et al., 1999; Ehrenmann et al., 2002; Dillmann, 2004; Nicolescu and Mataric, 2001 and 2003; Calinon and Billard, 2007), so it would be very difficult for these learning systems to adapt to learn new types of tasks. The checking and teaching of needed background knowledge for a new task have not been reported earlier, nor have they been integrated into a robot-task teaching and learning method. In this research, a separate step of checking and teaching needed background knowledge is considered essential before the teaching of a new task, and is therefore included as a key component of the system. This step can provide a channel for the teacher to understand what the robot has previously learned, what has to be taught to the robot before starting to teach new tasks, and what the teacher needs to address with extra effort. The understanding can thus help the teacher to plan and structure the task teaching. This step can also offer effective means for the teacher to teach any lacking background knowledge, to enable the robot to learn new types of tasks.

1.2.2 Task Expertise of the Teacher Not Well Exploited

For the ultimate research goal of service-robot teaching and learning, the robot must be intelligent and the user must help the robot learn new tasks. The only requirement for the user should be that they have expertise in the tasks to be taught at the task level. Much research needs to be done to exploit the expertise or knowledge of the user at the task level (Billard et al. 2008; Thomaz and Breazeal, 2008; Asfour et al., 2008), such as perceiving and understanding task-relevant object and environment attributes, important state changes, primitive skills, and task/subtask structures. Argall et al. (2007) state that “Most work in this area places the majority of the learning burden on the robot”. In addition, there has been little work done to make the robot represent the task in a similar way as a human would, which would allow the general user to interact easily and intuitively with the robot. This is considered to be extremely important in this research as service robots will coexist and interact with human beings throughout their life. In this research, it is considered essential that the teacher should be well exploited to introduce the task structure to the robot, provide cues and assistance in task segmentation, and help the robot perceive important task-related state changes, in ways that are natural and intuitive to the teacher.

1.2.3 Difficulty for General Users to Teach Tasks to Robots Effectively and Intuitively

In some of the past literature, robot teaching and learning methods, such as in Kuniyoshi et al. (1994), Dillmann et al. (1999) and Ehrenmann et al. (2003), lacked natural human-robot interaction. The teacher first quietly demonstrated the task to the robot, while the robot passively observed the demonstration and recorded the related data. Then, the robot started to analyze, abstract and generalize the taught task. In the post-processing, the teacher usually probed into related learning programs and tweaked some relevant underlying parameters. However, this kind of human supervision is not applicable for general users who are likely unknowledgeable in robotics and unable to interpret general robot internal knowledge representation. If the performance of the robot’s practice or execution was then not satisfactory, the teacher had to either re-demonstrate the task or retune relevant parameters. The robot was not able to refine its learned knowledge based on the feedback intuitively given by the teacher.

In the service-robot domain, human-inspired teaching and learning methods are very promising. The teacher can teach the robot by human-style tutelage, intuitively give the robot timely feedback, and the robot can then improve its knowledge accordingly, as in Nicolescu and Mataric (2003), Breazeal et al. (2004a, 2004b), Lockerd and Breazeal (2004), and Thomaz and Breazeal (2008). There may exist rich natural and intuitive interaction between the teacher and robot. However, the related research focused more on the conceptual application of human-style teaching and learning, and it is still in the early stages, simple, and far from a systematic service-robot teaching and learning method that enables a general user to easily teach a service robot new tasks. There has been little report of methods to enable general users to teach a task to a robot naturally, and furthermore, to effectively teach different tasks to the robot.

1.3 Research Objectives

The overall goal of this research was to develop an interactive and intuitive human-robot teaching system, which can allow general users to teach different object-manipulation tasks to a robot, and equip the robot to learn the taught tasks and adapt its learned tasks to new task setups. The teaching and learning processes should be interactive and effective. The service robot should actively participate in the teaching-and-learning processes by continuously revealing its learning state, for example, what it has understood thus far, and what it needs to further clarify. The teacher would then provide timely feedback on the robot learning performance and the robot should refine its learned knowledge accordingly. This would help to ensure that the robot has learned what the user intended. The following are the specific research objectives toward the overall goal:

(1) Develop a systematic human-style robot task teaching and learning method. The teaching method should enable the teacher to intuitively and effectively teach tasks to the robot. The expertise of the teacher in tasks to be taught should be exploited in ways that are intuitive to the teacher. The robot should actively involve the task teaching and learning process by actively observing the teacher's demonstration, listening for the teacher's requests, responses and comments, while revealing its learning progress and refining its knowledge based on the teacher's timely feedback. The task should be ultimately learned by the robot as expected by the teacher, even if the teacher is not initially satisfied with the performance of the robot. The robot should be able to ask the teacher for assistance if the robot finds an error during its learning process, and the teacher should be able to have the robot modify its task segmentation and learned task trajectory by intuitively giving relevant timely feedback to the robot. The robot also needs to have the capability to adapt its learned tasks to new task setups, which may be quite different from those used during the task demonstration.

(2) Develop a robot-task teaching and learning platform that implements the proposed human-inspired robot-task teaching and learning method. This platform would enable a general user to teach a robot different tasks easily and intuitively, and to continue any previously unfinished teaching of a task or an interrupted task demonstration. Natural and context-based speech dialogue would be the main media for human-robot interaction. The teacher could use their natural hand motion to teleoperate the robot for the demonstration of the task to be taught. The robot should have the ability to automatically explore its

workspace and detect and identify objects of interest using its stereo vision system. The robot's learned knowledge about the tasks and objects would be saved into a database, and the robot could retrieve the tasks from the database and perform them at any time.

1.4 Overview of Contributions

The main contributions that have been made through this research in robot-task teaching and learning are briefly described below for introductory purposes. A more detailed description of contributions is given in Section 7.3 Thesis Contributions in Chapter 7 Discussion.

- Introduction of a human-inspired robot-task teaching and learning method that enables general users to teach a robot new tasks intuitively and effectively, and empowers the robot to learn the taught tasks and adapt its learned tasks to new task setups (Wu and Kofman, 2008). This method provides a more systematic approach of the human style robot-task teaching and learning that includes the following:
    - two novel teaching and learning stages: checking and teaching robot's background knowledge needed for the task to be taught, and introduction of the overall task.
    - approaches to exploit the teacher's task expertise through natural and transparent human-robot interaction.
    - a flexible vote-based algorithm for segmenting demonstrated task trajectories (Wu and Kofman, 2008).
    - a probabilistic optimization method to ground or associate the user-introduced high-level task knowledge to the robot-sensor data collected during the task demonstration.
    - two blending schemes that combine task trajectories generated from different reference frames so that the robot can adapt its learned task to new setups (Wu and Kofman, 2008).
- Design and development of an agent-based framework as a realization of the proposed robot-task teaching and learning method. The framework consists of: the Robot Agent (RA), Teleoperation Agent (TOA), Speech Agent (SA), Vision Agent (VA), Task-Learning Agent (TLA), Task-Performing Agent (TPA), and a database. Except for the hand-tracking part of the TOA and calibration of the stereo-camera pair used for the VA, this system was newly developed as a part of this research.
- A user study to test the robot teaching and learning system with general users for the evaluation and validation of the feasibility and effectiveness in teaching tasks by general users, ease of use, and ability of the robot to learn taught tasks and adapt the learned tasks to new task setups.

1.5 Organization

The remainder of this thesis is organized as follows. A literature review of existing methods of robot-task teaching and learning is presented in Chapter 2. It mainly comprises two parts: the human teaching of tasks to a robot, and the learning of the taught tasks by the robot. Chapter 3 details the human-inspired robot task teaching and learning method. In Chapter 4, the implementation platform of the proposed method is described, including the knowledge organization of a learned task and an agent-based implementation of the task teaching and learning system. Experiments including the robot-task learning and adaptation (RTLA) experiments and user study, are given in Chapter 5, with the corresponding results presented in Chapter 6. Chapter 7 contains a discussion on the experiments and results, followed by the conclusion and future work in Chapter 8.

Chapter 2
Literature Review of Robot Task Learning from Human Teaching

2.1 Introduction

For a robot to accomplish a useful task, it has to execute a set of well-programmed software. This software can be created by programmers manually, by the robot system automatically through learning by self-exploration or from a user's teaching, or by a variation between the two extremes. Biggs and MacDonald (2003) presented a survey on robot programming systems. They classified robot automatic programming into three subcategories: learning systems, programming by demonstration (PbD), and instructive systems. The learning systems, also called inductive learning systems (Kuniyoshi et al., 1994), learn motion and sensing strategies through analyzing multiple user-provided examples or by system trial-and-error self-exploration. Such systems usually need a large number of trials or examples. PbD, also called learning from observation, learning from demonstration, or learning by imitation (Schaal, 1999; Billard et al., 2008), involves a teacher first demonstrating tasks to the robot and then the robot learning how to perform the task through watching the demonstration (Argall et al. 2009; Billard et al., 2008). Instructive systems are best suited for commanding robots to learn a new task that is composed of previously learned tasks or subtasks (Biggs and MacDonald, 2003).

In this literature review, only PbD automatic programming is reviewed. Instructive learning systems are excluded since the focus of this research was to develop a teaching and learning method that enables a robot to learn new tasks, where the subtasks may have not been previously learned, as opposed to using previously learned subtasks. Inductive learning systems are excluded since the research goal was to learn using minimal demonstration examples rather than multiple ones. Two recent reviews on robot learning from demonstration are presented in Argall et al. (2009) and Billard et al. (2008). However, in this thesis, the focus of the literature review is on robot task teaching and learning methods to enable a general user to teach different tasks to a robot. A schematic diagram of a general robot-learning-from-human-teaching system, such as those described in Dillmann et al. (1999) and Ekvall and Kragic (2008), is shown in Figure 2-1. Firstly a robot teaching and learning system is set up. Then the human teaches the robot relevant information about a new task, while the robot learns from the teaching. This learning may be simultaneous with the teaching or as a post-process to generate relevant parameters, and it is followed by robot practice. If the practice is not satisfactory, modifications to the system could be made or the teaching or learning process must be repeated as shown by the dotted-line arrows (Figure 2-1). The feedback may also be given by the teacher during the teaching and robot-learning stages (dotted-curve arrows). In the majority of the related literature, the teacher has to retune some underlying variables in the system or repeat the entire teaching and learning process again. Alternatively, the feedback can be given *intuitively* by the teacher, and the robot would then refine its knowledge accordingly (Nicolescu and Mataric, 2001 and 2003; Lockerd and Breazeal, 2004; Thomaz and Breazeal, 2008).

The review will be organized in three parts. The first includes existing task teaching methods, which describe how tasks have been taught to robots by human teachers. The second part will review task learning methods, which provide means for robots to learn the taught tasks from human teaching. Finally, a summary of the review is provided with prospects of the methods proposed in this research.

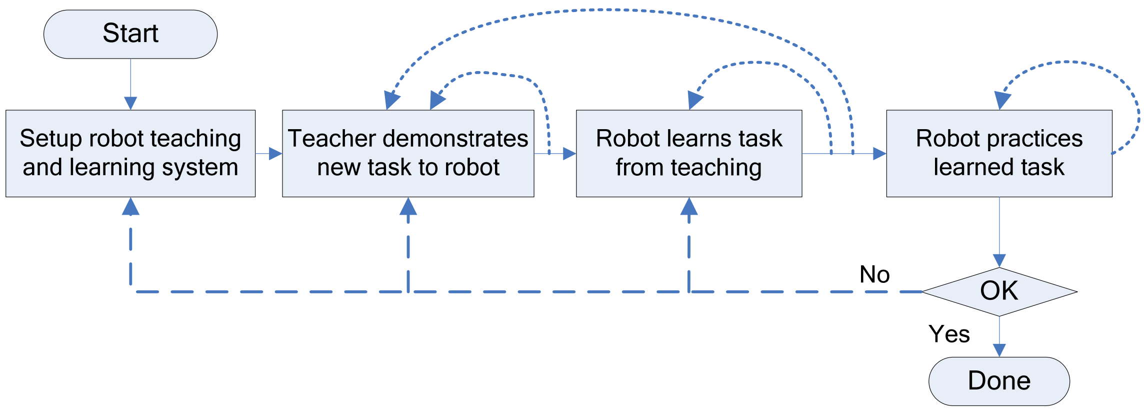


Figure 2-1: Schematic diagram of a general robot-learning-from-human-teaching system. During the teaching and learning process, the teacher might be able to provide some feedback on the performance of the robot's learning and practice, as indicated by the dotted arrows. The key issue is that whether the feedback can be given intuitively by a general user. If the practice result cannot reach a satisfactory level, the robot must go back to repeat the setup, teaching or learning process, shown by the dashed arrows.

2.2 Methods of Teaching Tasks to Robots by Human Teachers

Although the teaching and learning of a task are generally coupled together, methods of teaching tasks to a robot by a human, from the teacher's point of view, can be roughly categorized into three classes (Wu and Kofman, 2008; Thomaz and Breazeal, 2008; Biggs and MacDonald, 2003): teaching by guidance (TbG), teaching by passive human demonstration (TbPD), and human-style teaching (HST). HST usually incorporates rich human-robot interaction into the teaching by demonstration.

In this research, TbGs and TbPDs are considered as open-loop methods since the teachers are not actively involved in the processes of task teaching and learning, and they cannot give feedback intuitively on performance of the robot's learning and task practice to help the robot refine its relevant task knowledge accordingly. There is not much natural human-robot interaction during the process of the task teaching. Usually, the teacher first quietly guides the robot through or demonstrates the task, while the robot passively observes the demonstration and records the related data. Then, the robot starts to analyze, abstract and generalize the taught task. In the post-processing, the teacher often probes into related learning programs and tweaks some relevant underlying parameters. However, this kind of human supervision is not applicable for general users who are likely unknowledgeable in robotics and unable to interpret general robot internal knowledge representation. If the performance of the robot's practice or execution is not satisfactory, the teacher would either re-demonstrate the task or retune relevant parameters.

On the other hand, HST methods are deemed as closed-loop. There would be intuitive interaction between the teacher and robot to allow the robot to reveal its learning state, and to permit the teacher to track the learning progress of the robot and provide it timely feedback. The three categories of robot task teaching methods are reviewed below.

2.2.1 Robot-Task Teaching by Guidance (TbG)

Robot-task teaching by human guidance involves having an operator directly move the robot end-effector or robot joints along desired trajectories and activate relevant operations, such as opening or closing a gripper and turning on or off a tool, for example a painting spray gun, at specific instants. The robot would then typically repeat the recorded path and operations without additional information from external sensors (Kuniyoshi et al., 1994). Due to the simplicity and effectiveness, this approach is widely used for industrial robots. However, it is only suitable where personnel who teach the tasks to the robot are highly trained, and where the working environments are well-structured and controlled.

Guidance by operators is often carried out using a teach pendant. However, the trajectory is usually not smooth or optimal, the guidance is not intuitive for the operator, and motion-velocity information cannot be directly obtained for the robot to apply. More intuitive teaching devices have been developed, such as a six-degree-of-freedom (DOF) force sensor (Muto and Shimokura, 1994), six-DOF force mouse (Choi and Kim, 2000), joystick (Akanyeti at el., 2008; Dixon, 2004; Kristensen et al., 2001), force/moment direction sensors (Choi and Kim 2000, 2001), and kinesthetic techniques (Calinon et al., 2007). In addition, hand-gesture-based (Voyles et al., 1997), graphical-based (Uechi et al., 1999) and virtual reality (Aleotti and Caselli, 2006; Kunii and Hashimoto, 1997) teaching methods have been used. These techniques permit smoother motion as well as some level of intuitiveness. A quantitative evaluation of robot teaching methods using a force/moment direction sensor is presented in Choi and Lee (2003).

2.2.2 Robot-Task Teaching by Passive Human Demonstration (TbPD)

In teaching by human demonstration (TbD), also called programming by demonstration (PbD), the human teacher demonstrates a task to a robot naturally either directly (Kuniyoshi et al., 1994; Dillmann et al., 1999; Dillmann, 2004; Ekvall and Kragic, 2006; Chella et al., 2007; and Rusu et al., 2008) or through teleoperation via natural human body or arm motion (Peters et al., 2003; Lieberman and Breazeal, 2004; Wu and Kofman, 2008). Note that teaching through teleoperation using techniques other than natural human body or arm motion tracking is considered to be in the category of teaching by guidance since the involved teleoperation is less natural and intuitive to the teacher. In the methods of task teaching by passive human demonstration, the teacher usually quietly demonstrates the tasks with very limited or no natural interaction between the human teacher and the robot, both in the process of teaching and in the stage of learning and practicing. Except for the demonstration itself, the teacher has very limited influence on the robot's task learning in the teaching stage.

2.2.2.1 Teaching by Human Direct Passive Demonstration

The process of robot learning by passively watching human demonstration is also called programming by demonstration (PbD) in a narrow sense. It should be noted that the process of observing and learning may not run on the robot system, but instead on a separate system such as a learning center (Dillmann, 2004; Pardowitz et al., 2007) or a system where the robot is only one part of a network of sensors, actuators, and computers (Rusu et al., 2008). Usually the observing system is not portable, but it is has more processing capacity than the sensor system of the robot. The teacher quietly demonstrates tasks while the robot or a separate learning system passively watches, records, and measures parameters of the demonstration. There is no active revelation of the robot's internal learning states in forms that are understandable to general users, and the teacher cannot naturally and intuitively influence the learning process.

The common characteristics and steps of a PbD system, as described in Kuniyoshi et al. (1994), Chen and McCarragher (1998), Dillmann et al. (1999), Dillmann (2004), and Ekvall and Kragic (2008), can be summarized as follows: (1) A system for learning by passively watching human demonstration is set up at a station for task demonstration. The system includes sensory systems to observe the demonstration, especially a vision system, and necessary components for post-demonstration processing. (2) A teacher demonstrates the task, and the learning system records data flows from all sensors such as a data glove, vision system, and 3-D positioning system. (3) The learning system processes the recorded time-series data, extracts task-related features and states, and derives time-series sequences of hand motion and contact states with the objects. The processing may be done in real-time during the task demonstration (Kuniyoshi et al., 1994). (4) The demonstration is partitioned into different meaningful subsequences, segments or episodes based on the motion sequence and transitions of the states, such as approaching, grasping, lifting up, aligning, placing, and retreating (Kuniyoshi et al., 1994; Friedrich et al., 1999; Dillmann et al., 1999). (5) The segments are abstracted and generalized to a hierarchical symbolic representation of the demonstrated task (Card et al., 1983; Kieras, 1994). Each action segment contains preconditions, until-conditions and associated actions. (6) The learned generalized task plan is transferred to a robot, and instantiated and executed by the robot.

Although vision systems were commonly deployed to watch the human's task demonstrations (Kuniyoshi et al., 1994; Dillmann, 2004; Chella et al., 2006 and 2007; Asfour et al., 2008; Nakaoka et al., 2007; Akanyeti et al., 2008), sensing and human-robot communication used in human demonstrations (Step 2) have evolved into multi-modal methods (Biggs and MacDonald, 2003; Iba, 2004) using force/torque, tactile and position sensors, speech, virtual reality, inertial measurement units, and radio-frequency-identification (RFID)-enabled gloves (Ikezawa et al., 1997; Ehrenmann et al., 2001, 2002; Chen and Zelinsky, 2003; Asfour et al., 2008; Rusu et al., 2008). The transfer of the task knowledge (observed and learned in the dedicated demonstration-observation sensor systems) to the robot is very challenging (Knoop et al. 2007, 2008), due to the different observation perspectives (Breazeal et al., 2006), and the different physical body structure and dynamics between the robot and human demonstrator.

2.2.2.2 Teaching by Human Passive Demonstration through Teleoperation

Teleoperation of a robot by a human operator has been used to demonstrate tasks (Peters et al., 2003; Breazeal et al., 2004a; Lieberman, 2004; Lieberman and Breazeal, 2004; Wu and Kofman, 2008), where motion in multiple degrees-of-freedom needs to be taught simultaneously, where access by the teacher to the robot environment is limited, or when using a central operator remote from the robot. Here, only teaching via teleoperation using the teleoperator's natural body movements (Peters et al., 2003; Lieberman and Breazeal, 2004) or hand motions (Wu and Kofman, 2008; Kofman et al., 2005 and 2007) will be considered. Both kinds of teaching will be referred to as *hand-to-hand teaching* (Wu and Kofman 2008) or 'putting through' teaching (Otero et al., 2008).

Peters et al. (2003) demonstrated tool-use skills to the NASA Robonaut through full-immersion teleoperation, where positions of teleoperator's arm and head were determined via six-axis magnetic field sensors (Polhemus) mounted on a helmet and data gloves, and finger positions were measured with data gloves. Similarly in Lieberman and Breazeal (2004) and Lieberman (2004), the teacher, wearing a teleoperation suit ('Gypsy Suit') that measured 42 joint angles of the human operator, teleoperated the humanoid robot "Leonardo" to demonstrate a button-push skill. Outputs from tactile sensors, mounted in the robot hands, were also measured to be used as an augmentation for trajectory segmentation carried out later. As part of hand-to-hand instructive teaching in Wu and Kofman (2008), the teacher teleoperated a robot to demonstrate a pick-and-place task using natural hand motion that was tracked by a vision system using marker-based techniques.

Although demonstration via teleoperation is more natural for the teacher than physically moving the robot arms in teaching by guidance, it may be less natural and intuitive for the teacher than direct task demonstration without teleoperation. However, it is very import for the robot to directly observe and understand a taught task from its own perspective (Breazeal et al., 2006; Otero et al., 2008), due the robot's limited inference capacity. This also makes it much easier for the robot to transfer and instantiate, and then execute the learned tasks based on its own software and hardware capability. Teleoperation may therefore have additional merit in having the demonstration observed directly from the robot's own point of view and even with its own sensor system, such as joint-angle sensors embedded in the robot, as in the current research. Solving the motion correspondence between the robot and teacher is a challenging problem (Calinon and Billard, 2006; Calinon et al., 2007; Otero et al., 2008). By teleoperation, the goal-oriented motion correspondence between the robot and teacher can be established explicitly and earlier than without teleoperation. The teacher can handle the structural and dynamic differences between the human and robot, and ensure that the robot can achieve the required motions for the task, by adapting in real-time to the immediate robot motion feedback the teacher observes. A further advantage of teleoperation is that the teacher can teach the robot from remote sites. This can be advantageous for different applications such as telehealth and operating in hazardous environments.

Because of the above advantages, demonstration via teleoperation was used in the hand-to-hand instructive task demonstration in the current research. Although the current implementation uses vision-based tracking of the teacher's hand motion, other tracking techniques could be used, such as the six-axis magnetic tracking sensors and data gloves used in Ehrenmann et al. (2002), Peters et al. (2003) and Asfour et al. (2008).

2.2.3 Human-Style Robot-Task Teaching (HST)

The main drawback of the open-loop robot-task teaching and learning methods is that they lack natural and intuitive interaction between the teacher and robot to permit the two to build a common understanding and to enable the teacher to actively teach the task and further influence the robot's task-learning progress. These problems have motivated researchers to propose very promising approaches of human-style tutelage for robot teaching and learning. Early important work in these methods was done by Nicolescu and Mataric (2001, 2003), Lockerd and Breazeal (2004), and Breazeal et al. (2004). HST usually tightly integrates natural human-robot interaction with teaching by demonstration, and closely incorporates the teacher into the task teaching and learning loop. These methods allow the robot to reveal its learning states to the human teacher, and permit the teacher to provide timely and intuitive high-level feedback on the performance of the robot's task learning and practice. The robot can then refine its internal task knowledge based on the feedback so that it can improve its performance to a satisfactory level.

In a study of transferring elementary skills via human-robot interaction (Kaiser, 1997), a human teacher gave both positive and negative feedback to a robot on its skill performance, which was used in a skill refinement by exploration. This was an improvement over open-loop approaches; however, feedback was provided only after the robot performed the entire task. Argall et al. (2007) solved the credit-assignment problem by requiring the teacher to conduct an examination of the recorded robot performance and reward each small part of the executed task trajectory. However, the effect of the feedback on the performance cannot be reflected to the teacher online (in real-time), and this kind of delayed feedback verification may prevent the teacher from offering more effective and proper feedback. The delayed feedback response may have contributed to the up to 100 critiquing rounds needed in their experiments. Robot task learning methods inspired from human teaching and learning, allowed a teacher to give instructive task demonstration and offer timely feedback to the robot at any time during its practice in Nicolescu and Mataric (2001 and 2003) and later by Pardowitz et al. (2007). A scaffolding process was adopted in Calinon and Billard (2007) to gradually refine the robot's performance by physically moving relevant robot joints after some initial passive task demonstrations. Scaffolding and event structuring were broadly studied for the benefits brought to both the teacher and the robot in Otero et al. (2008). Here, event structuring meant allowing the teacher to emphasize important moments, states, and state changes in the task teaching and learning. It was found that appropriate segmentation of the task behaviors by the teacher was very helpful to the teacher to teacher and for the robot to effectively learn.

The richest human-robot interaction in robot task learning from human teaching has been developed in Breazel et al. (2004a, 2004b), Lockerd and Breazeal (2004) and Thomaz and Breazeal (2008) for the humanoid robot "Leonardo". In these works, human teaching and robot learning were treated as teamwork, and the robot and teacher were collaborating partners. Human-style tutelage and scaffolding were applied to robot teaching and learning, with rich interaction and timely feedback to allow the robot to express what it had learned and what remained in confusion. The method also permitted the robot to initiate requests for elaboration from the teacher in order to clarify ambiguities. This enables the teacher to maintain an awareness of the robot's current learning states, and adapt to them in the next task-teaching step. The robot can also refine its learned knowledge based on the teacher's timely feedback, better follow and comprehend the teaching, and better organize the taught knowledge. Furthermore, the teacher can intuitively, naturally and efficiently teach the robot, in a manner similar to teaching a person. The applied fundamental theories include collaborative systems, joint intention, teamwork, and task-orientated discourse dialogue (Cohen and Levesque, 1990, 1991; Grosz, 1996). However, their teaching and learning require the robot to have much advanced pre-existing knowledge, not only rich and subtle interaction skills but also object-manipulation skills. For example, their robot, Leonardo, can understand natural speech, recognize a human's gazing direction and hand pointing gesture, and reveal its internal state using sophisticated human-like social body-language expressions, such as shrugging shoulders, subtle head nods, eye gaze and blinks, different facial expressions, and arm opening and closing. In addition, the robot has in-built object-manipulation skills needed for the tasks to be taught, such as the button-pressing and sliding (Thomaz and Breazeal, 2008). In this research, the use of text-to-speech (TTS) synthetic voice is considered as a more explicit and cost efficient means for general service robots to speak out their internal states and requests. In this thesis, except for hand-to-hand task demonstration through teleoperation, the main human-robot interaction is via natural speech from the teacher and TTS synthetic voice from the robot.

While there have been great advances in robot teaching and learning, the current research on HST focuses more on the conceptual application of human-style teaching and learning, and it is still in the early stages. Current methods are still simple and a systematic service-robot teaching and learning method that enables a general user to easily teach a robot new tasks has yet to be achieved. As mentioned in Section 1.2, the robot's background knowledge needed for a task to be taught has not been checked and taught by the teacher at the beginning of the teaching of the task; and the task expertise of the teacher has not been well exploited, specifically in generation of a task structure and segmentation of the task.

There are only two known reports of user studies that evaluated robot-task learning methods from human teaching for object-manipulation tasks, from the user's perspective (Weiss et al., 2009). In Weiss et al. (2009), human subjects taught a robot two simple tasks "push box" and "close box", and they demonstrated each of the tasks more than three times on average to the robot. The subjects had to demonstrate the whole task again if the robot's task performance was not satisfactory, instead of providing timely feedback on the robot's performance when it was practicing the task. There was little natural vocal human-robot interaction. The mean system usability score was 65.8 out of 100 based on the subjects' evaluation. In one other study, Thomaz and Breazeal (2008) focused on socially-guided exploration. With pre-programmed primitive actions, the robot learned information about a puzzle box with the aid of subjects' vocal instructions and feedback. The resultant states after a primitive action under different initial states were learned. There was no subjective evaluation conducted.

The teaching method in this research, a continuation of previous working in Wu and Kofman (2008), closely imitates a human-to-human teaching approach, where the teacher often checks the students' background knowledge, teaches the students any lacking knowledge if necessary, and gives an overall introduction of what is to be taught. The teacher usually delivers difficult knowledge step by step, and provides an overall summary of the knowledge at the end. In addition, in-class problems are sometimes given to students and the teacher offers timely feedback on the students' problem-solving performance. In this research, a systematic approach and procedures are to be presented to allow general users to teach tasks to the robot effectively and intuitively. The teacher's expertise in the tasks will be well exploited to help the robot learn the taught task in a top-down approach and refine its learned task knowledge by giving pertinent timely feedback.

2.3 Methods for Robot Learning Tasks from Human Teaching

In robot task learning from human teaching, either a direct mapping function from observed states to robot actions, a reinforcement learning-based system model, or a goal-directed task plan is derived based on demonstration data (Argall et al., 2009; Billard et al., 2008). There have been approaches which minimize the required number of task demonstrations by the teacher, such as One-Shot-Learning (Dillmann et al., 1999; Ehrenmann et al., 2003; and Dillmann, 2004). Other approaches use multiple demonstrations to reveal the underlying optimal solution (Coates et al., 2008). The idea is that task knowledge extracted from multiple demonstrations may more closely reflect the teacher's true intentions. The simplest way for the robot to learn a task from human teaching is to repeat the taught task trajectories or determine the task paths by interpolating through demonstrated waypoints. The other learning methods can be briefly categorize into two groups: learning taught tasks in whole and learning taught tasks by segmentation into episodes. The two groups of methods are reviewed in this section. Argall et al. (2009) and Billard et al. (2008) provide more thorough reviews on robot task learning from human teaching.

2.3.1 Learning Task in Whole

In one type of robot task learning method, the demonstrated task or skill trajectory is learned as a whole. Machine learning techniques, such as neural networks (NNs), fuzzy logic and Hidden Markov Models (HMMs), are usually applied.

Fuzzy logic was successfully used to learn behavioral preferences and relevant activation rules from a task demonstration (Hoffmann, 2004) and simple assembly tasks (Myers et al., 2001). Neural networks were used for a mobile robot to learn sensor-motor coordination policies for navigation skills (Martin and Nehmzow, 1995). Sensor-motor policy for similar navigation skills were estimated by Non-linear Auto-Regressive Moving Average models with eXogenous inputs (NARMAX) in Akanyeti et al. (2008). In Aleotti and Caselli (2006), task trajectories of multiple demonstrations were first clustered using a distance-based geometric procedure, a HMM was used to indentify and select the most consistent demonstrated trajectory for each cluster, and a Non-Uniform Rational B-Spline (NURBS) module was then utilized to generate approximate trajectories for the task simulation and execution based on the trajectories selected by the HMM. HMMs were also trained by pre-processed key points of robot joints and end-effector trajectories to generalize the demonstrated task trajectories, in Asfour et al. (2008). Calinon et al. (2007) proposed a Gaussian Mixture Model (GMM) and Bernoulli Mixture Model (BMM) to represent dimension-reduced demonstrated task trajectories and related probabilities, respectively. The taught task trajectories were then generalized using Gaussian Mixture Regression (GMR). They further incorporated incremental learning ability into their models (Calinon and Billard, 2007). Although the robot showed some adaptation ability to new task setups, the robot's capacity was limited due to the data-dimension reduction and no generalization in different reference frames (Calinon and Billard, 2007). For each maneuver of a helicopter flight in Coates et al. (2008), an underlying generative trajectory model or hidden ideal trajectory was learned from multiple demonstrations by aligning each point of the demonstrated trajectories to a state of the hidden trajectory through dynamic time warping. The time warping was solved by maximizing the joint likelihood of the observed trajectories occurring with the pre-specified prior knowledge about the ideal trajectory while marginalizing over the hidden trajectory, using the expectation-maximization (EM) algorithm. In addition, a control policy along the estimated trajectory was learned.

Learning the task as a whole using the above mentioned machine learning techniques can permit learning of the sensorimotor policies for skills. However, a significant number of training datasets are needed to make the policies adaptable to new setups. Furthermore, it is difficult to reuse the learned control policies for different sensorimotor configurations (Dillmann et al., 1999), and it is also challenging to interpret the learned policies in ways that are understandable to general users (Breazeal et al., 2004). In addition, it might be unmanageable for the robot to learn a complex task if the task is not segmented into different episodes or subtasks and then learned starting from these subtasks.

2.3.2 Learning Task by Segmentation into Episodes

Partitioning a demonstrated task into different episodes and then learning and generalizing each episode give leverage to the robot to effectively learn complex tasks and adapt the learned tasks to different setups. The task can be further generalized and abstracted from the results learned in each episode.

2.3.2.1 Segmentation of Demonstrated Task

Segmentation of a task motion or trajectory into meaningful segments has commonly been achieved using motion breakpoints (Tso and Liu, 1995; Yeasin and Chaudhuri, 1997). These breakpoints have usually been determined based on instants of grasping and releasing (Delson and West, 1996; Friedrich et al., 1996a, 1996b, 1999; Ogawara et al., 2002; Pardowitz et al., 2007), evident features of force profiles and tactile feedback (Friedrich et al., 1999; Ehrenmann et al., 2002, 2003; Chen and Zelinsky, 2003; Lieberman, 2004); characteristics of the demonstrated trajectories (Chella et al., 2006, 2007; Nakaoka et al., 2007), such as velocity zero-crossing; and background knowledge related to the manipulation process, involved objects and environment (Pardowitz et al., 2007). Three two-value variables: gripper open/close, motion speed high/low, and object in-contact/not-in-contact, were used in Tso and Liu (1995). Position and velocity-based parameters have also been used for trajectory segmentation in various combinations: relative positions of the hand with respect to objects and absolute velocity of the hand obtained from a vision system (Kuniyoshi et al., 1994); changes in human-finger joint position, hand velocity, distance of the hand to objects, and force profile, respectively obtained from a data glove, 6-DOF position sensor, vision-based trinocular-camera system and tactile sensors (Friedrich et al., 1999; Ehrenmann et al., 2002); and key points of joint angles, end-effector locations and orientations (Asfour et al., 2008). A task segmentation method in navigation of mobile robots was proposed in Muslim et al. (2006), by using a modular network Self-Organizing Map (SOM) (mnSOM) to partition a demonstrated navigation into subsequences, such as “forward movement”, “left turn”, and “right turn”. Mobile navigation demonstrations have also been segmented by matching sensor data against pre-determined goals of a set of in-built robot behaviors (Nicolescu et al., 2007) or the variance of the feature data (Koenig and Mataric, 2006). Knowledge of known motions has also been applied to modify a HMM model in segmentation of full-body human motions (Kulic and Nakamura, 2008).

For natural human motions or movements of teleoperated robots using human operator’s natural motion, a very common auto-segmenting method is based on analysis of the mean-squared velocity (MSV) of the robot joints (Fod et al., 2002; Peters et al., 2003) or the robot end-effector (Lieberman, 2004; Lieberman and Breazeal, 2004; Kulic and Nakamura, 2008). The idea is based on the observation that human beings slow down their motions in the transition periods of action episodes. In Peters et al. (2003), the MSV was defined in robot joint space as:

$$MSV(t) = \sum_{i=1}^{n} \left(\frac{dq_i(t)}{dt} \right)^2 \tag{2-1}$$

where n is the number of robot joints, q_i is the i^{th} joint variable, and $MSV(t)$ is a function of time. Lieberman (2004) computed the MSV in the task space with an augment of the tactile feedback as:

$$MSV(t) = (\frac{dx_{EE}}{dt}^2 + \frac{dy_{EE}}{dt}^2 + \frac{dz_{EE}}{dt}^2) + \frac{1}{10000}\frac{dw(t)}{dt} \tag{2-2}$$

where $(x_{EE}(t), y_{EE}(t), z_{EE}(t))$ is the position of the end effector at time t, and $w(t)$ represents the tactile sensor value at time t. The condition for the beginning of an episode is:

$$MSV(t) > \zeta \quad and \quad MSV(t-1) \le \zeta \; , \tag{2-3}$$

while the condition for the termination of an episode is:

$$MSV(t) \le \zeta \quad and \quad MSV(t-1) > \zeta \; , \tag{2-4}$$

where threshold ζ is determined either by inspection (Fod et al., 2002, Peters et al., 2003) or automatic analysis of the $MSV(t)$ (Lieberman, 2004). To find true episodes, the general MSV analysis was augmented in Peters et al. (2003) with the maxima of the elbow joint position. In Peters et al. (2003), it was assumed that each trial contained the same number of episodes, while in Lieberman (2004) all true episodes were searched for, at the cost of getting false positives, and a method to discard false episodes was proposed. The next step in motion learning was to align the episodes by time-normalization (Peters et al., 2003) or correlation coefficient (Lieberman, 2004). Then a resultant motion sequence was obtained by averaging the aligned episodes to obtain a generalized time series either in the robot joint space (i.e. the joint trajectories) or in the task space (i.e. a trajectory of the robot end-effector). However, when the relative starting poses (locations and orientations) of the robot with respect to the objects were considerably different from those in the demonstrated trials, it was very challenging to adapt the trajectory episodes generalized in the joint space to the new task setups (Peters et al., 2003; Lieberman, 2004). This is because the sensor-motor data and target poses are strongly non-linearly coupled, and it is very hard to explicitly take into account the relative poses of the objects when generalizing the motion sequence with the sensory-motor coordination (SMC) data. To overcome the difficulties, Lieberman (2004) generalized the motion sequence of the end effector in task space and the object frame, respectively, at a cost of inverse kinematics computation. In this research, MSV of the robot end-effector task trajectory is selected as one of the signals to use to partition the demonstrated task trajectories.

The segmentation of the taught task trajectories usually produces a group of trajectory episodes (or segments) (Peters et al., 2003; Lieberman, 2004; Wu and Kofman, 2008), or a group of sub-goal-oriented primitive behaviors or actions and their pre-conditions and until-conditions (Thomaz and Breazeal, 2008; Pardowitz et al., 2007; Chella et al., 2006, 2007; Lockerd and Breazeal, 2004; Nicolescu and Mataric, 2003; Lee et al., 2002).

2.3.2.2 Abstraction and Generalization of Demonstrated Task

After a taught task is segmented, generalization is often needed as human demonstrations are often sub-optimal in nature (Coates et al., 2008; Chen and Zelinsky, 2003; Delson and West, 1994). The obtained task trajectory segments or primitive actions, operations, behaviors or subtasks are analyzed, abstracted and generalized to produce a task plan or structure, which includes the relevant pre-conditions, until-conditions, sub-goals, dependency, and other action-related parameters. A goal-driven task plan is usually developed, such as the learned task plan in Thomaz and Breazeal (2008).

By analyzing the sequence of sub-tasks, repeated fixed-order patterns of sub-tasks can be extracted, and this information may help the robot build a hierarchical task plan for the taught task in which a high-level sub-task may consist of several child sub-tasks, along with pre-conditions, post-conditions, state changes incurred, and new established relations (Pardowitz et al., 2007; Kuniyoshi et al., 1994). However, this kind of bottom-up deduction might produce a task structure, including the relevant task/subtask names, which are not similar to the ones the user had in mind. Furthermore, this non-natural task structure may make the human-interaction regarding the task more troublesome. For example, when a new user wants the robot to perform the task and doesn't have enough confidence that the robot could execute the task well, they can ask the robot to reveal the steps of the task using an utterance such as "please tell me the subtasks that it has". If the robot responds with an unfamiliar task breakdown and task/subtask names, it will not help the user to improve their confidence in the relevant ability of the robot.

The generalization across multiple task demonstrations is done by sub-goal comparison, and similar state mergence (Thomaz and Breazeal, 2008; Lockerd and Breazeal, 2004), finding common states among each demonstration (Nicolescu and Mataric, 2003), and reexamination of subtasks' pre-conditions and until-conditions (Pardowitz et al., 2007). In other words, the task generalization is the generalization of the task state representation and the task goal representation (Thomaz and Breazeal, 2008). The processes determine the essential and common actions and their dependency, and diversify required pre-conditions for involved actions so that adaptation to different task setups may be achieved (Asfour et al., 2008; Nicolescu and Mataric, 2003). The generated task structure needs to sufficiently describe and characterize the demonstrated task, and it can be instantiated in different setups to accomplish the taught task goals. An extended policy gradient algorithm was proposed in Cherubini et al. (2007) to find out the best behavior-composition strategy given a set of different strategies of accomplishing a task, and to determine related parameters for the involved behaviors corresponding to the best strategy. Here, the task was considered as a composition of different behaviors. The primitive operations and task plans might be further symbolized so that linguistic reasoning and planning of the task become possible (Ekvall and Kragic, 2006 and 2008; and Chella et al., 2006 and 2007).

It is also very common for a robot not to learn how to accomplish a task. Instead it learns what has been done from the human task teaching, i.e. the constitution of the sub-goals or state changes of a task (Ekvall and Kragic, 2006 and 2008; Nicolescu and Mataric, 2003). To accomplish these sub-goals in the actual task executions, low level actions or skills are automatically chosen based on their functionality and the goals.

In addition, methods for the robot to learn tasks incrementally and refine its task knowledge based on feedback or comments given by the teacher intuitively have also been proposed and studied, such as scaffolding and task knowledge re-generalization (Otero et al., 2008; Thomaz and Breazeal, 2008; Calinon and Billard, 2007; Pardowitz et al., 2007; and Nicolescu and Mataric, 2003).

2.3.3 Adaptation of Learned Tasks to New Setups

Adaptation of demonstrated tasks to new setups is still a challenging job (Asfour et al., 2008; Nakaoka et al., 2007). This includes adaption of both low-level sensor-motor skills and high-level actions (Cherubini et al., 2007). The adaption to new setups in most approaches, as in

Thomaz and Breazeal (2008), Ekvall and Kragic (2008) and Nicolescu and Mataric (2003), is done as follows: first, each subtask's pre-condition, sub-goal or until-condition, its dependency on the other subtasks, and the configuration and constraints of the current given task setup are all examined; then a feasible solution to achieve the task goal based on the current task configuration is found. The corresponding task trajectory is solved through the automatic task path planning of involved behaviors or actions. Only few reported how the taught task trajectories are adapted to new setups. Some focused more on the final results of the human demonstration, and the taught task trajectories were not utilized (Ekvall and Kragic, 2006, 2008; Meng and Lee, 2003; Lee et al., 2002; and Williams et al., 2000). A depth-first and left-to-right search strategy was described to reproduce a learned task with a hierarchical representation in Knoop et al. (2007). In Bentivegna et al. (2006), local and global features extracted from the demonstrations and practices of a maze game were used to encode the game and enable the robot to play the game on a different maze layout.

In this research, the task trajectories demonstrated by the human teacher are considered important as they usually include valuable information and reflect the teacher's preference to some degree. Specifically, the parts of the taught task trajectory when the robot gets close to objects of interest and engages with them, the related relative poses (locations and orientations) between the robot and objects, and the corresponding velocities are very important. The taught task trajectories should be generalized and then adapted to the new task setups as much as possible. Different types of Neural Networks and Gaussian Mixture Models (GMMs) are often used to generalize the multiple demonstrated task trajectories (Calinon and Billard, 2007; Lieberman and Breazeal, 2004; Martin and Nehmzow, 1995). Although these kinds of generalization can help the robot to adapt its learned task trajectories to different task setups, the robot may not be capable of adapting them to setups that are considerably different from those in the demonstration. In Coates et al. (2008), a generative model of the task trajectory, which was learned from multiple demonstrations, indeed showed its power and flexibility in allowing a helicopter to adapt its learned aerobatic maneuvers to complete air shows. However, it is not clear how to apply the related methods in the robot domain where different objects and targets may be involved. Another way to give the robot the ability to adapt its taught tasks is to describe the taught task trajectories in different reference frames (Calinon and Billard, 2007), particularly the object frame. In Ogawara et al. (2002), the demonstrated trajectories were represented as relative trajectories only in the object coordinate frame. However, further path planning is needed to bridge from the robot starting point to the task trajectory generated in the object frame based on the actual task setup when the setup is considerably different from that demonstrated. Lieberman (2004) partially solved this problem by generalizing taught task trajectories both in the world frame and the object frame. The actual task path was generated by blending the two generalized trajectories according to the actual setups. However, the adaptation ability might still be limited since the rotations of the task trajectories and objects, and robot starting points were not considered and the generalized task trajectories were encoded with radial basis function neural networks. If a task setup is considerably different from those demonstrated, some supposed task path points might be far from the receptive fields of all generated radial functions. As a preliminary part of this thesis research, Wu and Kofman (2008) extended the approach presented by Lieberman (2004) to enable a robot to generalize the task trajectory from a single demonstration and to fully adapt its learned task trajectories to different poses of objects, targets and robot starting points. There have been no reports in the literature on how to adapt the taught task motion or speed profile to new task setups.

In this research, the robot will be able to not only adapt the taught task paths to new task setups where poses of the objects, targets, and robot starting point may be remarkably different from those in the demonstration, but the robot will also apply the taught task motion or speed to the new setups.

2.4 Summary

The methods of teaching robot tasks by a human teacher were categorized and reviewed in three groups: teaching by guidance, teaching by passive demonstration, and human-style task teaching. In the context where a robot-task teaching method might enable a general user to effectively teach different tasks to a robot, teaching by guidance is the least intuitive and effective method while the human-style task-teaching methods hold the most promise.

Although teaching by passive human demonstration, specifically where the tasks are directly demonstrated by the teacher, might be the most simple and least time consuming for the user or teacher, the teaching process is not interactive and is unable to provide more valuable information to aid the robot to learn the taught task. In addition, hand-to-hand task demonstration through teleoperating the robot using the teacher’s natural body or hand motion brings much more leverage to help the robot learn the demonstrated task than the task demonstration done directly by the teacher. Human-style methods of teaching a robot tasks by a human teacher, where the human and robot are in a closed loop, hold the most promise that a general user could teach the robot tasks naturally, and timely feedback could be requested or offered intuitively. Only these methods seem to be feasible and practical for the ultimate goal of service-robot teaching and learning. However, current related research on HST still focuses more on the conceptual application, and it is still in the early stages, simple, and far from a systematic robot-task teaching and learning method that enables a general user to easily teach a service robot new tasks (Wu and Kofman, 2008).

With regard to the methods of robot learning tasks from human teaching, the learning task in whole approach is very suitable for the robot to learn primitive skills and simple tasks. Complex tasks would be better to be broken down into different subtasks first, and then learned and generalized starting from each individual subtask. This would permit hierarchical task plans to be learned and the robot could better adapt the learned task to new setups.

The following are important limitations of existing robot-task teaching and learning methods that are addressed in this research:

1) There is no known robot-task teaching and learning method that enables general users to teach different object-manipulation tasks to a robot intuitively and effectively. In this research, a systematic human-style task teaching and learning approach is proposed where the teacher is allowed to: check and teach background knowledge needed by the robot to learn a new task; introduce the overall task to the robot; teach the task step by

step; demonstrate the task in whole; and give timely feedback during the robot's task practice.

2) There is no opportunity for a teacher to check and teach the background knowledge needed by the robot to learn a new task. This is especially important in the service or personal robot domain, where robots are expected to serve a lifetime, be multi-functional, and constantly learn new tasks from general users' teaching.

3) The teacher's task expertise is not well exploited. Task expertise that can be exploited includes:

    - breaking down the task into a hierarchical task structure or plan,
    - determination of important task states,
    - segmentation of the taught task,
    - evaluation of the robot's performance when it is practicing the task.

    "Most work in this area places the majority of the learning burden on the robot" (Argall et al., 2007). In this research, the above task expertise will be exploited in ways that are intuitive to the teacher. Although timely feedback has been used in earlier work (Argall et al., 2007; Nicolescu and Mataric, 2001 and 2003), there is room to enhance the user's influence throughout the robot's learning process.

4) There lacks a flexible task-trajectory-segmentation algorithm that can utilize multiple types of signals and handle individual signal segmentation uncertainty and reliability. Lieberman (2004) utilized only speed information to segment trajectories. Although Peters et al. (2003) augmented the speed information with maximum and minimum robot elbow joint angle, they did not show how to combine the two types of information mathematically. Kulic and Nakamura (2008) segmented full-body motions into known motion primitives based on position and speed information.

5) Current methods of robot learning from human teaching still have difficulty in adapting the taught task trajectory to considerably different new task setups. The representation of the learned task trajectories in different reference frames and the use of blending schemes in Lieberman (2004) and Lieberman and Breazeal (2004) have clear advantages to enable the robot to adapt its task knowledge to different task setups. However, their approaches considered only the translations, and multiple demonstrations were needed. In addition, their radial-basis-functions-based trajectory encoding prevented the robot from adapting learned trajectories to new task setups considerably different from task setups in the demonstration.

6) There are very few user studies in the literature involving multiple users teaching object-manipulation tasks to a robot (Weiss et al., 2009). The proposed research includes a user study to evaluate the newly developed robot teaching and learning method and validate its potential usefulness as a robot-task teaching tool.

This research aims to develop a systematic method of human-style robot task teaching and learning that enables general users to teach different tasks to a robot effectively and intuitively, and allow the robot to learn the taught tasks as expected and to adapt the learned task to new setups, even those considerably different from the demonstration. The task expertise of the teacher will be well exploited to help the robot abstract the task knowledge, partition the taught tasks, and refine its task knowledge.

Chapter 3
Human-Inspired Robot Task Teaching and Learning

3.1 Introduction of the Robot-Task Teaching and Learning Method

In this section, the desirable characteristics for the proposed robot-task teaching and learning method, an overview of the method, and main assumptions in this research are elaborated. Details of the robot task teaching and learning method are given in Sections 3.2 and 3.3.

3.1.1 Desirable Properties of the Proposed Method

The proposed robot-task teaching and learning method aims to enable general users to teach a robot new tasks. There are several desirable properties of the method to make the task teaching and learning effective. These are briefly described in this section.

The method of robot task teaching and learning should be natural and intuitive to the general user so that users with no experience could easily learn to use the robot task teaching and learning system. To achieve this, the human-robot interaction should be familiar and ideally closely resemble human-human interaction. The users should therefore use natural and spontaneous utterances in response to requests from the robot and in giving feedback to the robot. The contents of the robot's requests, responses, and internal learning state revelation should be very understandable to general users.

The robot should be able to learn taught tasks effectively and robustly. The learning should be efficient enough to enable the robot to learn the taught task from a single human demonstration. This approach was called one-shot learning in Dillmann (2004), and it brings advantages over task learning methods that require the teacher to give multiple demonstrations (Calinon et al., 2007). Learning tasks robustly means that the robot should be able to learn the tasks as expected by the teacher under different conditions, and if the robot detects an error, it would be able to seek assistance from the teacher on related matters in ways that are natural to the general user. Also, the general user should be able to naturally change or give feedback on the robot's learned task knowledge at any appropriate time; and the robot should promptly refine its knowledge accordingly.

The teacher should have an understanding of the robot's current capacity and background knowledge needed for the new task, or determine it before teaching the task. At least, the teacher should be given the means to gain the understanding by interacting with the robot.

To make sure the task is learned by the robot as expected by the teacher, the teacher should be given the ability to monitor and control the progress of the task teaching and learning throughout the task-teaching and learning process. For example, the robot should ask the teacher if they are satisfied with the task demonstration right after completion of the demonstration of the task by the teacher. If the task demonstration is not satisfactory (subjectively determined by the teacher), the teacher will be given another chance to demonstrate the task to the robot. Otherwise, the robot and teacher will proceed to the next task teaching and learning stage.

The task teaching and learning process should be transparent (Grollman and Jenkins, 2007; Thomaz and Breazeal, 2006). To achieve this, the robot not only needs to actively reveal its internal learning states in ways that are easily understandable to general users, the robot should also quickly acknowledge its receipt and understanding of the teacher's instructions and promptly respond to the feedback from the teacher, for example, the timely feedback given during the robot task practice. The teacher should be allowed to ask the robot what knowledge it has and what capacity it has, such as which objects it can recognize. The teacher can then confidently track the learning progress of the robot, and better construct any further task teaching. The acknowledgement is especially helpful in the vocal human-robot interaction as the speech recognition engine may fail to recognize some of the human's utterances, and speech may be incorrectly recognized.

The task learning burden should be shared between the teacher and the robot. "Most work in this area places the majority of the learning burden on the robot", as stated by Argall et al. (2007). The task expertise of the teacher at the task level should be well exploited. The teacher should be able to help the robot understand tasks in a top-down fashion, for example, introducing a hierarchical task structure, and aid the robot to associate this high-level abstract task knowledge with low-level robot sensor data (such as robot end-effector position). During teaching, the teacher may direct the robot's attention to perceive the task-related key state changes. This will be called *human-directed perceiving* in this thesis.

Task knowledge should be represented in a way that is similar to what the teacher has in mind. The names of the tasks and subtasks and the related task structures have to make sense to both the users and the robot. Since the service robot will engage different users and serve a long time, its task knowledge has to be arranged in ways that can be easily understood by other users. This will enable the learned tasks and subtasks to be effectively reused in the process of new task teaching and learning.

3.1.2 Overview of the Method

The approach of teaching tasks to a robot in this thesis is inspired from human-to-human teaching in the classroom. Similar to a student needing prerequisite knowledge to enroll in a new course, the robot may need prerequisite knowledge to learn a new task. The teacher might check and teach any needed background knowledge if necessary, before focusing on the details of the new knowledge. A teacher also usually gives a brief introduction at the beginning of a course or lecture, and guides the students through large difficult problems step by step and then reviews the procedure. In addition, the teacher may pose questions after imparting some new important knowledge, give the students timely feedback based on their problem-solving performance, and make sure the students have successfully learned the new knowledge as expected. Meanwhile, some students may be actively involved in the learning process: energetically listening, making notes, revealing their learning progress, asking questions, and refining their knowledge based on the teacher's feedback. In this research, a similar approach is adopted to: allow general users to teach different object-manipulation tasks to a robot intuitively and effectively, and equip the robot to robustly learn the taught tasks and adapt its learned tasks to new task setups. The teacher (or user) and robot are in a closed loop such that the user can naturally teach the tasks, track the robot learning states and give timely feedback, while the robot can reveal its learning progress and refine its knowledge based on the teacher's feedback. This human-inspired method of robot-task teaching and learning, as shown schematically in Figure 3-1, consists of six stages as follows:

1) Teacher checks and teaches needed background knowledge. The teacher will first check if the robot has the needed background knowledge for the task to be taught, and then teaches the lacking knowledge to the robot if necessary. This step serves two purposes: (i) letting the teacher understand the robot's capacity to recognize related objects and perform pertinent tasks and subtasks; and (ii) providing needed prerequisite knowledge for the robot to learn new tasks and making it really possible for the robot to learn new types of tasks different from those pre-programmed by manufacturers.

2) Teacher introduces the overall task to the robot. The step of overall task introduction is a brief description of the task structure, involved objects, and robot hand actions. This step is carried out to exploit the teacher's expertise in the task before demonstration of the task. This gives the robot an overall top-down comprehension of the task to be taught and helps the robot construct a task structure that would be easily understood by other users. In addition, the information regarding the involved objects and hand actions can serve important roles in partitioning the later demonstrated task.

3) Teacher teaches the task to the robot step by step. This is an optional step whereby the teacher can selectively teach some important and complex subtasks to the robot, and direct the robot to perceive their crucial details. In addition, the subtasks taught in this step will offer valuable information to partition the later in-whole demonstrated task.

4) Teacher demonstrates the task in whole to the robot. The teacher demonstrates the whole task to the robot by teleoperating the robot using natural hand motion. One such teleoperation method uses vision-based hand-motion tracking, described in Kofman et al. (2005). The teacher may also provide some vocal subtask-segmentation cues at the subtask transitions using vocal utterances such as "first step" and "next subtask", which will be used to segment the demonstrated task trajectories.

5) Robot learns the newly taught tasks. The robot will first partition the demonstrated task trajectory, assign the partitioned trajectory segments or episodes to the introduced subtasks, and then generalize the taught task trajectory and task plan. If the robot has difficulties to learn the task, it will ask the teacher for help.

6) Robot practices the learned task and refines its task knowledge based on the teacher's timely feedback. The robot will explore its workspace and determine poses (locations and orientations) of objects and targets of interest, generate the actual task trajectory by adapting its learned task trajectory to the given task setup, and then perform the task while listening for the teacher's feedback and refining its task knowledge accordingly.

In each stage of the task teaching and learning process, rich and natural interactions are essential, not only to make the teacher comfortable to teach the robot, but also for the teacher to deliver and retrieve knowledge and keep the related processes transparent and effective.

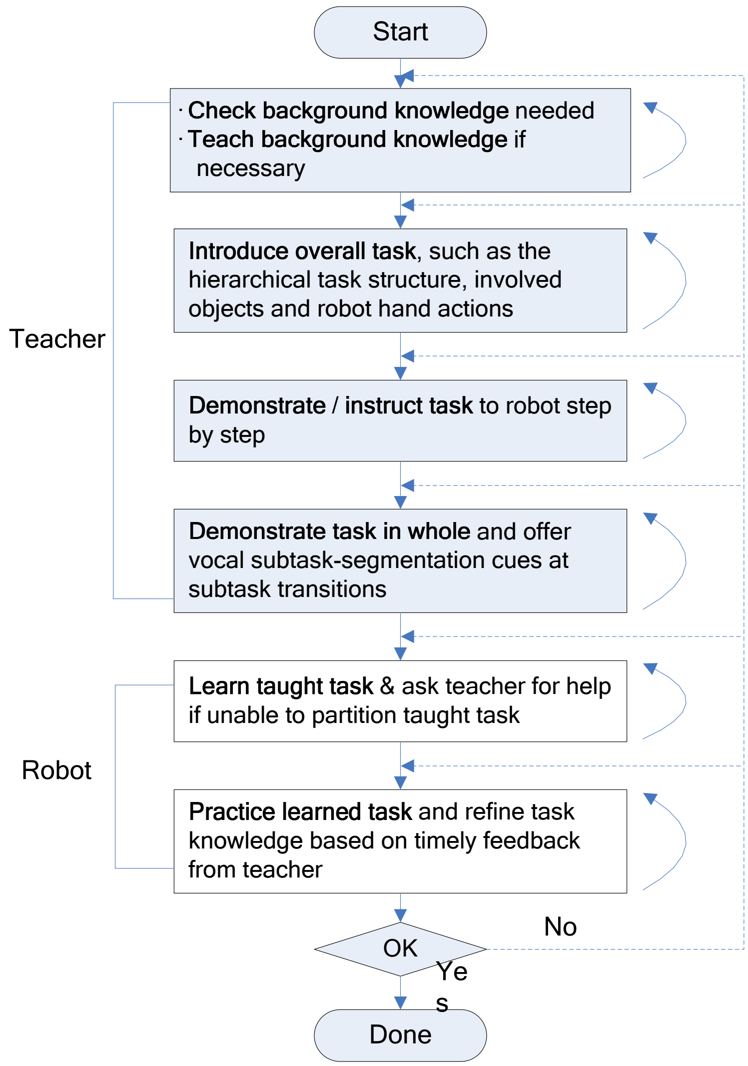


Figure 3-1: Schematic diagram of the proposed human-inspired robot-task teaching and learning method with stages carried out by the teacher and robot. Curved arrows indicate the timely and intuitive feedback given by the teacher. Dotted arrows are possible return paths to repeat related processes either totally or partially, if the practice result did not reach a satisfactory level.

3.1.3 Assumptions

Several important assumptions have been made in this research, related to task teaching and learning. Firstly, general users who would like to teach tasks to the robot are assumed to have expertise in the tasks at the task level, i.e. the users completely understand what object-manipulations or hand motions are needed to accomplish the task, and the involved individual steps or subtasks that compose the task.

The teaching of tasks to the robot is meant to be applied where there is a real need for the robot to perform the tasks. The teaching is not meant for entertainment purposes. The teacher needs to maximally use their task expertise at the task level to help the robot learn the tasks effectively and accurately as expected. The teacher has to carefully prepare the teaching just as a teacher would in preparing a lecture. The teacher should therefore conduct a task pre-analysis of: the task structure; involved basic robot hand actions; involved objects; expected important task-related states, such as the status of the robot hand; and task pre-planning, such as the task motion and motion transitions of subtasks.

The robot sensor system, which in this research is essentially the vision system that is responsible to sense and perceive the robot working environment, is presumed to be competent enough to detect and recognize objects of interest in the workspace and measure their poses (position and orientation). In this research, the detection and recognition of objects was simplified to a level that would enable the task teaching and learning to be carried out. Development of a more advanced robot-vision object detection and recognition system is beyond the scope of this thesis.

In addition, only object-manipulation tasks were considered in this research. Learning of other types of tasks were not addressed, although the presented robot-task teaching and learning method might be easily expanded to handle these types of tasks.

3.2 Human-Inspired Instructive Teaching of Object-Manipulation Tasks to Robots

In the human-inspired instructive teaching of object-manipulation tasks to robots, developed in this research, the main steps are: checking and teaching needed background knowledge, overall introduction of the task to be taught, teaching the task step by step, and demonstrating the task in whole. In addition, the human teaching is also tightly coupled with the robot learning process and task practices after the task demonstration. The teaching will be accomplished when the robot has learned the taught task as expected. Instructive teaching in this research means that the teacher is provided with the ability to give instructions to aid the robot to learn the task during the task teaching or demonstration. For example, the teacher can teach the robot subtasks transitions and modify information of the introduced subtasks.

3.2.1 Checking and Teaching Needed Background Knowledge

Before a task is taught, it is necessary for the teacher to make sure that the robot has the prerequisite background knowledge and the ability to acquire relevant information. Examples of background knowledge include knowledge about the objects that will be involved in the task, the working environment, and the basic object-manipulation skills. A novel step used in this research allows the teacher to check the robot’s background knowledge needed for the task to be taught and to teach what is lacking to the robot if necessary.

Regarding objects, the teacher may request the robot to reveal its knowledge of objects it has learned by simply saying to the robot "tell me what objects you know about". The teacher may also ask information about a specific object via an utterance, such as "what knowledge do you have about a fork?" In addition, to check if the robot is able to recognize an object of interest, the teacher can teleoperate the robot to point to the desired object and then ask the robot to identify the object by saying "tell me the name of this object". If the robot does not know the object name, the teacher can teach the name right away. The association of an object (i.e. the shape and surface features that are automatically extracted by the robot sensor system to represent the object) with a name can be explicitly learned by this kind of instruction from the teacher. The name can also be implicitly established in the process of learning the taught task (described below and in Section 3.3.2). The teacher may also check other object information such as surface-related and task-related information regarding a specific object. For example, its surface color, functional roles, and basic manipulation skills (such as grasping skill) may be checked. The teacher can teach or modify the object's name, color, functional roles, and typical place where the object can be found.

For the tasks and subtasks, the teacher can use a similar method to that used for objects to check the knowledge about all tasks or a specific task that the robot may have learned. The robot will respond with the task or subtask name and goals, its direct child subtasks, involved objects and robot hand actions. The teacher is allowed to change the relevant information except for the number of child subtasks that a task or subtask has.

For the working environment, it is important for the robot to have a clear understanding of its workspaces, and the spatial relationships between the objects of interest. However, the robot learning about the working environment was not one of the focuses in this research. A simple approach was developed and used for the robot to explore its workspace and detect objects of interest, as explained later in Section 4.2.4.

An example of a typical dialogue for the background-knowledge checking and teaching is given in Appendix A.2.

3.2.2 Overall Introduction of Task to Be Taught

A step of overall task introduction, a brief description of the task structure, is a novel contribution introduced in this research. The task introduction is carried out to exploit the teacher's expertise in the task before demonstration of the task. This gives the robot an overall top-down comprehension of the task to be taught and helps the robot construct a task structure that would be easily understood by other users. Kuniyoshi et al. (1994) used an approach where the robot automatically built the high-level task structure abstraction by analyzing the demonstrated tasks. However, there are two main reasons that it would be better for the human teacher to directly introduce the task structure to the robot: (1) it is very difficult for a robot to perform high-level abstraction, and (2) there would be no guarantee that the task structure generated by the robot itself would be similar to the task structure or plan that the teacher had in mind. This would in turn make it more difficult for any human-robot interaction and for the robot to reuse its learned knowledge.

Before performing the actual demonstration, the teacher/user would deliver a general introduction of the task structure that could include the task name, the number of the subtasks that compose the task, and information about the objects and actions involved. An suitable graphical user interface (GUI) or voice dialogue can be used for the introduction. The user would further give a concise description about each subtask, and child subtask, which includes similar types of information as for the overall task. For primitive subtasks, those that have no child subtask, information about robot-hand actions, (i.e. Open Gripper and Close Gripper) also needs to be specified. If voice dialogue is used by the teacher to introduce the task to the robot, the syntax of the utterances for the task/subtask names must be comply with one of the following two formats:

1): *ACTION* [*OBJ_MODIFIER*] *OBJECT* [from *SOMEWHERE*] [*DIR_PREP TARGET*]

e.g. "move this fork from table to placemat", and "place the knife onto table".

2): *ACTION DIR_PREP* [*OBJ_MODIFIER*] *OBJECT*

e.g. "pick up the knife" and "move up knife".

where each item in italics is one or several words from a list of expected relevant vocabulary or phrases; items inside square brackets are optional; *OBJ_MODIFIER* represents an object modifier such as "the", "my" and "three"; and *DIR_PREP* represents a directional preposition such as "to", "onto" and "down to". Thus, the information regarding the involved actions and objects in a task/subtask can be directly extracted from the task/subtask name by analyzing the syntax of the names. An example of a typical dialogue for the overall task introduction is given in Appendix A.2.

As a result of the task introduction by the teacher, the robot would establish a sketch of the hierarchical task structure. The information of involved objects, task/subtask actions (verbs), and robot-hand actions would serve as important information for task partitioning (discussed in Section 3.3.2). An example of a hierarchical task structure for a pick-and-place task is depicted in Figure 3-2. Two subtasks "pick up object" and "place object" compose the task. Subtask "pick up object" consists of three child subtasks that are also primitive subtasks "approach object", "grasp object" and "move up object". Subtask "place object" has two child subtasks that are also primitive subtasks "move object down to target" and "place object onto target". The primitive subtasks, indicated with rectangular blocks that have rounded corners in the figure, are those subtasks that have no child subtasks. This task will be used as an example to explain the proposed methods in the following sections. After the task introduction, the teacher will instructively demonstrate the task to the robot.

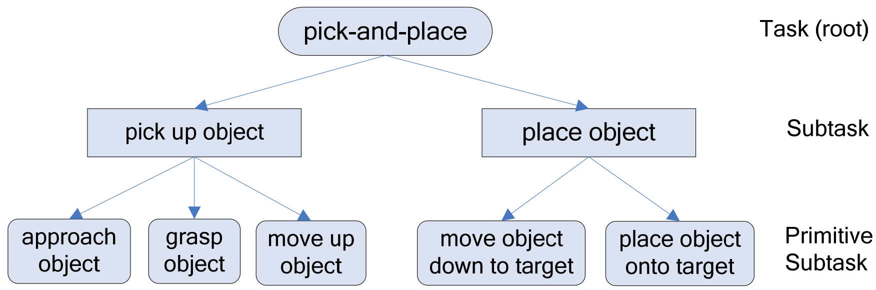


Figure 3-2: Hierarchical representation of a pick-and-place task. The overall task (root node) consists of two subtasks (sub-nodes): “pick up object” and “place object”, which both consist of primitive subtasks. Subtasks with rectangular blocks that have rounded corners are primitive subtasks; they have no child subtasks.

3.2.3 Teaching Task Step by Step

It may be too difficult for a robot to comprehend a complex task from just a single complete demonstration unless its learning system has been well customized and tuned for the task. Even when human beings learn tasks, they are usually taught step by step, for example, when learning a dance. It should be expected that for “less-developed” robots, learning step by step would be appropriate. As current robots have limited capability to perceive and understand the full details of a task from verbal or written instructions alone, it is reasonable to again exploit the teacher’s task expertise by having the teacher demonstrate the task. Inspired by the human teacher who takes the hand of a learner and guides them through a new task (a concept called *put through teaching* in Otero et al. (2008)), in this research, the teacher demonstrates a subtask to the robot through teleoperation using their own natural hand motion. In this thesis, this is termed *hand-to-hand* teaching.

In the stage of step-by-step task teaching, the teacher can selectively teach some subtasks or skip this teaching stage entirely. The order of the subtasks to be taught should be constrained by the nature of the subtasks. For example, some subtasks must be taught sequentially while others may be delivered in any order. The step-by-step teaching of a subtask is now detailed as illustrated in Figure 3-3.

For a given subtask, the teacher first tells the robot via voice the name of the subtask to be taught. Then, the robot responds to the teacher via synthesized voice by stating if it has learned the subtask. The robot then provides its detailed knowledge of the subtask learned in the overall task instruction. The teacher may correct the robot’s knowledge related to this subtask if needed. The teacher then demonstrates the subtask to the robot. While demonstrating the subtask, the teacher points out relevant state changes in the world and the subtask, and thus directs the robot’s attention to observe these changes. This is called *human-directed perception* in this research. For example, the teacher directs the robot to perceive the transitions of the child subtasks of this subtask by giving utterances such as “next subtask/ step” when the transitions occur during the subtask demonstration. At the same time, the robot watches the teacher’s demonstration, listens for instructions, senses its working environment and perceives salient state changes. It is essential for the robot to perceive and associate the real relevant state changes with each subtask. In reality, the robot may perceive some task-irrelevant state changes or omit some really important ones. Therefore, it is very important for the teacher to point out significant state changes to the robot and confirm the robot's related perception. Through this step-by-step demonstration, the robot can associate the introduced subtask with its acquired sensor data.

Once the subtask is finished being taught, the teacher would then teach the robot the next subtask in a similar way as described above, until all subtasks planned to be taught in this teaching stage have been completed. Then the teacher would demonstrate the task in whole in the next teaching and learning stage.

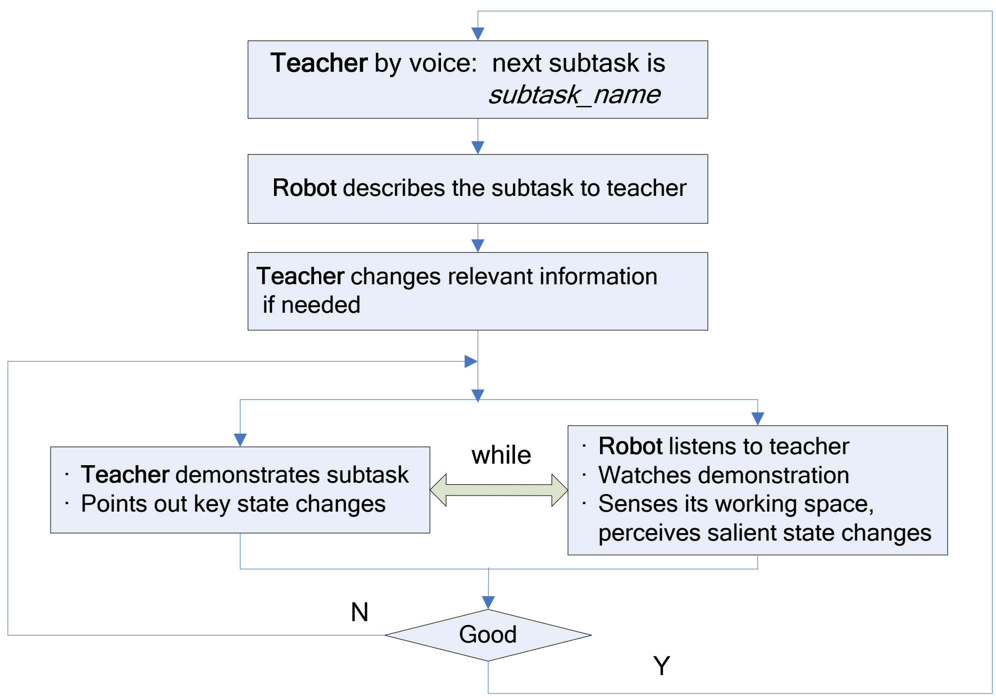


Figure 3-3: Diagram of step-by-step task teaching. The teacher first tells the robot the name of the subtask to be taught next. The robot then responses with its knowledge of this subtask obtained in the process of overall task introduction, and the teacher may revise relevant information if needed. The teacher then demonstrates this subtask to the robot.

3.2.4 Demonstration of Task in Whole

After step-by-step teaching, even after the robot has learned all subtasks of a task, the robot still lacks knowledge of the transitional motions between adjacent subtasks and knowledge of the overall task motion. The robot can obtain this information from an in-whole task demonstration, and then integrate it with the knowledge learned in the above step-by-step teaching stage.

The task would be demonstrated by the teacher with a focus on performing the task in a natural and fluent manner. The teacher only gives simple utterances, such as "first step", “then”, and “next step”, at the subtask transitions, based on their knowledge of the task and their possible prior experience in completing it. These simple organizational markers serve as useful cues for partitioning the task. Note that it is not necessary for the teacher to give any vocal segmentation cues at all in the in-whole demonstration stage and step-by-step teaching stage. However, the more cues given, the more the robot learning system can take advantage of the teacher’s expertise. At the same time, the robot actively observes the task, listens to the teacher’s instructions, immediately responds to the teacher’s requests, learns task-specific information such as changes of key states and alignment of objects, and records all pertinent data. After the demonstration, the robot will ask the teacher if they are satisfied with the demonstration. If the teacher considers the demonstration to be satisfactory, the robot will proceed to learn the taught task, including the task trajectory and detailed task structure, as explained in the next section. Otherwise, the robot will request the teacher to demonstrate the task again.

The human teaching continues to be coupled with the robot learning process and task practice until the robot has learned the taught task as expected.

3.3 Robot Learning of Taught Tasks

Once the teacher has finished the in-whole task demonstration, the robot will learn the introduced task by analyzing the observed task demonstration data in a bottom-up manner (i.e. directly starting from the robot sensor data and gradually building up the taught task), and associating the abstract task structure to the real robot sensor data. The learning of the taught task includes segmenting the demonstrated task trajectory, assigning the trajectory segments or episodes to the introduced task subtasks (ultimately each primitive subtask introduced by the teacher should have at least one trajectory episode assigned to it), and generalizing the taught task. The trajectory segmentation and assignment of trajectory episodes to primitive subtasks could be applied to trajectories demonstrated both in-whole and by step-by-step teaching.

3.3.1 Segmentation of Demonstrated Task Trajectory

Segmentation of the demonstrated task trajectory into episodes or segments is needed as each primitive subtask must be assigned at least one trajectory segment. A new flexible voting algorithm is developed for the robot to segment the taught task trajectory, and a probabilistic method is presented for the elimination of false trajectory episodes generated by the vote-based trajectory segmentation process.

3.3.1.1 Voting Algorithm for Task-Trajectory Segmentation

A new voting algorithm was developed for the robot to segment the demonstrated task trajectory. The method takes into account smooth and abrupt changes in signals, robot-environment relationships, and teacher-input information.

Candidate segmentation time instants along the taught trajectory can be generated as: (1) instants where signal values cross relevant thresholds (force or tactile sensor output could be useful; however, in the current system for this research, such sensor data were unavailable); (2) instants at which local minima or maxima of signal gradients occurred, such as minima of mean squared velocity (MSV) of robot end-effector motion; and (3) instants at which status changes or discrete events of interest happened, for example, the instant that a robot gripper closed or opened, or a teacher gave a vocal task-partition cue (the status of a door being changed from open to closed, or a light from on to off, are other examples not used in this thesis). The vote-based segmentation algorithm is based on the weighted sum of the votes from all of the above segmentation candidates.

The basic idea of using MSV to segment the human-teleoperated robot motion is based on the observation that human beings slow down their motions in the transition periods of action episodes (Lieberman, 2004). The MSV of the end-effector trajectory (not the MSV of the joint trajectories), is defined in this research as follows:

$$MSV(t) = \left(\dot{x}^2(t) + \dot{y}^2(t) + \dot{z}^2(t)\right) + \eta\left(\dot{\alpha}^2(t) + \dot{\beta}^2(t) + \dot{\gamma}^2(t)\right) \tag{3-1}$$

where $X(t) = [x(t)\ y(t)\ z(t)\ \alpha(t)\ \beta(t)\ \gamma(t)]^T$ represents the pose (location and orientation) of the end-effector at time t. α, β and γ are the rotation angles about axes x, y, z of the world frame, respectively. η is used to convert angular speed to linear speed. Segmentation candidates T_i^{MSV} from the MSV are time instants determined by the local minima of the MSV as follows:

$$T_i^{MSV} = t_j, \quad \text{if } \begin{array}{l} MSV(t_j) \text{ is a local minimum} \\ \cap\, MSV(t_j) < \delta_{MSV} \end{array} \tag{3-2}$$

where $i = 1, \cdots, n^{MSV}$; t_j is the j^{th} time sample; and δ_{MSV} is a threshold based on the average $MSV(t)$. n^{MSV} is the number of segmentation candidate instants found by analyzing the MSV.

Let T_i^k, $i = 1, 2, ..., n^k$ be the set of segmentation candidates generated by analyzing the k^{th} signal, where n^k is the total number of partition instants from the given signal. A vote for segmenting the task motion by the k^{th} signal, $v_k(t)$, is given by

$$v_k(t) = \mu_i^k e^{\left(-c_k\left(X(t) - X(T_i^k)\right)^2\right)}, \quad t \in \left[\frac{T_{i-1}^k + T_i^k}{2}, \frac{T_i^k + T_{i+1}^k}{2}\right) \tag{3-3}$$

where c_k determines the relative proximity of the segmentation candidate (obtained by analysis of the k^{th} signal) to the likely segmentation point. A greater c_k will influence the partition to be closer to T_i^k. For example, the trajectory for the pick-and-place task should be segmented very close to instants when events of gripper open or close occurred. However, the computed segmentation may not be so close to the instant where vocal segmentation cues were actually issued. This would be due to delays in teacher response and inability of the teacher to precisely determine the subtask transitions. c_k was determined based on the average speed of the demonstrated task trajectory. A corresponding scale factor was predetermined for each of the signals chosen for trajectory segmentation. μ_i^k is a factor that may be used to give a different treatment for the i^{th} segmentation candidate. The factor was used for the MSV segmentation candidates, where it would be set to 1 for MSV candidates adjacent to local maxima of *MSV*(t) and 0.7 for other MSV candidates. For signals other than MSV, the corresponding μ_i^k was always set to 1.

The overall vote $v(t)$ for segmenting the task motion is the weighted sum of the votes from all corresponding signals, and is defined by:

$$v(t) = \sum_{k=1}^{k=m} w_k v_k(t) \tag{3-4}$$

where m is the number of signals chosen to partition the task motion, and w_k is the weight of the vote by the k^{th} signal, predetermined based on its relative segmentation reliability as follows. Generally, votes from gripper open and close events are considered very reliable, votes from selected local minima of the MSV are least dependable, and votes from instants of vocal segmentation cues have median reliability. Equations (3-4) and (3-4) are flexible to include more segmentation signals.

The overall segmentation candidates, T_i, $i = 1,..., n$, can be determined by:

$$\begin{aligned} T_i = t_j, \text{ if } & v(t_j) \text{ is a local maximum} \\ & \cap\, v(t_j) > \delta_T \\ & \cap\ [\, D_A(t_j) > \delta_A \text{ before an object is grasped} \\ & \quad \cup\, D_N(t_j) > \delta_N \text{ before an object is placed} \,] \end{aligned} \tag{3-5}$$

where n is the total number of generated partition candidates, and threshold δ_T is based on the average $v(t)$. $D_A(t_j)$ is the distance between $X(t_j)$ (end-effector pose) and the position where the object is grasped along the robot-gripper approach direction before the grasp, while $D_N(t_j)$ is the distance between $X(t_j)$ and the position where the object is placed along the normal to the target surface before the object is placed. The corresponding thresholds, δ_A and δ_N, can be determined based on the length of the robot gripper or size of the robot hand. For example, δ_A and δ_N were set as the robot-gripper length and one third of the gripper length, respectively, in this research. The last condition in the equation shows that the task trajectory should not be segmented in the immediate region before an object is grasped or placed.

False segmentation and omission of segmentation may occur at some instants due to improperly selected δ_T. The occurrence of omitted partition instants can be reduced by decreasing δ_T at the cost of more possible false partition instants.

By the segmentation instants or candidates, the demonstrated task trajectory is divided into different episodes or segments. The i^{th} trajectory episode, E_i, is the trajectory segment,

$t \in [T_{i-1}, T_i]$. Some of the resulting episodes might be false. The following section deals with the elimination of the false trajectory episodes.

3.3.1.2 Elimination of False Trajectory Episodes

The outcome of the vote-based trajectory segmentation process is a set of the task trajectory episodes or segments. Some of the episodes may be considered false. A probabilistic method is presented to remove the false episodes.

Let $P(E_i)$ be the probability that the i^{th} trajectory episode E_i (i.e. the trajectory segment, $t \in [T_{i-1}, T_i]$) is a valid episode. $P(E_i)$ is defined as a function of the following factors: (1) whether the spatial length of the episode is long enough; (2) whether the duration of this episode is long enough; (3) whether the episode has clear characteristics, for example, primarily moving left/right/backward /forward/up/down, mainly turning left/right/up/down, or moving very fast; and (4) whether pertinent key events occurred in this episode, such as gripper closing or opening. E_k is considered to be a false episode as follows:

$$E_k \text{ is a false episode, if } P(E_k) < \delta_E \tag{3-6}$$

where δ_E is a threshold based on the average $P(E_i)$. If a false episode is found, E_k is merged to episode E_{k-1} if $v(T_{k-1}) > v(T_k)$, otherwise to episode E_{k+1}. Segmentation instants and episodes are re-indexed, and the elimination process is repeated until no more false segments can be found. The resulting segmentation instants are denoted as T_j, $j = 1, \cdots, n^E$, where n^E is the number of these instants.

The elimination should be conservative; otherwise some true segments might be removed. There might still be some false episodes remaining; however, this is not an issue as the main purpose is to perform a subtask-wise segmentation of the taught task.

3.3.2 Assignment of Trajectory Episodes to Introduced Primitive Subtasks

At the completion of task trajectory segmentation, the primitive subtasks of the introduced abstract task structure are not linked (or grounded) to any robot sensor data (e.g. end-effector pose). This grounding is solved by appropriately assigning the trajectory episodes (segments determined by the trajectory segmentation) to the primitive subtasks. The assignment of an episode to a particular primitive subtask depends on the joint probability that the subtask and episode mutually correspond. The joint probability can be expressed as:

$$P(E_i, S_j) = P(S_j \mid E_i) P(E_i) \tag{3-7}$$

where S_j is the j^{th} primitive subtask, $j = 1, ..., n^S$, $i = 1, ..., n^E$; and n^E and n^S are the total numbers of trajectory episodes and primitive subtasks, respectively. $P(E_i)$ is the probability that the i^{th} episode, E_i, is a true segment while $P(S_j \mid E_i)$ is the conditional probability that E_i should be assigned to S_j given E_i. $P(S_j \mid E_i)$ is a function of the following factors: (1)

how well the characteristics of E_i match the action and related directional preposition of the primitive subtask S_j, for example, if S_j is the subtask: "move object down to target", then the system can check whether the motion of E_i is mainly downward and the distance between the robot and target is considerably reduced (to match the directional preposition "down to") (note that the action here is the verb of the subtask name of S_j, not the hand action of the robot); (2) whether salient events that occurred in this episode match the action of S_j, for example, the subtask "grasp object" most likely expects a "close gripper" event to have happened in one of the assigned episodes; (3) whether the objects involved in this episode match the objects introduced in the primitive subtask S_j; and (4) whether the robot-hand actions specified to S_j by the teacher during the overall task introduction match the robot hand actions that occurred in the E_i. The first two matching relationships above can be statistically computed from the robot's previous experience. For example, when reviewing its experience, the robot would most likely find that a "Close Robot Hand" or "Close Gripper" action should occur in the trajectory episodes that were assigned to subtasks such as "grasp object" and "pick up object". As another example, an object should be raised up significantly compared to the motion along the other two orthogonal directions in episodes belonging to a subtask "move up object".

The final assignment of the trajectory episodes to primitive subtasks is solved through an optimization process, where the optimal assignment will produce a maximum score. The objective or cost function is given as follows:

$$
\begin{aligned}
J = &\sum_{j=1}^{n^S} P\left(S_j, \{E_i^j\}\right) - \kappa_1 \Omega\left(\{S_j\}, \{E_i^j\}\right) - \kappa_2 \sum_{j=1}^{n^S} \Psi\left(S_j, \{E_i^j\}\right) + \kappa_3 \sum_{l=1}^{n^V} \Phi\left(T_l^V, \{E_i^j\}\right) \\
&+ \kappa_4 \sum_{q=1}^{n^O} \mathrm{H}\left(O_q, \{S_j\}, \{E_i^j\}\right)
\end{aligned}
\tag{3-8}
$$

where $\{E_i^j\}$ is the set of trajectory episodes assigned to the j^{th} primitive subtask S_j; it may be empty although each primitive subtask should actually be assigned at least one episode. $P\left(S_j, \{E_i^j\}\right)$ represents the sum of the $P(S_j, E_k)$, where $E_k \in \{E_i^j\}, 1 \le k \le n^E$. $\Omega\left(\{S_j\}, \{E_i^j\}\right)$ is a penalty function that verifies if any subtask has not been assigned at least one trajectory episode. This should be avoided and the sign of the term is therefore negative in Equation 3-8. $\Omega\left(\{S_j\}, \{E_i^j\}\right)$ is defined as follows:

$$
\Omega\left(\{S_j\}, \{E_i^j\}\right) = \begin{cases} 1, & \text{if existing } \left\{E_i^j\right\} = \mathbf{0},\ j \in [1, \ldots n^s] \\ 0 & \textit{otherwise} \end{cases}
\tag{3-9}
$$

$\Psi\left(S_j, \{E_i^j\}\right)$ is also a penalty function that tests if robot hand actions that happened on trajectory episodes belonging to $\{E_i^j\}$ do not include all the robot hand actions that S_j is supposed to have, as specified by the teacher in the task introduction stage. This should be avoided and the sign of the term is therefore negative in Equation 3-8. $\Psi\left(S_j, \{E_i^j\}\right)$ is defined as follows:

$$\Psi\left(S_j, \{E_i^j\}\right) = \begin{cases} 1, & \text{if } S_j \text{ has at least one robot hand action specified by the teacher,} \\ & \text{and hand actions that happened in } \{E_i^j\} \text{ do not include all the} \\ & \text{specified actions} \\ 0, & \text{otherwise} \end{cases} \tag{3-10}$$

$\Phi\left(T_l^V, \{E_i^j\}\right)$ is used as a reward if a trajectory segmentation candidate, T_l^V, $l = 1,2,\cdots,n^V$, that is very close to a vocal subtask-segmentation cue indeed separates two groups of trajectory episodes, which were assigned to two consecutive primitive subtasks, respectively, where n^V is the number of this kind of segmentation candidate. $\Phi\left(T_l, \{E_i^j\}\right)$ is defined as follows:

$$\Phi\left(T_l^V, \{E_i^j\}\right) = \begin{cases} 1, & \text{if } T_l^V \text{ seperates two groups of episodes, for example,} \\ & \text{to } \{E_i^j\} \text{ and } \{E_i^{j+1}\} \\ 0, & \text{otherwise} \end{cases} \tag{3-11}$$

$\mathrm{H}\left(O_q, \{S_j\}, \{E_i^j\}\right)$ is also used as a reward if the q^{th} involved object, $O_q, q \in [1 \quad n^O]$, matches one of the objects specified by the corresponding primitive subtask name (e.g. two objects, "knife" and "placemat", indicated in subtask name "place knife onto placemat") based on the current trajectory-episode-primitive-subtask assignment. For example, if O_q was grasped in the i^{th} episode (E_i) and E_i is currently assigned to the primitive subtask S_j, O_q has to be one of the objects directly inferred from the name of S_j. $\mathrm{H}\left(O_q, \{S_j\}, \{E_i^j\}\right)$ is defined as follows,

$$\mathrm{H}\left(O_q, \{S_j\}, \{E_i^j\}\right) = \begin{cases} 1, & \text{if } O_q \text{ was manipulated in an episode belonging to } \{E_i^j\}, \\ & \text{and is one of the objects specified by the name of } S_j \\ 0, & \text{otherwise} \end{cases} \tag{3-12}$$

κ_1, κ_2, κ_3, and κ_4 are four weighting factors, constrained as $\kappa_1 >> \kappa_2 >> \kappa_3, \kappa_4 > 0$. Since the coefficient of $\Omega\left(\{S_j\}, \{E_i^j\}\right)$, κ_1, is very large, any non-zero value produced by $\Omega\left(\{S_j\}, \{E_i^j\}\right)$ will result in a very large negative value of the objective function.

The optimal episode-subtask assignment, χ^*, is determined by,

$$\chi^* = \arg\max_{\chi \in \mathbf{\Pi}} J(\chi) \tag{3-8a}$$

Where $\chi = \left\{\{E_i^1\}, \{E_i^2\}, \ldots, \{E_i^{N^S}\}\right\}$ is a candidate episode-subtask assignment for the N^S primitive subtasks. $\mathbf{\Pi}$ is the set of possible episode-subtask assignments with a constraint that if two adjacent episodes E_i and E_{i+1} are assigned to two different primitive subtasks S_j and S_k, respectively, it must be true that $k > j$.

A coarse-to-fine approach is used to search for the optimal trajectory-episode-to-primitive-subtask assignment. Brute force is applied in the fine search only. The optimization procedure is designed as shown in Table 3-1.

Table 3-1: Coarse-to-fine brute-force optimization process to assign obtained task trajectory episodes to introduced primitive subtasks.

1. Coarsely partition the sequences of the task-trajectory episodes and primitive subtasks into groups. Several hypotheses will be generated, and each hypothesis is a possible segmentation of the sequences. The adjacent primitive subtask groups *overlap* by one primitive subtask.
2. ***For*** each hypothesis:
3. ***For*** each group in the current hypothesis:
4. Compute the optimal episode-subtask assignment for the current group using Equation (3-8a), where $\mathbf{\Pi}$ is modified to represent all possible assignments for the current group with the same constraint.
5. ***End*** of "For each group in the current hypothesis".
6. Generate optimal assignment for the current hypothesis by combining the group optimal assignments (explained in text).
7. ***End*** of "For each hypothesis"
8. Compute the global optimal episode-subtask assignment (χ^*) using Equation (3-8a), where $\mathbf{\Pi}$ is modified to represent the set of all hypothesis optimal assignments (one from each hypothesis).

There are three possible ways to coarsely partition the sequences of the task trajectory episodes and primitive subtasks, as described in Line 1 in Table 3-1, based on: 1) robot hand actions that occurred in the in-whole demonstrated trajectory episodes and the robot hand actions that were specified by the teacher during the overall task-introduction stage; 2) key events (such as the robot hand actions) that occurred in the in-whole demonstrated episodes, and corresponding key events of *previously learned subtasks* that are now part of the taught task; and 3) key events (such as the robot hand actions) that occurred in the in-whole demonstrated episodes, and corresponding key events of *subtasks taught in the step-by-step teaching stage*. The last two approaches might not be possible to carry out if the robot has not previously learned the relevant primitive subtasks or if no subtasks were taught in the step-by-step stage.

The last primitive subtask in one group should be also be included in the next group since the task trajectory is not always partitioned right after the robot-hand action events (or other key events) in a primitive subtask-wise sense. Referring to Line 6 in Table 3-1, the combination of the group optimal assignments is the optimal assignment for the given hypothesis due to the group overlap.

For example, suppose the in-whole demonstrated task trajectory is segmented into episodes: $E_1, E_2, \cdots, E_{12}$; and a Close-Gripper action and Open-Gripper action happened in E_3 and E_{10}, respectively. Suppose also that the introduced primitive subtasks are: S_1, $S_2, \cdots, S_6$; and a Close-Gripper action and Open-Gripper action were specified by the teacher (during task introduction) for S_2 and S_5, respectively. With the first coarse partition approach, where the hand actions of the episodes should match the hand actions specified for the primitive subtasks, E_3 would be assigned to S_2, and E_{10} would be assigned to S_5. The episodes and primitive subtasks are thus broken into three groups by the hand actions. The hypothesis for this coarse assignment is: Group 1: E_1, E_2, and E_3 for S_1 and S_2; Group 2: $E_4, \cdots, E_{10}$ for $S_2, \cdots, S_5$; and Group 3: E_{11} and E_{12} for S_5 and S_6. Note that S_2 is in both Groups 1 and 2, and S_5 is in both Groups 2 and 3, as discussed above.

Except for the first group, it is possible that the first primitive subtask in a group may not be assigned with any trajectory episode. There would not be any penalty by $\Omega\left(\{S_j\}, \{E_i^j\}\right)$ for this case since this first primitive subtask is expected to be assigned at least one episode in the last group. For instance, as in the above example, S_2 in Group 2 and S_5 in Group 3, in the above example, may not be assigned any episode. S_2 and S_5 should be already assigned at least one episode in Group 1 and Group 2, respectively.

After the optimal episode-subtask assignment χ^* is found, if $J(\chi^*) < -\kappa_2 / 2$, the process is considered as having failed to properly assign the trajectory episodes to the primitive subtasks. The robot system may then lower δ_T in Equation (3-5) to generate more trajectory episodes and repeat the above processes if some primitive subtasks were not assigned at least one episode. Alternatively, the robot may directly ask the teacher for help to segment the demonstrated task via vocal human-robot dialogue. If $J(\chi^*) \geq -\kappa_2 / 2$, the robot will proceed to the next step. In this case, the assignments of trajectory episodes to primitive subtasks will be considered successful.

If the robot requests help, the robot can replay the demonstrated task, state its obtained segmentation and assignment results to the teacher, and ask for feedback from the teacher. The teacher may simply use utterances, such as "stop here", "move forward/ backward", and "the last/next episode should belong to *primitive_subtask_name*", to modify the trajectory segmentation and trajectory episode-to-subtask assignment.

With these episode-subtask assignments, the name of an unnamed object involved in the task can be inferred and associated with the group of object features that uniquely represent the object. The features could include surface-based features such as the color, and contour-based or shape-based futures such as roundness and moment of area. Each time when the robot is teleoperated to grasp an object in the task demonstration, the robot can determine which object (not its name) was just grasped based on the proximity of the locations of the object and robot hand when the object was grasped. Thus, the robot system should know which object was engaged in which task trajectory episode. The names of objects can be extracted from the subtask names according to the episode-subtask assignments. For example, if an object is grasped in a trajectory episode and the episode is assigned to the primitive subtask "grasp knife", then the name of the object should be "knife".

3.3.3 Generalization of Demonstrated Task

Once trajectory episodes are assigned to subtasks, generalization of the demonstrated task, including task trajectory and task structure, is performed. Generalization is performed to facilitate robot adaptation of its learned task to new task setups.

3.3.3.1 Generalization of Task Trajectory

A pick-and-place task is used as an example here to illustrate the task generalization method. A robot must dock (reach a specific pose relative to the object or target of interest) before starting to grasp an object or place an object on a target. The task trajectory generalization process begins by determining the docking poses. The task trajectory is then generated in different task-relevant Cartesian frames.

a) Determination of Docking Poses

A docking pose is defined relative to the object or target, and should be close to its corresponding grasping or releasing pose. The grasping pose or releasing pose is the relative robot pose at the segmentation instant corresponding to the related robot hand action, with respect to the pose of the engaged object or target, respectively. Transformation from the docking pose to grasping pose, as shown in Figure 3-4, only allows two translations, one in the end-effector frame along the approach direction (X_E in the figure) and one perpendicular to the plane defined by the approach (X_E) and lateral (Y_E) directions (Z_E not shown in the figure). Suppose the segmentation point (SP) or instant corresponding to the robot Close-Hand action to grasp an object is the i^{th} SP, T_i, as in the figure. The corresponding docking pose is computed by: 1) determining the relative pose $[x_d \; y_d \; z_d \; \alpha_d \; \beta_d \; \gamma_d]^T$ of the robot end-effector pose $X(T_{i-1})$ at T_{i-1} with respect to the grasping pose $X(T_i)$; 2) setting $\mathbf{A}_r = I_{4\times4}$ (4 by 4 identity matrix), and $\mathbf{A}_r[1][4] = x_d$ and $\mathbf{A}_r[3][4] = z_d$ (only translations along X_E and Z_E allowed); 3) $\mathbf{A}_D = \mathbf{A}_G \mathbf{A}_r$, where $\mathbf{A}_G$ is the homogeneous matrix representation of the grasping pose $X(T_i)$ in the world frame; and 4) determining the docking pose from $\mathbf{A}_D$.

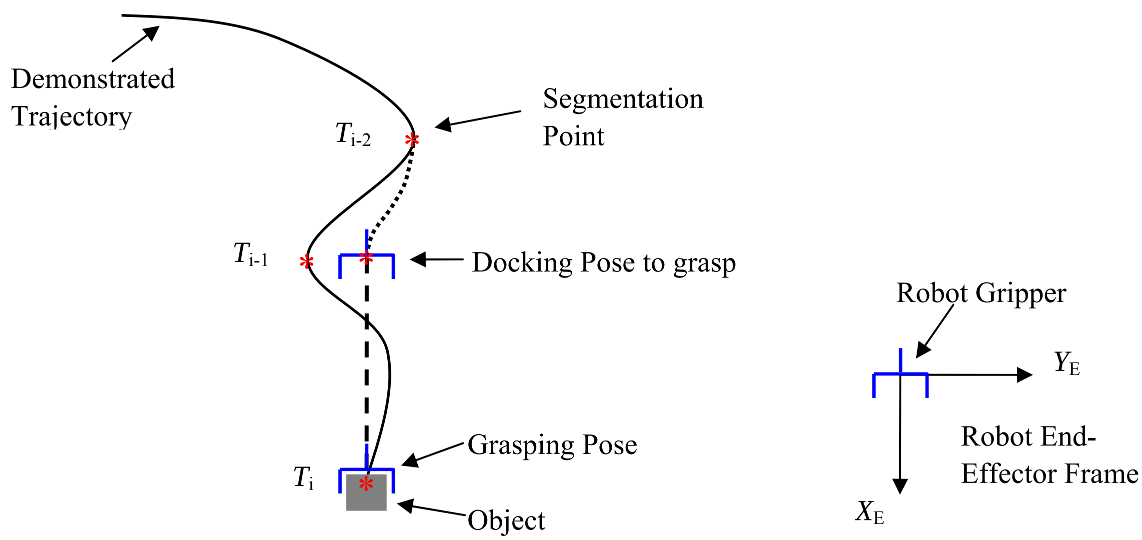


Figure 3-4: Determination of docking pose for the robot to grasp an object. Only translations along the approach direction (X_E) and Z_E (not shown in the figure) of the robot end-effector frame are allowed for transformations from the docking pose to its corresponding grasping pose.

The docking poses to release an object are determined in a similar manner to that used for docking poses to grasp except that the transformation from the docking pose to its corresponding final releasing pose only allows one translation direction, along the normal to the target surface (not shown in the figure).

The trajectory episode from the docking pose to its paired grasping pose or releasing pose is then reconstructed along the straight line trajectory using the same number of point samples as used in the demonstration (dashed line, Figure 3-4). The altered trajectory segments (i.e. from the newly determined docking pose to its paired grasping pose and from the newly determined docking pose to its paired final releasing pose) are considered in this research as *critical episodes* that must be truly followed in the task execution.

b) Generalization of Task Trajectory in Different Reference Frames

The demonstrated task trajectory is generalized in different reference frames: the world frame, object frames, and object-oriented frames to aid the robot to adapt the taught task trajectory to new task setups as described in Section 3.3.4. A new type of reference frame, called object-oriented frame, is introduced below. The robot end-effector pose is considered to be the same as the robot pose. The robot end-effector frame coincides with the gripper frame.

Generalization of task trajectory in the world frame. After having determined the docking pose for grasping or placing (or releasing), and having rebuilt the related critical episodes (dashed line in Figure 3-4), the adjacent episodes right before the critical episodes must also be altered to make related transitions smooth (dotted line in figure) as in the demonstration. Then the new task trajectories will be further smoothed and re-sampled based on the motion information given in the demonstration and constraints of robot speed, acceleration and jerk. This results in the generalized task trajectory $\mathbf{\Gamma}_W$ in the world frame $\mathbf{F}_W$ as follows:

$$\mathbf{\Gamma}_W = \{X_i\},\ i = 1,..,h \tag{3-13}$$

where $X_i = [x_i\ y_i\ z_i\ \alpha_i\ \beta_i\ \gamma_i]^T$ is the i^{th} robot end-effector pose of the generalized task trajectory expressed in $\mathbf{F}_W$, and h is the total number of sampled trajectory points.

Generalizing task trajectory in object frames. The k^{th} object frame is denoted as $\mathbf{F}_O^k$. $\mathbf{\Gamma}_W$ is expressed in the object frame $\mathbf{F}_O^k$ as follows:

$$\mathbf{\Gamma}_O^k = \{{}^O X_i^k\},\quad i = 1,..,h;\ k = 1,...,q \tag{3-14}$$

where ${}^O X_i^k = [{}^O x_i^k\ {}^O y_i^k\ {}^O z_i^k\ {}^O \alpha_i^k\ {}^O \beta_i^k\ {}^O \gamma_i^k]^T$ is the i^{th} robot end-effector pose, X_i, expressed in $\mathbf{F}_O^k$; and q is the number of objects involved in the taught task, including involved targets. Targets are treated as objects.

Generalizing task trajectory in object-oriented frames. A new type of reference frame, called *object-oriented frame*, $\mathbf{F}_G$, is now introduced. As illustrated in Figure 3-5, the first object-oriented frame $\mathbf{F}_G^1$ is constructed so that its vertical axis Z_G is parallel to the vertical axis of the world frame and the axis passes through the origin of the end-effector frame $\mathbf{F}_E$ when the robot is at its starting position, and its X_G axis passes through the origin of the object frame $\mathbf{F}_O$ defined on the object that the robot is approaching. After the first object engagement, Z_G of $\mathbf{F}_G^k$ (object-oriented frame $k > 1$) must pass through the origin of the object frame defined on the object that has just been engaged during the demonstration as shown by Z'_G.

The task trajectories generalized in the k^{th} object-oriented frame, $\mathbf{F}_G^k$, can be obtained as follows:

$$\mathbf{\Gamma}_G^k = \{{}^G X_i^k\},\quad i = 1,..,h;\ k = 1,...,q \tag{3-15}$$

where ${}^G X_i^k = [{}^G x_i^k\ {}^G y_i^k\ {}^G z_i^k\ {}^G \alpha_i^k\ {}^G \beta_i^k\ {}^G \gamma_i^k]^T$ is the i^{th} robot end-effector pose in the world frame, X_i, expressed in $\mathbf{F}_G^k$, and q is the number of the objects (including targets) involved in the taught task.

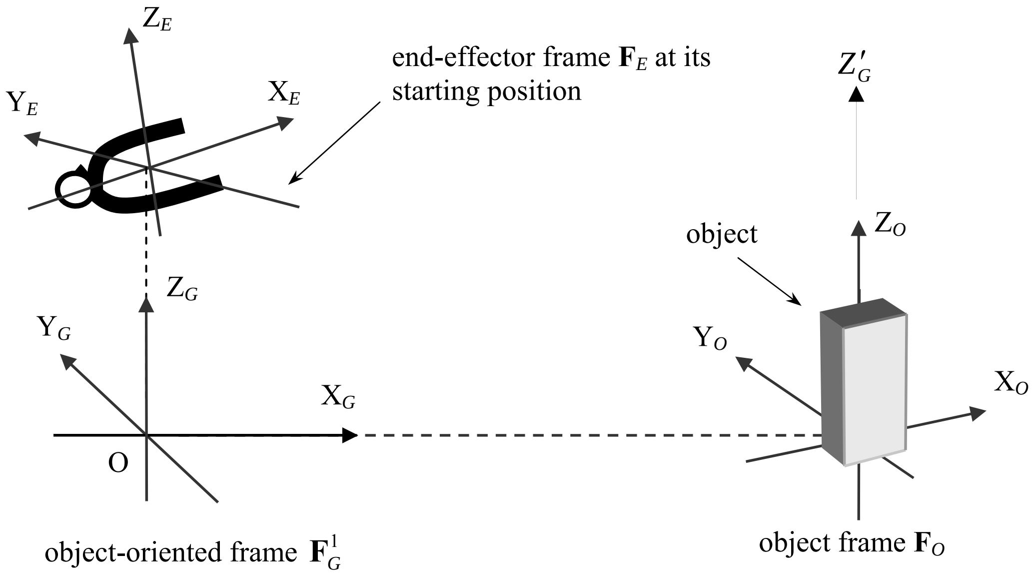


Figure 3-5: Definition of the first object-oriented frame $\mathbf{F}_G^1$ when the first object is approached by the robot. The Z_G axis of $\mathbf{F}_G$ is vertical and passes through the origin of the robot end-effector frame when the end-effector is at its starting position. The X_G axis passes through the origin of the object frame. Z'_G is the Z_G axis of the second object-oriented frame.

The advantage of the newly introduced object-oriented reference frame is that it can maintain the relative object-robot (i.e. object-gripper) orientation relationships during task practice and execution almost the same as those taught in the task demonstration. These relationships may encode desired observation perspectives for the robot's sensor system when the robot is in motion. As illustrated in Figure 3-6, the demonstrated trajectory (the blue line) of a grasping-object task was straight and the robot gripper (in red) directly faced the object in the demonstration, where the movement of the robot was constrained to a horizontal plane, i.e. two translations and one rotation about the normal to the plane. Two different task paths (originating from the robot starting pose) were generated for each of two task setups: the trajectory (in black) obtained by blending the trajectory produced in the current object frame $\mathbf{\Gamma}_O$ with the trajectory generalized in the world frame $\mathbf{\Gamma}_W$, and the task path (in green) by blending the trajectory $\mathbf{\Gamma}_O$ computed in the current object frame with the trajectory $\mathbf{\Gamma}_G$ generated in the current object-oriented frame. Based on the direction of the gripper overlaid on the generated task trajectories, the relative orientation relationship between the robot gripper and the objects was largely maintained along the green lines, while the gripper generally did not directly face the object along the black lines. Note that no docking was considered here; the blending rates were simply calculated as a proportion of the distance from the robot starting point to the final position where the object was grasped in the demonstration. The blending schemes are detailed in Section 3.3.4.2.

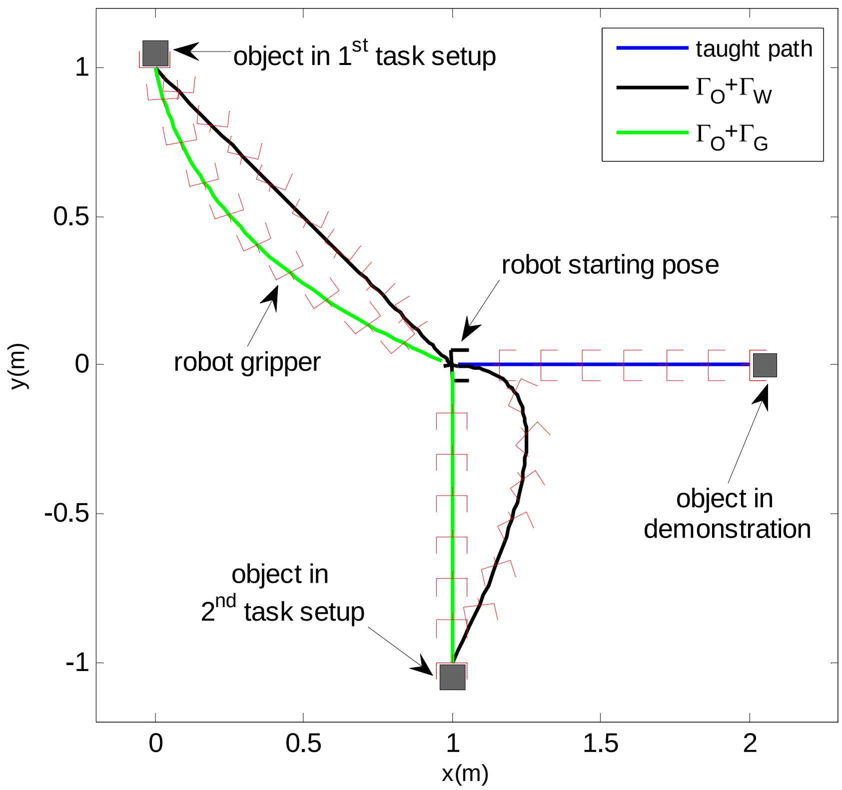


Figure 3-6: Advantage of the object-oriented frame ($\mathbf{F}_G$) for an object-grasping task: maintaining the taught (demonstrated) object-robot (object-gripper) relationship during task practice and execution. The robot end-effector has three DOFs: two translations in the horizontal plane and one rotation. The taught task path (in blue) is a straight line with the robot gripper (red) overlaid, and involves only translations. Two different task paths originating from the robot starting pose were generated for each of two task setups: one (black) by blending the trajectory, $\boldsymbol{\Gamma}_O$, produced in the object frame, and the trajectory, $\boldsymbol{\Gamma}_W$, generalized in the world frame; and the other one (green) by blending the $\boldsymbol{\Gamma}_O$ with the trajectory, $\boldsymbol{\Gamma}_G$, generated in the object-oriented frame.

3.3.3.2 Generalization of Task Structure

Generalization of the task structure includes determination of the pre-condition, until-condition (or subtask goal) and associated actions or primitive skills for each subtask, and execution orders of these subtasks.

The pre-condition of a subtask is a set of required prerequisites from which the subtask starts. The until-condition for a subtask is the goal or set of states that must be established before the subtask is considered to be accomplished. For example, for the subtask “approach object” in the pick-and-place task, the precondition is that the robot gripper must be open and the robot either knows the approximate location of the object of interest or knows how to search for the object. For the same subtask, the until-condition (subtask goal) is that the robot must be docked with respect to the object. The until-condition of the subtask “approach object” is the precondition of the next subtask “grasp object”.

Primitive skills or actions are extracted from the task demonstration before the task generalization. Approaches to recognize primitive skills from robot sensor data can be found in Bentivegna and Atkeson (2003) and Pardowitz et al. (2007). The resulting primitive skills are then associated with relevant primitive subtasks based on the episode-subtask assignment. The conditions upon which the primitive skills will be performed must also be specified. For example, for the pick-and-place task, two primitive skills, “Close Robot Hand (or Gripper)” and “Open Robot Hand (or Gripper)”, could be extracted from the task demonstration. The two primitive skills would then be associated with the primitive subtasks “grasp object” and “place object onto target”, respectively. The robot would close its gripper when the grasping pose is reached in the subtask “grasp object”, and it would open its gripper when the release pose is reached in the subtask “place object onto target”.

The relationships between subtasks affect the subtask execution order. The subtask order can be obtained by analyzing the dependency of the goals of the subtasks. The robot may also ask the teacher to confirm or clarify the relationships between subtasks. The relationships between sibling subtasks can be classified as follows: (1) the subtask should be strictly sequential; (2) some or all of the subtasks can be executed in any order, which means that the robot will have flexibility to decide which subtasks are executed first based on the configuration of the given task setup; and (3) some of the subtasks are optional, depending on the task setup configuration. The relationships between subtasks are very important, and should be incorporated into the task representation that is detailed in Section 4.1.

3.3.4 Adaptation of Learned Task to New Task Setups

It is crucial for the robot to be able to adapt its learned tasks to new task setups. This is especially true for service robots, where the possible task setups for a given task are often very different from that in the demonstration. To achieve the goal of a learned task in the new task setups, three important task-planning issues have to be solved, how to: adapt the learned task plan to the new task configuration and constraints (this is a high-level adaptation and involves determination of optimal subtask-execution order); adapt the learned task paths to new task setups; and maintain a similar motion (or speed) profile to that demonstrated.

3.3.4.1 Determination of Subtask Execution Order

To adapt the learned task to a new task setup, the execution order of the relevant subtasks must be determined. The execution order would be based on the subtask pre-conditions, until conditions (or subtask goals) and dependency on other subtasks, overall task goal, and current task configuration and constraints. Sometimes extra steps or actions would have to be taken. For example, “Object B” could be directly dependant on “Object A” according to the goal state of a task, while “Object A” is blocked by “Object B” in a current task setup. An extra action to move “Object B” away from “Object A” would be necessary in order to enable the robot to access “Object A”. More detail on task-plan reasoning and planning to adapt to new task setups can be found in (Ekvall and Kragic, 2006 and 2008; Chella et al., 2006 and 2007; Kuniyoshi et al., 1994). Determination of the optimal subtask-execution order has not been developed in this research, but would be important to be included in a future version of the robot-task teaching and learning system described in Section 4.2.

3.3.4.2 Generation of Task Path from Generalized Task Trajectories

Once the execution order of the subtasks of the task is determined, the robot needs to generate actual task paths based on the actual task setup and the learned task trajectories. Lieberman (2004) proposed a blending mechanism to adapt a learned task path to new task setups. However, their method considered only translations and blending of trajectories generalized in the world and object frames using two neural networks. The principle of blending is that the closer the robot is to the object/target, the more closely the taught trajectory generalized in the object/target frame should be followed. In this research, two blending schemes to generate task paths were developed.

Scheme I. Adapted from Lieberman (2004), Scheme I generates the task path by blending the paths produced in the world and object frames, respectively. The object/target frames are updated based on their current locations and orientations, and the actual path from the $(k-1)^{th}$ object to the k^{th} object is computed as:

$$\widetilde{X}_i = \left(1-\lambda_1(i)\right)X_i + \lambda_1(i)\ {}^{O}\breve{X}_i^k, \quad h_{k-1} < i \le h_k \tag{3-16}$$

where h_k is the time sample index at which the k^{th} object is manipulated (for example, the object is grasped or placed onto a target). ${}^{O}\breve{X}_i^k$ is the path point obtained by transforming ${}^{O}X_i^k$, the i^{th} robot end-effector pose generalized in the k^{th} object frame, from the current actual k^{th} object frame, $\mathbf{F}_O^k$ to the world frame $\mathbf{F}_W$. $\lambda_1(i)$ is a blending function that depends on the relationship between the path episode containing the path point X_i and the critical episodes and nature of the subtask that contains the path point X_i. For example, the robot must be at docking poses at the end of trajectory episodes right before critical episodes, i.e. $\lambda_1(i)$=1 at the end of the former episodes. As well, $\lambda_1(i)$=1 must always be true in the critical episodes. The robot can adapt to dynamic changes of the object or target locations using Equation 3-16. Although many forms of the blending functions exist (Volpe 1993), a six-order polynomial function used in Lloyd and Hayward (1991) was selected:

$$\lambda_1(i) = \begin{cases} 0, & h_{k-1} < i \le p_k \\ 6f_i^5 - 15f_i^4 + 10f_i^3, & p_k < i \le b_k - 1 \\ 1, & b_k \le i \le h_k \end{cases} \tag{3-17}$$

where the trajectory episode X_i, $i \in [b_k\ h_k]$, is the critical episode related to the k^{th} object that was engaged at X_{h_k} during the task demonstration. $f_i = (d_i - d_{p_k}) \big/ (d_{d_k-1} - d_{p_k})$, where d_i is the total distance traveled before the i^{th} sampling period in the task demonstration. The closer i is to $b_k - 1$ from p_k, the closer f_i is to one, and therefore the closer $\lambda_1(i)$ is to one.

Therefore, p_k is the time index of $\mathbf{\Gamma}_W$ after which the trajectory generated in the current k^{th} object frame will gradually have increased weight, before the k^{th} object is engaged.

Scheme II. This scheme generates the task path by blending the paths produced in the object-oriented frames and object frames, respectively. The object-oriented frames and object frames are updated online according to the current locations and orientations of the objects and the robot starting pose. The actual path from the $(k-1)^{th}$ object to the k^{th} object is obtained as:

$$\widetilde{X}_i = \left(1-\lambda_1(i)\right)\ {}^{G}\breve{X}_i^k + \lambda_1(i)\ {}^{O}\breve{X}_i^k, \quad h_{k-1} < i \le h_k \tag{3-18}$$

where ${}^{G}\breve{X}_i^k$ is the path point obtained by transforming ${}^{G}X_i^k$ (the i^{th} robot end-effector pose in the k^{th} object-oriented frame in the demonstration) from the current actual (practice or execution) k^{th} object-oriented frame $\mathbf{F}_G^k$ to the world frame $\mathbf{F}_W$, while ${}^{O}\breve{X}_i^k$ is the path point obtained by transforming ${}^{O}X_i^k$ from the current actual k^{th} object frame, $\mathbf{F}_O^k$, to $\mathbf{F}_W$. $\lambda_1(i)$ is calculated using Equation (3-17).

3.3.4.3 Consideration of Robot Current Position

During task practice or execution, the current robot pose, at the robot starting position or right after an object is grasped or released, could be distant to the closest path point generated by Equations (3-16) or (3-18). Therefore, it is reasonable to take the robot current position into account when generating the path point so that the robot can smoothly merge into the path. A modification to Equations (3-16) and (3-18) are respectively given as:

$$\begin{aligned}
\widetilde{X}_i &= \left(1-\lambda_1(i)\right)\left(\lambda_2(i)Y(t) + \left(1-\lambda_2(i)\right)X_i\right) + \lambda_1(i)\ {}^{O}\breve{X}_i^k, \quad i \le h_1, k=1 \\
\widetilde{X}_i &= \left(1-\lambda_1(i)\right)\left(\lambda_2(i)\ {}^{O}\breve{X}_i^{k-1} + \left(1-\lambda_2(i)\right)X_i\right) + \lambda_1(i)\ {}^{O}\breve{X}_i^k, \quad i > h_1, k>1
\end{aligned} \tag{3-19}$$

$$\begin{aligned}
\widetilde{X}_i &= \left(1-\lambda_1(i)\right)\left(\lambda_2(i)Y(t) + \left(1-\lambda_2(i)\right)\ {}^{G}\breve{X}_i^k\right) + \lambda_1(i)\ {}^{O}\breve{X}_i^k, \quad i \le h_1, k=1 \\
\widetilde{X}_i &= \left(1-\lambda_1(i)\right)\left(\lambda_2(i)\ {}^{O}\breve{X}_i^{k-1} + \left(1-\lambda_2(i)\right)\ {}^{G}\breve{X}_i^k\right) + \lambda_1(i)\ {}^{O}\breve{X}_i^k, \quad i > h_1, k>1
\end{aligned} \tag{3-20}$$

where $Y(t)$ is the current location of the robot, and h_1 the time sample index at which the first object was grasped or manipulated in the task demonstration. Equation (3-19) deals with the blending of the trajectory produced in the world frame with the trajectories generated in the object-frames while Equation (3-20) corresponds to the blending scheme between the trajectories produced in the object-oriented frames and the object-frames. After an object is grasped ($i > h_1$), it is desirable to follow a departure path used in the demonstration. Therefore, $Y(t)$ is replaced by ${}^{O}\breve{X}_i^{k-1}$ in Equation (3-19) and Equation (3-20) when the path to the $k^{th}(k>1)$ object is computed. $\lambda_2(i)$ is a blending function between X_i and $Y(t)$ when the robot approaches the first object (k=1), or between X_i and ${}^{O}\breve{X}_i^{k-1}$ when the robot advances to any k^{th} object afterward (where $k > 1$). $\lambda_2(i)$ has a similar form to (3-17):

$$\lambda_2(i) = \begin{cases} 1 - \left(6 f_i^5 - 15 f_i^4 + 10 f_i^3\right), & h_{k-1} < i \le q_k \\ 0, & q_k < i \le h_k \end{cases} \tag{3-21}$$

where $f_i = (d_i - d_{h_{k-1}})/(d_{q_k} - d_{h_{k-1}})$, where d_i is the total distance traveled before the i^{th} sampling period in the task demonstration. The closer i is to q_k from h_{k-1}, the closer f_i is to one, and therefore the closer $\lambda_2(i)$ is to zero. Therefore, from h_{k-1} to q_k, the trajectory generalized in the $(k-1)^{th}$ object frame if $k > 1$, (i.e ${}^{O}\breve{X}_i^{k-1}$) , or the robot current location if $k = 1$ (i.e. $Y(t)$), will increasingly be weighted less, while X_i in Equation (3-19) or ${}^{G}\breve{X}_i^{k}$ in Equation (3-20) will increasingly be weighted more. $\lambda_2(i)$=1 at $i = 0, h_1, h_2, h_3, \cdots$, and it rapidly decreases to 0 afterward. The influence of the current robot location $Y(t)$ immediately after the start of the task and of ${}^{O}\breve{X}_i^{k-1}$ immediately after the $(k-1)^{th}$ object is grasped is thus limited to a short period. The effect of $\lambda_2(i)$ is to let the robot gradually merge from the robot starting pose, into the task path directly generated from the trajectory blending based on Equation (3-16) or Equation (3-18). $\lambda_2(i)$ also ensures that the departure routes are similar to those demonstrated after relevant objects are grasped or placed.

3.3.4.4 Speed Profile of Generated Task Path

It may be important for the robot to keep a speed profile similar to that demonstrated during its task practice and execution. However, this has not been previously addressed in the literature. When the task setup is considerably different from that in the demonstration, as expected in a service-robot working environment, moving directly from $\tilde{X}_i$ to $\tilde{X}_{i+1}$ in one sampling period cannot ensure the similarity of the demonstrated motion profile. Interpolation between $\tilde{X}_i$ and $\tilde{X}_{i+1}$ or skipping one or more points becomes necessary in order to preserve the demonstration speed profile at X_i .

In the task practice or execution, it is likely that the robot will not be able to follow both the demonstrated linear and angular speeds at some points. The robot should have flexibility to deal with this. In this research, a progress-driven path-following means is adopted instead of a time-driven method to determine the actual trajectory speeds and the progress of the task path points computed using Equation (3-19) or (3-20). A non-decreasing progress factor, $s(t)$, is used to denote the current index of $\mathbf{\Gamma}_W$ (trajectory generalized in the world frame) in task execution. The progress-driven path-following method is adapted from Aarno at el. (2005), and has a similar function as the event-driven path planning used in Xi at el. (1996).

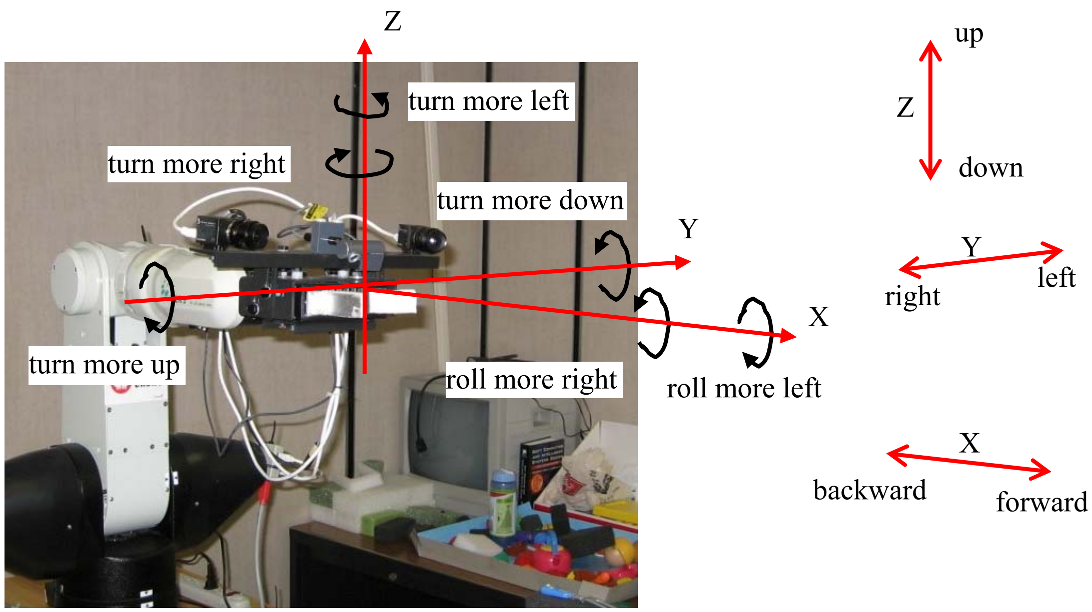
The following method is used to update the progress factor and location of the robot for its current sampling period. Denote v_i and ω_i, $i = \mathrm{s}(t)$, as the magnitudes of the taught linear and angular velocities, respectively, in the i^{th} sampling period. A search for the first planned path point (by using Equation (3-19) or (3-20)), $\tilde{X}_j$ ($j \ge i$), among $\tilde{X}_i$ and its following planned path points, is carried out so that the needed linear and angular velocities for the robot to move directly from the current robot location $Y(t)$ to $\widetilde{X}_j$ is equal or greater than v_i or ω_i. Here, $\widetilde{X}_j$ might skip over a few planned path points, i.e. $j > i$, or still be $\widetilde{X}_i$. The currently pursued point $\hat{Y}(t+1)$ (for the robot to move to) is obtained by interpolating from $Y(t)$ toward $\widetilde{X}_j$. $\hat{Y}(t+1)$ is determined under the conditions that both the linear and angular speed needed for the robot to move from $Y(t)$ to $\hat{Y}(t+1)$ in one sampling period are not greater than v_i and ω_i, respectively. Thus, it may take several sampling periods for the robot to reach $\widetilde{X}_j$. In this circumstance, the robot might relax the constraints of v_k and ω_k by gradually increasing the robot speed, if the nature of the current subtask permits. Meanwhile, the progress factor is updated as $\mathrm{s}(t+1) = j$. In addition, $\hat{Y}(t+1)$ may have to be further modified in order to avoid obstacles and collision.

3.3.5 Practice of Learned Task and Refinement of Task Knowledge Based on Timely Feedback

Robot practice of the learned task is performed to verify that the robot has learned the task as expected (by the teacher), to test the ability of the robot to adapt the learned task to new task setups. The practice also gives the robot a chance to refine its just learned task knowledge according to the teacher's timely feedback regarding its task-practice performance.

Once the needed task path for a given task setup is generated, the robot is ready to practice its just learned task. The robot is permitted to practice the learned task at two different speed levels. The first time, it practices the task at a speed scaled down from the taught speed. Before performing each primitive subtask, the robot tells the teacher the subtask name. The teacher can request the robot to practice the task multiple times with the same scaled down speed. Once the teacher has satisfied the task performance of the robot with the current speed, the robot practices the task again at normal speed for different task setups, and it does not need to disclose the current subtask name. The teacher can give feedback at any time. The teacher can influence the task trajectory: three translations and three rotations on the task path, and the trajectory speed. Since the teacher and the robot end-effector can potentially be at any location and orientation, the feedback on the task trajectory would be better to be interpreted in the robot tool (or end-effector) frame. For example, the teacher can give timely vocal feedback, such as "move faster/slower" for speeds, "move more forward/backward/left/right/up/down" for translation offsets, "turn more left/right/down/up" for orientation offsets about *Z* and *Y* axes and "roll more right/left" for orientation offsets about the *X* axis (Figure 3-7). The teacher can also offer timely feedback on the task segmentation by using utterances such as "the next/last task episode should belong to *primitive subtask name*" to modify the primitive subtask-wise task partition. The robot can then refine its knowledge accordingly. The practice-feedback-refine cycle has to be repeated until both the teacher and the robot reach an agreement that either the process of task teaching and learning has been accomplished or they must go back to some previous teaching or learning stages, modify related knowledge, and practice the task again.

In the task practice, the focus is on the refinement or small adjustment of the robot's just learned task knowledge, rather than the significant modification to the learned task such as in Calinon and Billard (2007). If considerable changes are needed to improve the robot's task performance, the teacher and robot may have to go back to some previous teaching and learning stages, and re-teach and re-learn the task according to the nature of the performance deficit.



Robot tool frame and rotations

Translations in robot tool frame

Figure 3-7: Definitions of the robot tool (or end-effector) frame, and directions of the translations and rotations used for the teacher's vocal feedback. Vocal instructions include: "move more forward/ backward/left/right/up/down" for translations; and "roll more right/left" for rotation about the X axis and "turn more down/up/left/right" for rotations about Y and Z axes in the robot tool frame.

Chapter 4
Agent-Based Implementation of Robot-Task Teaching and Learning

4.1 Goal-Oriented Hierarchical Task Representations

A goal-oriented hierarchical task representation is described in this section while an agent-based framework for the implementation of the robot-task teaching and learning method presented in Chapter 3 is described in Section 4.2.

Humans often represent, organize, and plan a task in a top-down manner, from high-level subtasks to low-level subtasks, and from the coarse to fine (Kieras, 1994). In contrast, a robot or machine builds the task structure from its sensor data usually in a bottom-up manner, from detail to abstract. Hierarchical task representations are often used to represent the learned tasks (Thomaz and Breazeal, 2008; Zoellner et al., 2005; and Breazeal et al., 2004a). In this research, a goal-oriented hierarchical task representation, a variation of the Goals, Operators, Methods, and Selection Rules (GMOS) model (Kieras, 1994), was adopted, as illustrated in Figure 4-1. Every task and subtask is associated with a specific goal or until-condition, a pre-condition, relevant salient state changes, and primitive actions or skills.

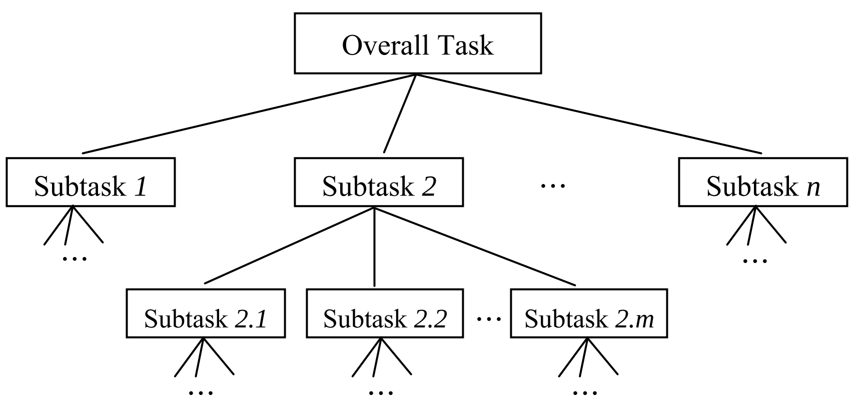


Figure 4-1: Hierarchical representation of a task. The overall task (root node) consists of *n* subtasks (sub-nodes). Each sub-node has its own child sub-nodes unless the sub-node is a leaf node or a primitive subtask. For example, Subtask 2 itself is composed of *m* child subtasks.

A sketch of a subtask, named SUBTASK, is illustrated in Figure 4-2, as it used in the object-oriented software program developed in this research. The main attributes of a subtask include its subtask name, pre-condition, goal (until-condition), start and end indices of the part of the task trajectory assigned to the subtask in the complete task trajectory, the sequence of state changes occurring in the subtask, a pointer to its parent subtask, and a list of its child subtasks and their relationships if the subtask is not a primitive subtask. SUBTASK is designed to accommodate the goal-oriented hierarchical task structure and to provide easy access to the parent subtasks and child subtasks. The definition of a task, named TASK, described in Figure 4-3, consists of a task ID that is unique for each task in the database, its task trajectories generalized in the world reference frame, related object frames and objectedoriented frames, the task's involved objects and primitive skills, and the attributes defined for SUBTASK since TASK is derived from SUBTASK. Two important member functions are also defined for TASK: *InstantiateTask* (…) and *SaveTaskToDatabase*(…) . The former function instantiates a TASK from the task knowledge base (i.e. database) based on a given task name or task ID, while the latter function saves the task knowledge to the database. The involved objects and primitive skills in each subtask can be easily retrieved based on the start and end indices of the subtask.

The task trajectories generalized in object frames and object-oriented frames have their own effective time periods. For example, as shown in Equations (3-19) and (3-20), the trajectory generated in the k^{th} object frame (${}^{O}\breve{X}_{i}^{k}$) only influences the parts of task trajectory from the $(k-1)^{th}$ object to the k^{th} object ($\lambda_1(i)\ {}^{O}\breve{X}_{i}^{k}$) and then to the $(k+1)^{th}$ object in the task practice and execution ($\lambda_2(i)\ {}^{O}\breve{X}_{i}^{k}$ corresponding to $\lambda_1(i)\ {}^{O}\breve{X}_{i}^{k+1}$). Primitive skills are detail-oriented and play import roles in accomplishing tasks, such as the skills for the robot to open and close its hands. Another example of primitive skills involved in accomplishing relevant tasks is that when a robot pours liquid from a bottle to a cup, the pouring skill is used to manipulate the bottle to maintain the bottle and cup in a desirable relative relationship. The pouring skill is also used to determine when to terminate the pouring. In this research, needed primitive skills are assumed to have been pre-built into the robot system. Approaches to recognize primitive skills from robot sensor data can be found in Bentivegna and Atkeson (2003) and Pardowitz et al. (2007). The involved primitive skills in a task will be executed during the task execution when the task progress factor $s(t)$, described in Section 3.3.4.4, reaches certain values learned in the task teaching. In other words, some conditions must be satisfied before a corresponding primitive skill will be performed. For example, the robot needs to reach the grasping pose of an object before the close-hand skill is executed for the robot to grasp the object.

```
class SUBTASK
{
    /*** attributes ***/
    // subtask name, pre-conditions and until conditions
    subtask-id        := ID of this subtask, unique in the task; 0 for root node
    subtask-name      := name of this subtask
    pre-conditions    := prerequisite states for this subtask to be executed
    goal              := state or until-condition that the subtask has to achieve
    start-index       := the index of complete task trajectory from which the subtask begins
    end-index         := the index of complete task trajectory the subtask ends
    // hierarchical task representation structure
    parent-task       := a pointer to the closest high level subtask this subtask directly belongs to,
                             if this is the overall task (i.e. root node), then := NULL
    number-of-child-subtasks      := number of its child subtasks that the current subtask has
```

```
    list-of-child-Subtasks          := set of the subtasks this subtask consists of. If this is a
                                       primitive subtask, then := NULL
    child-subtask-relationships     := relationships among these child subtasks, i.e. sequential,
                                       parallel, and/or optional.
    // states that change in the task/subtask
    sequence-of-state-changes       :=  sequence of state changes that happen in this subtask

    /*** member functions ***/
    GetPrimitiveSubtasks (…)        // get the primitive subtasks of this subtask
    Clone (…)                       // make a copy of this subtask
};
```

Figure 4-2: A brief definition of a subtask, named SUBTASK, as it is used in the object-oriented software program. SUBTASK is designed to accommodate the tree-shape hierarchical task structures: easy to access the parent subtasks and child subtasks. *start-index* and *end-index* represent the start and end indices of the part of task trajectory assigned to this subtask in the complete task trajectory, respectively.

```
class TASK,  derived from class SUBTASK
{
    /*** attributes ***/
    //task ID and attributes of the Subtask, class Subtask is a subset of class Task
    task-id                   := ID of the Task, unique id for each task in the database
    attributes-of-subtask     := refer to the definition of "class SUBTASK", implemented via
                                 class inheritance
    //task trajectories generalized in world frame, object frames, and object-oriented
    frames
    task-trajectory           :=  task trajectory generalized in world frame
    trajs-in-obj-frames       := a list of the task trajectories respectively generalized in object
                                 frames
    trajs-in-obj-oriented-frames      := a list of the task trajectories respectively generalized in
                                         object-oriented frames
    //involved objects, robot hand actions, and primitive skills in the task
    involved-objects                  := a list of objects, including targets, involved in this task
    involved-primitive-skills         := a list of primitive skills involved in this task

    /*** member functions ***/
    Clone (…)                     // make a copy of this task
    InstantiateTask (…)           // instantiate this task from knowledge saved in database
    SaveTaskToDatabase(…)         //save the task knowledge into database
    AssignSubtaskIDs(…)           //assign IDs to subtasks, a unique # to each subtask in the task
}
```

Figure 4-3: A brief definition of a task, named TASK, based on class SUBTASK as it is used in the object-oriented software program. TASK not only includes the attributes defined for SUBTASK, but it also consists of task trajectories generalized in the world frame, object frames and object-oriented frames; as well as involved objects and primitive skills. In addition, the member functions for instantiating a task given by the task name or task ID based on the task knowledge saved in the database, and for saving the task knowledge into the database are also defined.

4.2 Agent-Based System Architecture

An agent-based architecture was developed to implement the proposed robot teaching and learning method. As illustrated in Figure 4-4, the robot teaching and learning system consists of six different agents: Robot Agent (RA), Teleoperation Agent (TOA), Speech Agent (SA), Vision Agent (VA), Task-Learning Agent (TLA), and Task-Performing Agent (TPA). It also includes a database (DB) that is used to save and manage learned task knowledge and the language vocabulary used in the human-robot vocal interaction. The communication between the agents is through TCP/IP on an Intranet or a local area network (LAN). All agents were entirely newly developed as part of this research except for stereo-camera calibration in the VA, and the tracking-hand-motion functional unit in the TOA.

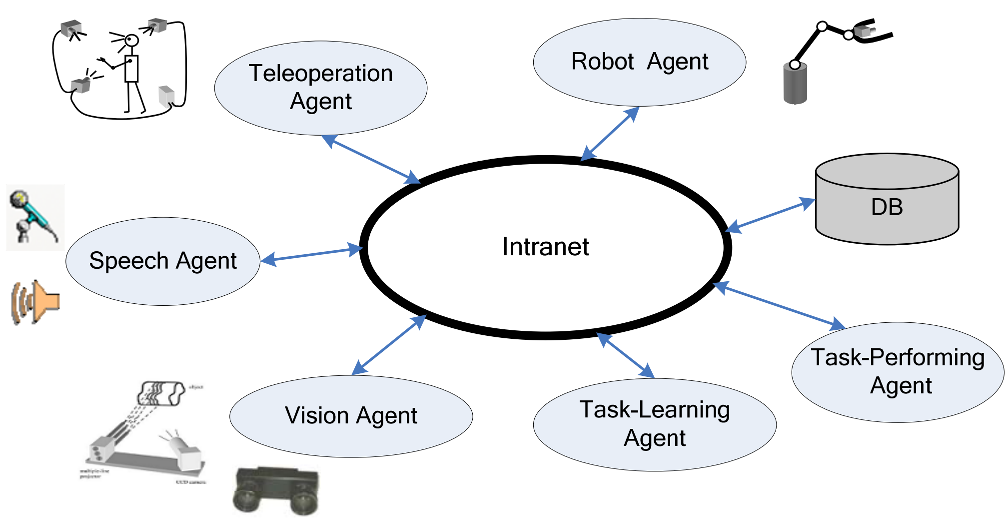


Figure 4-4: Multi-agent-based architecture of the robot task teaching and learning system.

4.2.1 Robot Agent

The Robot Agent (RA) directly controls and interacts with the physical robot, which is a six-axis robot manipulator (Thermo CRS model A465), in this research. As shown in Figure 4-5, the agent consists of five major functional units: TCP/IP Communications Unit, Robot Status Query, Robot Forward/Inverse Kinematics, Robot Motion Control Unit, and Robot Safety Unit.

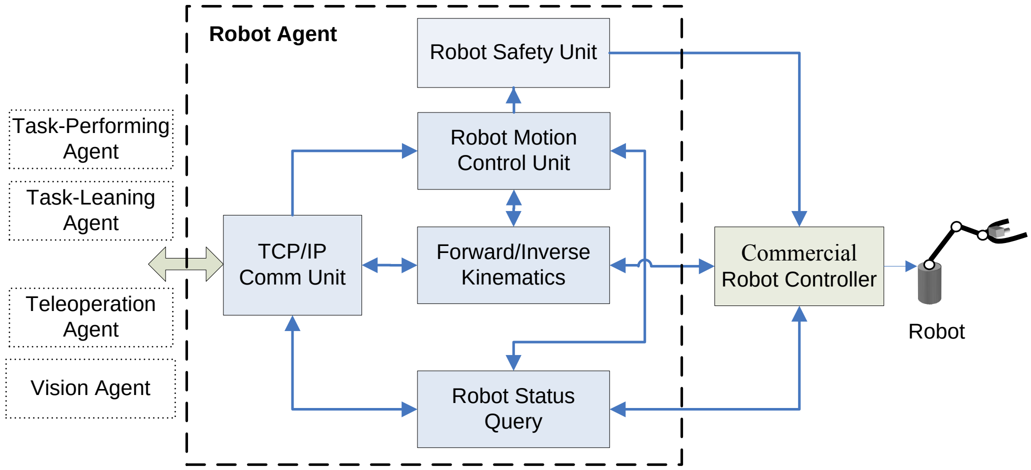


Figure 4-5: Block diagram of the Robot Agent, which consists of five main functional modules: TCP/IP Communications Unit, Robot Status Query, Robot Forward/Inverse Kinematics, Robot Motion Control Unit, and Robot Safety Unit.

The TCP/IP Communications Unit listens to requests from other agents, such as requests regarding the current robot status and the poses that the robot has to move to, dispatches the requests to other relevant functional units, and sends computed responses to the corresponding request-sending agents. The agents with which the RA mainly interacts include the TPA, TLA, TOA, and VA.

The Robot Status Query Unit deals with requests from both external agents and other functional units of the RA, such as requests for the current robot pose, robot joint angles, and robot hand status.

The Robot Forward/Inverse Kinematics Unit is basically used by the Robot Motion Control Unit and TPA to check the robot's reachability to a given robot end-effector pose. If the given pose is unreachable for the robot, the Motion Control Unit will skip this pose and directly proceed to the next requested pose, while the TPA will modify the section of the planed task path around this pose so that a reachable task path will be generated. Note that in-built software functions in the Thermo CRS software package are called to compute the forward and inverse kinematics in this research.

Through the Robot Motion Control Unit, the RA only takes care of the motion control instructions from the agent that has obtained an exclusive right to control the robot. An agent has to obtain the exclusive right from the RA before it is allowed to control the robot, and must return the right to the RA if it has finished its robot-control work. This unit makes sure the robot end-effector poses, which are sent to the commercial robot controller, much be reachable for the robot. Sometimes, interpolation is necessary in order to keep the robot speeds and acceleration within predetermined limits.

Keeping the robot safe, such as avoiding collisions, and movements beyond preset limits, is the responsibility of the Robot Safety Unit.

4.2.2 Teleoperation Agent

The Teleoperation Agent (TOA) is responsible for tracking the hand motion of the teacher, mapping the hand motion from the hand-motion-tracking stereo-camera frame to the robot frame, and sending the motion information expressed in the robot frame to the robot. These functions carried out by the TOA permit the robot to imitate the teacher's motion. Three modules (Figure 4-6), Tracking Hand Motion, Mapping Hand Motion to Robot Frame, and TCP/IP Communications Unit, are implemented to achieve these three main responsibilities, respectively. The hand tracking module and the motion mapping module currently run on two different computers, and communicate to each other by TCP/IP. This agent mainly cooperates with the Robot Agent, Task-Learning Agent, and Speech Agent. The TOA is activated and deactivated by the user uttering, "teleoperation system follow me" and "teleoperation system stop following me", respectively, with the aid of the Speech Agent.

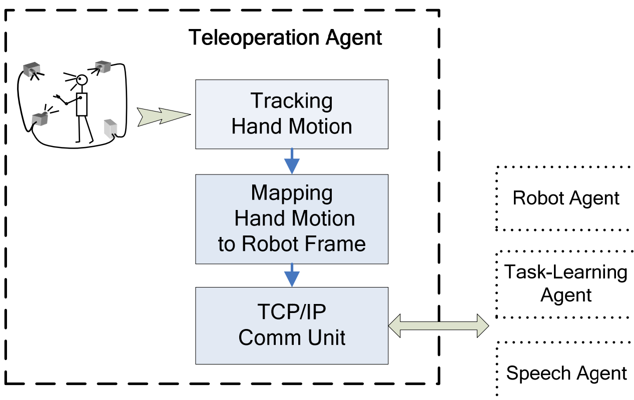
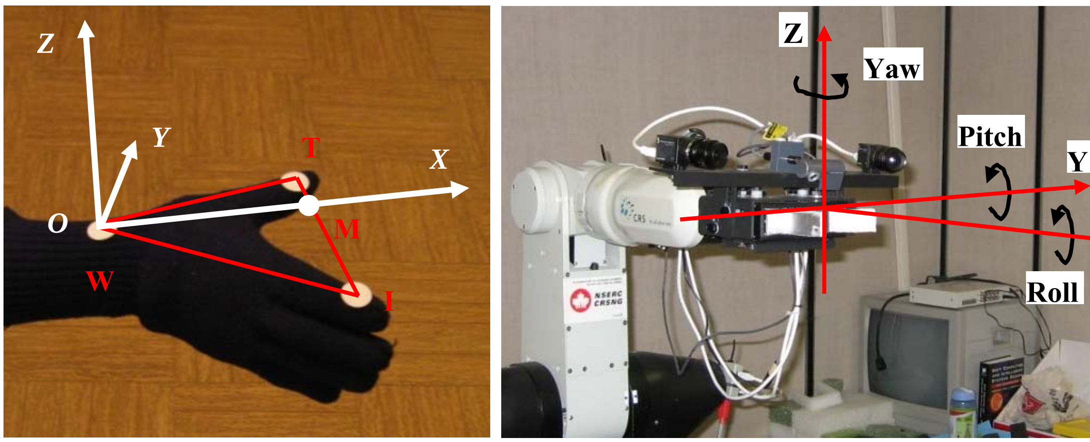


Figure 4-6: Block diagram of the Teleoperation Agent, which consists of three main functional modules: tracking teacher's hand motion, mapping hand motion from stereo-camera frame to the robot frame, and the TCP/IP Communications Unit that sends the mapped motion information to the Robot Agent for the robot to imitate the motion. The agent also interacts with the Task-Learning Agent and Speech Agent during the process of task teaching and learning.

When teleoperating the robot, the teacher or human operator wears a black glove on which three circular markers are placed at the wrist, thumb and index finger, respectively, as shown in Figure 4-7a. The markers are tracked by a stereo-vision system with three cameras respectively mounted on three walls. A teacher-hand frame, illustrated in Figure 4-7a, is defined by: the $\boldsymbol{X}$ axis from the wrist marker W to the point M located at one fifth the distance from the thumb marker T to the index-finger marker I; the $\boldsymbol{Z}$ axis, concurrent with the centroid of the wrist marker W, and normal to the plane defined by the three markers T, I, W; and the $\boldsymbol{Y}$ axis, the last orthogonal axis through W, perpendicular to Z and X. The X, Y, Z axes of the robot end-effector (or tool) frame are shown in Figure 4-7b, with the roll, pitch, yaw rotation directions. Let A_0^H and A_0^R denote the homogenous matrices to transform the teacher-hand frame and the robot end-effector frame, respectively, to the hand-tracking stereo-camera frame and the robot frame at the beginning of the teleoperation, and denote

A_t^H as the current teacher-hand frame and A_t^R as the current robot end-effector frame. A_t^R is computed as: $A_t^R = A_0^R\left((A_0^H)^{-1} A_t^H\right)$, i.e. the movement of the teacher-hand with respect to the hand starting pose is imitated by the robot with respect to the robot's starting pose.

Note that the tracking-hand-motion functional unit was developed outside this research.



a) Human-hand frame b) Robot end-effector (tool) frame

Figure 4-7: Definitions of (a) the human-operator hand reference frame, and (b) the robot end-effector (or tool) frame. For the hand, three white markers are placed on the wrist (*W*), index finger (*I*), and thumb (*T*) of the human hand, respectively. The *X* axis of human hand frame is from *W* to a point *M* on the line segment *IT*, where *MT/IT* = 0.2. The *Z* axis is the normal to the plane *WIT*, and the origin, *O*, is located at the centroid of *W*. Note that the stereo camera mounted on the robot end-effector is used by the Vision Agent (Section 4.2.4) to detect objects of interest in the robot workspace.

4.2.3 Speech Agent

The Speech Agent (SA) is responsible for both recognizing the teacher or user's natural speech (using Microsoft speech recognizer), and speaking out the robot's requests, responses and instructions to the teacher (using Microsoft text-to-speech (TTS)), in the daily human-robot engagement and the process of robot-task teaching and learning.

The Speech Agent plays an instrumental role in the robot task teaching and learning, and task practice and execution. The vocal human-robot interaction is the main communication between the teacher and robot, specifically with the Task-Learning Agent, Task-Performing Agent, and Teleoperation Agent. The SA recognizes the teacher's speech based on activated grammars. Each grammar in the context of speech recognition essentially defines a set of utterances to recognize. The agent then notifies the agents of interest its recognized results. The Speech Agent also speaks out requests and responses from other agents using synthetic voice. In addition, the agent changes the grammars based on the stage of the task teaching and learning. The expected utterances are changed based on the current task teaching and learning stage and updated based on the introduced task knowledge and vocabulary saved in the database.

In the user study experiments, which is discussed in the next chapter, vocal human-robot interaction scripts and a demonstration video of using the developed system to teach a task to the robot were provided to the human participants for training purposes. The vocal interaction scripts as well as human-robot dialogues excerpted from the video for the teaching stages "check and teach needed background" and "overall task introduction" are described in Appendix A.1 and A.2. Note that the implemented vocal human-robot interaction is much more flexible and richer than the provided scripts.

4.2.4 Vision Agent

The Vision Agent (VA) helps the robot explore its workspace, using a robot stereo-camera vision system as shown in Figure 4-8. The vision system includes a stereo-camera pair (circled), a multiple-line laser projector (circled middle), and a six-axis robot manipulator (Thermo CRS model A465). The stereo cameras and laser projector are mounted on the robot end-effector. The laser projector is currently not used in this research. The capability of the VA and measurement accuracy of the robot-stereo-camera measurement system are described in the following two subsections.

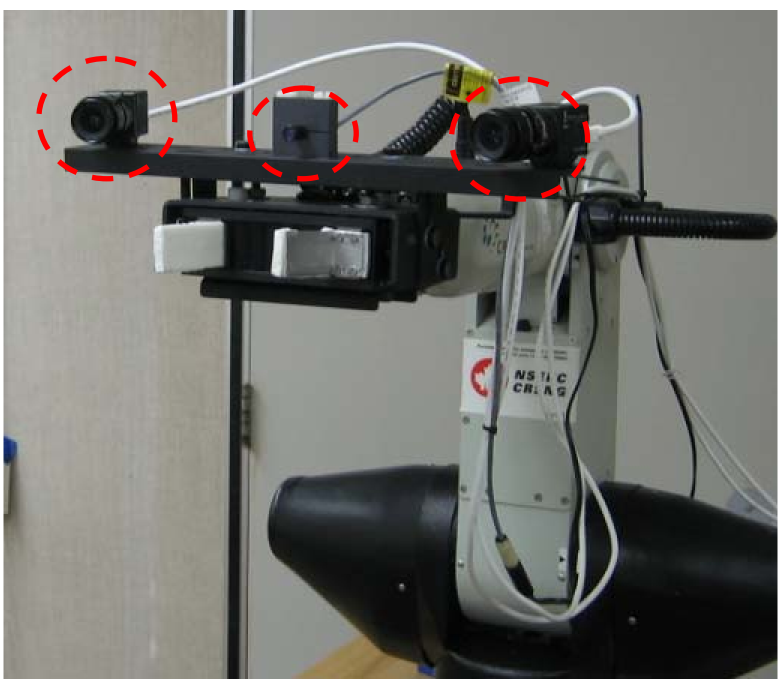


Figure 4-8: The robot stereo-camera vision system, including a stereo-camera pair (circled), a multiple-line laser projector (circled middle), and a six-axis robot manipulator (Thermo CRS model A465). The stereo-camera pair and laser projector are mounted on the robot end-effector. The laser projector is not used in this research.

4.2.4.1 Capability of the Vision Agent

The Vision Agent (VA) performs the functions of exploring the robot workspace by observation to detect objects in the space and determine their locations and orientations in the robot frame (i.e. the world frame in this research), and identify previously learned objects.

The agent mainly includes two function modules: exploration-route planner, and object detector. The exploration-route planner develops exploration routes for the robot to fully explore its workspace, and then moves the robot (end-effector) to numerous poses along the planned routes. The object detector unit automatically detects objects in the robot workspace, extracts and groups their surface and shape features that uniquely identify the objects. The detector unit then compares the features to those of objects that the robot has already learned, associated with object names, and saved in the database. The result of the exploration is available to the Task-Learning Agent and Task-Performing Agent upon request. There are two ways for the teacher to help the robot associate features of an object with its name via the Task-Learning Agent. One method is by explicitly moving the robot (via teleoperation) to indicate the object and then telling the robot the object name. This would be done at the stage of checking and teaching needed background knowledge. Alternatively, the teacher could implicitly teach the object name during the task demonstration. These associations of the groups of features to object names are then sent to the VA by the Task-Learning Agent. The VA subsequently saves the features with the associated object name as attributes of the object in the database.

The VA performs two types of space exploration: active exploration and passive exploration. In the active exploration mode, for example, upon request from the Task-Performing Agent before executing a task, the VA requests the robot (end-effector) to move along the planned exploration route. Once the robot reaches a requested pose along the route, the object detector unit is called to detect objects seen by the cameras. However, for the passive exploration, the Vision Agent does not influence the movement of the robot, and the object detector unit keeps detecting objects in the field of view of the cameras while the unit constantly asks the robot agent for the latest robot pose. With passive exploration, some objects of interest in the robot workspace may not be detected. In this research, only the active exploration mode was used, i.e. the VA does not perceive the environment when the robot is not under its control. Before the robot performs a task or starts each of the following teaching and learning stages: “check and teach needed background knowledge”, “teach task step by step”, and “demonstrate task in whole”, the Task-Performing Agent and the Task-Learning Agent have to obtain the latest information regarding locations and orientations of objects in the robot workspace through the Vision Agent.

The Value (V) plane is commonly used as one of the planes of the hue, saturation, value (HSV) colour model in colour imaging, and is determined as the maximum of the Red, Green, Blue (RGB) values). In the VA agent, V plane images taken by the end-effector mounted stereo cameras are binarized with thresholds automatically computed based on the histogram of the V-plane images. Contours of the objects are extracted and then analyzed. The following features are computed for each object:

(1) $extent = \dfrac{area \text{ of object contour}}{area \text{ of minimal bounding box of contour}}$.

(2) $roundness = \dfrac{perimeter \text{ of a circle whose area equals convex hull of contour}}{perimeter \text{ of object contour}}$.

(3) $solidity = \dfrac{area \text{ of object contour}}{area \text{ of convex hull of contour}}$.

(4) Mean RGB color of the object.

(5) The number of direct child contours that the object contour includes (for example a hole contour within an object perimeter contour).

(6) Hu moment invariants (Hu, 1962) of the object contour.

(7) Center of area of the contour.

(8) Principal vectors of the object contour through principal component analysis (PCA).

The first six features above and epipolar distances of the centers of area of the object contour are used to solve for the correspondences between the two images of a stereo pair while the first six features are utilized to match objects in consecutive pairs of images. Then, the detected objects are matched with objects previously learned and saved in the database also using the first six features, based on a weighted sum of the differences between the corresponding features. The centers of mass and principal directions of the object contours are employed to calculate the pose (locations and orientations) of each object in the robot frame. Differences between object poses computed from consecutive pairs of images are used as additional criteria to help match objects in consecutive pairs of images.

When a 3D object is observed by the cameras from different viewpoints, the extracted contour-based features of the 3D object may greatly vary. This occurred with objects such as a cup and bottle, used in the experiments of this research. Therefore, the extracted contour-based features are limited to identify objects of small depth (perpendicular to the camera) such as the knife, fork, spatula, and spoon, used in the experiments. Because of this limitation, for the bottle and cup, different patterns printed on a white background were attached to the objects, as shown in Figure 4-9. The bottle and cup were identified by a black rectangle and pattern of three black markers, respectively. The bottle location was calculated as the average location of the four rectangle corners. The bottle orientation was determined by the average of the four surface normals to the planes defined by each corner and its two adjacent corners. The cup location was calculated as the average of the locations of the three markers while its orientation was the normal to the plane defined by the three markers.

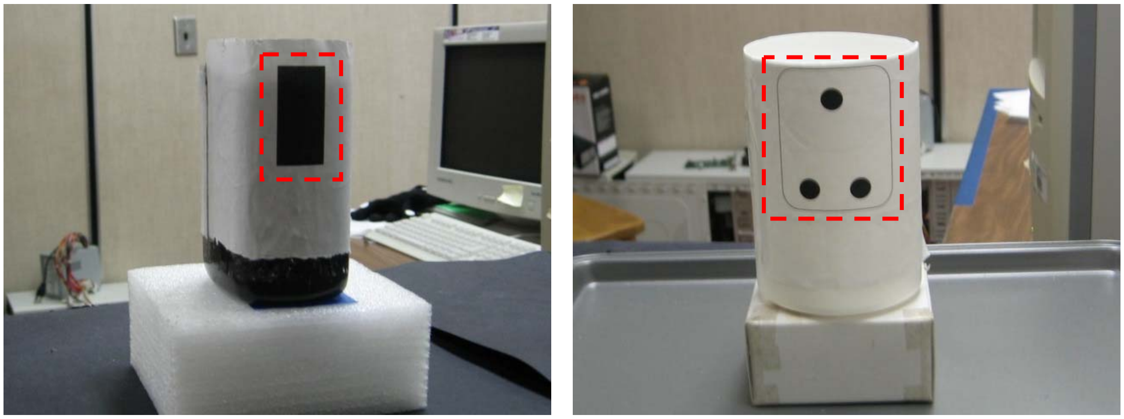


Figure 4-9: Patterns attached to the bottle (left) and cup (right). The black rectangle and the three black markers were used to identify the bottle and cup, respectively.

The focus of this research was not on development of the Vision Agent. The simplified feature extraction techniques of the VA were used in this research to enable experimentation on the teaching and learning aspects of the robot-task teaching and learning methods and system.

Note that the stereo cameras were calibrated separately outside this research, while the determination of the transformation matrix from the stereo-camera frame to the robot end-effector frame was included as part of this research.

4.2.4.2 Measurement Accuracy of the Robot Stereo-Camera Vision System

The accuracies of the stereo camera pair and the robot stereo-camera vision system were determined as detailed below.

a) Measurement Accuracy of the Stereo Cameras

The measurement accuracy of the stereo-camera pair mounted on the robot end-effector (Figure 4-8) was computed as the mean difference between the true distance (215.5 mm) and seventeen measured distances between two points on a flat plate. The measured distances were for seventeen different poses of the plate, respectively, in the working volume of the stereo cameras. The mean and standard deviation of the differences were 1.68 mm and 0.46 mm, respectively. This accuracy was acceptable to carry out the experiments in this research.

b) Measurement Accuracy of the Robot-Stereo-Camera Vision System

The measured location of a fixed point in the world frame (robot frame in this research) should ideally be constant regardless of the pose of the robot end-effector and stereo-camera pair. Based on this, the measurement accuracy of the robot-stereo-camera vision system was calculated. A black-white checkerboard was fixed in the robot frame while the end-effector was moved to ten different poses to cover the stereo-camera working volume. A selected region of the board (5×5 squares with 20 mm sides, and 36 grid corners) was used for the measurement as follows. The location of each grid corner was measured in the stereo-camera frame, and then transformed to the robot end-effector and robot frames, for each end-effector pose. For each of the 36 grid corners, the average of the ten measured locations as well as the distances from the average measurement to each of the ten measured locations were calculated. The mean and standard deviation over all 360 distances grouped together were finally computed as 2.83 mm and 1.17mm, respectively, and ideally should be zero. This accuracy of the robot stereo-camera vision system was considered acceptable to carry out the experiments in this research.

4.2.5 Task-Learning Agent

The Task-Learning Agent (TLA) is one of the core agents that is used to realize the new method for the robot to learn tasks from a user's natural teaching. The TLA actively collaborates with the Speech Agent (SA), Teleoperation Agent (TOA), Vision Agent (VA), Robot Agent (RA), Task-Performing Agent (TPA), and the database (DB) to accomplish the main research goal. The TLA implements the following teaching and learning stages (Figure 4-10): 1) teacher checks and teaches background knowledge needed for the task to be taught, 2) teacher introduces the overall task to the robot, 3) teacher teaches the task step by step; 4) teacher demonstrates the task in whole, 5) TLA learns and generalizes the taught task, and 6) TLA requests the TPA to practice the learned task and TLA refines the task knowledge that has been altered by the TPA based on the timely feedback from the teacher during the task practice. The TLA also saves the learned task knowledge to the database. The TLA works closely with the SA to carry out the natural vocal human-robot interaction and dialogue in the process of task teaching and learning. Examples of human-robot conversations used in the first two teaching stages (check and teach background knowledge, and introduce task) are given in Appendix A.2.

To make the teaching and learning progress similar to human-style teaching, the teacher and robot are allowed to skip some teaching and learning stages to a specific stage, as indicated by the dotted arrows in Figure 4-10. For example, the teacher may want to skip a specific teaching stage, cut short the current teaching stage, go back to reintroduce the task, re-demonstrate some parts of the task in the step-by-step teaching stage, re-demonstrate the whole task, or request that the robot re-practice its learned task. For these purposes, the following vocal instructions are adopted:

go to the ***next/last/previous*** *teaching and learning stage*

go to the ***check background knowledge*** *stage*

go to the ***introduce overall task*** *stage*

go to the ***teach task step by step*** *stage*

go to the ***demonstrate task in whole*** *stage*

go to the ***practice task*** *stage*

To make the teacher's teaching schedule more flexible, a record of the current teaching and learning progress after each teaching and learning stage will be taken and saved so that the teacher can terminate the teaching and learning process at any time, and pick up the unfinished task teaching and learning process at a more convenient time. An example of a conversation used for the teacher to continue teaching a task that was left unfinished is:

Teacher: Hi robot.

Robot: Hi, what can I do for you?

Teacher: I would like to teach you a task.

Robot: That is great. Let's start.

The last task, TASK_NAME, has not been completely taught and learned yet. Do you want to continue this teaching?

Teacher: Yes, I do.

……

Right after the teacher's confirmation, the teaching and learning process will jump to the stage where the teacher terminated their previous teaching.

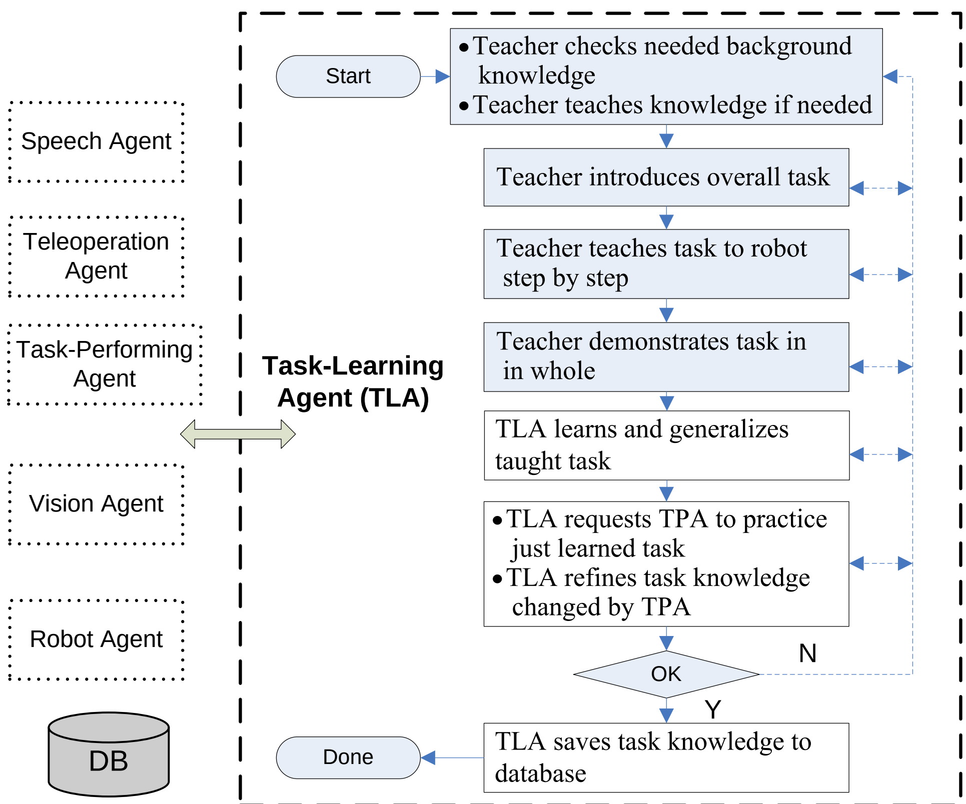


Figure 4-10: Block diagram of the Task-Learning Agent (TLA), which is composed of seven main functional modules. The TLA works together with the Speech Agent, Teleoperation Agent, Task-Performing Agent, Vision Agent, and Robot Agent, and the database. The dotted arrows indicate possible paths in the sequence of teaching and learning stages depending on the teacher's corresponding vocal instruction.

4.2.6 Task-Performing Agent

The Task-Performing Agent (TPA) is in charge of planning and executing specified tasks that the robot learned previously. The determination of the optimal subtask-execution order, based on the given task setup and learned subtask dependencies, is not currently included in the TPA. Instead, the TPA executes the subtasks of the task in the same order as demonstrated. The agent takes measures to: re-plan unreachable parts of the task path generated using Equation (3-19) or (3-20); prevent the robot from collision with the experiment platform; and prevent potential collision between the stereo cameras and the robot. Currently, a task can be successfully performed as long as the critical task trajectory episodes generated using Equation (3-19) or (3-20) can be followed by the robot. The normal procedure to perform a task by the TPA is described as follows.

The TPA obtains the task knowledge either from the Task-Learning Agent (TLA) upon the task-practice request from the TLA or from the database upon the user's vocal request to perform a task. The TLA can also specify the scale of the taught task speeds to be applied in the task practice. The TPA will ask the teacher or user to first place the involved objects and targets in place in the robot workspace and to then say "it is ready" to the robot. Then, the TPA will request the Vision Agent (VA) to explore the robot workspace.

After the VA's exploration, the TPA checks if all objects and targets involved in this task have been detected. If so, the TPA instantiates the task according to the current task setup (i.e. the current locations and orientations of the objects, targets, and robot starting point), and computes the adapted task path using Equation (3-19) or (3-20). Then, each computed task-path point is examined if it is reachable by the robot. If only some parts of the non-critical task trajectory episodes are unreachable, these parts will be re-planned so that they become reachable. If some objects involved in the task are not detected or some parts of the critical task trajectory episodes are unreachable, the TPA will tell the teacher which object is unreachable or undetected, ask the teacher to rearrange the involved objects, and request the VA to explore the robot workspace again. In addition, the generated task path is also checked for potential collisions of the robot with the experiment platform or the stereo cameras mounted on the robot end-effector. Some modification to the task path is necessary if the potential collisions were detected.

If the TPA is requested to execute the task with scaled down speeds, the agent will remind the teacher or user (via the Speech Agent) of the vocal instructions available to them to give feedback on the current task segmentation and trajectory at the beginning of the task execution. The TPA will also state the name of each primitive subtask at the beginning of the primitive subtask execution.

At the beginning of each time sampling period, the TPA determines the next robot pose based on the current robot pose, progress factor $s(t)$, corresponding taught task speeds, and the planned task path, using the approach described in Section 3.3.4.4. Then, the agent sends the computed pose to the Robot Agent.

The TPA actively listens to the teacher or user for their feedback while it is executing the given task. Examples of such vocal feedback include: "move faster/slower" for speeds, “move more left/right/up/down/forward/backward” for translations, “turn more left/right/down/up” for orientation offsets about the Z and Y axes of the robot end-effector frame and “roll more right/left” for orientation offsets about the X axis as defined in Figure 3-7, “the last/next path segment should belong to *primitive-subtask-name*” for the segmentation of the task, and “pause here” and “segmentation move forward/backward” to change the direction of the motion. The feedback on the translations and orientations is interpreted in the robot end-effector frame. The agent immediately responds to the feedback by modifying the task knowledge accordingly, including changes on the task segmentation, task path, and trajectory motions. The agent reflects the relevant refinement in the next sampling period.

Chapter 5
Experiments

5.1 Introduction

Experiments were conducted to evaluate how well the developed robot-task teaching and learning system would enable: general users to teach different tasks to a six-axis robot manipulator, the robot to learn the taught tasks, and the robot to adapt its learned tasks to new task setups. Two series of experiments were carried out: 1) Robot-task learning and adaptation (RTLA) experiments, performed by the student investigator, to verify the learning process of the system for the robot to learn tasks from human teaching and test the ability of the robot to adapt its learned tasks to new task setups; and 2) a user study, to determine how well the system enables general users to teach tasks to the robot and how well the robot learns the taught tasks from them. The experimental setups, procedures for the RTLA experiments, user study protocol, and data collection for the two series of experiments are described in the following sections.

5.2 Experimental Setups

The setup of the human-robot system (i.e. the robot-task teaching environment and configuration of related software and hardware), the taught tasks, experimental parameters, and task setups in the experiments are briefly discussed in the following subsections.

5.2.1 Human-Robot System Setup

For the RTLA and user study experiments, the same robot-task teaching environment and configuration of related software and hardware were adopted. The experimental environment is shown in Figure 5-1. For the robot teleoperation used for task demonstration, the teacher teleoperated the robot from one room to manipulate objects in its environment in an adjacent room. The involved robot was a 6-DOF manipulator, Thermo CRS model A465. The teacher utilized direct viewing to sense the robot and its robot workspace. There was no other visual feedback provided to the teacher.

A vision-based human-robot teleoperation system was employed for hand-to-hand task teaching. As shown in Figure 5-1, the teleoperation system used three cameras mounted on the walls to track the hand motion of the teacher. The teacher wore a black glove and three markers were placed on the wrist, thumb and index finger, respectively, as shown in Figure 4-7 and Figure 5-1. More detail about the teleoperation system is described in Section 4.2.2 on the Teleoperation Agent. A computer (Pentium D930 3.0 GHz Dual-Core CPU and 1.0 GB RAM) (Computer 1) ran the hand-motion tracking part of the Teleoperation Agent.

A stereo-camera vision system with two color cameras mounted on the robot end-effector (Figure 4-8), was used by the Vision Agent to detect objects of interest, measure their locations and orientations in the robot frame, and identify the objects (i.e. determine the names of the objects) by comparing their extracted features to those of previously learned objects. The Vision Agent ran on a Pentium 4 1.8 GHz CPU computer (Computer 2) with 1.5GB RAM.

The Speech Agent included a speech recognition engine (Microsoft English (U.S.) v6.1 Recognizer) and a text-to-speech engine (Microsoft Speech SDK 5.1) used for the vocal human-robot interaction. The Speech Agent and all other agents except the Vision Agent and hand-motion tracking part of the Teleoperation Agent ran on a Pentium 4 1.8GHz CPU computer (Computer 3) with 896 MB RAM. A hands-free headset (Motorola MotoStart HS850 Bluetooth Handsfree Headset) was used by the human teacher to communicate with the robot. It included a wireless microphone and speaker, and it was connected to the computer via a Motorola Bluetooth USB PC Adapter PC850.

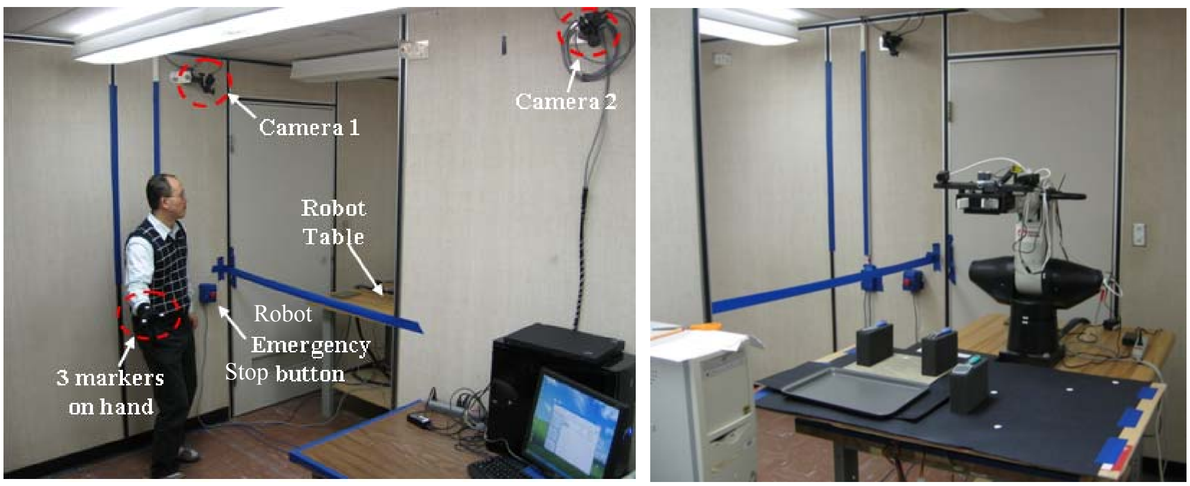


Figure 5-1: Robot teleoperation environment for task demonstration showing operator hand-tracking site (left) and robot site (right). Three white circular markers are placed on the teacher/operator's wrist, thumb and index fingers, respectively. The markers are tracked by three wall-mounted cameras. The third camera not shown, is mounted on the wall opposite Camera 2.The relative motion of the hand in the stereo-camera frame (defined in Figure 4-7) with respect to its initial pose is mapped to the relative movement of the robot end-effector (or tool) in the robot frame with respect to the initial end-effector pose.

5.2.2 Tasks Taught in Experiments

Two tasks were taught in the experiments and they were named: "Lay out Table" and "Pour Liquid". Note that experiment participants might call the tasks with different names, such as "Make Table" and "Pour Juice". For both tasks, to prevent risk of damage to the robot during experiments, a platform surface made of cardboard raised on blocks approximately 10 cm above the real robot table was used, as shown in the right figure in Figure 5-1. The platform was referred as table in the experiments.

The goal of the first task "Lay out Table" was that a knife and a fork be picked up from a table and placed onto a placemat in a conventional layout, as shown in Figure 5-2. Note that some human subjects in the user study chose a spoon and spatula instead of the knife and fork for this task, and all the objects (i.e. knife, fork, spoon and spatula) were mounted on foam blocks (approximately 11 cm in height) to facilitate grasping by the two finger gripper and for robot safety. The robot should adapt the learned task to new locations and orientations of the knife, fork, placemat, and the robot starting point.

The goal of the task “Pour Liquid” was that a bottle of beads simulating liquid should be picked up and moved close to a cup, some beads should be poured into the cup, and then the bottle should be placed onto the table (platform surface), as illustrated in Figure 5-3. To avoid losing tracking of the markers placed on the teacher’s hand, the teacher could only rotate their hand about the ***X*** axis of the hand frame by approximately 20 deg. The teacher therefore could not completely control the pouring process. Instead, the robot would start to rotate the bottle to pour after the teacher would: teleoperate the robot end-effector to be brought to a position ready to pour, and give the robot a vocal command “start to pour to right/left”. Upon receiving the instruction, the robot would slowly rotate its end-effector 95 deg about the ***X*** axis of the robot end-effector frame in the direction requested. The robot would then hold the pouring pose until the teacher would issue the instruction “stop pouring”. In the rotating process before the holding, the teacher has the ability to control the position of the bottle via teleoperation, but not its orientation. After the task would be learned, the robot should adapt the learned task to new locations and orientations of the bottle, cup, and the robot starting point.

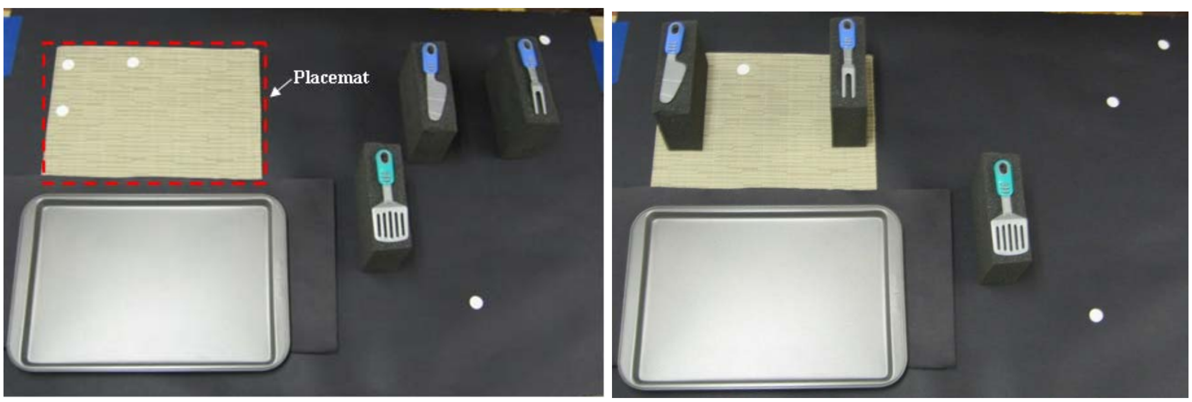


Figure 5-2: Sample setup of the task “Lay out Table” showing: initial locations and orientations of the knife and fork (left); and final placements on the placemat (right).

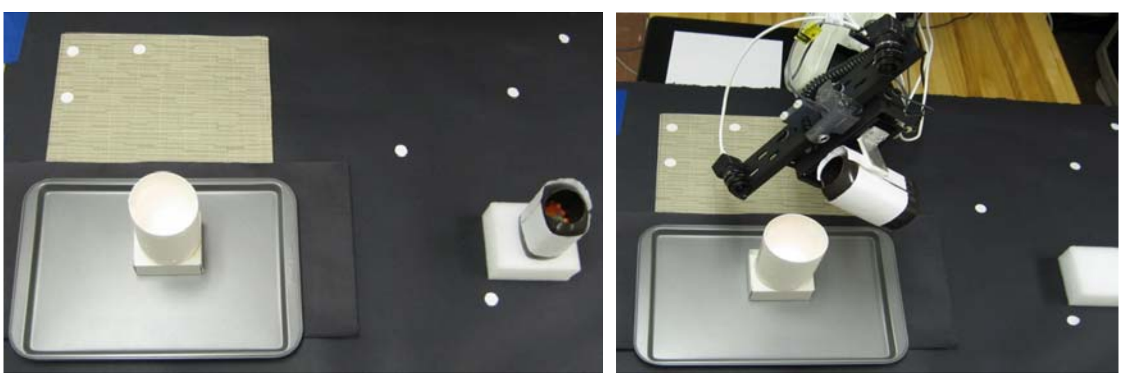


Figure 5-3: Sample setup of the task "Pour Liquid". The initial locations and orientations of the cup (left) and bottle (right) are shown in the left figure. An image of the pouring action is shown in the right figure. After the pouring action, the bottle is placed on the table (on the black platform).

5.2.3 Experimental Parameters

The parameters used in the robot-task teaching and learning for all experiments are described in this section. The same parameters were used for the RTLA and user study experiments.

For the task "Lay out Table", three signals, MSV of the taught (end-effector) trajectory, status of the robot hand or gripper, and vocal subtask-segmentation cues, were selected to segment the demonstrated task trajectory. The segmentation candidates (or points) of the task trajectory contributed from each of the three signals were respectively chosen as: the instants where the *MSV* was at a local minimum and less than a threshold based on Equation (3-2), instants where the status of the robot hand (or gripper) was changed, and instants where vocal subtask-segmentation cues were given by the teacher in the task demonstration. For the task "Pour Liquid", in addition to the above mentioned three signals, the instant that the vocal command to pour liquid, "start to pour to left/right", was issued by the teacher to the robot was also used to segment the demonstrated task trajectories.

The recorded trajectories were filtered with an averaging filter and the MSV of the robot end-effector trajectory was computed using Equation (3-1) with $\eta = 70^2$ mm^2 (based on 70 mm robot gripper length) to convert squared angular speed to squared linear speed. Segmentation candidates (or points) from the MSV were chosen using Equation (3-2) with $\delta_{MSV} = \mu_{MSV}$, where μ_{MSV} is the mean MSV. The parameters c_k used to determine if the task trajectories should be partitioned in the vicinity of the trajectory-segmentation candidates obtained by analyzing the k^{th} segmentation signals in Equation (3-3), were selected as: $c_1 = 1/(50\overline{V}^2)$, $c_2 = 1/(2\overline{V}^2)$, $c_3 = 1/(1800\overline{V}^2)$, $c_4 = 1/(200\overline{V}^2)$, where c_1, c_2, c_3, and c_4 were applied to the MSV, robot gripper status, vocal subtask-segmentation cues, and vocal command to pour liquid, respectively. $\overline{V}$ is the mean linear speed of the end-effector along the taught trajectory. For non-MSV signals (robot gripper status, vocal subtask-segmentation cue, and vocal command to pour liquid), their corresponding μ_i^k was set to 1.0 in Equation (3-3). For MSV segmentation candidates, μ_i^k was set to 1.0 for candidates adjacent to local maxima of *MSV*(*t*); 0.7 for candidates with MSV $\leq(\mu_{MSV} - \tilde{\sigma}/2)$; 0.3 for candidates with $(\mu_{MSV} - \tilde{\sigma}/2) <$ MSV $\leq (\mu_{MSV} - \tilde{\sigma}/4)$; and 0.1 for all other candidates (with MSV $< \delta_{MSV}$), where $\tilde{\sigma} = \mathrm{MIN}(0.8\mu_{MSV}, \sigma_{MSV})$, and σ_{MSV} is the MSV standard deviation. To compute the overall vote $v(t)$ using Equation (3-4), the following weights were used: $w_1 = 0.10$, $w_2 = 0.56$, $w_3 = 0.15$, and $w_4 = 0.19$, where w_1, w_2, w_3 and w_4 were applied to the MSV, robot gripper status, vocal subtask-segmentation cues, and vocal command to pour liquid, respectively. The threshold δ_T in Equation (3-5) to determine the overall trajectory segmentation candidates was set to 0.6 times the mean $v(t)$ over the entire trajectory. For the third condition in Equation (3-5), the minimum spatial length along the robot approach direction (+X direction of the robot tool frame) of the trajectory episode right before an object was grasped, δ_A, was set to 70mm (gripper length). The minimum spatial length along the normal to the target surface (Z direction of the world frame) of the trajectory episode right before an object was placed onto a target, δ_N, was set to 70/3 mm. δ_E in Equation (3-6), the threshold to remove false episodes, was 0.3 times the average of all probabilities of trajectory episodes being valid. The weights used in Equation (3-8) to calculate the optimal assignment of the segmented episodes to primitive subtasks were: κ_1=10000 (penalty for the case where at least one primitive subtask was not assigned any trajectory episodes or at least two neighboring episodes were either allocated to two non-adjacent primitive subtasks or cross-assigned to two neighboring primitive subtasks), κ_2= 100 (penalty for the case where the robot hand actions that happened in trajectory episodes that were currently assigned to a primitive subtask did not include all the robot hand actions that were specified by the teacher to this primitive subtask), κ_3= 1.0 (reward for cases where the given vocal segmentation cues indeed partitioned the task in a primitive subtask-wise sense), and κ_4= 5.0 (reward for cases where the manipulated objects in the episodes that were currently assigned to a primitive subtask were used in the naming of this primitive subtask). p_k in Equation (3-17) was determined as the sampling index so that $(d_{p_k} - d_{h_{k-1}})/(d_{b_k} - d_{h_{k-1}})$ = 0.4 and $h_{k-1} < p_k < b_k$; where d_{p_k} was the total distance the end-effector traveled over the entire trajectory before the ${p_k}^{th}$ sampling period in the task demonstration, h_{k-1} was the sampling index when the $(k-1)^{th}$ object was engaged, b_k was the first sampling index of the critical segment for the k^{th} object. q_k in Equation (3-21) was the sampling index so that $(d_{q_k} - d_{h_{k-1}})/(d_{b_k} - d_{h_{k-1}})$ = 0.25 and $h_{k-1} < q_k < b_k$.

5.2.4 Task Setups in Experiments

5.2.4.1 Task Setups for RTLA Experiments

One of the purposes of the RTLA experiments was to test the ability of the robot to adapt its learned tasks to new task setups. For each of the two tasks, “Lay out Table” and “Pour Liquid”, ten different task setups were selected to approximately cover the whole robot workspace. The executed task setups (locations and orientations of objects and robot starting point) for the task “Lay out Table”, relative to the setup in the in-whole task demonstration, are reported in Table 5-1. Large relative setup locations and orientations are highlighted in bold, if the values are close to or more than 295 mm for translations and 26° for rotations. Note that the robot’s maximum possible working radial distance was 864 mm. The task setup in the demonstration was (x, y, z; α, β, γ) (mm, mm, mm; deg, deg, deg): knife at (600, 186, 119; 0, 0, 1), fork at (601, 282, 119; 0, 0, 0), placemat at (536, -316, 118; 0, 0, 44), and robot starting pose (489, 0, 446; 0, 50, 0). The alignment of an object on the placemat was defined by two translations (x, y) and an orientation, as illustrated in Figure 5-4. Two coordinate frames were defined on the left and right bottom corners of the placemat, respectively. Two translation components (x, y) were used to define: the coordinates of the left bottom corner of the object foam block placed on the left side of the placemat (the object was the fork here) in the left coordinate frame, and the coordinates of the right bottom corner of the object foam block placed on the right side of the placemat (knife object used here) in the right coordinate frame. The angle from the short edge of the placemat to the long side of the object was represented by θ. The triplet (x, y,θ) was used to represent the alignments of the knife and fork on the placemat.

Table 5-1: Different executed task setups for the task "Lay out Table", relative to the setup in the in-whole task demonstration. Large relative setup locations and orientations are highlighted in bold, if the values are close to or more than 295 mm for translations and 26° for rotations. The setup in the demonstration was (x, y, z; α, β, γ): knife at (600, 186, 119; 0, 0, 1), fork at (601, 282, 119; 0, 0, 0), placemat at (536, -316, 118; 0, 0, 44), and robot starting pose (489, 0, 446; 0, 50, 0).

m	Task Setups Relative to Setup in Demonstration Differences in location: (Δx, Δy, Δz) (mm, mm, mm) Differences in orientation: ($\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$) (deg, deg, deg)											
	Knife			Fork			Placemat			Robot starting point		
1	-49	90	2	-86	94	2	11	22	1	0	0	0
	0	0	-3	0	0	20	0	0	-23	0	0	0
2	33	81	0	-74	21	0	2	17	0	0	0	0
	0	0	-4	0	0	**54**	0	0	**-42**	0	0	0
3	10	**-456**	2	-15	56	2	79	**352**	-4	0	0	0
	0	0	-3	0	0	-4	0	0	**-112**	0	0	0
4	37	**-297**	-3	-30	**-518**	-3	84	**371**	-3	0	0	0
	0	0	**-53**	0	0	**-34**	0	0	16	0	0	0
5	-2	**-364**	-2	-19	**-561**	-5	140	**500**	-5	0	0	0
	0	0	**43**	0	0	-23	0	0	21	0	0	0
6	9	**-355**	-7	-84	**-619**	-7	24	**492**	-3	0	0	0
	0	0	-6	0	0	**32**	0	0	-1	0	0	0
7	-19	**-329**	-6	-61	**-568**	-6	121	**434**	-4	63	150	-26
	0	0	**-58**	0	0	-24	0	0	**33**	0	**-35**	20
8	-22	65	49	-103	81	49	25	69	48	56	158	-56
	0	0	**27**	0	0	**28**	0	0	1	4	-4	-1
9	5	**-384**	43	-119	79	40	36	**303**	49	-5	188	-56
	0	0	**-29**	0	0	**36**	0	0	0	30	-5	22
10	-41	-271	46	-64	**-541**	46	14	**514**	44	-79	-392	-55
	0	0	2	0	0	**-36**	0	0	9	4	-4	**-60**

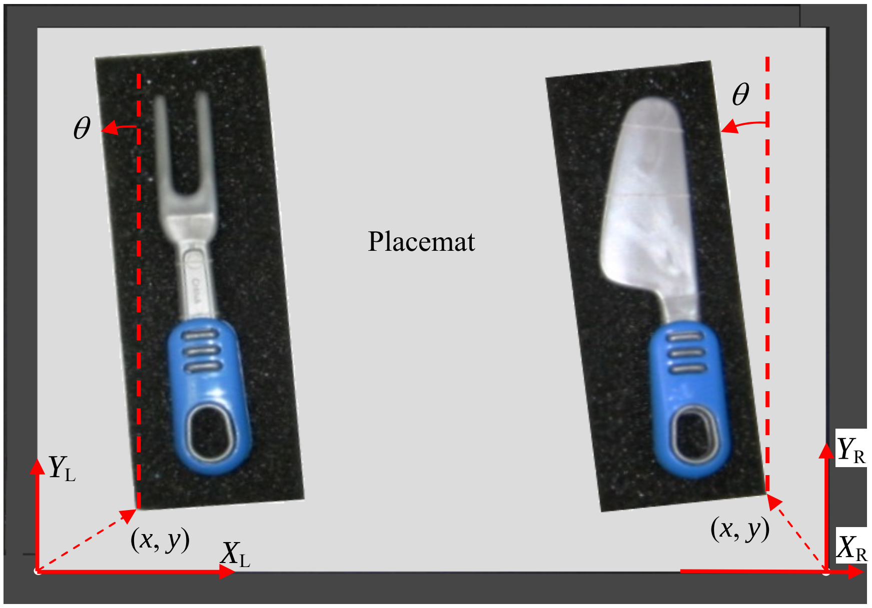


Figure 5-4: Definition of the alignments of objects on the placemat for the task “Lay out Table”. Two coordinate frames are respectively defined on the left and right bottom corners of the placemat. Two translation components, (x, y), are used to define the coordinates of the left bottom corner of the fork foam block in the left coordinate frame, and the coordinates of the right bottom corner of the knife foam block in the right coordinate frame. The angle from the short edge of the placemat to the long side of the object is represented by θ. The triplet (x, y, θ) is used to represent the alignments of the knife and fork on the placemat.

The ten different executed task setups for the task “Pour Liquid”, relative to the setup in the in-whole task demonstration, are reported in Table 5-2. Large relative setup locations and orientations are highlighted in bold, if the values are close to or more than 250 mm for translations and 20° for rotations. The setup in the demonstration was $(x, y, z; \alpha, \beta, \gamma)$ (mm, mm, mm; deg, deg, deg): bottle at (746, 251, 239; 0, 0, -3), cup at (768, -245, 230; 0, 0, -2), table at (580, 0, 109; 0, 0, 0), and robot starting pose (575, -4, 609; 0, 0, 0). The table pose was not reported in Table 5-2 since except for its Z component, the table pose was pre-set.

Note that the poses of the objects, target and robot starting point in all task setups were determined by the Vision Agent and unknown to the teacher or user until the completion of the task teaching and learning (or the task execution if there was no task teaching involved).

Blending Scheme II (Equation (3-20)) was used to generate the task paths for all the task setups. Blending Scheme I (Equation (3-19)) was also tested using two sample setups, one from each task: Setup 7 in Table 5-1 and Setup 8 in Table 5-2.

Table 5-2: Different executed task setups for the task "Pour Liquid", relative to the setup in the in-whole task demonstration. Large relative setup locations and orientations are highlighted in bold, if the values are close to or more than 250 mm for translations and 20° for rotations The setup in the demonstration was (x, y, z, α, β, γ): bottle at (746, 251, 239; 0, 0, -3), cup at (768, -245, 230; 0, 0, -2), table at (580, 0, 109; 0, 0, 0), and robot starting pose (575, -4, 609; 0, 0, 0) with units of (mm, mm, mm, deg, deg, deg).

m	Task Setups Relative to Setup in Demonstration Differences in location: (Δx, Δy, Δz) (mm, mm, mm) Differences in orientation: ($\Delta\alpha$, $\Delta\beta$, $\Delta\gamma$) (deg, deg, deg)								
	Bottle Locations			Cup Locations			Robot starting Locations		
1	0	56	1	4	-37	-1	0	0	0
	0	0	-2	0	0	-3	0	0	0
2	-8	-37	-1	15	140	-1	0	0	0
	0	0	-2	0	0	-3	0	0	0
3	22	**-454**	3	-46	**439**	-2	0	0	0
	0	0	-4	0	0	-2	0	0	0
4	-18	34	49	2	29	49	1	4	-4
	0	0	-4	0	0	1	0	0	0
5	-14	58	49	-52	-45	49	0	-1	0
	0	0	0	0	0	0	0	0	0
6	-71	112	50	-58	-10	50	1	4	-4
	0	0	**27**	0	0	-15	0	0	0
7	-45	**-519**	79	-57	**540**	48	1	4	-4
	0	0	-28	0	0	**26**	0	0	0
8	-134	185	79	-101	-54	50	3	-114	**-319**
	0	0	**34**	0	0	**-33**	0	15	-12
9	-38	51	28	-49	-13	0	-37	182	-127
	0	0	**23**	0	0	**-20**	-7	10	-1
10	-11	**-540**	31	-35	**496**	-3	-3	**-260**	123
	0	0	**-33**	0	0	**23**	-7	10	**-31**

5.2.4.2 Task Setups for User Study

In the user study, all the task setups were set up by the student investigator for each human subject. The objects that were grasped by the robot in the in-whole task demonstration were placed so that the objects were aligned approximately parallel to the line of sight from the teacher to the objects. This alignment helped the subjects have a good direct view to teleoperate the robot to grasp the objects. For task setups in the robot-task practice with 50% of the demonstration speed, objects to be grasped were placed at the approximate locations where they were in the demonstration. For the task setups in the robot-task practices with full demonstration speed, the poses of the objects and targets were considerably different from those in the demonstration, to purposely show the subjects the ability of the robot to adapt the just learned tasks to new setups. The subjects were asked for their evaluations of this adaptability in the questionnaire (Appendix B.5) at the end of their participation. For all task setups, only Blending Scheme II (Equation (3-20)) was used to generate the task paths.

5.3 Procedures for the Robot-Task Learning and Adaptation Experiments

In the RTLA experiments, nine steps were adopted by the student investigator to teach the "Lay out Table" task to the robot, as follows:

1) Pre-analyze the task to be taught, including its hierarchical task structure and its approximate task path in the space.
2) Check and teach the robot needed background knowledge via voice dialogue, i.e. information regarding objects and tasks/subtasks of interest. The teacher can move the robot end-effector via teleoperation over an object, and ask and/or teach the name of the object.
3) Introduce the overall task to be taught to the robot via voice dialogue, i.e. the task structure and involved robot hand actions. The robot can then build the task structure.
4) Teach the task to the robot step by step. The teacher is allowed to select the subtasks to be taught in this step. The only constraint is that if one child subtask of a sequential subtask is chosen, all its sibling subtasks must be selected and sequentially demonstrated. In the subtask demonstration, the teacher might direct the robot's attention to the transitions of the subtask by giving some vocal subtask-segmentation cues in the subtask transitions, by utterances such as "first subtask" and "next step".
5) Demonstrate the task in whole while giving some vocal subtask-segmentation cues in the subtask transitions, by utterances such as "first subtask" and "next step".
6) Robot: Learn taught tasks. The robot would learn tasks by segmenting the demonstrated task trajectories, assigning the partitioned trajectory segments to the introduced task structure, extracting involved primitive skills, and generalizing the taught task trajectories and task structures. If the robot would have difficulty in the segmentation and assignments, it would initiate a request to the teacher for help on these matters.
7) Robot: Practice the learned task with two speed levels: 50% and 100% of the demonstrated speeds. When using 50% of the taught speed, the robot would say the name of subtask to be executed at the beginning of each primitive subtask. During the robot task practice, the teacher could give timely feedback on the robot's task performance, and the robot should refine its task knowledge accordingly and reflect the refinement instantly in the next sampling period.
8) Robot: Save to the database its learned task knowledge, including the hierarchical task structure, task trajectories in different frames, blending weights, involved primitive skills or actions, and involved objects.

9) Request the robot to execute the just learned task in ten different task setups described in Section 5.2.4. The robot would instantiate the task according to the given setups using the task knowledge saved in the database.

For the task "Pour Liquid", all the above steps except Step 4 (step-by-step task teaching) were used by the investigator to teach the task to the robot.

5.4 User Study Protocol

Ten human subjects participated in the user study to evaluate how well the developed system enabled general users to teach robot tasks, and how well the robot could learn the taught tasks from the general users. The study was approved by the Office of Research Ethics, University of Waterloo (Approval–Appendix B.1, Recruitment–Appendix B.2). All ten subjects were asked to teach the task "Lay out Table" to the robot while Subjects 4, 5, 6, 8 and 10 were also requested to teach the robot the task "Pour Liquid". The first two subjects were used for pilot testing. Each subject was instructed to follow the protocol below:

- Read an information letter (Appendix B.3) and upon consent sign two consent forms (Appendix B.4).
- Complete two standard training sessions of the Microsoft Speech Recognition system.
- Be given a brief introduction to the system and sample scripts to use for the human-robot vocal interaction (Sample scripts are found in Appendix A.1).
- Watch a demonstration video of the robot-task teaching and learning. The vocal human-robot interaction in the teaching stages "check/teach needed background knowledge" and "overall task introduction" was excerpted from the video, and is given in Appendix A.2.
- Pre-analyze the object-manipulation task: write out its task structure by breaking down the task into subtasks based on their own knowledge, naming relevant subtasks with vocabulary of their own choice (i.e. the verbs, directional prepositions, object names, and target names in the subtask names); specify the involved robot hand actions for each primitive subtask; and consider its approximate task path in space.

  The student investigator would then examine the names of the task and subtasks to make sure that syntax of the task/subtask names, as described in Section 3.2.2, is honored. Proper specification of the involved robot hand actions would be verified for each primitive subtask. The names of relevant targets would be checked to properly appear in the names of primitive subtasks that involve targets. If the names of some objects, verbs or directional prepositions in the pre-analyzed task and subtask names would not be recorded in the database, the investigator would input these names into the database.
- Request the robot to learn a task from human teaching. The robot would initiate an exploration of its workspace by asking the teacher to: place the involved objects and target in place, and say "it is ready" to the robot. The investigator would place the

objects and the subject would then say “it is ready” to the robot, and the robot would start the exploration.

- Check the robot’s background knowledge for the task to be taught and teach lacking knowledge if necessary.
- Introduce the overall task to the robot via vocal human-robot dialogue.
- Practice the robot teleoperation using their hand motion to teleoperate the robot to demonstrate the task. At the beginning of each of these practices and the in-whole task demonstration, the robot would carry out an exploration of its workspace.
- Demonstrate the task in whole to the robot by teleoperation.
- Help the robot partition the taught task upon request from the robot if the robot found it had failed to do so.
- Help the robot refine its learned task knowledge by giving timely feedback during the robot task practice. (At the beginning of each task practice, the investigator would set up the involved objects and targets for the subject).
- Complete a questionnaire for their subjective evaluation of the robot teaching and learning system.
- Receive a letter of appreciation from the student investigator.

5.5 Data Collection

The data collected in both the RTLA and user study experiments were as follows:

1) time used for each task teaching and learning stage;

2) actual number of robot requests and recognized teacher’s responses during the overall task introduction;

3) introduced task structure;

4) instants that the subtask-segmentation cues were issued by the teacher during the task demonstration;

5) instant the vocal command given by the teacher for the robot to start to pour beads was issued in the task “Pour Liquid”;

6) demonstrated task and subtask trajectories and robot hand status throughout the demonstration;

7) data computed in the learning process: computed MSV, individual votes by chosen signals and overall vote for segmentation of the demonstrated task trajectory, blending rates $\lambda_1(i)$ and $\lambda_2(i)$, assignment of the obtained task trajectory episodes to the introduced primitive subtasks; task trajectories generalized in the world frame, object frames and object-oriented frames; extracted primitive skills, and generalized task structure;

8) planned, practiced and executed task trajectories;

9) number of recognized feedback cues issued by the teacher during the robot's task practice;

10) refined task knowledge if timely feedback was issued during robot task practices: refined blending rates $\lambda_1(i)$ and $\lambda_2(i)$, assignment of the obtained task trajectory episodes to the introduced primitive subtasks, task trajectories generalized in the relevant frames, and generalized task structure;

11) starting locations and orientations of involved objects, targets, and robot in the task demonstrations, robot-task practices, and robot-task executions;

12) final alignments of objects on target (only for the RTLA experiments)

13) time used for human subjects to practice the robot teleoperation (only for the user study);

14) subjective evaluations of the robot-task teaching and learning system, through a survey using a questionnaire (Appendix B.5) (only for the user study).

The minimum number of responses that the subject has to give to the robot to introduce the overall task were computed based on the introduced structure: four responses for a non-primitive subtask (two types of responses from the subject - the subtask name and number of child subtasks that the subtask has, and for each there would be an initial response and a confirmation response from the subject), six responses for a primitive subtask (four as for a non-primitive subtask plus an additional initial response and confirmation response regarding the involved robot hand actions) (See sample human-robot dialogue in Appendix A.2 for examples of responses).

Chapter 6
Experimental Results

6.1 Introduction

Two series of experiments, the robot-task learning and adaptation (RTLA) experiments and user study, were conducted. In the RTLA experiments, the student investigator taught the tasks "Lay out Table" and "Pour Liquid" to the robot. After the robot learned each of the taught tasks, it was then requested to execute the task in ten different task setups to test the robot's ability to adapt its learned task to new task setups in Table 5-1and Table 5-2.

In the user study, subjective evaluations of the robot-task teaching and learning system were performed by a convenience sample of ten human subjects. The subjects evaluated how well the system enabled them to teach tasks to the robot and how well the robot learned the tasks from their teaching. The results of the two series of experiments are described in the following two sections.

6.2 Results of Robot Task Learning and Adaptation Experiments

In the RTLA experiments, the two tasks, "Lay out Table" and "Pour Liquid", were taught by the investigator to the robot using the experimental procedures described in Section 5.3. Results of the experiments for the two tasks are reported in the following subsections. Note that the step-by-step teaching stage was only used for "Lay out Table", and not used for the task "Pour Liquid".

6.2.1 Experimental Results for the Lay-out-Table Task

In the first teaching and learning stage, "check and teach needed background knowledge", the robot responded promptly to the recognized teacher's vocal requests and instructions. The robot's response was easily understandable to the teacher. The following conversation is an example.

Teacher: What objects do you know about?

Robot: There are six objects I can identify: table, knife, fork, placemat, bottle, and cup. There are also ten objects I cannot identify: object, target, block, teapot, spoon, spatula, juice, container, water, and banana.

Note that if the robot claimed it could identify an object, the name of this object existed in the database and the object features (extracted by the vision agent) of this object were already associated with the object name. If the robot claimed it could not identify an object, only the name of this object existed in the database.

The teacher was able to ask and modify attributes of objects, tasks, and subtasks quite naturally. The teacher moved the robot end-effector over the object "*spatula*" via teleoperation, and asked for the name of the object with an utterance "tell me its name". The robot responded with "I cannot recognize this object. Please give me its name first." Then the teacher told the robot, “Its name is *spatula*”. The robot associated the given name with the group of features extracted from the spatula, and saved the association in the database.

The taught task structure of the “Lay out Table” task is depicted in Figure 6-1. The task consisted of two subtasks: “Lay out Knife” and “Lay out Fork”. “Lay out Knife” had two child subtasks: “Pick up Knife” and “Place Knife”, while “Lay out Fork” had two primitive subtasks “Pick up Fork” and “Place Fork onto Placemat”. “Pick up Knife” was further decomposed into three primitive subtasks: “Approach Knife”, “Grasp Knife”, and “Move up Knife”, while “Place Knife” was made up of two primitive subtasks: “Move Knife down to Placemat” and “Place Knife onto Placemat”. Note that the structures of the subtasks “Lay out Knife” and “Lay out Fork” were purposely designed to be different. Their use verified that the robot can handle different task structures.

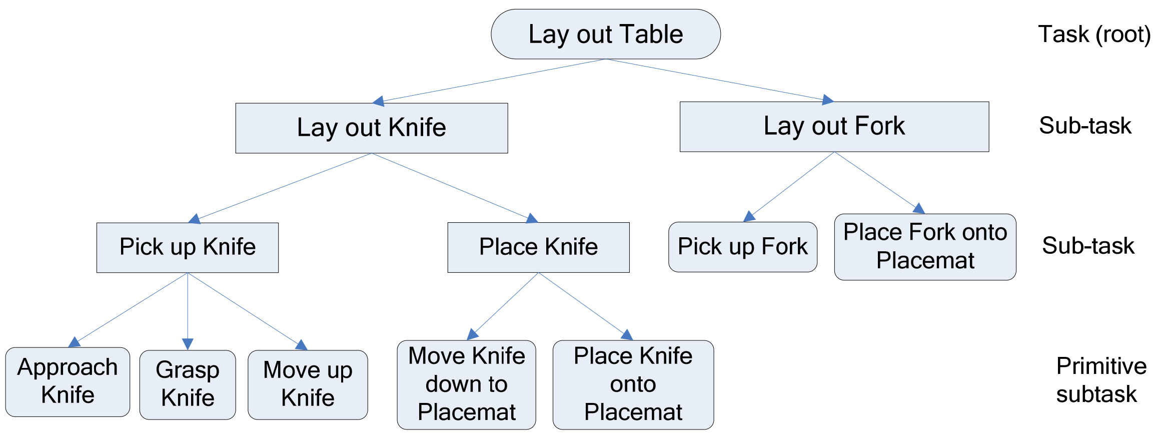


Figure 6-1: Hierarchical structure of the task “Lay out Table” used in RTLA experiments. Subtasks with rectangular blocks that have rounded corners are primitive subtasks; they have no child subtasks.

The robot controlled the human-robot vocal dialogue in the overall-task-introduction teaching and learning stage, and asked for acknowledgement for every unconfirmed response in order to achieve a human-robot conversation with reliable information exchange. An example of the conversation is given as follows:

Robot: Now, we are to build the ***first*** *step of the task: lay out table. Please tell me the* ***name of this subtask***

Teacher: The subtask name is ***lay out knife****.*

Robot: the subtask name is ***lay out knife****. Is this right?*

Teacher: Yes, it is.

If the acknowledgement was positive, the robot moved to request information for the next item. Otherwise, the robot repeated the current request. There were: $(4\times5)+(6\times7) = 62$ minimum number of responses required since the introduced task includes 5 non-primitive subtasks and 7 primitive subtasks (Figure 6-1). The actual number of responses made by the teacher during the experiment was 66. Only human responses that were recognized by the robot were counted. Information about the task/subtask actions (i.e. the verbs in the task/subtask names), the involved objects and targets, as well as the directional prepositions (e.g. the "up" in the subtask "Move up Knife") in the task or subtask names, were automatically extracted by analyzing the syntax of the names.

In the stage of step-by-step teaching, four subtasks, "Pick up Knife", "Place Knife", "Pick up Fork" and "Place Fork onto Placemat", were taught sequentially. When the teacher told the robot the name of the subtask to be taught next, the robot responded by stating the subtask's name, number of its child subtasks, involved objects, robot hand actions, and whether this subtask had been previously learned.

The teacher then demonstrated the task in whole to the robot and offered some vocal subtask segmentation cues at the subtask transitions. By analyzing the collected data from the in-whole task demonstration, the robot automatically extracted four primitive skills: Close-Robot-Hand skill, Open-Robot-Hand skill, Close-Robot-Hand skill, and Open-Robot-Hand skill; and the instants they were initiated. The robot also found four objects involved in this task: Knife, Placemat, Fork and Placemat, and the instants at which they were engaged. Note that the Placemat was counted twice here with different instants of engagement. At this point, the relationships between the primitive subtasks and parts of the demonstrated trajectory were not yet established. In other words, the introduced task structure was not yet associated with the robot end-effector trajectory data.

The MSV of the whole demonstrated task trajectory, as well as the instants that the robot gripper status changed, and the instants the teacher offered vocal subtask-segmentation cues are shown in Figure 6-2. The transitions between the demonstrated task motions (at different speeds) are indicated by the valleys of the MSV.

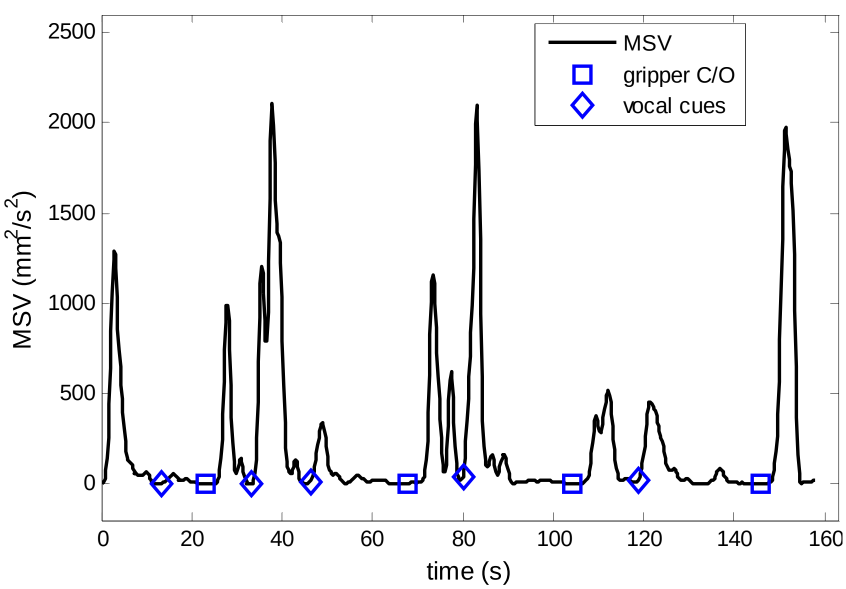


Figure 6-2: Mean squared velocity (MSV) of the demonstrated task trajectory for the task "Lay out Table". The gripper status changes (gripper Close/ Open actions), indicated by "gripper C/O", and the instants that the teacher gave vocal subtask-segmentation cues are marked by □ and ◊, respectively.

The overall vote $v(t)$ and individual weighted votes by the MSV, gripper status, and vocal subtask-segmentation cues were then calculated. The votes generated from the in-whole task demonstration are illustrated in Figure 6-3. In the figure, local peaks of the individual votes by the three signals reflect the instants the local minima of the MSV were selected, instants the gripper status changed, and instants that the teacher offered vocal cues in the task demonstrations. Note that the gripper actions and vocal cues corresponding to the individual peaks in Figure 6-3 are the same as those marked in Figure 6-2. For the in-whole demonstrated task trajectory, fifteen final segmentation points (SPs) and four removed false SPs, resulting from applying Equations (3-5) and (3-6), are also indicated in the figure.

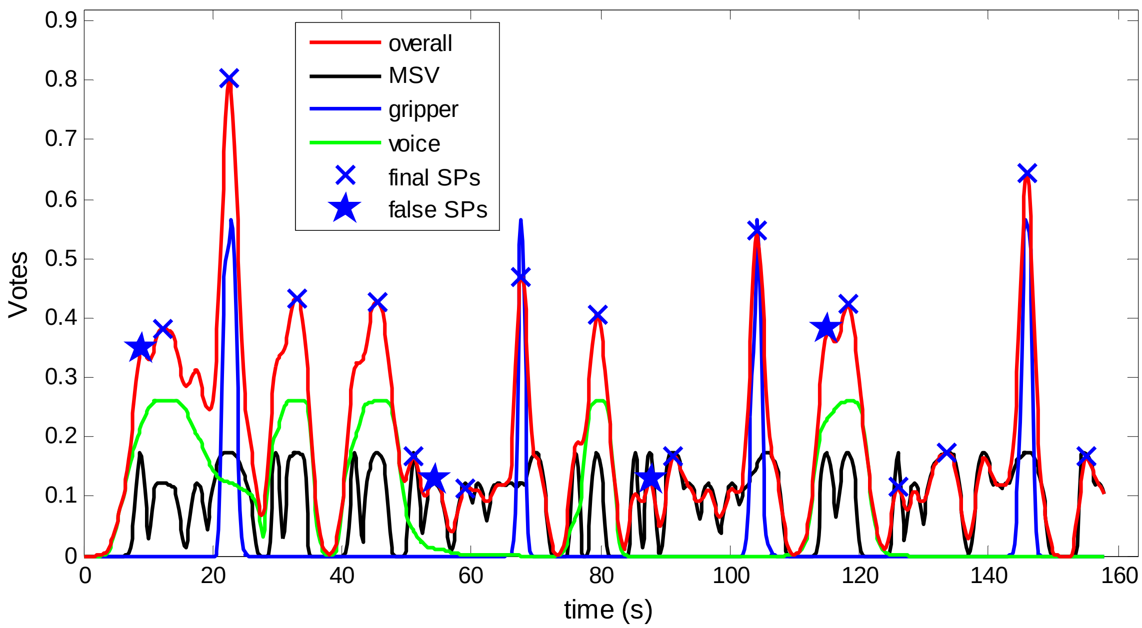


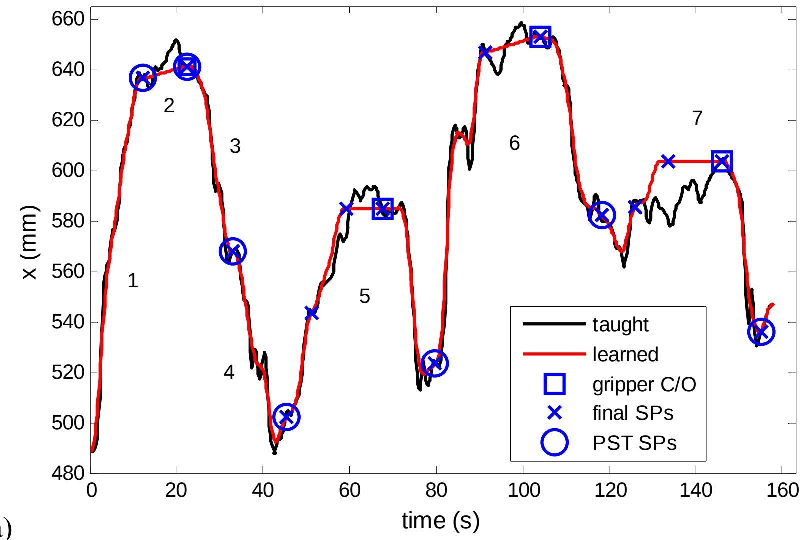
Figure 6-3: Votes for segmenting the demonstrated task trajectory for the task "Lay out Table", showing overall vote, and weighted votes by: MSV, gripper state changes, and voice cues. Resulting segmentation points (SPs) are indicated by the asterisk *, and removed false SPs by the five-pointed blue stars.

Assignment of the generated task trajectory episodes (or segments) to the primitive subtasks was achieved by applying the optimization process defined in Equation (3-8), using coarse-to-fine and brute force optimization methods. Two coarse assignment hypotheses were generated: one produced by comparing the subtask trajectories taught in the step-by-step teaching with the in-whole taught task trajectory; and the other created by directly matching primitive subtasks for which robot hand actions were specified to the task trajectory episodes, based on information of robot hand actions and involved objects. Further optimization of the two coarse hypotheses using Equation (3-8) produced two identical optimal assignments. The final assignment is shown in Figure 6-4, indicated by the blue circles. In addition, the gripper Close/Open events and the final SPs are also marked in the figure. It seems that the primitive subtask "Move Knife Down to Placemat" (task segment 4 in Figure 6-4c) may have been incorrectly assigned the trajectory segment as the component of the trajectory in the vertical (*Z*) direction was small and not essentially “down”. However, the teacher intuitively and seemingly correctly gave a vocal segmentation cue after making a much larger movement than the following smaller movement toward the docking pose for knife placement (Figure 6-4b). This vocal subtask-segmentation cue (Figure 6-2) greatly influenced the assignment.

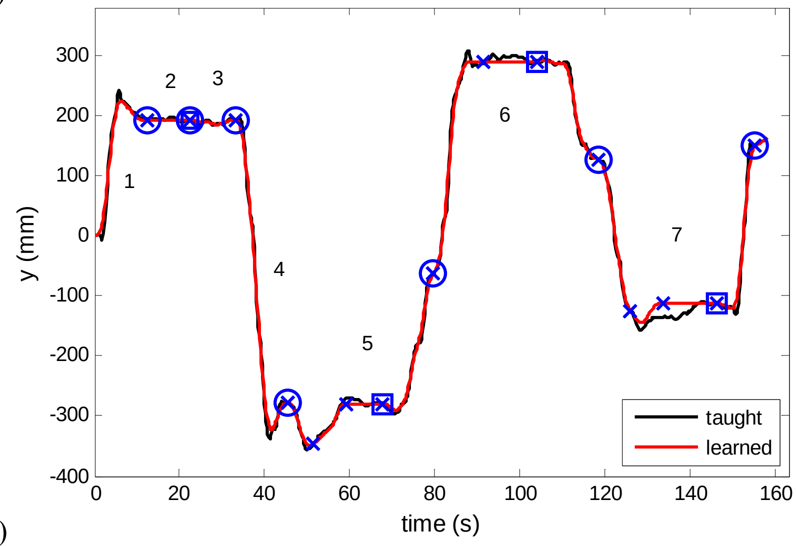
The demonstrated task trajectory was generalized in the world frame (i.e. the robot frame), and then transformed into four object frames and four object-oriented frames for the four objects. The demonstrated and corresponding generalized task trajectories in the world frame for each translation and rotation component are also shown in Figure 6-4. The demonstrated motions of the robot in the critical trajectory segments (segments before robot-gripper status changes) were modified to comply with the relevant transformation constraints from a docking pose to its paired grasping pose or releasing pose in the resulting generalized trajectory. This can be seen where the teacher made fine adjustments before an object was grasped or placed (t = 136 s to 145 s). The trajectory episodes adjacent to the critical episodes were also modified to smooth the transitions to the docking poses and from the grasping or releasing poses (for example, the *X* component in the episode from t = 125 s to 136 s and the all rotations in most of these episodes). A rotation during a critical episode was considered to be unintentional and then set to zero in the generalization if the rotation was less than five deg. The generalized task trajectories were much smoother than the demonstrated ones.

The computed blending weights $\lambda_1(i)$ and $\lambda_2(i)$ between the generalized trajectories in Equations (3-19) and (3-20), are shown in Figure 6-5. Before starting the critical trajectory segments (segments before robot gripper status changes), $\lambda_1(i)=1$ was true before reaching relevant docking poses. $\lambda_2(i)$ decreased rapidly to zero right after the robot departed from its starting point or positions where relevant objects were grasped or placed.

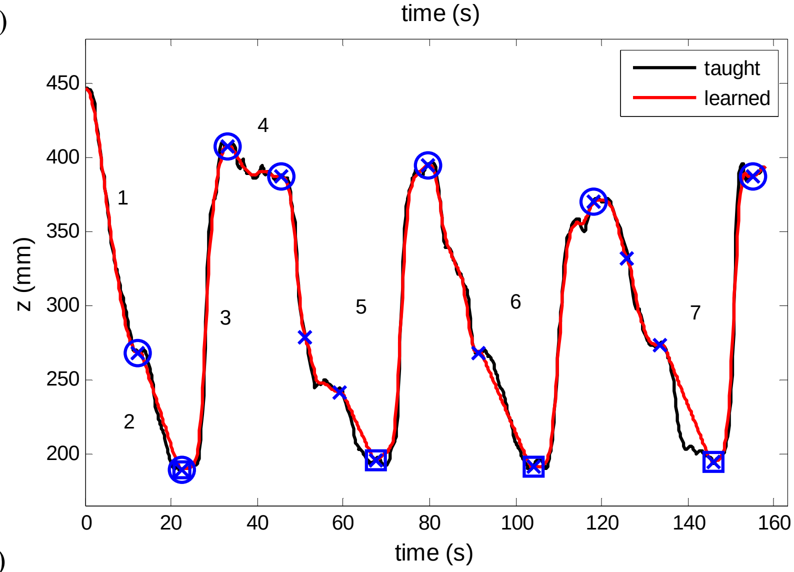
For generalization of the task structure, the subtasks “Lay out Knife” and “Lay out Fork” could be executed in any order since there was no direct dependency between the knife and fork alignments on the placemat. The child subtasks of each of these subtasks were sequential subtasks and must therefore be executed in specified order. The primitive subtask “Move Knife down to Placemat” might be skipped by the robot during task execution because no associated key state changes, such as change of the gripper status, occurred in this subtask, and it was also not adjacent to any critical segments.


x (mm)
time (s)
taught
learned
gripper C/O
final SPs
PST SPs
1
2
3
4
5
6
7

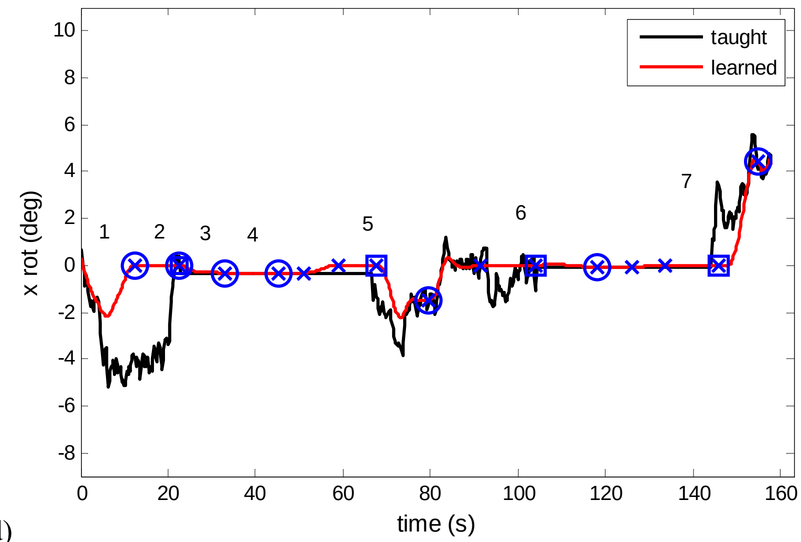
a)


y (mm)
time (s)
taught
learned
1
2
3
4
5
6
7

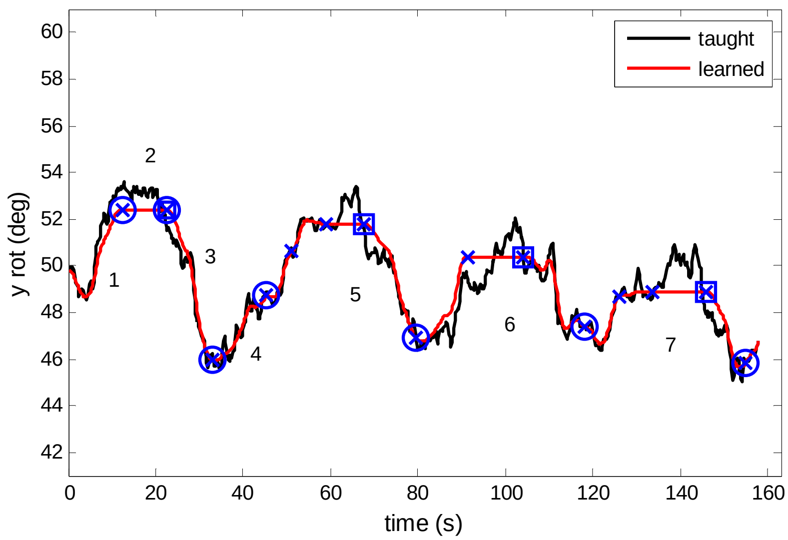
b)


z (mm)
time (s)
taught
learned
1
2
3
4
5
6
7

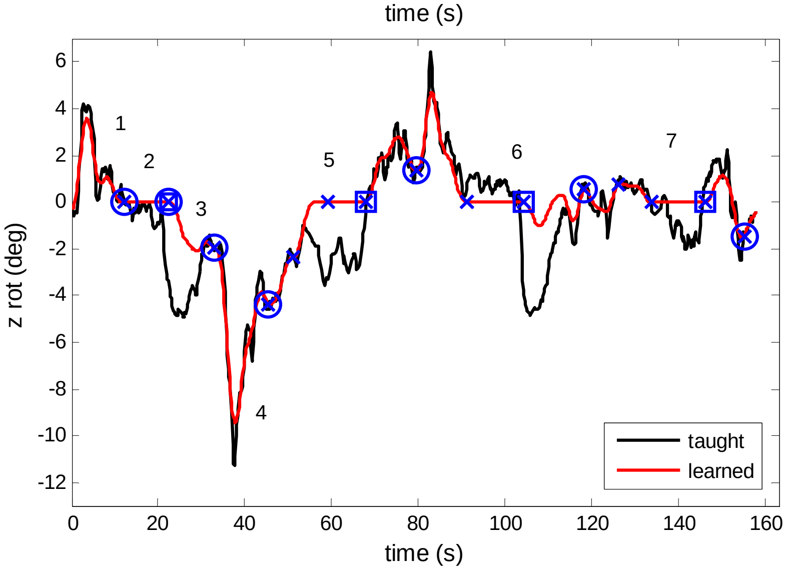
c)


taught
learned
x rot (deg)
time (s)
1
2
3
4
5
6
7

d)


taught
learned
y rot (deg)
time (s)
1
2
3
4
5
6
7

e)


taught
learned
z rot (deg)
time (s)
1
2
3
4
5
6
7

f)

Figure 6-4: Taught (demonstrated) and corresponding learned task trajectories in the world frame for "Lay out Table", for (a)-(c): *X*, *Y*, *Z* translations, and (e)-(f) *X*, *Y*, and *Z* rotations, respectively. Primitive subtask (PST)-based segmentation points (SPs) are indicated by blue circles. The primitive subtask-based task trajectory segments 1, 2, 3, 4, 5, 6 and 7, which are delimited by the PST SPs, are assigned to primitive subtasks: "Approach Knife", "Grasp Knife", "Move up Knife", "Move Knife down to Placemat", "Place Knife onto Placemat", "Pick up Fork" and "Place Fork onto Placemat" (Figure 6-1) , respectively. The instants that the robot gripper changed its status are indicated by blue squares.

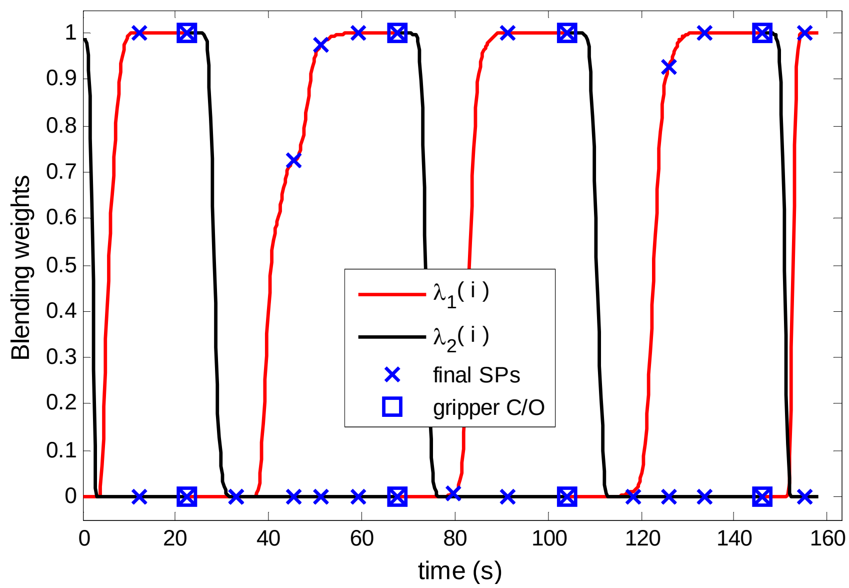


Figure 6-5: Blending weights computed in Equations (3-19) and (3-20) for the task "Lay out Table". $\lambda_2(i)$ are the weights of the robot current pose when approaching the first object or the weights of the trajectory generalized in the last object frame when they are blended with the trajectory generalized in the world frame or the trajectory generalized in the object-oriented frame of the object being approached to produce an intermediate trajectory. $\lambda_1(i)$ are the weights of the trajectory generalized in the approaching object frame when it is blended with the intermediate trajectory.

The task partitioning without considering the vocal subtask-segmentation cues and step-by-step task teaching is shown in Figure 6-6. Compared to the task partition with the cues and step-by-step task teaching included (Figure 6-4c), a new trajectory segment that ended at 85.1 s was added, the fourth primitive subtask "Move Knife down to Placemat" was extended to 58.9 s, and the fifth primitive subtask "Place Knife onto Placemat" finished earlier at 67.5 s.

The learned task was practiced twice by the robot: once with half the demonstrated speed and the second practice with the full demonstrated speed. The investigator found that the resulting task partition and performance of the task practice were quite satisfactory, and did not issue any feedback to the robot.

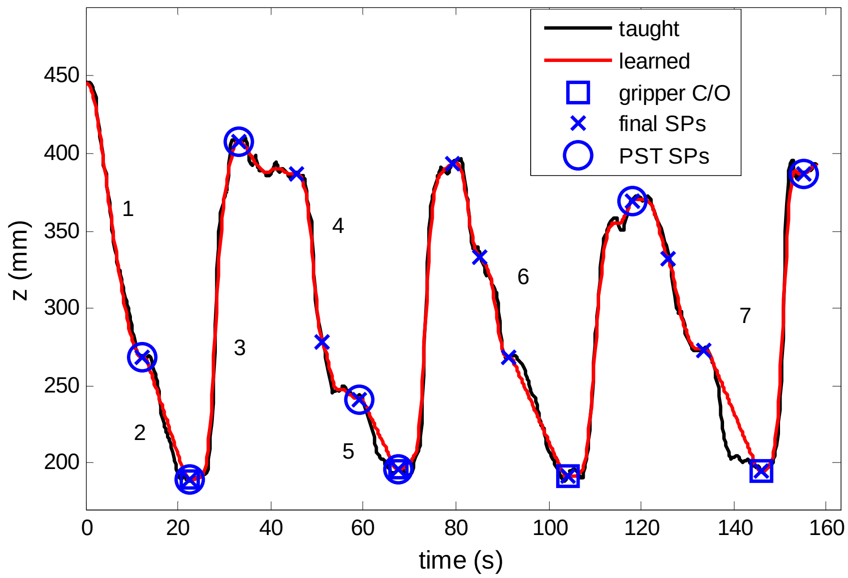


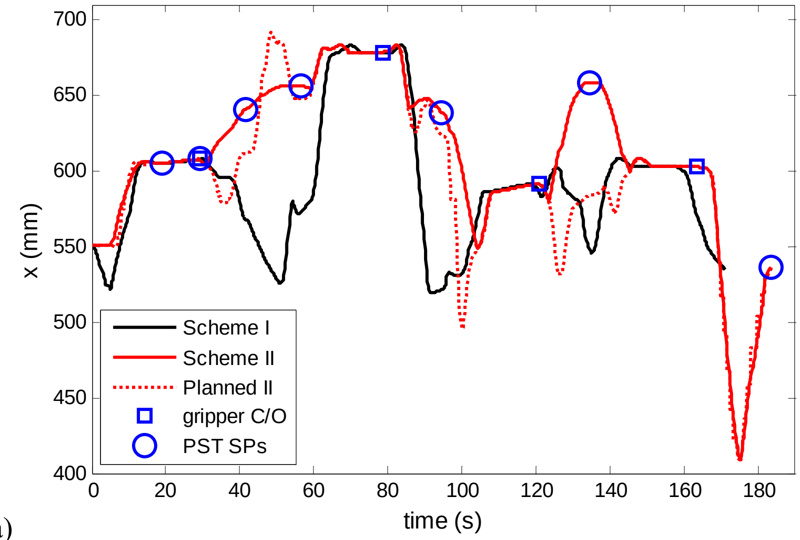
Figure 6-6: Task partition result without considering the vocal subtask-segmentation cues and step-by-step task teaching for “Lay out Table”. The segmentation points (SPs) of the trajectory segments and primitive subtasks (PST) and the gripper close/open (“gripper C/O”) instants are marked on the *z* coordinates of the taught and generalized task trajectories.

In the tests of the robot’s ability to adapt its learned task to the ten different task setups (Table 5-1) using Blending Scheme II (Equation (3-20)), the robot succeeded in executing the task for all the setups; the knife and fork were properly grasped and placed on the placemat as taught. The differences of alignments (Figure 5-4) in placing objects on the target for each task setup from the taught (demonstrated) alignments, (-40.0, 11.5, 1) for the knife, and (19.5, 26.5, -1) for the fork in units (mm, mm, deg), as well as the means and standard deviations (SD) of these differences are reported in Table 6-1. The alignments were generally very good to acceptable, and only one translational alignment difference was more than 10 mm (in Setup 9) and the maximum rotational difference was only 3 deg (in Setups 3, 4, and 5). The alignment results were very close to the taught alignments, and this was expected due to the use of Blending Scheme II. It could also be expected that task trajectories generated using Blending Scheme I (Equation (3-19)) would have achieved similar good results since the both schemes generate the same task path between the docking pose and grasping pose or docking pose and releasing pose, for a given setup.

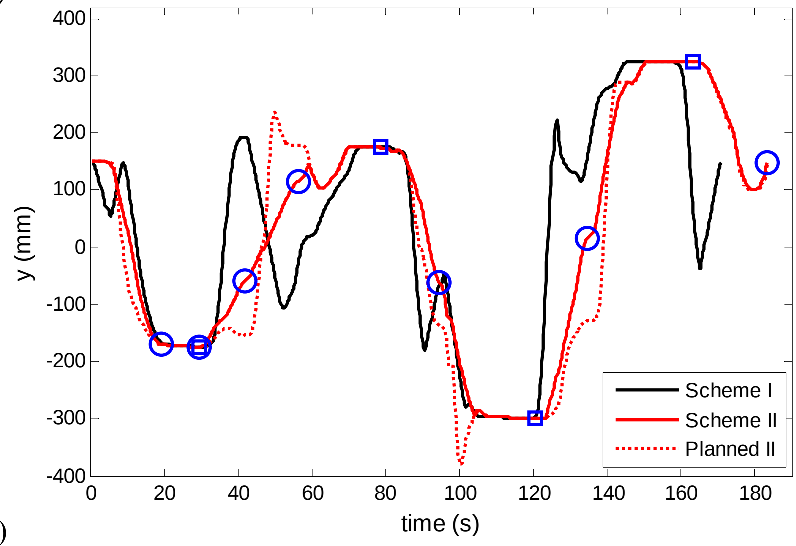
Task setup 7 (Table 5-1) was a very challenging setup for the robot to adapt the learned task to, as the differences from the taught (demonstrated) setup were great: the poses of knife, fork, placemat and robot starting point were all considerably different from those in the task demonstration. In this setup, the placemat was moved from the robot’s right side (as in the demonstration) to the left side, and the two objects (knife and fork) were initially placed on the robot’s right side instead of its left side as in the demonstration. The executed task trajectories using the two blending schemes are shown in Figure 6-7 with solid lines. The planned task trajectories generated by directly applying Equation (3-20) (Blending Scheme II), are indicated by the red dotted lines. These planned trajectories (red dotted lines) were further modified by the robot respectively in the time periods (in seconds): [0, 16.5], [30.5, 59.2], [85.1, 103.7] and [122.1, 145.1] for the problems of the robot's reachability, and [55.1, 59.5] and [169.9, 175.6] due to the possible collision of the stereo cameras mounted on the robot end-effector with the robot's fourth link (i.e. the forearm link). The modified trajectories (Scheme II, red solid line) were then executed. Note that no changes were made to the trajectories generated by Blending Scheme I because no unreachability or collision problems were detected. The task trajectories oscillated noticeably in *Y* direction (in the time periods 3-10 s, 36-59 s, 85-95 s, 125-135 s, and 162-170 s). The differences between the trajectories corresponding to the two blending schemes are notable, such as the translation in the *X* and *Y* directions, and the rotations around the Z axis. The executed speed profile of the task trajectory related to Blending Scheme II and the corresponding taught counterpart are also shown in Figure 6-8. In the critical trajectory episodes, the executed trajectory speeds were very close to the corresponding taught speeds. In the non-critical episodes, the executed trajectory speeds were quite different from the corresponding taught speeds. These differences were due to the matching relaxation between the execution and taught speeds in the non-critical episodes.

Table 6-1: Relative alignment differences compared to the taught alignments for the task "Lay out Table". The measurements of the alignments of the knife and fork on the placemat are represented by (x, y, θ), as defined in Figure 5-4. The alignment differences, $(\Delta x, \Delta y, \Delta\theta)$, in placing an object on the target for each task setup from the demonstrated alignments: (-40.0, 11.5, 1) for the knife and (19.5, 26.5;-1) for the fork in units of (mm, mm, deg), as well as the means and standard deviations of these differences are reported in the table.

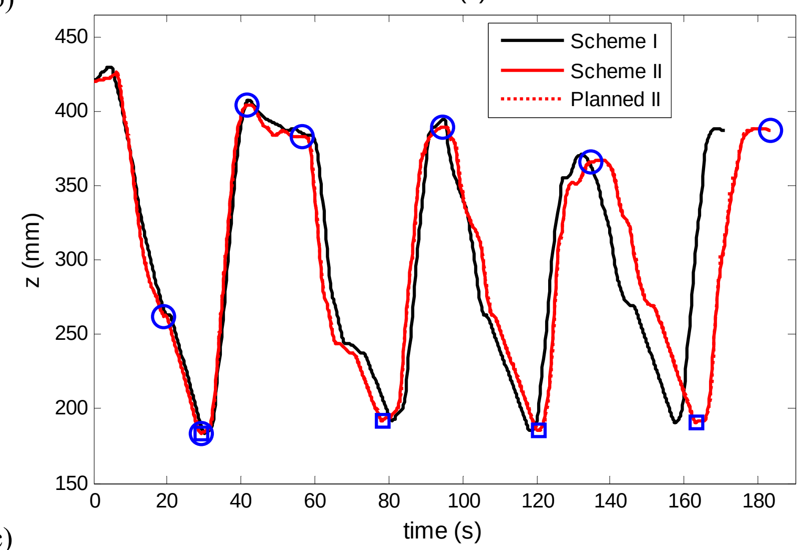
m	Alignment Differences Relative to Taught Alignments					
	Knife with Placemat $(\Delta x, \Delta y, \Delta\theta)$ (mm, mm, deg)			Fork with Placemat $(\Delta x, \Delta y, \Delta\theta)$ (mm, mm, deg)		
1	0.5	2.5	1	5.0	3.0	2
2	0.0	6.0	1	5.5	7.0	1
3	1.0	7.5	2	2.5	-3.5	3
4	5.0	4.5	3	3.0	5.5	-1
5	-1.0	4.5	-1	4.0	5.5	3
6	2.5	10.0	0	9.0	5.0	1
7	1.5	-0.5	1	-0.0	4.5	-1
8	-6.0	8.5	-2	6.5	3.0	0
9	-2.0	8.5	0	1.0	15.0	0
10	-0.5	7.5	-1	4.0	5.0	-1
Mean	0.1	5.9	0.4	4.1	5.0	0.7
SD	2.8	3.0	1.4	2.5	4.3	1.5


x (mm)
time (s)
Scheme I
Scheme II
Planned II
gripper C/O
PST SPs

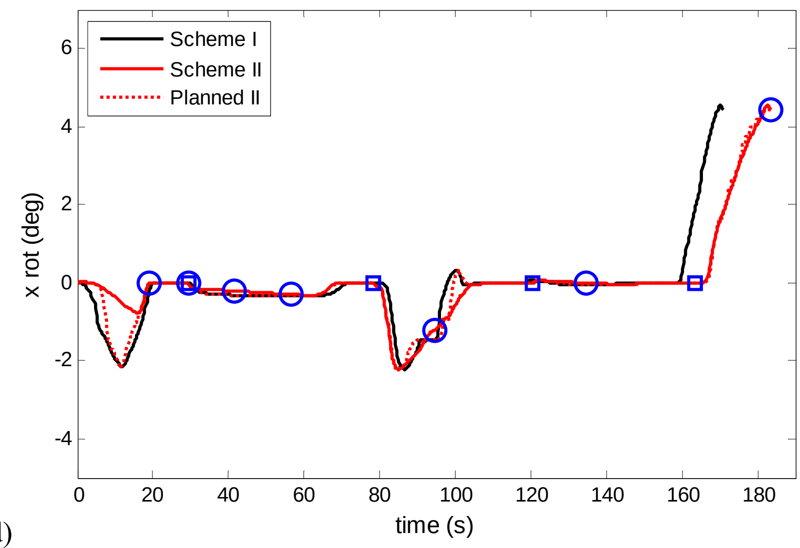
a)


y (mm)
time (s)
Scheme I
Scheme II
Planned II

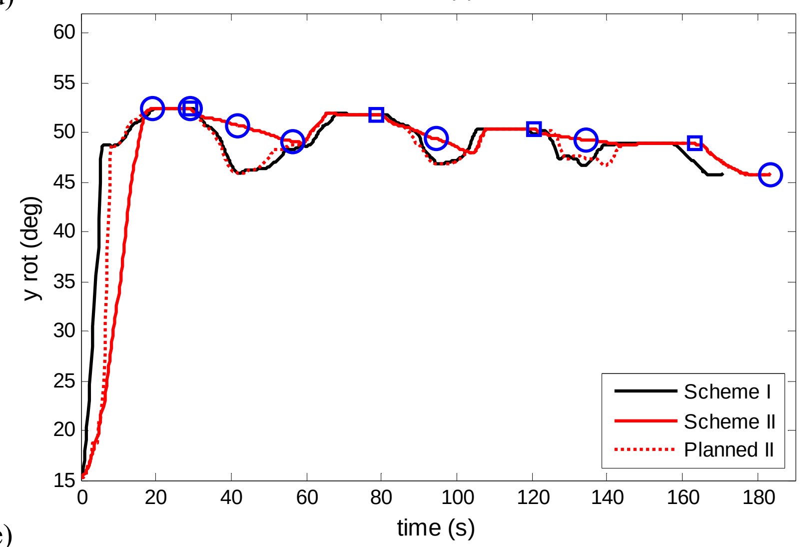
b)


z (mm)
time (s)
Scheme I
Scheme II
Planned II

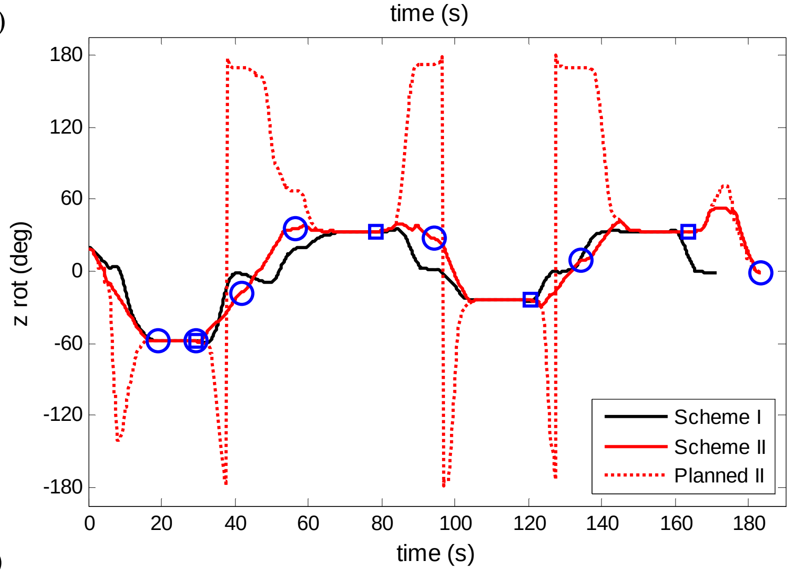
c)


Scheme I
Scheme II
Planned II
x rot (deg)
6
4
2
0
-2
-4
0
20
40
60
80
100
120
140
160
180
time (s)

d)

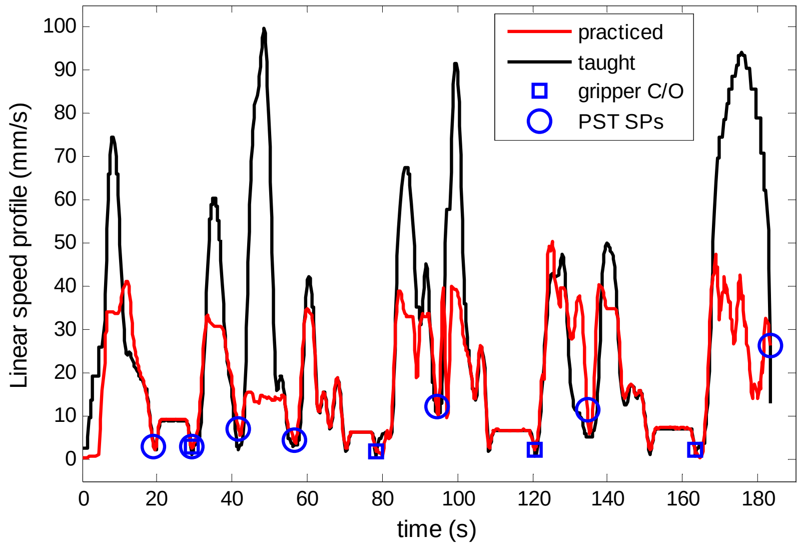

y rot (deg)
60
55
50
45
40
35
30
25
20
15
0
20
40
60
80
100
120
140
160
180
time (s)
Scheme I
Scheme II
Planned II

e)

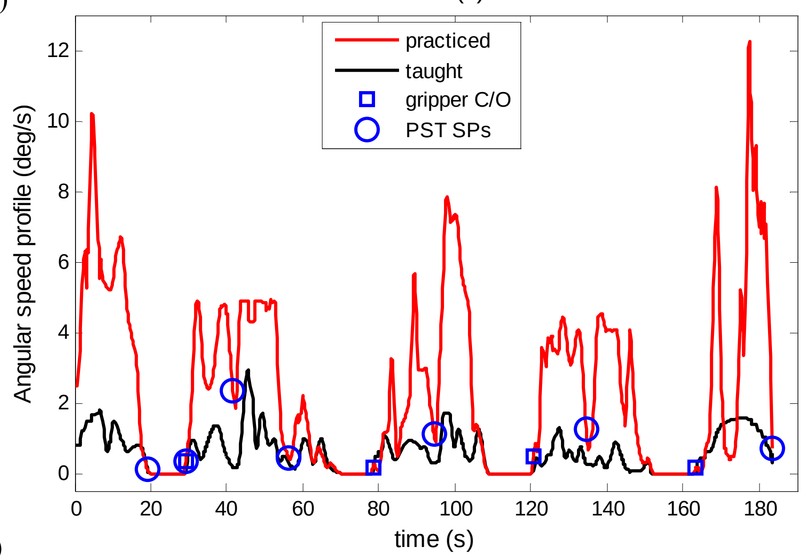

z rot (deg)
180
120
60
0
-60
-120
-180
0
20
40
60
80
100
120
140
160
180
time (s)
Scheme I
Scheme II
Planned II

f)

Figure 6-7: Executed task trajectories using the two blending schemes for the task "Lay out Table", given Setup 7 in Table 5-1. The black solid lines illustrate the executed task trajectories corresponding to Blending Scheme I (blending taught task trajectories generated in the world and object frames, respectively, based on the actual task setup), while the red solid lines corresponding to Blending Scheme II (blending taught task trajectories generated in the object-oriented and object frames, respectively). The red dotted lines are the planned task trajectories generated by directly applying Equation (3-20) (Blending Scheme II). These planned trajectories (red dotted lines) were further modified by the robot for the problems of the robot's reachability and possible collisions of the stereo cameras mounted on the robot end-effector with the robot's fourth link (forearm link). The modified trajectories (Scheme II, red solid line) were then executed.



a)



b)

Figure 6-8: Speed profile of the executed (practiced) task trajectory for the task "Layout Table" by using Blending Scheme II, for Setup 7 in Table 5-1. The linear speed profile is shown in a) and the angular speed profile is in b). The taught (demonstrated) speed profile is shown by black lines. The robot relaxed its motion matching from the taught counterparts in non-critical trajectory episodes if the executed task setup was very different from that in the demonstration.

6.2.2 Experimental Results for the Pour-Liquid Task

The taught task structure of “Pour Liquid”, an essentially sequential task, is depicted in Figure 6-9. The task consisted of only three primitive subtasks: “Pick up Bottle”, “Move this Bottle to Cup”, and “Place this Bottle onto Table”. Near the end of the “Move this Bottle to Cup”, the robot was instructed to pour the beads from the bottle to the cup using a command “start to pour to right”. “Close Gripper” and “Open Gripper”, were specified by the teacher in the task introduction, as robot hand actions for “Pick up Bottle” and “Place this Bottle onto Table”, respectively. The minimum number of needed responses from the teacher to introduce the task structure was: (4×1)+(6×3) = 22. The actual number of responses was 26.

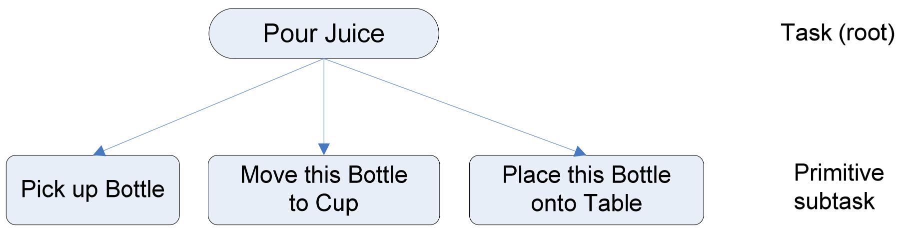


Figure 6-9: Hierarchical structure of the task “Pour Liquid”. Near the end of the “Move this Bottle to Cup”, the robot was instructed to pour the liquid from the bottle to the cup.

Three primitive skills were automatically extracted from the demonstration, Close-Robot-Hand, Pour-Liquid and Open-Robot-Hand, and the instants they were initiated. The robot also found three objects involved: bottle, cup and table, and the instants they were engaged.

The MSV of the robot end-effector trajectory computed using Equation (3-1) is shown in Figure 6-10. The transitions between the demonstrated task motions (at different speeds) are indicated by the valleys of the MSV.

The overall votes $v(t)$, as well as the individual weighted votes by the MSV, gripper status, vocal subtask-segmentation cues, and the vocal pour-liquid (bead) cue, respectively, were then calculated, as illustrated in Figure 6-11. Local peaks of the individual votes by the four signals reflect the instants that the local minima of the MSV were selected, instants that the gripper status changed, instants that the teacher offered the vocal subtask-segmentation cues in the task demonstrations, and the instants that the vocal pour-liquid trigger was issued. Ten final segmentation points (SPs) and four removed false SPs, resulting from applying Equations (3-5) and (3-6), are also indicated in the figure.

The final assignment of the trajectory episodes to the introduced primitive subtasks is shown in Figure 6-12, indicated by the blue circles. Some primitive subtasks were assigned more than one trajectory episode. The taught task trajectory was generalized in the world frame, bottle frame, cup frame, table frame, bottle-oriented frame, cup-oriented frame, and table-oriented frame, respectively. In the generalization of the task structure, the robot determined that the taught task was a sequential task.

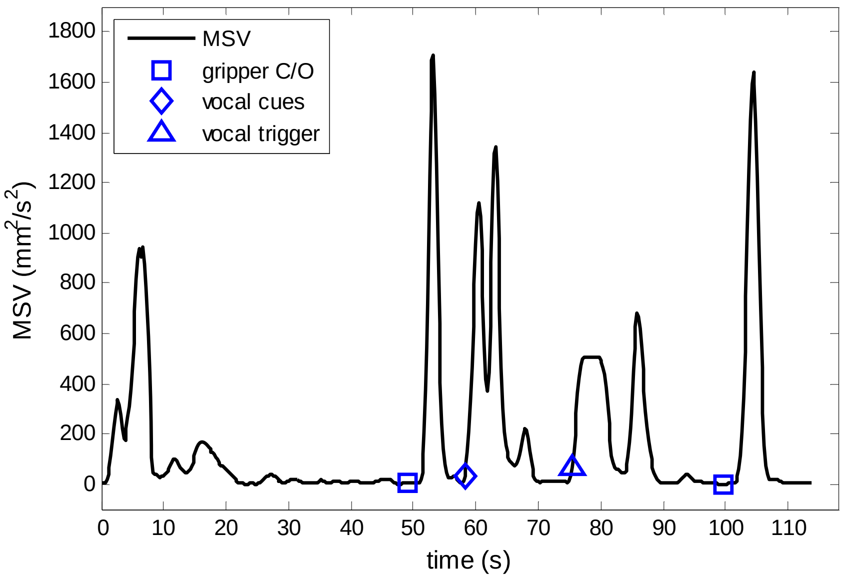


Figure 6-10: Mean squared velocity (MSV) of the demonstrated task trajectory for the task "Pour Liquid". Gripper Close/Open ("gripper C/O") actions, instants that vocal subtask-segmentation cues and the vocal trigger for the robot to start pouring that were issued are marked by □, ◊ and Δ, respectively.

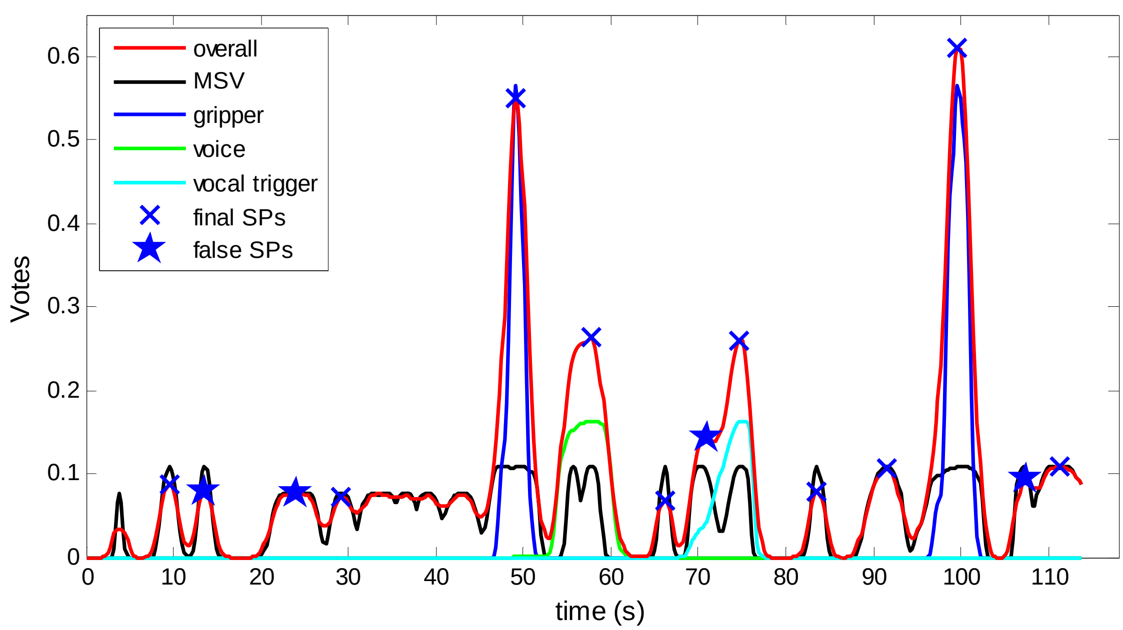
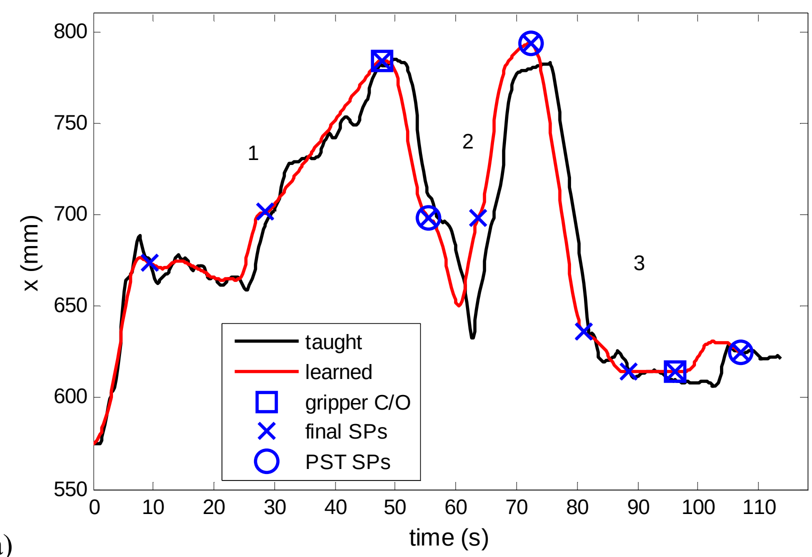
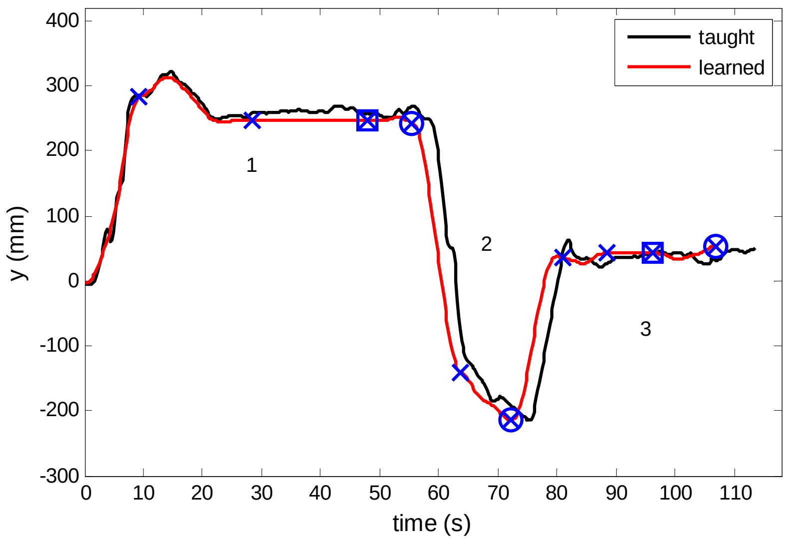


Figure 6-11: Votes for segmenting the demonstrated task trajectory for the task "Pour Liquid", showing overall vote, and weighted votes by: MSV, gripper state changes, voice cues, and vocal trigger for the robot to start pouring. Resulting segmentation points (SPs) and removed false SPs are indicated by the asterisk * and five-pointed blue stars, respectively.

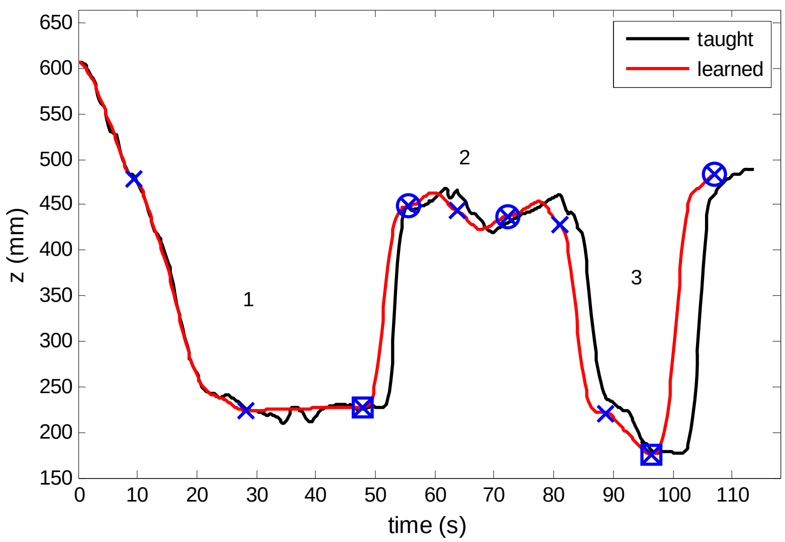
The learned task was first practiced with half the speed used in the demonstration and the task setup similar to that in the demonstration. During the practice, nine feedback utterances were issued to the robot: three "move more right" instructions were given when the robot was moving from the bottle's docking pose to its grasping pose so that the clearance between the two robot gripper fingers and the bottle was more evenly distributed; four "move more forward" instructions were issued to adjust the relative pose between the bottle and the cup before the robot started to pour the beads from the bottle to the cup, and two "move more left" instructions were provided to the robot right after the bottle was placed on the table so that the robot gripper finger would not collide with the bottle when retreating from the bottle. The refined task trajectory was then generalized and transformed to relevant frames, and the corresponding segmentation of the task, blending rates, task structure, subtask pre-conditions and goals, and primitive skills were also updated. The refined task trajectory, generalized in the world frame, and its demonstrated counterpart are shown in Figure 6-12. In the figure, the adjustment of the task trajectory based on the provided teacher feedback can be clearly observed: changes in $+X$ direction at 72 s, and offsets in $-Y$ from 27 to 47 seconds. Then, the robot practiced the task at the full speed used in the demonstration using the refined and re-generalized task trajectory. The performance of the practice was satisfactory and no further feedback was issued by the teacher. The corresponding blending weights, $\lambda_1(i)$ and $\lambda_2(i)$ in Equations (3-19) and (3-20), were also computed for the refined task trajectories and are shown in Figure 6-13.



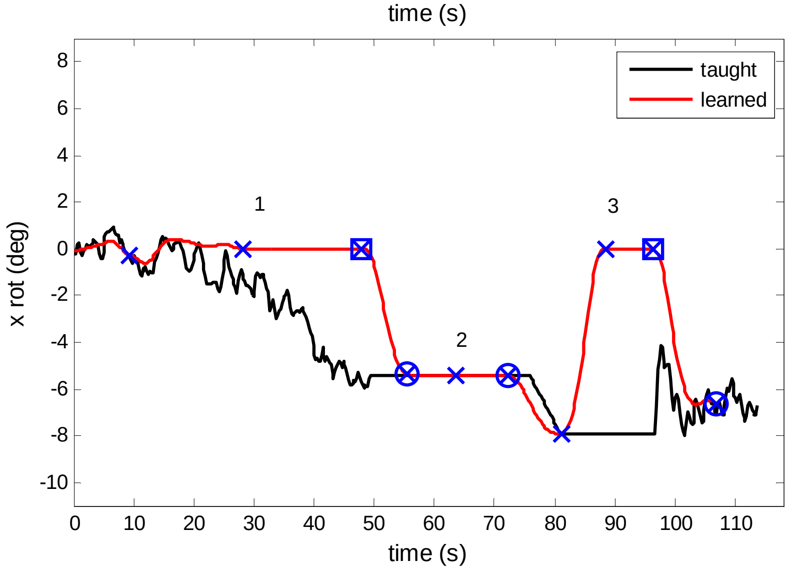
a)


taught
learned
y (mm)
time (s)
1
2
3

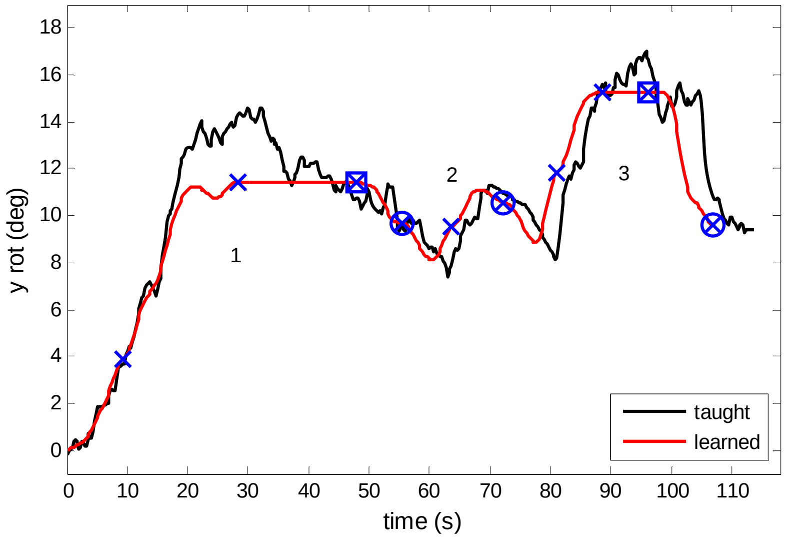
b)


taught
learned
z (mm)
time (s)
1
2
3

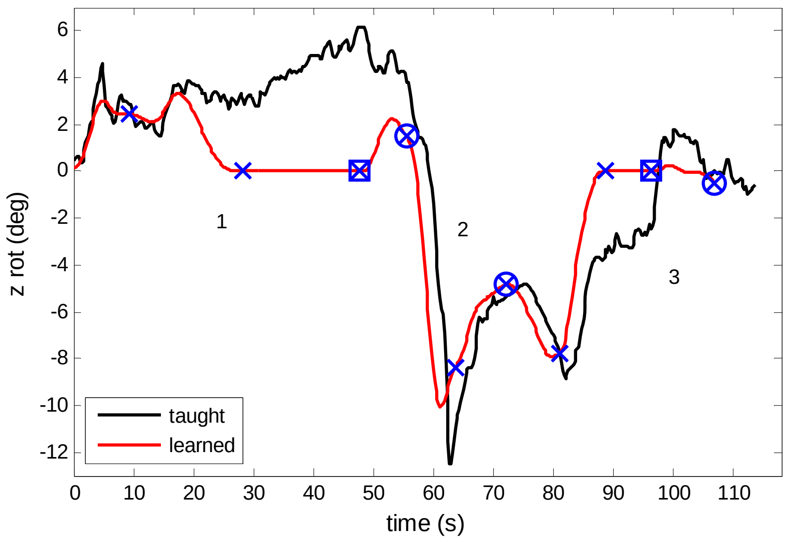
c)


taught
learned
x rot (deg)
time (s)
1
2
3

d)



e)



f)

Figure 6-12: Learned task trajectories and their demonstrated counterparts in the world frame for the task “Pour Liquid”. Primitive subtask (PST) segmentation points (SPs) are indicated by blue circles. The instants that the robot gripper changed its status are indicated by blue squares. The PST task trajectory episodes 1, 2, and 3, were assigned to primitive subtasks: “Pick up Bottle”, “Move this Bottle to Cup”, and “Place this Bottle onto Table”, respectively. Some primitive subtasks were assigned more than one trajectory episode.

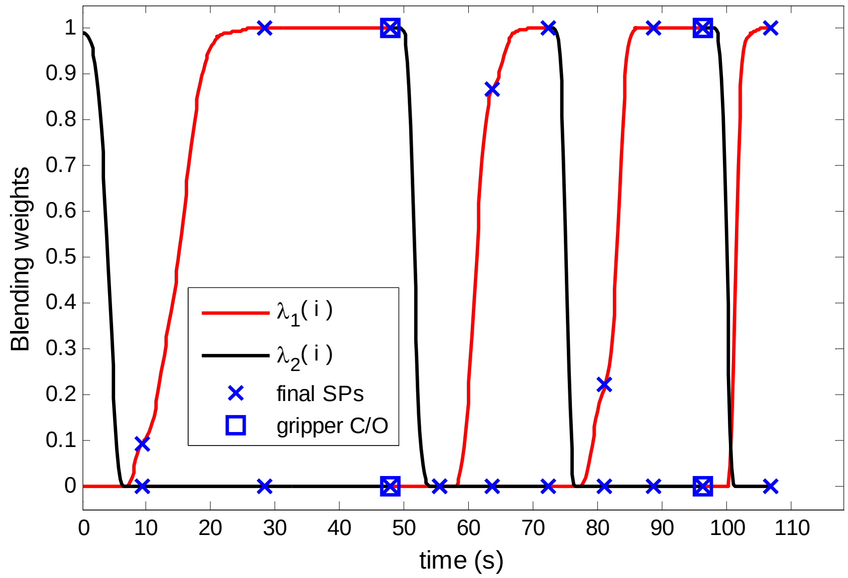
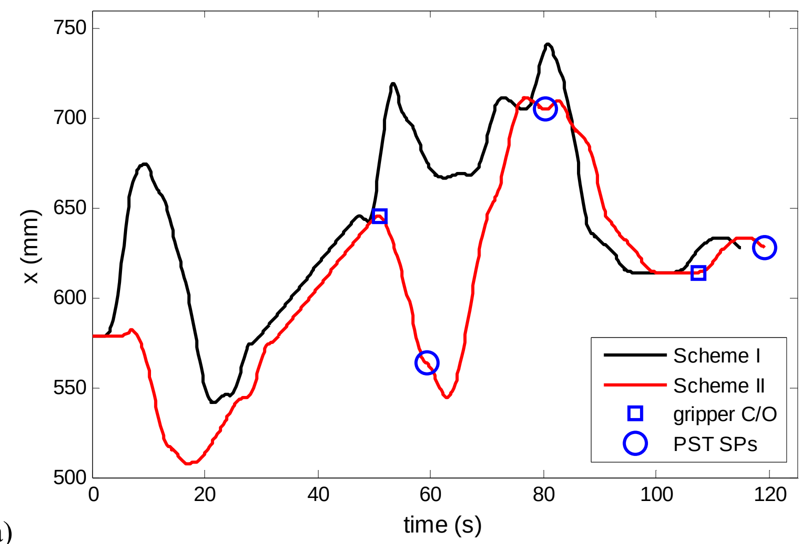


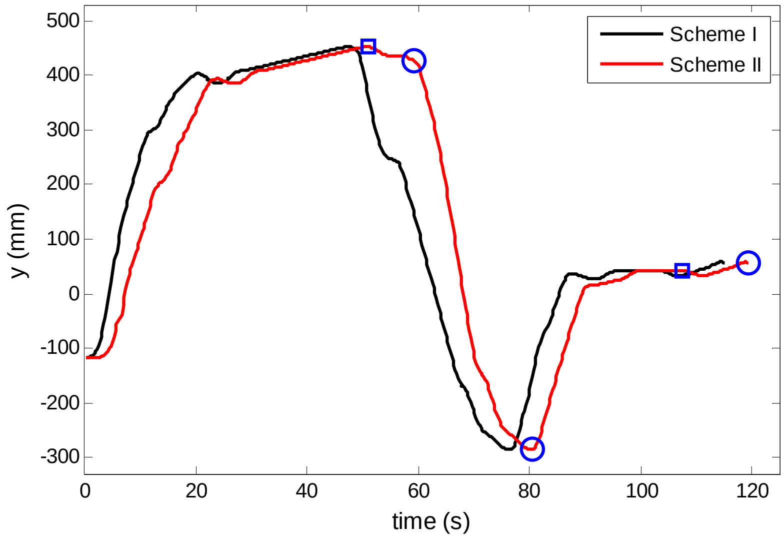
Figure 6-13: Blending weights in Equations (3-19) and (3-20) for the task "Pour Liquid". $\lambda_2(i)$ are the weights of the robot current pose when approaching the first object or the weights of the trajectory generalized in the last object frame when they are blended with the trajectory generalized in the world frame or the trajectory generalized in the object-oriented frame of the object being approached to produce an intermediate trajectory. $\lambda_1(i)$ are the weights of the trajectory generalized in the approaching object frame when it is blended with the intermediate trajectory.

Ten task setups (Table 5-2) were used to test the robot's ability to adapt its learned task to new setups. Blending Scheme II was used for these tests. The robot succeeded in performing the task in all setups, meaning that it did pour some beads into the cup for each setup. However, a rotation of approximately 180 deg by the robot's fourth link (i.e. the forearm link) occurred in Setups 7 and 10 when the robot was approaching the table after the pouring action. The robot motion for the two setups had to be interrupted to prevent possible collision of the stereo cameras mounted on the end-effector with the table. For each of the ten task setups, the relative pose between the robot end-effector and bottle when the bottle was grasped, and the relative pose between the end-effector and cup when the robot started to pour beads from the bottle into the cup were computed. Note that the latter relative pose (between the robot end-effector and cup) was used instead of the relative pose between the bottle and cup since the bottle pose could not be directly measured during the process of pouring, in the current system implementation. The accuracy of the computed relative pose was directly affected by the accuracy of the robot forward kinematics, and the measurement accuracy of the robot stereo-camera vision system. The measurement accuracy of the robot vision system was 2.83 mm with a standard deviation of 1.17 mm, as reported in Section 4.2.4.2. For the ten tests, the differences between the computed (and executed) relative poses and their corresponding learned relative poses were within the system accuracy and precision (less than 0.26 mm in translation and 0.05 deg. in rotation). These excellent results demonstrate that the blending schemes were highly effective in adapting the learned robot-object engagements (i.e. the grasping and pouring process here) to the new task setups.

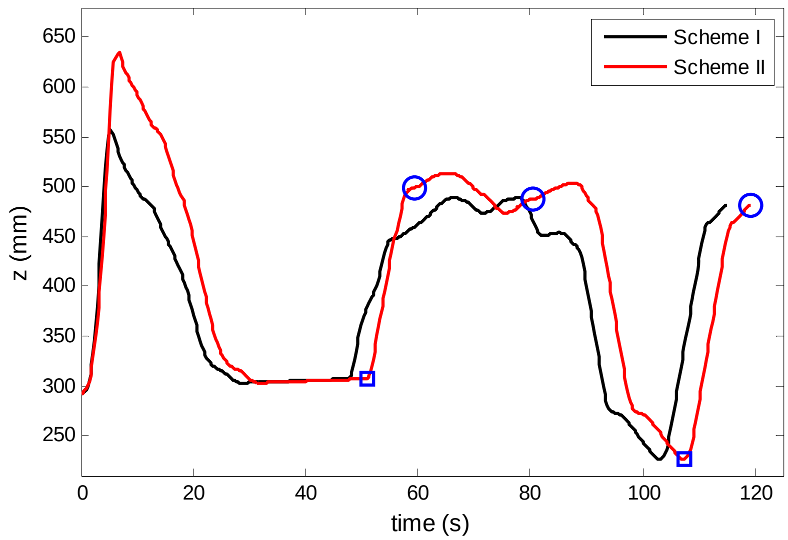
The executed task trajectories for Setup 8 (Table 5-2), generated by Blending Schemes I and II, are shown in Figure 6-14. No changes were made to the task trajectories generated directly by the two blending schemes using Equations (3-19) and (3-20), since every part of the trajectories was reachable for the robot and possible collisions between the robot and table or robot and stereo-camera vision system were not detected along the trajectories. The differences between the trajectories corresponding to the two blending schemes are notable in the *X* direction for the following reason. Both the bottle and cup were placed more than 100 mm in the -*X* direction relative to their locations in the demonstration. The taught task trajectory generalized in the world frame in Blending Scheme I tried to bring the robot back to the object locations in the demonstration. However, the corresponding task trajectory generalized in the object-oriented frames in Blending Scheme II tried to maintain a similar robot-object relationship as taught during the task execution instead of bringing the robot back to the object locations in the demonstration. The speed profile of the executed task trajectory corresponding to Blending Scheme II and the corresponding taught speed profile are illustrated in Figure 6-15. In the critical episodes and trajectory parts adjacent to the critical episodes, the executed trajectory speeds were very close to the taught counterparts. In the trajectory parts that were neither the critical episodes nor the parts adjacent to the critical episodes, the executed trajectory speeds were quite different from the corresponding taught counterparts. These differences were due to the matching relaxation between the execution and taught speeds in the non-critical episodes.



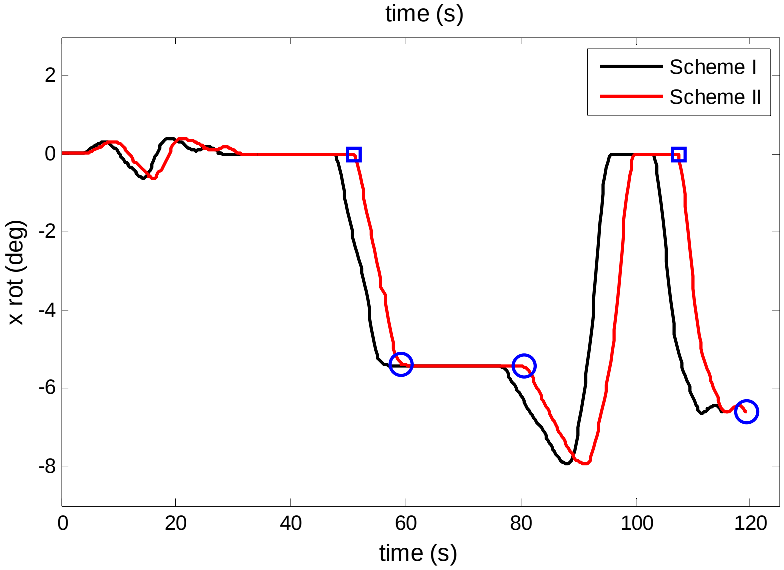
a)


Scheme I
Scheme II
y (mm)
500
400
300
200
100
0
-100
-200
-300
0
20
40
60
80
100
120
time (s)

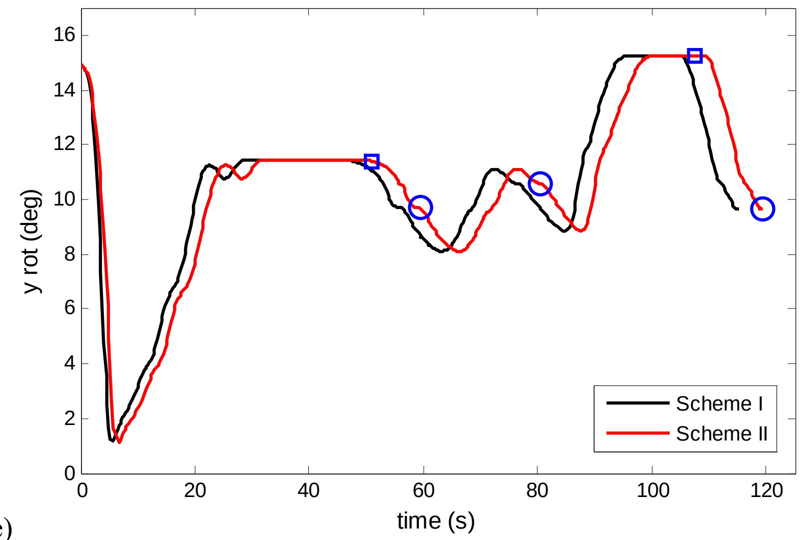
b)


Scheme I
Scheme II
z (mm)
650
600
550
500
450
400
350
300
250
0
20
40
60
80
100
120
time (s)

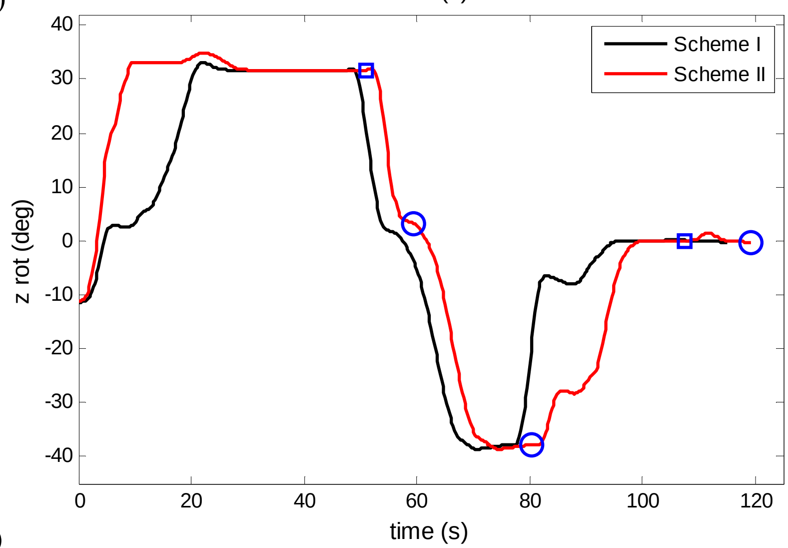
c)


Scheme I
Scheme II
x rot (deg)
2
0
-2
-4
-6
-8
0
20
40
60
80
100
120
time (s)

d)

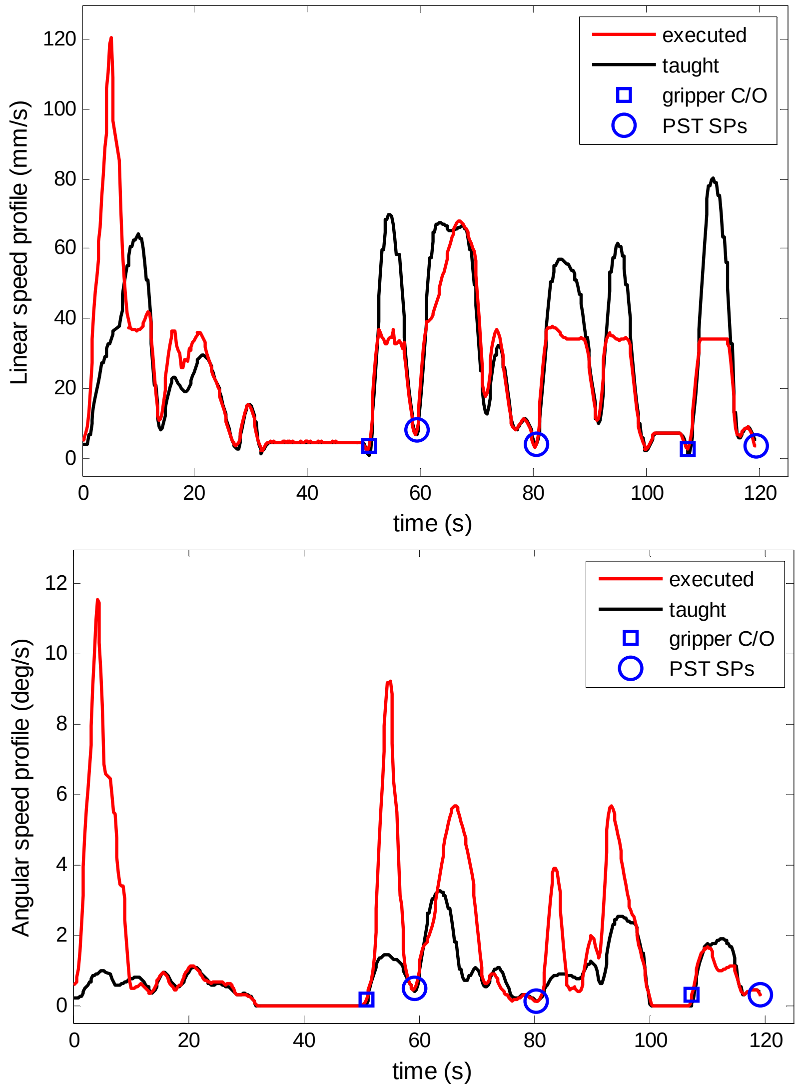


e)



f)

Figure 6-14: Executed task trajectories using the two blending schemes for the task “Pour Liquid”, for Setup 8 in Table 5-2. The black lines represent the executed task trajectories corresponding to Blending Scheme I (Equation (3-19)). The red lines represent the executed trajectories corresponding to Blending Scheme II (Equation (3-20)).



a)

b)

Figure 6-15: Speed profile of the executed task trajectory for the task “Pour Liquid” using Blending Scheme II, for Setup 8 in Table 5-2 and corresponding taught speed profile: a) linear speed profile and b) angular speed profile.

6.3 Results of User Study

In the second series of experiments, the user study, quantitative and qualitative evaluations were collected to reflect how well: a) the developed robot-task teaching and learning system enabled the human subjects to teach two different tasks “Lay out Table” and “Pour Liquid” to the robot; b) the robot responded to the subjects’ requests, instructions, and feedback; c) the robot learned the taught tasks; and d) the robot adapted the learned tasks to new task setups.

The pilot study with the first two subjects, 1 and 2, revealed some need for improvement in training with the Microsoft Speech Recognition. For instance, multiple training sessions were necessary even for native English speakers, and some training passages seemed more effective than others to improve the rate of speech recognition. Some revisions to the expected utterances to be used by the human subjects in providing feedback during the robot's task practice were also made to make the utterances more natural and less cognitively demanding for the subjects. Note that the parameters described Section 5.2.3 were not tuned in the entire user study, including the pilot study.

Generally, the subjects and robot-task teaching and learning system collaborated very well in the processes of task teaching, learning, and practice. Quantitative evaluations were reported for subjects 3, 4, 5, 6, 7, 8, 9 and 10 only since subjects 1 and 2 were used for the pilot study and the related protocol parameters (voice training) and human-robot interaction utterances that were modified would influence the measurements. One additional (eleventh) subject withdrew their participation after introducing the task structure and beginning the task demonstration. Qualitative evaluations were reported for all ten subjects that completed the task teaching. The time used in each robot-task teaching and learning stage and the time for subjects' robot-teleoperation training are presented for the two tasks "Lay out Table" and "Pour Liquid" in Table 6-2 and Table 6-3, respectively. The time used for the overall-task-introduction teaching stage by each subject was normalized by the minimum number of the responses needed for the subject to introduce their pre-analyzed task structure to the robot. The normalized time is shown in parentheses in the two tables beside the time used by each subject for the overall task introduction. The mean time used in each teaching and learning stage except the stage of robot's task practice ranged from 3.87 to 8.32 min for "Lay out Table" and from 1.10 to 4.93 min for "Pour Liquid". The normalized time in the overall task introduction was from 0.14 min to 0.20 min with a mean time of 0.17 min and a standard deviation of 0.02 min over the two tasks.

The vocal interaction efficiency in the overall task introduction, number of vocal subtask-segmentation cues given in the subjects' task demonstrations, and number of feedback cues given by the subjects in the first and second task practice are reported in Table 6-4 and

Table 6-5 for the two tasks "Lay out Table" and "Pour Liquid", respectively. Subjects could ask the robot to practice a task at each speed level (i.e. 50% or 100% of the taught speed) as many times as they wanted, and some subjects had the robot practice more than twice. For the two tasks, the mean vocal interaction efficiency was near 0.9, while the ideal efficiency is 1.0. For the "Lay out Table" task, all subjects demanded the robot to practice twice for them to help the robot refine its task knowledge to a satisfactory level by giving timely feedback on the robot's performance. For the "Pour Liquid" task, Subjects 5, 6 and 10 requested the robot to practice the task three times, while the other subjects asked the robot to practice twice. The average numbers of timely feedback cues given by the subjects were 11.88 in the first task practice and 3.0 in the second practice for the task "Lay out Table", while the corresponding numbers were 8.20 and 7.00 for the task "Pour Liquid". The declining trend of the number of feedback cues given in the two task practices indicates that the improvements of the task performance by the robot were achieved due to refinement based on the timely feedback. The case where Subject 5 issued 36 feedback cues in the first robot task practice for the task "Lay out Table" indicates that the teacher's task expertise could be effectively exploited by the robot learning system to help the robot refine its task knowledge.

The averages and standard deviations of the rating scores from subjective evaluations of the robot-task teaching and learning system, using the questionnaire in Appendix B.5, are given in Table 6-6. The average response from all the subjects were greater than or equal to 4.0 out of 5 except the scores on "System is *intuitive* to use" (3.9) and "Robot responded to your feedback *without delay*" (3.8). All subjects were quite satisfied with the task-partition results presented by the robot in the task practices (4.6 out of 5), and the subjects did not issue any feedback cues in attempt to change the task partitions. However, as reported below in this section, Subject 5 was requested by the robot to help it partition the demonstrated task "Lay out Table" in the task learning stage and the subject did issue multiple task partition cues at that stage.

Table 6-2: Time (in minutes) used for each robot-task teaching and learning stage and the teleoperation training in the user study for the task "Lay out Table".

Subjects	Time used for task: Lay Out Table (min)					Training on tele-operation
	Check & teach background knowledge	Overall task introduction ‡	Demonstration of task in whole	Robot's task learning	Robot's task practice	
3	3.89	6.87 (0.19)	8.24	0.30	17.56	17.40
4	5.87	5.99 (0.17)	9.10	0.58	21.20	19.16
5	2.48	17.36 (0.20)	4.34	25.20*	21.64	0†
6	4.89	5.01 (0.14)	8.44	0.21	19.79	28.31
7	3.91	7.87(0.16)	0^	0^	0^	0^
8	2.91	6.31(0.18)	5.58	0.10	17.88	0†
9	3.33	8.74(0.15)	11.36	0.45	19.13	0
10	4.89	4.26(0.15)	11.17	0.22	19.62	0
Mean	4.02	7.80(0.17)	8.32	3.87	19.54	12.97
SD	1.06	3.86(0.02)	2.43	8.71	1.42	11.22

* The robot task learning system failed to segment the demonstrated task automatically. The robot system then asked the teacher for help, and three replays of the task by the robot were used for the robot to correct the task segmentation; the third replay was for teacher's confirmation on the obtained task segmentation.

^ Corresponding data were not available due to data corruption, and not used in relevant computation.

† Subject 5 and 8 had previous training in robot teleoperation; the 0 min reflects no additional training and was not included in mean and SD.

‡ Time normalized by minimum number of human responses in parentheses

Table 6-3: Time (in minutes) used for each robot-task teaching and learning stage and the teleoperation training in the user study for the task “Pour Liquid”.

Subjects	Time used for task: Pour Liquid (min)					Training on tele-operation
	Check & teach background knowledge	Overall task introduction‡	Demonstration of task in whole	Robot’s task learning	Robot’s task practice	
4	0.09	4.16 (0.19)	6.18	0.11	14.51	0
5	5.05	4.10 (0.19)	6.34	0.30	29.84*	6.64
6	0.12	3.91 (0.18)	5.54	0.23	20.72*	7.94
8	0.13	3.46(0.16)	2.56	0.05	12.87	0†
10	0.13	3.56(0.16)	4.02	0.07	24.39*	0
Mean	1.10	3.84(0.17)	4.93	0.15	20.47	3.65
SD	1.97	0.28(0.01)	1.44	0.09	6.27	3.67

‡ Time normalized by minimum number of human responses in parentheses.

* Subjects 5, 6 and 10 requested the robot to practice the task three times.

† Subject 8 had previous training in robot teleoperation; the 0 min reflects no additional training and it was not included in the mean and SD.

Table 6-4: Efficiency of vocal interaction in the overall task introduction, number of subtask segmentation cues, and the number of feedback cues given by the teacher in the first and second robot task practice, for “Lay out Table”.

Subjects	Task: Lay Out Table			
	Vocal interaction efficiency in overall task introduction	Number of subtask segmentation cues	Number of cues in 1st task practice	Number of cues in 2nd task practice
3	0.86	3	3	5
4	0.90	2	14	2
5	0.86	2	36	8
6	0.97	3	9	1
7	0.86	0	1	3
8	0.90	2	5	2
9	0.94	0	6	3
10	1.00	2	21	0
Mean	0.91	1.75	11.88	3.00
SD	0.05	1.09	10.94	2.35

Table 6-5: Efficiency of vocal interaction in the overall task introduction, number of subtask segmentation cues, and the number of feedback cues given by teacher in the first and second robot's task practice, for "Pour Liquid".

Subjects	Task: Pour Liquid			
	Vocal interaction efficiency in overall task introduction	Number of subtask segmentation cues	Number of cues in 1st task practice	Number of cues in 2nd task practice
4	0.81	2	5	4
5	0.92	0	13	4
6	0.92	0	5	11
8	0.92	1	5	6
10	0.85	2	13	10
Mean	0.88	1.00	8.20	7.00
SD	0.04	0.89	3.93	3.00

The related task structures and task/subtask names were largely determined by the subjects through their task pre-analysis. Their obtained task structures and names were different from each other. For example, subject 2 named the "Lay out Table" task as "Make Table", which consisted of two subtasks: "Put Knife" and "Put Fork". The "Put Knife" subtask was composed of two primitive subtasks "Pick up Knife" and "Put Knife onto Placemat" while The "Put Fork" was further broke down into four child subtasks: "Approach Fork", "Pick Fork", "Move Fork to Placemat" and "Put Fork onto Placemat", as shown in Figure 6-16. Subject 5 decomposed the "Lay out Table" even further, as illustrated in Figure 6-17.

The robot-task teaching and learning system only experienced difficulty in automatically segmenting the task "Layout Table" demonstrated by Subject 5 (Figure 6-17). Specifically, the robot improperly segmented "Move to Spatula" and "Grab Spatula" and was unable to assign a trajectory segment to "Move up Spoon". The robot asked the subject for help in the trajectory segment assignment when the system found $J(\chi^*) < -\kappa_2/2$, where χ^* is the optimal episode-subtask assignment based on Equation (3-8a). It took the robot only two replays (not task practice) of the demonstrated task for the subject to correct the segmentation of the task and a third replay for the subject to confirm that the robot had achieved the right task partitioning. The system enabled the subject to intuitively and effectively issue multiple task partition cues to the robot to correct task partitions when the robot was replaying the demonstrated task. Although subject 5 rated the performance of the robot's automatic task partitioning poorly, the subject highly rated the system on the intuitiveness, effectiveness, and response to feedback overall and without delay.

Table 6-6: Subjective evaluations of the robot-task teaching and learning system based on all tasks taught. All ten subjects performed the task "Lay out Table" or similar task name, Subjects 4, 5, 6, 8 and 10 also performed task "Pour Liquid". A scale of 1 (do not agree) to 5 (strongly agree) was used.

Questionnaire Statements	**Average response**	**Standard Deviation**
Provided *training* is adequate	4.9	0.30
System is *intuitive* to use	3.9	0.54
System is *effective* overall	4.2	0.40
Demonstrated tasks were *partitioned* as expected	4.6	0.92
Robot responded well to your *feedback* overall	4.1*	0.74*
Robot responded to your feedback *without delay*	3.8	0.87
Robot responded to your feedback with *appropriate magnitude* (e.g. move distance, change speed)	4.0	1.10
Robot responded to your feedback with motion in the *direction* you expected	4.2	0.60
Given vocal *utterances* for your feedback were *intuitive* to you	4.4	0.49
Robot refined its task knowledge and applied the new modification later on during a subsequent practice of the task	4.4	1.02
System adapted its learned task to new setups well	4.6	0.49
Used the system without frustration overall	4.0	0.63

*Subject 1 did not respond to this statement

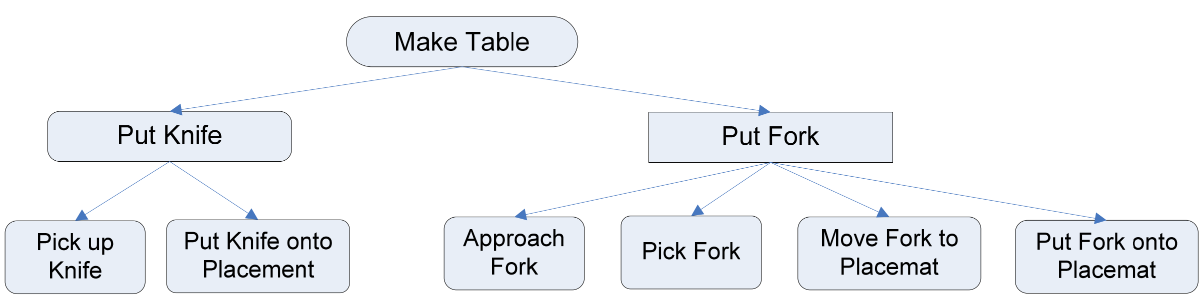


Figure 6-16: The task structure of task "Make Table", generated by Subject 2 through pre-analysis.

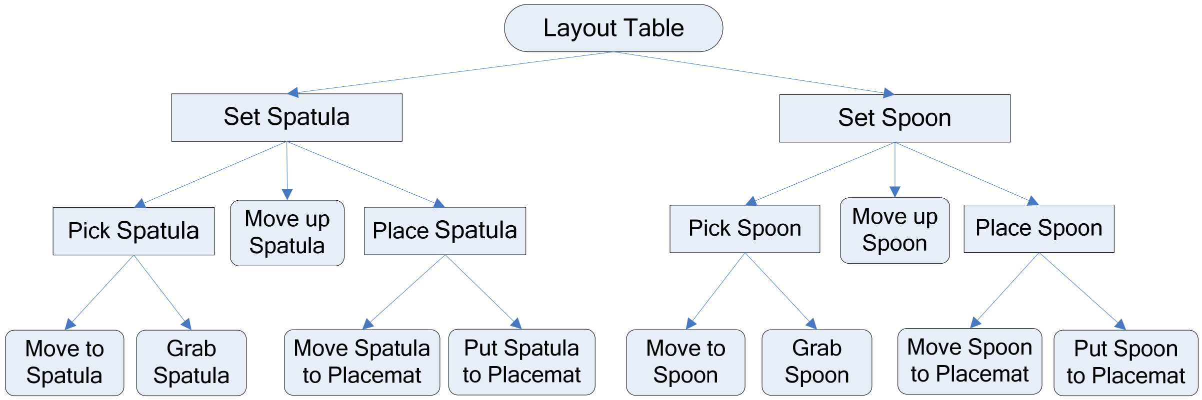


Figure 6-17: The task structure of task "Layout Table", generated by Subject 5 through pre-analysis.

Chapter 7
Discussion

7.1 Discussion of Experimental Results

Two series of experiments were carried out to test how well the developed system enabled: 1) general users to teach tasks to the robot; 2) the robot to learn the taught tasks; and 3) the robot to adapt its learned tasks to new task setups. The experimental results showed that the proposed human-inspired robot-task teaching and learning approach was practical and effective for the human subjects to teach tasks to the robot. The newly developed system enabled the robot to learn the taught tasks, refine its task knowledge based on the feedback given by the teachers, and adapt its learned tasks to new setups. This chapter discusses the instructive and hand-to-hand task teaching, robust task learning by the robot, and robot adaptation of the learned task to new task setups. Other important findings from the user study, not described earlier, are discussed separately.

7.1.1 Instructive and Hand-To-Hand Task Teaching Method

The system was developed to have the teacher involved in every stage of the robot-task teaching and learning. In the user study, these involvements for the human subjects proved to be natural and effective. The subjective results in the user study (Table 6-6) showed that the overall system was intuitive (3.9 out of 5) and effective (4.2) for the subjects to teach tasks to the robot. Teaching tasks to the robot using this system was without frustration (4.0) for the subjects. The study results also demonstrated the effectiveness of the system in permitting the teacher to teach the robot a task. The efficiency to introduce the task structure was nearly 90% and the mean time used for each teaching stage was less than 9 minutes. In this section, these evaluations are further discussed in the context of each teaching and learning stage.

7.1.1.1 Checking and Teaching Needed Background Knowledge

The stage of checking and teaching the robot's needed background knowledge for the task to be taught was a novel method. This stage was developed and included in the robot task teaching for a number of purposes that were beneficial to both the robot and teacher. First, it aided the teacher in understanding what objects and tasks/subtasks the robot had learned, including object names, colour, and task-relevant attributes, information about whether the robot could identify these objects and the task/subtask names, numbers of child subtasks, involved objects and robot hand actions. Secondly, the teacher was able to prepare the robot to be ready to learn a new task from their teaching, by teaching the needed background knowledge before focusing on teaching the details of the task. The teacher could modify the information about existing objects and tasks/subtasks, and teach the names of new objects by moving the robot end-effector over them. This was done by the student investigator in the experiment for "Lay out Table". The user study showed that the subjects were indeed allowed to naturally inquire about the robot's knowledge and teach the information that was lacking or modify the existing information if they chose, by using some utterances as given in Appendix A.1. The information the robot responded with was understandable to the human teachers. In addition, the process of changing or teaching the robot's relevant knowledge was transparent to the teacher. For example, the robot would respond with "the color of the object has just been changed from *red* to *blue*" if the original object color was *red* and the teacher changed it to *blue* using a utterance such as "its color should be *blue*".

7.1.1.2 Overall Task Introduction

The human teachers in all the experiments were required to give an introduction of the task to be taught to the robot. Essentially, this involved teaching the hierarchical task structure to the robot. The robot controlled the dialogue flow by asking questions and having the teacher respond. An example of a typical dialogue for overall introduction of the task is found in Appendix A.2. For the task (root) node and each subtask, the robot requested the task/subtask name and the number of child subtasks it had. The robot also asked for information regarding the involved robot hand actions for each primitive subtask. In addition, the teacher was given the ability to navigate through the task structure using utterances such as "go to its last/previous/next/child/parent subtask".

In the user study, it was found out that the voice dialogue for the overall task introduction was natural to the subjects. Also, the human-robot dialogue was efficient in terms of the ratio of the minimum number of needed teacher responses to the actual number of recognized responses to introduce the task structure. The mean of the ratios was high, at or near 0.9 (Table 6-4 and

Table 6-5) (the ideal value is 1). It is important to note that the human subjects were not provided any opportunity to practice any of the utterances used for the human-robot interaction in the entire process of robot-task teaching and learning.

The system was able to help the teacher focus on the subtask to be introduced by reminding the teacher using utterances such as "*Now, we are to build the* ***first step*** *of the subtask:* ***lay out knife****. Please tell me the* ***name of this subtask***".

The dialogue between teachers and the robot was generally very robust to prevent errors from being uncorrected. A confirmation from the teacher was required for all teacher responses to robot requests (except confirmation responses). This allowed the teacher to correct any of their responses and any misunderstanding by the robot. Errors would therefore not be carried forward in the teaching and learning process. However, this robustness to prevent errors was achieved at some cost. At least four responses for each non-primitive subtask and six responses for each primitive subtask were required from the teacher. This situation might be improved by asking for decreasingly fewer confirmation requests if the robot receives a high percentage of positive confirmations from the teacher at the beginning of the human-robot vocal interaction.

7.1.1.3 Hand-to-Hand Task Demonstration

Step-by-step teaching was not used in the user study but was adopted by the student investigator for the task "Lay out Table". The investigator was able to start to teach a subtask by saying to the robot “the next subtask is *subtask-name*”. The robot would then respond with the introduced knowledge of this subtask and whether this subtask had already been learned if the robot successfully recognized the given subtask as one of the introduced subtasks. The teacher helped the robot perceive important state changes - the transitions of the child subtasks for the current subtask. This was done while demonstrating the subtask, either implicitly by naturally slowing down at the transitions, or explicitly by giving some vocal subtask-segmentation utterances such as “next step” at the transitions. The outcomes of the step-by-step teaching stage (subtask trajectories, segmentation cues, and robot hand actions) were used in the task learning stage to assist the robot in the coarse assignment of the episodes of the in-whole demonstrated task trajectory to the subtasks taught in the step-by-step stage.

In the current implementation, the hand-to-hand task demonstration was carried out by the teacher teleoperating the robot using marker-based vision techniques to track their hand motion. The vision-based teleoperation required training by the subjects (mean time of 12.97 min) before they were comfortable to demonstrate tasks by teleoperation. Although this implementation proved to be feasible, the use of vision-based tracking caused one major problem to use the robot-task teaching and learning system. All three hand markers had be visible by at least two of the three cameras during the teleoperation. This constrained the motion of the teacher to a confined area and restricted some of the hand motion to permit free six DOF hand motion. This was particularly a problem for the “Pour Liquid” task where the pouring action had to be done autonomously by the robot following a “start to pour to right/left” command from the teacher, rather than allowing the teacher to define more personally the pouring trajectory of the end-effector. The limited motion of the teacher also prevented them from having a better direct view of the robot and the robot’s workspace. Although the hand motion was sufficient to teleoperate a six-axis robot, pose information of the arm, body and head may be needed to teleoperate a robot with more than six axes.

In the user study, the in-whole task demonstration stage confirmed that the teaching process had several advantages. It was transparent to the teacher when the robot would respond “yes” whenever it recognized a vocal subtask-segmentation cue. It was helpful to the teacher when the robot reminded them about the relevant procedures. If the demonstration was interrupted due to problems in hand-motion tracking or if the teacher considered the demonstration to be not satisfactory, it was convenient that the teacher could continue or start over the demonstration, or even re-start the entire robot-task teaching and learning system and directly go to the in-whole demonstration. In addition, the robot instructed the teacher to place the objects in place and then notify the robot with “it is ready” to start a new exploration of the robot workspace at the beginning of each demonstration.

7.1.1.4 Teaching in Task-Learning Stage

In the learning stage, the robot might request the teacher for help to partition the taught task if it found that it had failed to assign the obtained episodes of the in-whole demonstrated task trajectories to the introduced primitive subtasks. In all experiments, only Subject 5 in the user study encountered this request initiated by the robot in the task learning stage. The process for the subject to help the robot assign the task trajectory to the primitive subtasks was confirmed to be effective, transparent, and flexible in its use. While replaying the demonstrated task trajectory, the robot revealed its currently obtained assignment at the beginning of each primitive subtask using utterances such as “the next task trajectory segment belongs to *primitive-subtask-name*”. If a primitive subtask had not been assigned any trajectory segments, the robot would notify the teacher at the beginning of the next primitive subtask. The teacher could easily request the robot to pause, move forward and backward, using utterances such as “pause here” and “segmentation move forward/backward”. If the teacher wanted to change the assignment of a primitive subtask, they could use utterances such as “the LAST/NEXT trajectory segment should be assigned to *primitive-subtask-name*”. The robot would then determine the beginning and end of the trajectory segment based on the time the instruction was given, and then assign the segment to the specified primitive subtask accordingly.

7.1.1.5 Teaching in Robot-Task Practice

During the robot-task practice, the teacher was allowed to modify the learned task trajectory by giving timely feedback for three translations, three rotations, and one speed. The teacher could also easily modify the robot’s task-partition results during the task practice. The human subjects highly rated the process of helping the robot refine its learned task knowledge as follows: robot responds well to the subject’s feedback overall (4.1), without delay (3.8), with appropriate magnitude (e.g. distance moved, speed changed) (4.0), and with motion in the direction as expected (4.2) (Table 6-6). The subjects also highly rated the intuitiveness of the provided vocal utterances for feedback to the robot (4.4), and the robot’s refinements according to given feedback and application of the new refinements later on during a subsequent task practice (4.4). The demonstration of the task by the teacher did not have to be perfect as the teacher was given the ability to refine the learned task during the task practice. Two main factors also made it impossible for the teacher to achieve perfect object grasping and releasing in the task demonstration: 1) the width of the robot gripper (68 mm) was only slightly larger than the thickness of the involved objects in “Lay out Table” (50 mm) and the bottle in “Pour Liquid” (57 mm); and 2) the teacher’s view of the robot and its workspace was not ideal. However, the intuitive feedback that the subjects in the user study could give, aided them in making up for these limitations adequately.

The teacher was not able to modify the task structure or the task/subtask names in the robot-task practice stage. Instead, in the current system implementation, the task structure and subtask names could be taught and modified by the teacher by having the robot go back to the overall-task-introduction stage. The subtask names could also be changed in the step-by-step teaching stage.

7.1.2 Robot Task Learning

The system enabled the robot to effectively learn the taught tasks. This section discusses the top-down and bottom-up task-learning approaches, refinement of task knowledge during robot task practice, object modeling, error detection and recovery, and effectiveness of the involved task-learning parameters.

7.1.2.1 Learning Tasks with Both Top-Down and Bottom-Up Approaches

For the top-down approach, the robot was able to successfully build an abstract task structure through the overall introduction by the teacher. By analyzing the syntax of the introduced task/subtask names, the robot successfully extracted the task/subtask actions (the verbs in the names), directional prepositions, and the involved objects for each subtask. Currently, there are several limitations of the syntax of task/subtask names, specifically, the names of primitive subtasks. The task and subtask names must include one verb and at least one object. The objects indicated in the primitive subtask names must be recognizable by the robot, i.e. the robot had to be able to extract a set of visual features of each object that uniquely differentiated the object from other objects. In other words, the robot would not be capable of distinguishing two objects with similar visual features but with different names. For a given object, only one name should be given. Although the articles, pronouns, and modifiers before an object name, such "*my* telephone" and "*three* apples", can be recognized and extracted, currently, this information was not used. For example, the subtask name "move bottle to cup" was treated the same as "move this bottle to the cup". In addition, the robot was not able to perform analysis of semantics on the subtask names. If the robot would have the ability to do semantics analysis, once the robot would be instructed a subtask name, it would immediately know some task-relevant implications from the name. For example, given the subtask name "grasp knife", the robot would immediately know that a Close-Hand action would likely be involved in this subtask based on the meaning of the verb "grasp".

The robot also successfully learned the taught tasks in a bottom-up approach by analyzing its sensor data such as the demonstrated task trajectories and robot-hand status, as described in Section 3.3. The voting algorithm developed for segmenting the demonstrated task trajectories was able to utilize different types of signals. For each signal, the ways to compute possible trajectory segmentation points were very dependent on the nature of the signal. For discrete signals, such as the status changes of the gripper and the instants the vocal sub-task segmentation cues were given, there were no difficulties. However, it was more complicated to find the possible segmentation candidates by analyzing the MSV. The approaches to segment the trajectory based on the MSV, presented in Peters et al. (2003) and Lieberman (2004), might discard some valuable parts of the trajectory due to their very low speeds. The trajectory parts with low MSV values were considered neither meaningful nor important by the authors. In this research, every part of the demonstrated task trajectory is considered potentially important and should not be discarded. For example, the teacher usually would slow down their motion in the part of the trajectory between an object's docking pose and its grasping pose. The local MSV minima, whose values were below the given threshold, were selected as possible segmentation point candidates. The MSV candidates adjacent to local MSV maxima were treated more favorably than other MSV candidates. This differentiation was evident by the different heights of the local peaks of the votes by the MSV as shown in Figure 6-3 and Figure 6-11. For each signal, its relative reliability for segmentation (its voting weight, w_k) and scale factors for relative proximity (c_k) of its segmentation candidates to likely partition points had to be pre-determined in order to compute the overall trajectory-segmentation votes in Equations (3-4) and (3-3). In the currently used system, the votes by the gripper status changes were deemed to be the most reliable ones, while the votes by the MSV segmentation candidates were the least dependable. For the relative proximity (c_k), the changes of robot gripper status were expected to be very close to some true trajectory segmentation points (SPs), while the instants that the vocal segmentation cues were given might be somewhat distant from the true SPs. The degrees of reliability and proximity of each chosen signal were evident by the maximum peak values of the votes and the widths of the votes' local bell shapes, respectively (Figure 6-3 and Figure 6-11).

It was important to find the proper trajectory segments from an object's docking pose to its grasping pose or from a docking pose of a target to its corresponding releasing pose. If a docking pose was spatially far from its corresponding grasping pose or releasing pose, the corresponding critical episode would be very long spatially. This might cause problems of reachability by the robot in the task practice or execution, since the critical episode must be fully followed when the robot adapts the learned task to new task setups. This problem occurred with Subject 3 in the user study, where the critical episode for grasping the knife was rather long spatially. The robot warned the subject that the knife was unreachable several times when the robot tried to instantiate the task based on the actual new task setup. The unreachability problem resulted from parts of the critical episode being unreachable.

In the robot task learning, it was very important to connect the task knowledge learned in the top-down and bottom-up approaches, respectively. In other words, it needed to associate the learned abstract task structure with the obtained robot-sensor data. This was done through a probabilistic optimization process in Equation (3-8a) to assign the segmented task trajectory episodes to the introduced primitive subtasks. This process utilized the knowledge learned in the top-down approach (information of the subtask actions, directional prepositions, involved objects, and specified robot gripper actions), and the knowledge learned in the bottom-up approach (such as the observed status changes of the robot gripper, vocal segmentation cues and vocal command for the robot to pour beads, and characteristics of the trajectory episodes). Although the degrees of matching between the characteristics of a trajectory episode (related to speed and direction) and the action and directional preposition extracted from a subtask name (e.g "move up" from "move up knife") could be statically computed through analysis of the robot's previous experience, the relationship between characteristics of episodes and a very limited number of directional prepositions was currently pre-determined due to the limited previous experience of the robot. The adopted coarse-to-fine optimal episode-subtask assignment process was not very time consuming as indicated by the maximum 0.58 min used for the robot task learning stage in the user study (Table 6-2 and Table 6-3), except for the 25.2 min used in this stage for Subject 5 to help the robot partition the taught task "Lay out Table". The user subjects seemed to be quite satisfied with the episode-subtask assignments automatically obtained by the optimization process, as indicated by the score of 4.6 out of 5 for the statement "Demonstrated tasks were *partitioned* as expected" (Table 6-6). For the task "Lay out Table" taught by Subject 5, the robot recognized that it failed the episode-subtask assignment, and then initiated a process to ask the subject for help (Section 6.3). The task structure given by the subject was rather complicated and consisted of ten primitive subtasks, while just two subtask-segmentation cues were given by the subject during the task demonstration. The process of helping the robot partition the task was intuitive and effective for the subject. Even when the robot had no difficulty in partitioning a task, the teacher was always allowed to correct the partitioning during the robot's task practice. The robot was also able to infer names for previously unnamed objects that were involved in a task based on episode-subtask assignments.

For the task "Lay out Table" taught by the investigator, the results from the step-by-step teaching were used to coarsely divide the primitive subtask and trajectory episodes into groups, as described in Section 6.2.1. Although the coarse groups were different from those generated by comparing the robot hand actions that occurred in relevant episodes to those specified in primitive subtasks, optimization of the two different coarse assignments produced identical final optimal assignments. When the vocal subtask-segmentation cues were not taken into account, the generated episodes from the segmentation of the in-whole demonstrated task trajectory were very similar to the results when the cues were considered, except that one new episode was produced, as shown in Figure 6-6. The assignments of the episodes to the primitive subtasks without considering the vocal cues and the step-by-step teaching results were somewhat different from the assignments with that information considered (Figure 6-4c). The assignments to primitive subtasks "Move Knife down to Placemat" and "Place Knife onto Placemat" were affected. The assignment to "Move Knife down to Placemat" seemed more reasonable when the vocal cues and the step-by-step teaching results were excluded. This indicates that the information from the teacher is not always optimal (Coates et al., 2008). Although it was not necessary for the vocal subtask-segmentation cues to be given by the teacher at each subtask transition, the more cues given, the more the system could take advantage of the teacher's task expertise.

7.1.2.2 Effective Refinement of Robot Task Knowledge during Robot Task Practice

The robot refined its task knowledge instantly when a feedback cue was recognized during its task practice, and the robot reflected the refinement in the next sampling period. The results from both series of experiments indicate that the refinement process was quite effective. The human subjects rated highly the statement "Robot refined its task knowledge and applied the new modification later on during a subsequent practice of the task" (4.4 out of 5). The refinement increments on the translations, rotations and speed of the robot trajectory based on teacher feedback were constant within and across all experiments. A more effective method of refinement might be to adjust the motion increment based on the number of the same type of feedback cues received by the robot within a specified time period. In addition, the directions of the robot motion could not be directly changed through a given feedback. For example, the teacher was not allowed to request the robot to move right when the robot was moving to the left.

7.1.2.3 Object Modeling

An object model was developed to include information regarding an object's name, contour-based shape features, surface color, functional roles, paired docking poses and grasping poses. Although this information was adequate for the current experiments, a more comprehensive task-oriented object modeling would be needed for more general object-manipulation tasks. The task-oriented object model would consist of information about the object's 3D shape, surface-related features, functional parts (such as the mouth and handle of a teapot); typical places it would be found; and task-related functional roles, object states, and manipulation actions that could be applied to the object.

7.1.2.4 Error Detection and Recovery in Task Teaching and Learning Processes

In all the experiments, except "Lay out Table" taught by Subject 5 (Figure 6-17), the system was able to automatically learn the taught task in the learning stage. The system was designed to not rely only on a "single shot" (single opportunity) for the teacher to successfully teach the task and for the robot to learn the taught task. Instead, in the experiments, the robot system could early recognize and expose problems in the process of teaching and learning and could provide natural and effective means for the teacher to help the robot correct and improve its learned task knowledge.

In the overall-task introduction stage, the robot spoke out its understanding of the teacher's vocal response to its request, and asked the teacher to confirm its understanding. This mechanism helped to expose speech recognition errors made by the robot or mistakes made by the teacher. This gave the teacher a chance to correct the errors and mistakes at the task-representation level. In the step-by-step teaching and in-whole task demonstration stages, the robot's utterance "yes" to each recognized vocal subtask-segmentation cue helped the teacher deliver the vocal cues with confidence. In the learning stage, the robot was able to recognize its failure of its automatic task partition, and initiate a process for the teacher to help. In the task practice stage, when instantiating a learned task, if an involved object were not present, the robot would tell the teacher which object was not present in its workspace. If an object were unreachable, the robot would tell the teacher which object was unreachable. The robot would then re-instantiate the task after relevant corrections by the teacher. When the robot would practice the task with 50% of the taught speed, the robot would state its obtained task partition results. The teacher could easily give timely feedback regarding the task partition and 6-DOF robot-task trajectory. The robot's immediate responses to the feedback helped the teacher correct relevant errors and provide a useful amount of refinement to the task trajectory. However, more work needs to be done to detect and recover relevant errors at different levels in the current system.

7.1.2.5 Effectiveness of Task-Learning Parameters

For the parameters used in the experiments as described in Section 5.2.3, although some heuristic tuning was done on these parameters during the system-building phase where only the student investigator was involved, the same values of all parameters were used for both tasks, all involved task setups, and all human users in both the RTLA and user study experiments. The positive experimental results showed that the single set of parameter values were effective for all of these cases, and the robot system was robust to differences in the two tasks, all task setups, and the subjects' task teaching (this includes the entire teaching and learning process: hierarchical task structure, demonstrated motion - trajectories and speeds, and human-robot interaction). Except for the gripper length used for η in Equation (3-1) and δ_A and δ_N in Equation (3-5), the parameters for task learning were not explicitly dependent on the embodiment of the robot; they were set in the task space. These parameters are therefore expected to be applicable to other robots for learning an object-manipulation task. Task instantiation and execution do depend on the specific embodiment of a robot. Related parameters (for example, reachable distances) would need to be adjusted when importing the current system to a different robot.

7.1.3 Adaptation of Learned Task to New Task Setups

The results from the robot-task learning and adaptation (RTLA) experiments showed that the robot had a very good ability to adapt its learned task to new setups, including different poses of objects, targets, and the robot end-effector starting point. Some of the setups were considerably different from those in the task demonstration, as indicated in Table 5-1 and Table 5-2. In the user study, the subjects also rated highly the statement on "System adapted its learned task to new setups well" (4.6 out of 5). The robot's task-adaptation ability was enabled by generalization of the taught task trajectory in different frames, use of the blending schemes, and consideration of robot starting points.

As long as the related critical trajectory episodes generated for the given task setup could be fully followed, the robot was allowed to interpolate and modify the planned task path directly as generated using Equation (3-19) or (3-20). These modifications and interpolations were made based on the ability of the robot to reach each of the computed task path points, and safety considerations such as possible collision between the robot and table and between the fourth robot link (forearm link) and the stereo cameras mounted on the robot end-effector. The interpolations and modifications are shown in the task trajectories in Figure 6-7 (Section 6.2.1) where modifications due to robot unreachability and safety considerations were elaborated. When an object or a part of the related critical trajectory episode was unreachable by the robot, the robot would tell the teacher which object needed to be adjusted.

Based on the results of the RTLA experiments for the task "Layout Table" (Table 6-1), the differences of alignment of the knife and fork on the placemat compared to the taught alignment, were considerably small considering the experimental environment. Only one translation difference was greater than 10 mm and the maximum orientation difference was 3 deg. The mean translation difference was 3.8 mm with a standard deviation of 3.9 mm while the mean rotation difference was 0.6 deg with a standard deviation of 1.5 deg. The two robot gripper fingers are padded with a 5 mm thick layer of foam, and in the experiments, the robot grasped objects (knife, fork, spoon, and spatula) by grasping 50 mm thick foam blocks on which the objects were fixed. With the simplified Vision Agent, the poses of these objects, specifically the Z (vertical) component, could not be accurately determined using the current contour-based method. The main reason is that the object contours were different when the object was viewed by the cameras from different perspectives. The variation in the contours caused variation in the contour centre and main contour directions. Objects might therefore have been grasped from different positions than in the demonstration. The use of objects on foam blocks made fine manipulation of objects difficult, partly due to unexpected reaction forces between the foam blocks and other objects, gripper, and placemat, and from the object being accidentally moved when the robot was retreating after placement. In the task "Pour Liquid", the locations and orientations of the bottle and cup could be measured accurately. The robot-gripper pose with respect to the bottle when it was grasped and the pose with respect to the cup when the robot started to pour beads into the cup were almost identical to the poses used during the teaching.

The blending weights were found to play a crucial role in determining the shapes of the actual executed task paths. The weights would determine when a trajectory generated in a specific frame should be dominant. This was particularly important when the task setup was very different from the one in the demonstration. If the trajectories produced in the object frames were allowed to dominate too early, the overall quality of the taught task path might not be preserved. For example, some small differences of the object's orientations might cause considerable difference of the overall planned task path. Alternatively, if the trajectories produced in the object frames dominated very late, the related transitions of the task path could be very sharp. As shown in Figure 6-7a, the sharp transition (solid red line) was evident along the *X* direction when the robot retreated back to its initial position at approximately 175 s, after placing the object at approximately 165 s. The statistical approach of analyzing multiple task demonstrations used by Lieberman (2004) may have been useful to determine blending weights between generalized task trajectories. However, in this research, it was desired to permit only a single demonstration of a task by the teacher.

In the RTLA experiments for task "Layout Table", the executed task trajectory generated by Blending Scheme I (Figure 6-7b) for task setup 7 in Table 5-1 oscillated noticeably in the *Y* direction (in the time periods 3-10 s, 36-59 s, 85-95 s, 125-135 s, and 162-170 s) because the objects (knife and fork) and target had switched sides (in the *Y* direction) and their poses were considerably different from those in the demonstration. These oscillations, which occurred when the robot approached an object or target, resulted because the taught task path generalized in the world frame tried to move the robot end-effector back to where the object or target was during the demonstration. Due to the same side-switch effect of the objects and targets, the actual object-oriented frames (corresponding to the approximate directions from the placemat to the knife and fork) were almost 180° (about the *Z* axis of the world frame) different from those in the demonstration. The trajectories generated in the object-oriented frames were nearly opposite to those in the demonstration (Figure 6-7b and Figure 6-4b). This caused parts of the task paths directly produced by Blending Scheme II (planned path indicated by the red dotted lines in the figures) to need modification for reachability and safety. If the robot base were mobile, these problems could be eliminated, and the planned path by Blending Scheme II would prevail over Blending Scheme I, since it could greatly maintain the relative poses of the robot with respect to the objects and targets as taught.

The two blending schemes themselves do not depend on the embodiment of the robot. However, the robot embodiment is the key factor in the subsequent processes of checking reachability of the task trajectory planned by the two schemes and re-planning of the unreachable trajectory segments. Changes of robot embodiment will affect these processes.

For instance, if a two figured robot hand is replaced by a three fingered hand, the related robot hand primitive skills must be updated. If the relevant primitive grasping and releasing skills are complicated, they should be pre-built. On the other hand, simple skills can be learned from a skill demonstration by a general user. There is no need for the robot to learn again the whole task that involves these skills. The learned task knowledge except for the related skills should be reused.

The executed speed profiles in the critical task trajectory episodes (i.e. episodes right before gripper close-open hand actions) were very close to those in the demonstration, as shown in Figure 6-8 and Figure 6-15. The robot was required to follow the demonstrated task speed with higher fidelity when it was engaging the objects. However, it was not necessary to always keep similar motion speeds as taught since the executed trajectories might greatly differ from those taught (due to different task setups) and the period for the robot to complete the task should be kept somewhat similar to that in the demonstration. With the consideration of this, the task motions in the non-critical episodes were allowed to be different from their taught counterparts. This gave the robot flexibility in its motion speed when it was far from engaged objects.

The determination of optimal subtask-execution order based on subtask dependencies and constraints in the given task setup has not been included in the Task-Performing Agent of the current system, partially due to the limited capacity of the Vision Agent. In the current implementation, the robot executed the subtasks of its learned tasks in the same order they were taught. In addition, the robot was not able to detect the occurrence of object occlusion or estimate potential collisions between the robot and objects or obstacles in the task execution.

In addition, the eye-in-hand camera configuration (mounting the stereo cameras on the robot end-effector) made it very difficult for the robot to perceive objects in its workspace during its task execution. Instead, an active exploration of the workspace was executed at the beginning of each task demonstration, practice, and execution. Therefore, the current robot system was not capable of handling dynamic environments where the objects and targets might not be stationary. Task executions were performed with the objects and targets at the same positions as they were in the exploration.

The knowledge representation of a learned task in this research makes it possible to use a learned task as a template to achieve other similar tasks and subtasks in the future. For example, after having learned a task to lay out a knife on a placemat, the robot may be ready to lay out a spoon on the placemat as long as the robot knows the grasping pose of the spoon and location on the placemat where the spoon should be placed. Here, the placement of the spoon at the correct target location is the task goal needed to be achieved. The knowledge of picking up and placing the knife may also be reused in other tasks such as “Clean the Table” and “Put Dishes into a Dishwasher”.

7.1.4 Other Important Findings of User Study

Overall, the subjective evaluations on the robot-task teaching and learning system were rather positive, as discussed in Section 7.1.1. There were several other interesting findings from the user study that might be useful for improving the presented robot-task teaching and learning method, further development of the agent-based system, and further user testing of the system.

There was no requirement that the subjects have any more knowledge than having expertise at the task-level for the task to be taught. The developed method and system did not require subjects to have robot knowledge and computer skills beyond being able to complete the voice recognition training. For the ten human participants, only one subject was familiar with the proposed robot-task teaching and learning method. Only two subjects had previous experience in robot teleoperation using natural hand motion. All subjects had little or very limited experience in human-machine interaction via voice and programming a robot.

The training provided for the subjects was considered very adequate (4.9 out of 5 as shown in Table 6-6). The training phase for the subjects (not including teleoperation practice) required approximately 65 min: 30 min for Microsoft Speech Recognition Training, 5 min for quick browsing through the provided vocal human-robot interaction scripts (Appendix A.1), and 30 min for watching a demonstration video of the use of the robot-task teaching and learning system to teach a task to the robot. When watching the video, the subjects were provided with the task structure introduced in the video as depicted in Appendix A.2. The training time could be halved if the subject used the Microsoft speech recognition system previously, and their related training profile was available for reuse in the study. There was no dedicated training effort to educate the human subjects to pre-analyze the task to be taught, such as the breaking down of the task into subtasks, and specification of the involved robot hand actions for each primitive subtask. The subjects may have obtained a sense how to break down a task from the presented example task structure (Appendix A.2). There was no difficulty observed for human subjects to break down the tasks into their own hierarchical structures. The only problem for some subjects to pre-analyze the tasks was that they sometimes did not name the subtask names based on the given syntax as described in Section 3.2.2 and Appendix A.1. For example, Subject 5 originally named the primitive subtask "move up spatula" (Figure 6-17) as "retreat". After the student investigator pointed out that an object name must be included in the subtask name, the subject changed the name to "retreat robot". Since the object "robot" could not be recognized by the robot's vision agent, the subtask name was then changed to "move up spatula". In addition, some subjects did not include target names in primitive subtask names when the subtask did involve a target. For example, for a primitive subtask of putting a knife onto a placemat, the name "put knife" was not acceptable since the target name "placemat" was not included in the name, but "put knife onto placemat" was acceptable.

Proper and adequate voice training was very important and necessary even for native English speakers. In the pilot study, two human subjects, one a native English speaker and the other with a foreign accent, did quite well at the end of their first voice training session. However, when they proceeded with their vocal interaction with the robot, the robot responded very poorly because the majority of their requests and responses could not be recognized by the robot. The two subjects had a considerably high level of frustration with the poor recognition of their speech. The participants had to go back to train their voice profile with different standard training passages provided by Microsoft Windows. Two specific training passages were found to be very effective for the two subjects, and these passages were then used for all other subjects to train their voice profiles. Results of the human-robot vocal dialogues were so encouraging that the subjects did not need to have any vocal interaction practice with the robot, and they directly went to teach tasks to the robot after voice recognition training. Human subjects were instructed by the robot to not speak while the robot was speaking; however, if they did speak simultaneously, the subjects were required to repeat their utterances. In addition, it was found that the utterances for the teacher to interact with the robot must consist of at least two words in order for the robot to greatly avoid incorrectly recognizing the teacher's speech.

The vision-based robot teleoperation was found to be the major problem and frustration for subjects in using the robot-task teaching and learning system. The vision-based teleoperation system prevented the subjects from having full six-DOF hand motion and the ideal direct view to the robot and its workspace. Indirect visual feedback (by video monitor) of the robot and its workspace was purposely not provided to the human subjects to allow potential system portability and enable use of the system in unstructured environments without mounted cameras in the robot workspace. An alternative hand motion tracking technique such as the data glove (Peters et al., 2003), would allow the teacher to have full 6-DOF hand motion and stand closer to the robot during the task demonstration to have a better view of the robot and its workspace. A backdrivable robot manipulator that is safe for human contact would allow the user to be even closer for the best view of the robot environment. In the current system, the operator had to be in a safe area out of the robot's reach. As the goal of this research was to develop and test the human-robot task learning system and not the robot-teleoperation system, human subjects were instructed not to weigh their frustration with the teleoperation system too heavily in their evaluation of the overall robot-task teaching and learning system.

Managing or reducing the cognitive load required for general users in the human-robot interaction is important for the user to be able to naturally and intuitively interact with the robot. The following means were taken to achieve this: 1) making the interaction transparent so that users can easily track the robot's learning progress and be aware if the robot has understood their requests or responses; 2) making the conversations as plain and flexible as possible, such as by allowing "task path" to be used as an alternative utterance for "task trajectory" and "path segments" for "path/trajectory episodes"; 3) having the robot system remind the users of the normal procedure and commonly used vocal commands at the beginning of each teaching and learning stage; and 4) adopting natural and simple utterances. An example of the latter in the pilot study, occurred when two subjects were initially asked to give timely feedback to the robot on the orientation of the robot end-effector during the robot task practice and execution, with the following utterances "rotate yaw/pitch/roll more right/left". These were found to be very awkward and hard to use because the user had to have four cognitive interpretations before issuing a related feedback cue: comprehension of the "rotate", definition of "yaw", "pitch" and "roll", the plus and minus directions, and the mapping from the desired refinement to the proper vocal instructions. These utterances were changed to "turn more right/left" "turn more up/down" and "roll more right/left" for yaw, pitch and roll rotations of the robot end-effector, respectively. This change was well received in the user study that followed.

The MSV information for segmentation might not have truly reflected the teacher's intention if the user did not get used to the teleoperation system or if the views of the robot end-effector and objects of interest were not adequate. In future development, this issue may be avoided if a better alternative teleoperation system is adopted so that the teacher could have full 6-DOF hand motion and have a better view of the robot and its workspace when teleoperating the robot to demonstrate a task.

During the robot's task practice, the human subjects seem to have changed their voice pitch or speed when they felt some urgency to issue feedback to refine the robot performance. This would occur, for example, when the relative pose between the robot hand and an object of interest was not satisfactory, when the robot was about to grasp the object. The change of voice pitch or speed might have resulted in unsuccessful voice recognition of the issued feedback. In future development, the voice recognition system could be further trained to understand the user's words even with the change of pitch or speed. Furthermore, the change of pitch could be understood to represent urgency and the step size of the motion could be adjusted accordingly. For example, a high pitch may cause an immediate stopping of robot end-effector motion, or an immediate retreat to a previous position.

7.2 Comparison of Robot-Task Teaching and Learning Method to Literature

The newly developed robot-task teaching and learning method in this research is among several human-style robot-task teaching and learning methods recently developed. Comparing to the human-style robot task teaching and learning methods in Thomaz and Breazeal (2008), Lockerd and Breazeal (2004), and Breazeal et al. (2004), and Nicolescu and Mataric (2001, 2003), this research introduced a more complete systematic human-inspired robot-task teaching and learning method that enables general users to teach a robot new tasks intuitively. The method permits the robot to learn the taught tasks and adapt its learned task knowledge to new task setups. In comparison to the relevant methods in the literature, the following improvements have been achieved.

The burden for the robot to learn new object-manipulation tasks from human teaching has been shared more between the teacher and robot compared to earlier work. The teacher's expertise in the task to be taught was well exploited to help the robot learn new tasks. The overall task introduction, not used in earlier work, helps the robot build a hierarchical task structure and comprehend the task in a top-down manner. This information is highly useful for the robot to partition the demonstrated task later on. The task introduction also helps the robot build the task structure in a form that is easily understandable to other users. The human-directed robot perception (i.e. use of vocal subtask-segmentation cues during task demonstration) and timely feedback on the performance of the robot's task practice are also examples of taking advantage of the teacher's expertise in ways that are natural and intuitive to the teacher. Although Breazeal et al. (2004a) and Thomaz and Breazeal (2008) presented a step-by-step teaching approach, the teacher could not help the robot build a hierarchical task representation. Their methods might be more suitable for simple and sequential tasks. In

Nicolescu and Mataric (2001, 2003), the teacher indeed demonstrated the task and gave feedback during the robot's task practice. However, they did not attempt to help the robot build a hierarchical task plan, the instructions given in the teaching stage before the robot's task practice were not designed to help the robot better understand the demonstrated task but merely to demonstrate the task, and the teacher's influence on the robot's refinement of the task knowledge was fairly limited.

In this research, a new flexible vote-based algorithm that permits use of different types of signals for segmenting demonstrated task trajectories (Wu and Kofman, 2008) was developed. The closest related work in the literature was done by Lieberman (2004) and Lieberman and Breazeal (2004) as described in Equations (2-2) to (2-4). However, they only used the MSV with an augment of the tactile feedback (Equation (2-2)) to segment the demonstrated trajectory. Their chosen signals to segment the trajectory were limited to continuous signals, where their derivatives were utilized for trajectory segmentation.

A probabilistic optimization method to ground the user-introduced high-level abstract task knowledge to the robot-sensor data acquired in the task demonstration was developed. This is essentially the assignment of segmented trajectory episodes to the introduced primitive subtasks.

Two blending schemes, which blend different task trajectories generated from different reference frames for the robot to adapt its learned task trajectory to new setups, have also been presented. These schemes take into account the robot starting pose, poses of the objects and targets, and the demonstrated departure trajectory after an object is engaged. In addition, a new reference frame, the object-orientated frame, is introduced. This new reference frame brings an important advantage over the world frame, by enabling the robot to maintain similar relative orientations between the robot and object as those demonstrated, no matter where the object is placed. The closest related work in the literature is by Lieberman (2004) and Lieberman and Breazeal (2004). However, they considered translations only, generalized the demonstrated trajectories in the world frame and object frame with two radial-basis-function neural networks, and did not consider the different robot starting poses and the desired departure trajectory from an engaged object. Their method therefore needed multiple demonstrations compared to the single demonstration required in this research. Furthermore, their methods may have difficulties in adapting the demonstrated task trajectory to significantly different new task setups. Also, further task trajectory interpolations may be needed to handle the different robot starting poses and the transition trajectory from one object to another. In addition, they did not address the issue of how to apply the demonstrated speed profile to new task setups.

The step of checking and teaching the robot's background knowledge needed for the task to be taught is another novelty of the developed method. The purpose of this step is to enable the teacher to understand the robot's capacity, including its knowledge about objects and previously learned tasks or subtasks, and to teach the robot the lacking knowledge if necessary. This step makes it possible for the robot to learn new tasks if the needed prerequisite knowledge can be taught to the robot first.

Comparing to the user study in Weiss et al. (2009), the user study in this thesis included more quantitative and qualitative evaluations. The method developed in this research seems to be more effective since human subjects in this study demonstrated each task only once to the robot while the users in the other study demonstrated the whole task more than three times on average. In addition, the two tasks taught in this research were more complicated than the two tasks in that study ("push box" and "close box"). The main difference in the robot's learning capacity was that users in their method had to demonstrate the whole task again if the robot's task performance was not satisfactory, whereas in the current study, timely feedback was used to help the robot refine its task performance. Thomaz and Breazeal (2008) did not obtain subjective evaluations of their exploration-based task-learning system.

7.3 Thesis Contributions

The main contributions made through this research in the area of robot-task teaching and learning are as follows:

- <u>A human-inspired robot-task teaching and learning method that enabled general users to teach a robot new object-manipulation tasks intuitively and effectively, and permitted the robot to learn the taught tasks and adapt its learned tasks to new task setups.</u> This method was effective as the subjects in the user-study experiments demonstrated each task only once to the robot, and were allowed to help the robot improve its task performance to a satisfactory level during robot task practice. This method also allows the general user and robot to share the task learning burden. The method would be applicable to general object-manipulation tasks and different embodiments of robots.
- <u>Two novel teaching and learning stages: checking and teaching the robot's background knowledge needed for the task to be taught, and introduction of the overall task.</u> The former stage allows the teacher to check if the robot has the needed background knowledge to learn a new task. It also makes it possible for the teacher to teach the robot the lacking needed background knowledge, in order to prepare the robot to be ready to learn the new task. The latter stage makes use of the teacher's task expertise in breaking down the task plan into a hierarchical structure, and helps the robot build a task representation that is understandable to other users.
- <u>Approaches to exploit the teacher's task expertise through natural and transparent human-robot interaction.</u> Compared to relevant methods in the literature, originality of the task-expertise exploitation in this thesis includes: teaching the robot needed background knowledge, introducing the task structure to the robot, providing vocal subtask-segmentation cues during the task demonstration, and helping the robot partition the taught task upon the robot's request if it recognizes its failure to partition the task automatically. In addition, improvements on existing approaches were made in allowing the teacher to give timely feedback during robot task practice on both the task partition and the 6-DOF robot-task trajectory.

- A new flexible and effective vote-based algorithm for segmenting demonstrated task trajectories. The main improvement over similar methods in the literature is that this algorithm is flexible to use different types of signals and considers their individual uncertainty and reliability for segmentation. The algorithm was found to be effective, requiring only a single task demonstration.
- A probabilistic optimization method to ground the user-introduced high-level abstract task structure (or plan) to demonstrated sensor data. There is no report of similar methods in the domain of robot-task teaching and learning.
- Two blending schemes to blend different task trajectories generated from different reference frames so that the robot can adapt its learned task knowledge to new setups. Compared to similar approaches in the literature, improvement was made to take into account the robot starting pose, demonstrated departure trajectory after an object is engaged, and taught task speed, when the robot adapts it learned task trajectory to new task setups. A new type of reference frame, which relates the position and orientation of the robot to the object or target being approached, was also developed.
- Design and implementation of an agent-based framework to realize the robot-task teaching and learning method. The framework consists of: the Robot Agent (RA), Teleoperation Agent (TOA), Speech Agent (SA), Vision Agent (VA), Task-Learning Agent (TLA), Task-Performing Agent (TPA), and a database. Only the hand tracking part of the TOA and calibration of the stereo-camera pair for the VA were developed outside of this research.
- Evaluation and validation of the effectiveness and intuitiveness of the developed agent-based robot-task teaching and learning system by a user study. Only two previous user studies to evaluate methods of robot teaching and learning of object manipulation tasks were reported (Weiss et al. 2009). Human subjects' evaluation of the system was as follows (scores out of 5): effective (4.2), intuitive (3.9), partitioned taught tasks as expected (4.6), responded overall well to their timely feedback (4.1), refined its task knowledge and applied the refinement in subsequent task practice (4.4), adapted its learned task to new setups well (4.6), and without frustration (4.0).

7.4 Limitations of the Research

While this research introduced a new systematic human-inspired robot-task teaching and learning method that enables general users to teach a robot new tasks, and permit the robot to adapt its learned tasks to new task setups, there are some limitations of the currently developed system and experiments.

There are limitations in the current use of task and subtask names. The verbs, object names, and directional prepositions of the task/subtask names have to exist in the database at the beginning of the task teaching. Currently, the teacher cannot add words online. In addition, the names of primitive subtasks of a task must only involve objects that are potentially observable and detectable to the robot. For instance, "Pour Water" cannot be the name of a primitive subtask because the robot is currently not capable to extract the features of “Water”. However, the names of unobservable objects are permitted to appear in non-primitive subtasks, such as the tasks “Pour Liquid” and “Pour Juice” taught in the experiments. Also, the name of a subtask or task must include a verb and an object name, and the target names must be properly included in the primitive subtasks when targets are involved. These naming requirements allow the robot to extract needed information about the subtask actions (verbs) and objects to be manipulated in the current system. However, these requirements limit the flexibility to name the task/subtask names in the task pre-analysis. In the future, the strict requirement to include the names of involved objects or targets into the subtask names might be relaxed if the robot is able to infer the information from: the adjacent subtask names, teacher’s task demonstration, and the robot’s previous experience.

The task structure and subtask names could not be modified by the teacher in every teaching and learning stage. In the current system, the task structure could only be taught and modified in the overall task introduction stage, and the task/subtask names can be changed only in the task introduction stage and step-by-step teaching stage.

In the current system, low-level primitive skills or actions, such as Close Hand, Open Hand, and Pour Liquid, are the building blocks of primitive subtasks. However, this research has not addressed the issues of how to recognize and extract these skills from the teacher’s task demonstration, and how the user can teach new primitive skills to the robot. The three primitive skills were pre-built for the robot to learn the tasks taught in the experiments, although Close Hand and Open Hand could have been taught by the teacher. However, teaching a robot a complicated low-level skill may involve knowledge in robotics that general users would not have. Such skills should be pre-built. For example, the skill to insert a peg into a hole involves force control and different strategies to adjust the relative pose between the peg and hole. In the domain of service-robot task teaching and learning, primitive skills must be pre-built, if the knowledge extracted from a limited number of task demonstrations by a general user is not sufficient to enable the robot to adapt the skills to new task setups.

While the robot was able to perform tasks that involved picking and placing objects, the lack of force and torque sensors on the robot prevented the robot from learning and performing more complex tasks involving contact, such as the peg-in-hole tasks.

The vision agent obtained only limited 3D information about objects of interest, using the stereo-cameras mounted on the robot end-effector, and simple contour-related features. However, the vision system was not one of the foci of this research. These limitations prevented the robot from recognizing a variety of objects and computing their locations and orientations. The limitations also prevented the robot from having the capabilities of obstacle avoidance, potential collision estimation, and determination of an optimal execution order of subtasks for high-level task adaptation. Having such capabilities would enable the robot to better adapt its learned tasks to a given task setup based on the constraints of the setup and dependencies of the subtasks. In addition, the eye-in-hand camera configuration made it difficult for the robot to perceive objects in its workspace while it was being teleoperated or executing a learned task. The robot currently cannot handle dynamic environments where the objects and targets are not stationary.

The teacher was not able to have free 6-DOF hand motion or an ideal direct view to the robot and its workspace during robot teleoperation with the current vision-based teleoperation system. These limitations could be alleviated using alternative motion tracking techniques such as a data glove (Peters et al., 2003).

The implementation of the current learning algorithms could be more flexible regarding thresholds for valid trajectory episodes, such as their temporal and spatial lengths. For example, the minimum temporal length of a valid trajectory segment is currently set at 5 s. This threshold caused a problem when Subject 5 was helping the robot partition their demonstrated task "Layout Table". The system could not accept the trajectory segment that was being assigned to primitive subtask "Put Spatula to Placemat" as the temporal length was less than 5 s.

The reuse of previous learned tasks or subtasks to learn a new task, called bootstrap learning (Thomaz and Breazeal, 2008; and Calinon and Billard, 2007), was not included in this research. In future development, the teacher could be able to ask the robot to perform a previously learned task or subtask in the background-knowledge checking and teaching stage. In the step-by-step teaching stage, the teacher could also be able to examine the robot performance of its earlier learned tasks or subtasks that are parts of the new task by requesting the robot to perform them. The in-whole task demonstration would even be unnecessary if the robot had learned all related subtasks and its performance on these subtasks were satisfactory.

In all task teaching and learning stages, the teacher was not permitted to help the robot extract and specify object features that uniquely identify the object. Instead, the Vision Agent autonomously extracted features of an object and combined the feature information for object identification. If an object could not be distinguished from other objects based on its group of extracted features, the robot would not successfully learn a task that involved the related object.

Further development of the object modeling is needed for 3D shape and surface-related features, functional parts, typical places objects would be found, task-related functional roles, object states, and manipulation actions that could be applied to the object.

The determination of optimal subtask-execution order was not included in the Task-Performing Agent of the current system. Instead the robot executed the subtasks of its learned tasks in the same subtask orders they were taught. High-level subtask adaptation that depends on determining optimal subtask-execution order was not implemented.

The testing of the ability of general users to use the human-guided robot task learning system was limited to an initial pilot test with two subjects and the user study with four users. While these studies provided a useful evaluation of the teaching and learning system, once the robot task teaching and learning system is improved, further evaluation of the system with more subjects should be conducted.

Chapter 8
Conclusion and Future Work

8.1 Conclusion

A human-inspired robot-task teaching and learning method was developed and implemented in a new robot teaching and learning system. Experiments suggest that the research objectives in developing a new robot task learning system have been met. The system enabled general users to teach two different object-manipulation tasks to the robot intuitively. The teacher's task expertise was well exploited to help the robot learn the taught task and the robot was able to learn the taught tasks and adapt its learned tasks to new task setups.

8.1.1 Intuitive and Effective Robot-Task Teaching

The developed robot-task teaching and learning system enables general users to teach tasks to the robot intuitively and effectively. Human subjects in the user study evaluated the system as being effective (4.2 out of 5), intuitive (3.9), and without frustration (4.0). The vocal interaction efficiency during the overall task introduction was nearly 90%. The average time used in each teaching and learning stage before the robot's task practice was less than 9 min.

The developed human-inspired robot-task teaching and learning method and system does not require the users to be knowledgeable in robotics or computer science. There is no requirement that the user or teacher have any more knowledge than having expertise at the task-level for the task to be taught. In the pre-analysis stage, it is not difficult for the teacher to break down the tasks into subtasks, represent the tasks with hierarchical task structures, and specify robot-hand actions to primitive subtasks. The designed task or subtask name syntax is also fairly flexible for the teacher to name the tasks and subtasks, and allows the robot to extract valuable information (such as the actions and involved objects) from the names. However, there are also some limitations for the name syntax. A legitimate task or subtask name must include an action (verb) and an object name. The target object names have to be included in the primitive subtask names properly. The objects mentioned in the primitive subtask names must be detectable to the robot. In addition, the missing vocabulary (such as verbs, object names and directional prepositions) must be input in the database before the teacher commences the task teaching to the robot.

In the stage of checking and teaching needed background knowledge, the teacher is allowed to ask the robot for information about its previously learned objects, tasks and subtasks using natural utterances such as those in Appendix A.1. The teacher is also allowed to modify relevant information easily, such as an object's name and task-relevant attributes, and the name, involved objects and robot-hand actions of a task or subtask. Currently, the teacher is not able to request the robot to perform a task or subtask, or teach a basic object-manipulation skill to the robot.

During the overall task introduction, the designed robot-guided dialogues enable the teacher to follow the robot-human interaction very well, and respond to the robot's requests efficiently. The teacher is able to easily instruct the robot to go back to a just introduced subtask. Part of the flexibility and richness of the human-robot interaction is enabled by the wide range of permitted utterances, shown in the vocal human-robot interaction scripts (Appendix A.1) and the excerpt (Appendix A.2) of the dialogues from the demonstration video. Currently, the robot asks for acknowledgement for every teacher's non-confirmation response to achieve a reliable information exchange. This request-response-request (for acknowledgement)-confirmation cycle may be lengthy, and unnecessary if the rates of positive acknowledgement are very high at the beginning of the task introduction. To deliver a complicated task structure (such as the one in Figure 6-17) to the robot via vocal dialogues is challenging. The teacher needs to focus on the current subtask. To make this easier for the users, at the beginning of the introduction of a child subtask, the robot tells the teacher which child subtask of which subtask is to be built. This is done by the robot using utterances such as "Now, we are to build the **second** step of the subtask: ***Lay out Knife***.". Although this measure is helpful, GUI-based interaction might be necessary to complement the vocal dialogue-based approach.

The step-by-step teaching stage is designed for the teacher to teach important and complex subtasks to the robot and direct the robot to perceive important task-relevant states. In this stage, the teacher is able to check and change information about the introduced task, and teach chosen subtasks to the robot. Because of the limited capacity of the robot sensor system, the teacher is limited to direct the robot to perceive the transitions of the child subtasks by giving vocal subtask-segmentation cues such as "next step" at the transitions while the teacher is demonstrating a subtask. The reuse of the robot's previously learned task subtask is not included in the current system. In this stage, the robot could not perform the tasks or subtasks upon request from the teacher. Therefore, the teacher could not examine the subtask performance of the robot. The potential bootstrapping teaching and learning by re-using previous experience needs to be explored and developed in future development.

In the in-whole task demonstration, the teacher is allowed to successfully demonstrate the whole task and give subtask-segmentation cues at the subtask transitions. The vocal cues are not necessary for the task learning algorithm to segment the taught task, but they are certainly helpful to influence the robot. The visual marked-based robot teleoperation system, which is used for the hand-to-hand task demonstration in the current system, is practical and enables the teacher to demonstrate tasks to the robot. However, the teleoperation system was found to be the major contribution to the subjects' frustration to teach tasks to the robot in the user study. Other motion tracking techniques such as the data glove in Peters et al. (2003) and Ehrenmann et al. (2002) would overcome the drawbacks of the current teleoperation system.

In the task learning stage, the system enables the teacher to intuitively and effectively help the robot partition the taught task upon request of the robot if the robot has found it failed to do so automatically.

The teacher is able to intuitively give timely feedback on the robot task performance during the robot's task practice. The teacher can give timely feedback on the task partition, and translations, rotations and speed of the task trajectory while the robot is practicing the task. The robot is allowed to effectively and instantly refine its task knowledge accordingly, and reflect the refinement in the next sampling period. The feedback-refinement process was found to be transparent, intuitive and effective. However, neither the task structure nor the task/subtask names could be modified in the task-practice stage in the current system.

In addition, the developed system helps the teacher teach the task as desired, and helps the robot learn the task as expected. The teacher can request the robot to go back or jump over some teaching and learning stages to a specific stage under some constraints. At the end of each task demonstration, the teacher is explicitly asked if they are satisfied with the demonstration. If not, the teacher is required to re-demonstrate the task. At the end of each robot task practice, the teacher is also asked if the robot task performance is acceptable. If not, the robot can practice the learned task again. At the end of the task teaching and learning, the robot asks the teacher if they are satisfied with the learned result. If not, the robot would ask the teacher which teaching and learning stage the robot and teacher should go back to.

There is not much demand for the teachers to memorize all the robot-task teaching and learning procedures and utterances for the vocal human-robot interaction. First of all, the proposed method closely imitates the human-to-human teaching paradigm. The teaching and learning procedures are not strange to the general users. The teacher is reminded of the procedures by the robot throughout the task teaching and learning process. In most cases, the utterances used for vocal human-robot interaction are designed to be context-based, natural, and flexible. At the beginning of each robot-task teaching and learning stage, the robot reminds the teacher which teaching and learning stage they are in, and the relevant procedures and utterances that would most likely be used.

8.1.2 Exploitation of Task Expertise of the Teacher

The developed robot task teaching and learning method has a fairly well shared teaching and learning burden between the teacher and robot. This is very important since the ability of a robot to automatically adapt to a vast variety of working environments and to learn arising new tasks may still be limited in the foreseeable future. In this research, the proposed method greatly exploits the task expertise of the teacher at the task level to help the robot learn the taught tasks as expected and organize the task knowledge in ways that are easily understandable to other users. Specifically, the measures to take advantage of the teacher's expertise include: introducing the overall task to be taught, teaching the task hand-to-hand through teleoperation using natural hand motion, human-directed robot perceiving, helping the robot partition the demonstrated task upon the robot's request, and offering timely feedback on the robot's task segmentation and task trajectory during the robot's task practice. All of these measures were found to be very natural and effective in the user study.

8.1.3 Robot Task Learning from Human Teaching

The developed system enables the robot to successfully learn new tasks from a general user's teaching by employing together top-down and bottom-up approaches. Human subjects evaluated the robot as follows: partitioned taught tasks as expected (4.6 out of 5), responded overall well to their timely feedback (4.1), and refined its task knowledge and applied the refinement in subsequent task practice (4.4).

The robot learns the abstract task structure and information about robot-hand action for each subtask from the teacher's overall task introduction. Through the designed syntax of the task/subtask names, the robot can also extract information regarding the task/subtask actions (the verbs in the names), directional prepositions, and involved objects and targets from the introduced task/subtask names. The introduced task structure helps the robot represent its learned task in ways that are easily understandable to other users. The information about the subtask actions, directional prepositions, involved objects, and specified robot-hand actions is very important in grounding or associating the introduced abstract knowledge to the robot sensor data (such as task trajectory and robot hand status) collected in the task demonstration. However, the current system has limited ability to obtain information about the involved objects in a subtask from adjacent subtasks and the robot's previous experience. Instead, this information must currently be included in the primitive subtask names. In addition, the robot can be directed by the teacher during task demonstration to perceive important states, such as the transitions of subtasks.

The robot can learn the taught task by analyzing the observed task demonstration data in a bottom-up approach. The developed vote-based algorithm helps the robot segment the demonstrated task trajectory into meaningful episodes. The method takes into account smooth and abrupt changes in signals (gripper status, MSV), robot-environment relationships (gripper-object distance), and teacher-input information (segmentation cues). For each signal, the relative proximity of its segmentation candidates to likely partition points, and the relative segmentation reliability have to be determined. To ground the task knowledge learned in the top-down manner to the results learned in the bottom-up approach, a probabilistic optimization process is presented to assign the obtained task trajectory episodes to the introduced primitive subtasks. This process utilizes the information about the involved objects and robot-hand actions (obtained both in the task introduction and task demonstration), vocal subtask-segmentation cues provided in the demonstration, and the degree of matching of the subtask action (the verb) and directional preposition in a subtask name with the characteristics of a trajectory episode. Although the degrees of matching between a verb - directional-preposition pair and an episode characteristic could be statistically computed from the robot's previous experience, such information was currently pre-determined due to the robot's limited experience. If the robot recognizes that it failed to partition the task automatically, it would ask the teacher for help. The experimental results demonstrated the capability and flexibility of these methods to learn the tasks taught by different human subjects.

In addition, the robot could also refine its task knowledge during its task practice based on the timely feedback given by the teacher, including the task partition, and translations, rotations and speeds of the task trajectory.

There are several mechanisms incorporated into the system to help the robot learn the task as expected by the teacher. In the overall task introduction stage, the pattern of robot request, teacher response, robot verification of its understanding, and teacher confirmation, is designed to help the robot learn the hierarchical task structure as expected by the teacher. The robot's acknowledgement of recognized teacher's vocal subtask-segmentation cues during the task demonstration also helps the teacher deliver the cues successfully. During the robot task practice, the robot's transparent and immediate response to the teacher's feedback on the task partition and trajectory proves to help the robot effectively refine its task knowledge according to the teacher's feedback.

In addition, due to the current limited capacity of the robot sensor system, for instance, the lack of the force and torque sensors, the types of the tasks the robot is ready to learn are limited mainly to tasks that pick and place objects.

8.1.4 Adaptation of Learned Tasks to New Task Setups

The robot has the ability to adapt its learned tasks to new task setups, even for setups considerably different from those in the demonstration. This is achieved by generalization of taught task trajectories in different reference frames and the use of two blending schemes. The two blending schemes take into account: the poses of involved objects, targets, and robot starting point; and the departure trajectory right after an object is engaged in the demonstration. In addition, the taught task speed profile is also applied in the task practice and execution, specifically when the robot gets close to manipulate objects. The experimental results demonstrate the strong capacity of the robot to adapt its learned task to new task setups.

The limited capacity of the current robot vision system does not allow the robot to determine an optimal execution order of subtasks based on subtask dependencies and task configurations,

8.2 Future Work

Further work needs to be done to improve the human-inspired robot-task teaching and learning method and agent-based system. The main future work is described as follows:

- adopt a hand-motion tracking technique, such as magnetic-field and inertial based location sensors, so that the teacher could teleoperate the robot with full 6-DOF hand motion, and position themselves where they can have an ideal direct view to the robot and its workspace when teleoperating the robot.
- enable the robot to perform its learned tasks or subtasks upon the teacher's request in the background-knowledge checking and teaching stage. The teacher would also be able to teach basic object-manipulation skills to the robot in this stage.
- adopt bootstrap learning techniques for the robot to learn new tasks by reusing its previously learned tasks and subtasks. In the step-by-step teaching stage, the teacher would be able to examine the robot performance of its earlier learned tasks or subtasks that are parts of the new task by requesting the robot to perform them. The teacher would just teach subtasks that are new to the robot or that the robot did not perform satisfactorily.

- enable the robot to perform semantics analysis of the introduced task/subtask names.
- make the robot more capable to perceive and interact with its environment by using stereo vision, 3D range sensing, and force and torque sensors. This would be greatly useful for the robot to build complete 3D models of its learned objects, understand constraints of its workspace, estimate possible collisions, and avoid obstacles.
- enable the robot to have more capacity to perceive important task states in response to the teacher's instruction during the step-by-step task teaching.
- give the robot the ability to determine an optimal subtask-execution order of a task, based on the learned subtask dependencies and constraints of given task setups.
- test the system with more human subjects and new types of tasks.

Bibliography

Aarno D., Lingelbach F. and Kragic D. (2005), "Constrained path planning and task-consistent path adaptation for mobile manipulators", *Proc of 2005 Int. Conf. on Advanced Robotics*, p 268-273.

Argall B., Chernova S., Veloso M., Browning B. (2009), "A survey of robot learning from demonstration", *Robotics and Autonomous Systems*, v 57, n 5, p 469-483.

Akanyeti O., Nehmzow U. and Billings S.A. (2008), "Robot training using system identification", *Robotics and Autonomous Systems*, v 56, n 12, p 1027-41.

Aleotti J. and Caselli S. (2006), "Robust trajectory learning and approximation for robot programing by demonstration", *Robotics and Autonomous Systems*, v 54, n 5, p 409-13.

Argall B., Browning B., and Veloso M. (2007), "Learning by demonstration with critique from a human teacher", *Proc. of the 2007 ACM/IEEE Conf. on Human-Robot Interaction*, p 57-64.

Asfour T., Azad P., Gyarfas F. and Dillmann R. (2008), "Imitation learning of dual-arm manipulation tasks in humanoid robots", *Int. Journal of Humanoid Robotics*, v 5, n 2, p 183-202.

Muslim A. M., Ishikawa M., and Furukawa T. (2006), "A new approach to task segmentation in mobile robots by mnSOM", *2006 Int. Joint Conf. on Neural Networks*, p 3510-17.

Bentivegna D.C. and Atkeson C.G.. (2003), "A framework for learning from observation using primitives", *RoboCup 2002: Robot Soccer World Cup VI*, v 2752, p 263-70.

Bentivegna D.C., Atkeson C.G., and Cheng G. (2004), "Learning tasks from observation and practice", *Robotics and Autonomous Systems*, v 47, n 2-3, p 163-9.

Bentivegna D.C., Atkeson C.G. and Cheng G. (2006), "Learning similar tasks from observation and practice", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, p 2677-83.

Biggs G. and MacDonald B. (2003), "A survey of robot programming systems", *Proc. of the Australasian Conf. on Robotics and Automation*, CSIRO, Brisbane, Australia.

Billard A., Callinon S., Dillmann R., and Schaal S. (2008), Robot programming by demonstration, in: Siciliano B., Khatib O. (Eds.), Handbook of Robotics, Springer, USA, 2008 (Chapter 59).

Breazeal C., Brooks A., Gray J., Hoffman G.., Kidd C., Lee H., Lieberman J., Lockerd A., and Mulanda D. (2004a). "Humanoid Robots as Cooperative Partners for People", *Int. J. Humanoid Robotics*, v 1, n 2, p 1–34.

Breazeal C., Hoffman G. and Lockerd A. (2004b), "Teaching and working with robots as a collaboration", *Proc. the Third Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, v 3, p 1030-1037.

Breazeal C., Berlin M., Brooks A., Gray J. and Thomaz A.L. (2006), "Using perspective taking to learn from ambiguous demonstrations", *Robot and Autonomous Systems*, v 54, n 5, p 385-93.

Calinon S, Guenter F. and Billard A. (2007), "On learning, representing, and generalizing a task in a humanoid robot", *IEEE Trans. Syst, Man, Cybernetics, Part B*, v 37, n 2, p 286-98.

Calinon S. and Billard A. (2007), "Active teaching in robot programming by demonstration", *16th IEEE Int. Conf. on Robot and Human Interactive Communication*, p 702-7.

Card S., Moran T. and Newell A. (1983), The Psychology of Human-Computer Interaction. Hillsdale, New Jersey: Erlbaum Associates.

Chang S., Kim J., Kim I., Borm J.H., and Lee C. (1999), "KIST teleoperation system for humanoid robot," *in Proc. 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, p 1198–1203.

Chella A., Dindo H. and Infantino I. (2006), "Learning high-level tasks through imitation", *2006 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, p 3648-54.

Chella A., Dindo H. and Infantino I. (2007), "Imitation learning and anchoring through conceptual spaces", *Applied Artificial Intelligence*, v 21, n 4-5, p 89-105.

Chen J. and McCarragher B. (1998), "Robot programming by demonstration-selecting optimal event paths", *in Proc. 1998 IEEE Int. Conf. on Robotics and Automation*, v 1, pt 1, p 518-23.

Chen J. and Zelinsky A. (2003), "Programming by demonstration: coping with suboptimal teaching actions", *Int. J. of Robotics Research*, v 22, n 5, p 299-319.

Cherubini A., Giannone F., Iocchi L. and Palamara P.F. (2007), "An extended policy gradient algorithm for robot task learning", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, p 4121-6.

Choi M. H. and Kim S. J. (2000), "Force/moment direction sensor and its use for human-robot interface in robot teaching", *Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics*, v 3, p 2222-7.

Choi M. H. and Kim S.J. (2001), "A Force/Moment Direction Sensor and Its Application in Intuitive Robot Teaching Task", *Trans. on Control, Automation, and Systems Engineering*, v 3, n 4, p 236-241.

Choi M. H. and Lee W. W. (2003), "Quantitative evaluation of an intuitive teaching method for industrial robot using a force / moment direction sensor", *Int. J. of Control, Automation and Systems*, v 1, n 3, p 395-400.

Coates A., Abbeel P. and Ng A. (2008), "Learning for control from multiple demonstrations*", in Proc. of the 25th Int. Conf. on Machine Learning*, p 144-151.

Cohen P.R. and Levesque H.J. (1990), "Intention is choice with commitment", *Artificial Intelligence*, v 42, n 2-3, p 213-61.

Cohen P.R. and Levesque H.J. (1991), "Teamwork", *Nous, Special Issue on Cognitive Science and AI*, v 25, n 4, p 487-512.

Delson N. and West H. (1994), "Robot programming by human demonstration: the use of human variation in identifying obstacle free trajectories", *in Proc. 1994 IEEE Int. Conf. on Robotics and Automation*, v 1, pt 1, p 564-71.

Delson N. and West H. (1996), "Segmentation of task into subtasks for robot programming by human demonstration", *Proc. of the Japan-USA Symposium on Flexible Automation*, pt 1, v 1, p 41-7.

Dillmann R., Rogalla O., Ehrenmann M., Zöllner R., and Bordegoni M. (1999), "Learning Robot Behaviour and Skills Based on Human Demonstration and Advice: the machine learning paradigm", *Proc. of the 9th Int. Symposium of Robotics Research*, Utah, USA, p 229–38.

Dillmann R. (2004), “Teaching and learning of robot tasks via observation of human performance”, *J. of Robotics and Autonomous Systems*, v 47, n 2-3, p 109-116.

Dixon K.R. (2004), Inferring User Intent for Learning by Observation, doctoral dissertation, Department of Electrical & Computer Engineering, Carnegie Mellon University.

Ehrenmann M., Rogalla O., Zöllner R., and Dillmann R. (2001), “Teaching service robots complex tasks: programming by demonstration for workshop and household environments”, *Proc. of the IEEE Int. Conf. on Field and Service Robotics*, Finnland.

Ehrenmann M., Zollner R., Rogalla O., and Dillmann, R. (2002), “Programming service tasks in household environments by human demonstration”, *Proc. of 11th IEEE Int. Workshop on Robot and Human Interactive Communication*, p 460-7.

Ehrenmann M., Zöllner R., Rogalla O., Vacek S., and Dillmann R. (2003), “Observation in Programming by Demonstration: Training and Execution Environment”, *Int. Conf. on HumanoidRobots*, Karlsruhe, Munich, Germany.

Ekvall S. and Kragic D. (2004), “Interactive grasp learning based on human demonstration”, *IEEE Int. Conf. on Robotics and Automation*, pt 4, v 4, p 3519-24.

Ekvall S. and Kragic D. (2006), “Learning task models from multiple human demonstrations”, *15th IEEE Int. Symposium on Robot and Human Interactive Communication*, p 358-363.

Ekvall S. and Kragic D. (2008), “Robot learning from demonstration: A task-level planning approach”, *Int. Journal of Advanced Robotic Systems*, v 5, n 3, p 223-234.

Fod A., Mataric M.J., and Jenkins O.C. (2002), “Automated derivation of primitives for movement classification”, *Autonomous Robots*, v 12, n 1, p 39-54.

Friedrich H., Dillmann R., and Rogalla O. (1999), “Interactive robot programming based on human demonstration and advice”*, Int. Workshop on Sensor Based Intelligent Robots*, v1724, p 96-119.

Friedrich H., Kaiser M., and Dillmann R. (1996a), “PBD - the key to service robot programming”, *Acquisition, Learning and Demonstration: Automating Tasks for Users*. *1996 AAAI Symposium (TR SS-96-02)*, p 23-9.

Friedrich H., Munch S., Dillmann R., Bocionek S., and Sassin M., (1996b), “Robot programming by demonstration (RPD): supporting the induction by human interaction”, *Machine Learning*, v 23, n 2-3, p 163-89.

Grollman D.H. and Jenkins O.C. (2007), “Learning robot soccer skills from demonstration”, *2007 6th IEEE Int. Conf. on Development and Learning*, p 276-81.

Grosz B.J. (1996), “Collaborative systems”, *AI Magazine*, v 17, n 2, p 67-85.

Hoffmann F. (2004), “Fuzzy behavior coordination for robot learning from demonstration”, *Annual Meeting of the North American Fuzzy Information Processing Society*, pt 1, v 1, p 157-62.

Hu M. K. (1962), “visual pattern recognition by moment invariants”, *IRE Transactions on Information Theory*, v 8, Issue 2, p179 – 187.

Iba S. (2004), Interactive Multi-Modal Robot Programming, doctoral dissertation, tech. report CMU-RI-TR-04-50, Robotics Institute, Carnegie Mellon University.

Ikezawa K., Konishi Y., and Ishigaki H. (1997), “Development of a teaching system for an industrial robot using stereo vision”, *in Proc. of the SPIE*, v 3203, p 23-32.

Kaiser M. (1997), “Transfer of elementary skill via human-robot interaction”, *Adaptive Behavior*, v 5, n 3-4, p 249-80.

Katagami D. and Yamada, S. (2003), “Active teaching for an interactive learning robot”, *Proc. of 12th IEEE Int. Workshop on Robot and Human Interactive Communication*, p 181-6.

Kieras D. (1994), “A Guide to GOMS Task Analysis”, University of Michigan Technical Report.

Knoop S., Pardowitz M., and Dillmann R. (2007), “Automatic robot programming from learned abstract task knowledge”, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, p 1651-7.

Knoop S., Pardowitz M., and Dillmann R. (2008), “From abstract task knowledge to executable robot programs”, *J. of Intelligent and Robotic Systems*, v 52, n 3-4, p 343-362.

Koenig N. and Mataric M. J. (2006), “Behavior-based segmentation of demonstrated tasks,” in *Proc. of the Int. Conf. on Development and Learning*.

Kofman, J., Verma, S. and Wu, X. (2007). “Robot-Manipulator Teleoperation by Markerless Vision-Based Hand-Arm Tracking”. *Int. J. of Optomechatronics*, n 1(3), p 331-357.

Kofman J., Wu X., Luu T., and Verma S. (2005), "Teleoperation of A Robot Manipulator Using A Vision-Based Human-Robot Interface", *IEEE Trans. on Industrial Electronics*, v 52, n 5, p 1206-19.

Kristensen S., Horstmann S., Klandt J., Lohnert F., and Stopp A. (2001),”Human-friendly interaction for learning and cooperation”, *Proc. of IEEE Int. Conf. on Robotics and Automation*, v 3, p 2590-2595.

Kulic D. and Nakamura Y. (2008), “Scaffolding on-line segmentation of full body human motion patterns”, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, p 2860-6.

Kunii Y. and Hashimoto H. (1997), “Tele-teaching by human demonstration in virtual environment for robotic network system”, *in Proc. 1997 IEEE Int. Conf. on Robotics and Automation*, v 1, pt 1, p 405-10.

Kuniyoshi Y., Inaba M., and Inoue H. (1994), “Learning by watching: extracting reusable task knowledge from visual observation of human performance”, *IEEE Trans. on Robotics and Automation*, v 10, n 6, p 799-822.

Lee M.H., Meng Q., and Holstein H. (2002), “Learning and reuse of experience in behavior-based service robots”, *Proc. of the 7th Int. Conf. on Control, Automation, Robotics and Vision*, v 2, p 1019-102.

Lieberman J. (2004), Teaching a Robot Manipulation Skills Through Demonstration, Masters of Science thesis, Dept. of Mechanical Engineering, MIT.

Lieberman J. and Breazeal C. (2004), “Improvements on action parsing and action interpolation for learning through demonstration”, *in Proc. 2004 4th IEEE/RAS Int. Conf. on Humanoid Robots*, v 1, pt 1, p 342-65.

Lloyd J. and Hayward V. (1991), “Real-time trajectory generation using blending functions”, *in Proc of IEEE Int. Conf. on Robotics and Automation*, v 1, p 784-789.

Lockerd A. and Breazeal C.(2004), “Tutelage and socially guided robot learning”, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pt 4, v 4, p 3475-80.

Martin P. and Nehmzow U. (1995), “Programming by teaching: Neural network control in the Manchester mobile robot”, *in Proc 1995 2nd IFAC Conf. on Intelligent Autonomous Vehicles*, p 277-82.

Meng Q. and Lee M.H. (2003), “Adapting home service robot behaviors by experience reuse and interaction with humans”, *Proc. of 2003 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, pt 1, v 1, p 673-8.

Muto S. and Shimokura K. (1994), “Teaching and control of robot contour-tracking using contact point detection”, *in Proc. 1994 IEEE Int. Conf. on Robotics and Automation*, v 1, pt 1, p 674-81.

Myers D.R., Pritchard M.J., and Brown M.D.J. (2001), “Automated programming of an industrial robot through teach-by showing”, *in Proc. 2001 ICRA. IEEE Int. Conf. on Robotics and Automation*, v 4, pt 4, p 4078-83.

Nakaoka S., Nakazawa A., Kanehiro F., Kaneko K., Morisawa M., Hirukawa H., and Ikeuchi K. (2007), “Learning from observation paradigm: leg task models for enabling a biped humanoid robot to imitate human dances”, *Int. Journal of Robotics Research*, n 8, p 829-44.

Nicolescu M.N. and Mataric M.J. (2001), “Learning and interacting in human-robot domains”, *IEEE Transactions on Systems, Man & Cybernetics*, Part A, v 31, n 5, p 419-30.

Nicolescu M. N. and Mataric M. J. (2003), “Natural Methods for Robot Task Learning: Instructive Demonstrations, Generalization and Practice”, *Proc. of the Int. Conf. on Autonomous Agents*, v 2, p 241-248.

Nicolescu M., Jenkins O.C. and Stanhope A. (2007), “Fusing robot behaviors for human-level tasks”, *2007 6th IEEE Int. Conf. on Development and Learning*, p 340-5.

Ogawara K., Takamatsu J., Kimura H. and Ikeuchi K. (2002), “Generation of a task model by integrating multiple observations of human demonstrations”, *IEEE Int. Conf. on Robotics and Automation*, v 2, p 1545-50.

Otero N., Saunders J., Dautenhahn K. and Nehaniv C. (2008), “Teaching robot companions: the role of scaffolding and event structuring”, *Connection Science*, v 20, n 2-3, p 111-34.

Pardowitz M., Knoop S., Dillmann R. and Zollner R.D. (2007), “Incremental learning of tasks from user demonstrations, past experiences, and vocal comments”, *IEEE Trans. on Systems, Man and Cybernetics,* Part B, v 37, n 2, p 322-32.

Peters R.A. II, Campbell C.L., Bluethmann W.J., and Huber E. (2003), “Robonaut task learning through teleoperation”, *IEEE Int. Conf. on Robotics and Automation*, pt 2, v 2, p 2806-11.

Rusu R.B., Gerkey B. and Beetz M. (2008), "Robots in the kitchen: exploiting ubiquitous sensing and actuation", *Robotics and Autonomous Systems*, v 56, n 10, p 844-56.

Schaal, S (1999). "Is imitation learning the route to humanoid robots?", *Trends in Cognitive Sciences*, v 3, n 6, p 233-242.

Thomaz A. L. and Breazeal C. (2006), "Transparency and socially guided machine learning", *in proc of 5th Int. Conf. on Development and Learning (ICDL)*.

Thomaz A.L. and Breazeal C. (2008), "Experiments in socially guided exploration: lessons learned in building robots that learn with and without human teachers", *Connection Science*, v 20, n 2-3, p 91-110.

Tso S.K. and Liu K.P. (1995), "Automatic generation of robot program codes from perception of human demonstration", *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, pt 1, v1, p 23-8.

Uechi M., Ogata H., Nakamura Y., and Mizukawa M. (1999), "A component architecture for customizing robot-teaching systems", *in Proc. 1999 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, v 3, pt 3, p 1625-32.

Volpe R. (1993), "Task space velocity blending for real-time trajectory generation", *in Proc of IEEE Int Conf on Robotics and Automation*, v 2, p 680-687.

Voyles R.M., Morrow J.D., and Khosla P.K. (1997), "Towards gesture-based programming: shape from motion primordial learning of sensorimotor primitives", *Robotics and Autonomous Systems*, v 22, n 3-4, p 361-75.

Weiss A., Igelsboeck J., Calinon S., Billard A. and Tscheligi M. (2009), "Teaching a Humanoid: A User Study with Hoap3 on Learning by Demonstration", *Proc. of the IEEE Int. Symposium on Robot and Human Interactive Communication.*

Williams T.G., Rowland J.J., Lee M.H., and Neal M.J. (2000), "Teaching by example in food assembly by robot", *Proc. 2000 IEEE Int. Conf. on Robotics and Automation*, v 4, pt 4, p 3247-52.

Wu, X. and Kofman, J. (2008). "Human-Inspired Robot Task Learning from Human Teaching", *proc. of IEEE Int. Conf. on Robotics and Automation (ICRA 2008)*, Pasadena, CA, USA, p 3334-39.

Xi N., Tarn T. and Bejczy A.K. (1996), "Intelligent planning and control for multirobot coordination: an event-based approach", *IEEE Trans on Robotics and Automation*, v 12, n 3, p 439-452.

Yeasin M. and Chaudhuri S. (1997), "Automatic robot programming by visual demonstration of task execution", *Proc. 8th Int. Conf. on Advanced Robotics* (*ICAR'97*), p 913-18.

Zoellner, R., Pardowitz, M., Knoop, S., Dillmann, R. (2005), "Towards cognitive robots: building hierarchical task representations of manipulations from human demonstration", *IEEE Int. Conf. on Robotics and Automation (ICRA 2005)*, Barcelona, Spain, p 1535–1540.

Appendix A: Training Material for User Study

A.1 Sample Human-Robot Vocal Interaction Scripts Provided to Human Participants for Training Purpose

The human-robot vocal interaction scripts below are only a sample of the available utterances (vocabulary and phrases) allowed and understood by the human-robot teaching and learning system.

Script for Vocal Robot-Human Dialogues

Notes:1) ***[...]*** means content inside the square bracket is optional
2) ***(...)*** means content inside the parenthesis cannot be separated
3) **+** means the continuation of the last line
4) ***/*** means you must select one of them
5) ***ACTION***, or others in the blue color and bold font, means to select one from a list of pre-specified actions or others, either through database or pre-learning.
5) ***ROLES***, or others with under line and in the blue color and bold font, takes whatever the user speaks, i.e Dictation

1. Common Dialogues

1.1 Task Name

Format 1: ***ACTION [the] OBJECT***
e.g. "place the knife", "grasp knife", and "approach the knife"

Format 2: ***ACTION [the] OBJECT DIRECTION_PREPOSITION [the] TARGET***
e.g. "place the knife onto the placemat"

Format 3: ***ACTION DIRECTION_PREPOSITION [the] OBJECT***
e.g. "pick up knife"

1.2 Acknowledgement

Positive:
yes, it is
yes, please
yes, I do
yes, you are [right]
…

Negative:
no, it is not [right]
no, it isn't [right]
no, I do not
no, I don't
no, you are wrong

no, you are not right

1.3 Teleoperation

Request the robot to follow or imitate your hand motion:

Teleoperation system follow me
Teleoperation follow me

Request the robot NOT to follow or imitate your hand motion

Teleoperation system stop following me
Teleoperation stop following me

1.4 Trigger to Pour Liquid

Start to pour to left:

start to pour to left

Start to pour to right:

start to pour to right

Stop pouring:

stop pouring

1.5 Navigation through Different Teaching and Learning Stages

go to the next teaching and learning stage
go to the last teaching and learning stage
go to the previous teaching and learning stage

go to the check background knowledge stage
go to the introduce overall task stage
go to the demonstrate task in whole stage
go to the practice task stage

2. Greeting

2.1 Greeting

hi/hello robot
good morning/afternoon robot
robot good morning/ afternoon

2.2 Request Robot to Perform a Task

please perform the task TASK_NAME for me
please execute the task TASK_NAME for me

2.3 Tell Robot to Teach It a Task

i'd like to teach you a task
i would like to teach you a task
i am going to teach you a task
i will teach you a task

```
i am to              teach you a task
```

3. Teaching

3.1 Check/Teach Background Knowledge

3.1.1 Switch Focus

```
switch to objects
switch to tasks
```

Note 1: the default focus is on objects, which means that when you first enter this stage you are only allowed to check/teach background knowledge of the robot regarding the objects.

Note 2: if you want to focus on tasks, please say to the robot “switch to tasks”. Please say to the robot “switch to objects” if you want to focus on objects.

3.1.2 Regarding Objects

```
<!--background knowledge about object includes:
  * name
  * colors
  * functional role,
  * grasp pose (or graspable)
  * ask robot to give/(teach robot) name of object pointed
-->
```

a). Check general knowledge of all objects or a specific object

```
what objects do you know about
tell me what objects you know about

what do you know         about OBJECT_NAME
do you know anything     about OBJECT_NAME
tell me anything         about OBJECT_NAME
tell me your knowledge   about OBJECT_NAME
```

b). Check knowledge of object pointed or specified by name

```
tell me its name/(main color)
tell me its docking/grasping pose
tell me its typical staying places
  # i.e. where you would normally find it
tell me where you would normally find it

tell me the object’s name/(main color)
tell me the object’s docking/grasping pose
tell me the object’s typical staying places
tell me where you would normally find the object

do you know its name/(main color)/(docking pose)/(grasping pose)
do you know its typical staying places
do you know where you would normally find it
```

```
do you know the object's name/(main color)
do you know the object's docking/grasping pose
do you know the object's typical staying places
do you know where you would normally find the object

tell me the name/(main color)      of the object
tell me the docking/grasping pose  of the object
tell me the typical staying places of the object

tell me the main color             of OBJECT_NAME
tell me the docking/grasping pose  of OBJECT_NAME
tell me the typical staying places of OBJECT_NAME

do you know the name/(main color)      of the object
do you know the docking/grasping pose  of the object
do you know the typical staying places of the object

do you know the main color             of OBJECT_NAME
do you know the docking/grasping pose  of OBJECT_NAME
do you know the typical staying places of OBJECT_NAME
```

c). Teach or modify knowledge of object pointed or pre-specified

```
# object name
  the [object] name is        OBJECT_NAME
  the [object] name should be OBJECT_NAME

# main object color
  the main [object] color should be COLOR_NAME
  the main [object] color is        COLOR_NAME

# functional roles of the object
  the/its functional role  is         ROLES
  the/its functional roles are        ROLES
  the/its functional roles should be  ROLES

# typical places of the object
  the/its typical places is                   PLACES
  the/its typical staying places is           PLACES
  the/its typical staying places should be    PLACES
```

Notes: 1). #: leads a line of a comment
2). to point to an object, you must first say "teleoperation system follow me"; then teleoperate the robot end effector over to the object; and next, say "teleoperation system stop following me"

3.1.3 Regarding Tasks

```
<!-- background knowledge about task or subtask:
  * task or subtask name
```

```
   * # of child subtask or steps
   * actions involved
   * objects involved
-->
```

a). Check general knowledge of all tasks or a specific task

```
what tasks do you know about
tell me what tasks you know about

what do you do you know about TASK_NAME
do you know anything about    TASK_NAME

tell me anything about       TASK_NAME
tell me your knowledge about TASK_NAME
```

b). Check knowledge of a pre-specified task or the task specified by the given task name

Overall task/subtask knowledge

```
tell me your knowledge about the task/subtask
```

The number of subtasks

```
tell me the number of subtasks/steps the task/subtask has/includes
tell me how many      subtasks/steps the task/subtask has/includes

do you know the number of subtasks/steps the task    has/includes
do you know the number of subtasks/steps the subtask has/includes
do you know how many      subtasks/steps the task    has/includes
do you know how many      subtasks/steps the subtask has/includes

tell me the number of subtasks/steps of TASK_NAME
tell me how many      subtasks/steps    TASK_NAME has/includes

do you know the number of subtasks/steps of TASK_NAME
do you know the number of subtasks/steps    TASK_NAME has/includes
do you know how many      subtasks/steps    TASK_NAME has/includes
```

Task/subtask name

```
tell me     the/its task/subtask  name
do you know the/its task/subtask  name

tell me the [task]    name of the task
tell me the [subtask] name of the subtask

do you know the [subtask] name of the subtask
do you know the [task]    name of the task
```

Involved robot hand actions

```
tell me the/its                involved actions
tell me the task's/subtask's  involved actions
```

```
do you know the/its               involved actions
do you know the task's/subtask's  involved actions

tell me     the involved actions of TASK_NAME
do you know the involved actions of TASK_NAME
```

Involved objects

```
tell me the/its               involved objects
tell me the task's/subtask's  involved objects

do you know the/its               involved objects
do you know the task's/subtask's  involved objects

tell me     the involved objects of TASK_NAME
do you know the involved objects of TASK_NAME
```

c). Teach or modify knowledge of pre-specified task or subtask

Task or subtask name

```
the task/subtask name is/(should be) TASK_NAME
```

The number of subtasks

```
the task/subtask has         X  steps/subtasks
the task/subtask includes    X  steps/subtasks
the task/subtask consists of X  steps/subtasks

the subtask has 0 steps
the subtask is a primitive subtask (i.e. 0 steps)
```

It has no subtasks.

```
the subtask is a primitive subtask
```

Involved robot hand actions

```
it                   has no hand action involved
the step/subtasks    has no hand action involved
there is  no hand action [ involved in the subtask ]

the involved [robot] hand action is open/close hand

the involved [robot] hand actions include close hand and open
  hand
the involved [robot] hand actions include open  hand and close
  hand

the involved [robot] hand actions are     close hand and open
  hand
the involved [robot] hand actions are     open hand  and close
  hand
```

Note: **hand** and **gripper** are interchangeable here.

Involved object

```
the involved object  is/(should be)   OBJECT_NAME
the involved objects are/(should be)  OBJECT_NAME and OBJECT_NAME
the involved objects include         OBJECT_NAME and OBJECT_NAME
```

Note: you can specify at most two involved objects and robot hand actions, for a given task or subtask

3.2 Overall Task Introduction

You first need to know that the robot will basically control the dialogue in this teaching and learning stage, i.e. you just follow the dialogue flow. Of course, you can navigate through the tree-shape task structure using proper voice instructions.

The content of the questions the robot will ask and your allowable responses are listed as follows:

Task name

```
the task/subtask name is TASK_NAME
its              name is TASK_NAME
it                    is TASK_NAME
```

Number of subtask(s)/step(s)

```
the task/subtask has         X  steps/subtasks
the task/subtask includes    X  steps/subtasks
the task/subtask consists of X  steps/subtasks

the task/subtask has         X  child steps/subtasks
the task/subtask includes    X  child steps/subtasks
the task/subtask consists of X  child steps/subtasks

it               has         X  steps/subtasks
it               includes    X  steps/subtasks
it               consists of X  steps/subtasks

it           is a primitive subtask    #i.e. the subtask has 0 steps
the subtask  is a primitive subtask
```

Involved robot hand actions, only asked for primitive subtasks

```
it                has no hand action involved
the step/subtask  has no hand action involved
there             is  no hand action [involved in the subtask]

the involved [robot] hand action is open hand
the involved [robot] hand action is close hand

the involved [robot] hand actions include/are close hand
 + and open hand
the involved [robot] hand actions include/are open  hand
 + and close hand
the involved [robot] hand actions consist of  close hand
```

+ and open hand
the involved [robot] hand actions consist of open hand
+ and close hand

Note: **hand** and **gripper** are interchangeable here.

\# You may use follow instruction to navigate through task structure

go to its child/parent subtask/step
go to its next/last/previous subtask/step

3.4 Demonstration in Whole

a). To demonstrate the task, at the beginning, you have to say to the robot:

teleoperation system follow me

which asks the robot to imitate your human hand motion. When the demonstration is finished, you must say to the robot:

teleoperation system stop following me

which ask the robot to stop imitating your human hand motion.

b). During the demonstration, you can give some vocal segmentation cues in the subtask transitions, using the following utterances:

next step
next subtask

c). Pouring liquid task

When ready to start to pour liquid to left, say to robot:

start to pour to left

When ready to start to pour liquid to right, say to robot:

start to pour to right

To terminate the pouring, say to the robot:

stop pouring

Note: during the pouring, you can still control the position of the liquid container (by moving your hand), but not the orientation. You should not rotate your hand.

3.5 Task Generalization and Learning

If the robot has difficulty to segment the demonstrated task, it will ask for help by repeating your demonstration and showing its current segmentation of the task. During the robot's demonstration, you can give following vocal instructions:

1). Ask robot to pause at its current position

stop here

2). Ask the robot to move forward

segmentation move forward

3). Ask the robot to move backward

segmentation move backward

4). Assign the last or next task path segment to a primitive subtask

```
the last/next [path] segment/episode should belong      to TASK_NAME
the last/next [path] segment/episode belongs            to TASK_NAME
the last/next [path] segment/episode is assigned        to TASK_NAME
the last/next [path] segment/episode should be assigned to TASK_NAME
  # e.g. the last episode should belong to "pick up knife"
the [path] segment just shown [to me] belongs           to TASK_NAME
the [path] segment just shown [to me] should belong     to TASK_NAME
```

Note: ***TASK_NAME*** means one instructed primitive subtask name, here.

3.6 Task Practice

1). Modify task segmentation

During practice trials with half the demonstrated speed, the robot speaks out the primitive subtask name at the beginning of its execution. You, the teacher, can modify the segmentation as needed, with vocal instructions as in Section 3.5.

2). Timely feedback on task trajectory motion

7 degrees of freedom — 3 translations + 3 rotations + 1 speed.

All the translations and rotations are defined in the robot tool (gripper or end-effector) frame, as illustrated in the following Figure 1. In other words, your feedback instructions will be interpreted with regarding to the robot end-effector frame.

Translations in robot tool frame

```
move [more] forward    # + X
move [more] backward   # - X

move [more] left       # + Y
move [more] right      # - Y

move [more] up         # + Z
move [more] down       # - Z
```

Rotation in robot tool frame

```
roll [more] right      # +X rotation
roll [more] left       # -X rotation

turn/title [more] down       # +Y rotation
```

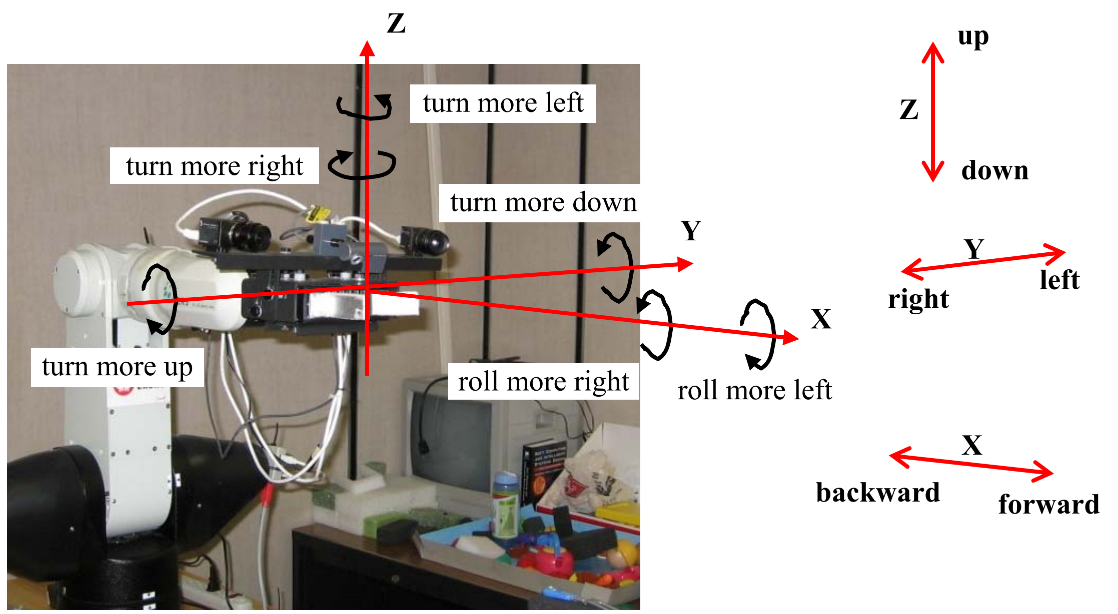
```
turn/title [more] up        # -Y rotation

turn [more] left        # +Z rotation
turn [more] right       # -Z rotation
```

\# to increase the motion speeds

```
go/move   faster
```

\# to reduce the motion speeds

```
go/move   slower
```



Robot tool frame and rotations ***Translations in robot tool frame***

Figure 1: Definitions of the robot tool (gripper / end-effector) frame, and directions of the translations and rotations.

A.2 Excerpt of Vocal Human-Robot Interaction from Training Demonstration Video

The task taught in the demonstration video for training purposes in the user study was "Lay out Table", and the task structure introduced in the overall task introduction teaching stage is depicted in Figure 2.

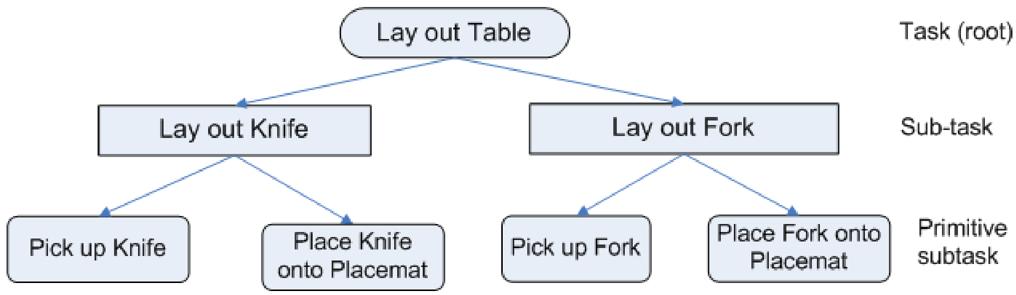


Figure 2: Task structure of the task, “Lay out Table”, which was taught in the demonstration video for training purposes in the user study.

The vocal human-robot interaction in the first two teaching and learning stages (checking/ teaching needed background and overall task introduction) in the demonstration video, is illustrated as follows:

Greet and request the robot to learn a task from human teaching

Teacher (**T**): *Hi robot*

Robot (**R**): *Hi, what can I do for you?*

T: *Good afternoon, robot*

R: *Good afternoon, what can I do for you?*

T: *I am to teach you a task*

R: *That is great. Let's start.*

Please stand by, the robot is to explore its workspace, and detect objects of interest.

Please put involved objects in place. Then say 'It is ready' to me.

T: *It is ready*

Robot went to explore objects in its working space

Teaching Stage I: Check and Teach Needed Background Knowledge

R: *Do you want to test my background knowledge needed for the new task?*

T: *Yes, I do.*

R: *You can now ask me questions about my background knowledge regarding objects and tasks, such as object name, color, typical places it would be found, functional roles; task name, goal, number of subtasks it has, and so on.*

If you want to go to next teaching stage, please say to me 'go to next teaching stage'.

Note that if you want to move the robot hand to point an object, you have to say 'Teleoperation follow me' at beginning of the movement and then say 'Teleoperation stop following me' at the end of the movement.

We are now focusing on Objects. If you want to ask questions regarding Tasks, please say 'switch to tasks' to me.

T: *What objects do you know about?*

R: *There are 7 objects I can identify: table, knife, fork, placemat, bottle, cup, and spatula. There are also 9 objects I can not identify: object, target, block, teapot, spoon, juice, container, water, and banana.*

T: *Tell me your knowledge about knife.*

R: *The name of the object is KNIFE. Its task-related functional role is CUT FOOD. It has one DOCKING POSE and GRASPING POSE respectively.*

T: *Tell me its COLOR.*

R: *Its main color is PINK.*

T: *Tell me its typical staying places.*

R: *Its typical staying places are not specified yet.*

T: *Its typical staying places is TABLE*

R: *The typical staying places are changed to TABLE.*

T: *Tell me its functional role.*

R: *Its task-related functional role is CUT FOOD.*

T: *Tell me your knowledge about FORK*

R: *The name of the object is FORK. Its task-related functional role is IT FOR. Its typical staying place is TABLE. It has one DOCKING POSE and GRASPING POSE respectively.*

T: *Switch to tasks*

R: *We are now focusing on Tasks. If you want to focus on Objects, please say 'switch to objects' to me.*

T: *Tell me what tasks you know about.*

R: *I have learned 2 tasks: SET THE TABLE and POUR JUICE.*

T: *Tell me your knowledge about POUR JUICE.*

R: *Its name is POUR JUICE. Its goal is POUR JUICE. It has 3 subtasks: PICK UP BOTTLE, MOVE BOTTLE TO CUP, and PLACE BOTTLE ONTO TABLE. TWO robot gripper actions are involved. THREE objects are involved.*

T: *How many subtasks does it have?*

R: *It has 3 subtasks: PICK UP BOTTLE, MOVE BOTTLE TO CUP, and PLACE BOTTLE ONTO TABLE.*

R: *Exploration is finished. 4 objects are detected.* **←Robot finished an exploration**

T: *Go to next teaching and learning stage.*

Teaching Stage II: Introduce Overall Task

R: *We are now in the teaching stage of overall task introduction. You are going to use voice dialogues to introduce the task structure to me.*

Please speak carefully, use complete sentences only, and do NOT give your response while I am speaking.

- Building the root node of the task structure, "Lay out Table"

R: *Please tell me the NAME of this task.*

T: *The task name is LAY OUT TABLE.*

R: *The task name is LAY OUT TABLE. Is this right?*

T: *Yes, you are.*

R: *How many STEPs does this task include?*

T: *It has TWO steps.*

R: *The task has TWO steps. Is this right?*

T: *Yes, it is.*

- Building the FIRST subtask of the task: "Lay out Table"

R: *Now, we are to build the FIRST step of the task: lay out table.*

R: *Please tell me the NAME of this subtask.*

T: *The subtask name is LAY OUT KNIFE.*

R: *The subtask name is LAY OUT KNIFE. Is this right?*

T: *Yes, it is.*

R: *How many steps does this subtask include?*

T: *It has TWO steps.*

R: *The subtask has TWO steps. Is this right?*

T: *Yes, you are right.*

- Building the SECOND subtask of the task: "Lay out Table"

R: *Now, we are to build the SECOND step of the task: lay out table.*

R: *Please tell me the NAME of this subtask.*

T: *It is LAY OUT FORK.*

R: *The subtask name is LAY OUT FORK. Is this right?*

T: *Yes, you are.*

R: *How many STEPs does this subtask include?*

H: *It includes TWO subtasks.*

R: *The subtask has TWO steps. Is this right?*

H: *Yes, you are right.*

- Building the FIRST child subtask of the subtask, "Lay out Knife"

R: *Now, we are to build the FIRST step of the subtask: lay out knife.*

R: *Please tell me the NAME of this subtask.*

T: *The first subtask is PICK UP KNIFE.*

R: *The subtask name is PICK UP KNIFE. Is this right?*

T: *Yes, it is.*

R: *How many STEPs does this subtask include?*

T: *It is a PRIMITIVE subtask.*

R: *The subtask has ZERO steps. Is this right?*

T: *Yes, you are right.*

R: *Please tell me the HAND ACTION involved in this primitive subtask.*

T: *The hand action is CLOSE GRIPPER.*

R: *The involved robot hand action is CLOSE GRIPPER. Is this right?*

T: *Yes, you are right.*

- Building the SECOND child subtask of the subtask, "Lay out Knife"

R: *Now, we are to build the SECOND step of the subtask: lay out knife.*

R: *Please tell me the NAME of this subtask.*

T: *It is PLACE KNIFE ONTO PLACEMAT.*

R: *The subtask name is PLACE KNIFE ONTO PLACEMAT. Is this right?*

T: *Yes, it is.*

R: *How many STEPs does this subtask include?*

T: *It has ZERO subtasks.*

R: *The subtask has ZERO steps. Is this right?*

T: *Yes, you are right.*

R: *Please tell me the HAND ACTION involved in this primitive subtask.*

T: *The involved hand action is OPEN HAND.*

R: *The involved robot hand action is OPEN GRIPPER. Is this right?*

T: *Yes, it is.*

- Building the FIRST child subtask of the subtask, "Lay out Fork"

R: *Now, we are to build the FIRST step of the subtask: lay out fork.*

R: *Please tell me the NAME of this subtask.*

T: *The first subtask is PICK UP FORK.*

R: *The subtask name is PICK UP FORK. Is this right?*

T: *Yes, it is.*

R: *How many STEPs does this subtask include?*

T: *It is a PRIMITIVE subtask.*

R: *The subtask has ZERO steps. Is this right?*

T: *Yes, you are right.*

R: *Please tell me the HAND ACTION involved in this primitive subtask.*

T: *The involved hand action is CLOSE HAND.*

R: *The involved robot hand action is CLOSE GRIPPER. Is this right?*

T: *Yes, you are.*

- Building the SECOND child subtask of the subtask, "Lay out Fork"

R: *Now, we are to build the SECOND step of the subtask: lay out fork.*

R: *Please tell me the NAME of this subtask.*

T: *The subtask name is PLACE THIS FORK ONTO PLACEMAT.*

R: *The subtask name is PLACE THIS FORK ONTO PLACEMAT. Is this right?*

T: *Yes, you are right.*

R: *How many STEPS does this subtask include?*

T: *It is a PRIMITIVE subtask.*

R: *The subtask has ZERO steps. Is this right?*

T: *Yes, you are right.*

R: *Please tell me the HAND ACTION involved in this primitive subtask.*

T: *The involved hand action is OPEN HAND.*

R: *The involved robot hand action is OPEN GRIPPER. Is this right?*

T: *Yes, you are right.*

R: *Thank you. We are going to the next teaching and learning stage.*

A.3 Demonstration Video of a Task Teaching and Learning

Video is available upon request.

Appendix B: User Study Support Document

B.1 Institutional Review Board Ethics Approval Certificate

The ethics clearances for the user study application and modification to the recruiting process for human participants are attached in the following two subsections.

B.1.1 Ethics Clearance (ORE # 15219)

From: ORE Ethics Application System ()
To: jkofman
Date: Tuesday, December 2, 2008 12:09:52 PM
Cc: x9wu
Subject: Ethics Clearance (ORE # 15219)

Dear Researcher:
The recommended revisions/additional information requested in the ethics review of your ORE application:

Title: Human-Inspired Service Robot Task Teaching and Learning
ORE #: 15219
Faculty Supervisor: Jonathan Kofman
Student Investigator: Xianghai Wu

have been reviewed and are considered acceptable. As a result, your application now has received full ethics clearance.

A signed copy of the Notification of Full Ethics Clearance will be sent to the Principal Investigator or Faculty Supervisor in the case of student research.

****
Note 1: This clearance is valid for four years from the date shown on the certificate and a new application must be submitted for on-going projects continuing beyond four years.

Note 2: This project must be conducted according to the application description and revised materials for which ethics clearance have been granted. All subsequent modifications to the protocol must receive prior ethics clearance through our office and must not begin until notification has been received.

Note 3: Researchers must submit a Progress Report on Continuing Human Research Projects (ORE Form 105) annually for all ongoing research projects. In addition, researchers must submit a Form 105 at the conclusion of the project if it continues for less than a year.

Note 4: Any events related to the procedures used that adversely affect participants must be reported immediately to the ORE using ORE Form 106.

Best wishes for success with this study.

Susanne Santi, M. Math.,
Senior Manager
Office of Research Ethics
NH 1027
519.888.4567 x 37163

B.1.2 Ethics Clearance of Modifications

From: ORE Ethics Application System ()
To: jkofman
Date: Friday, December 19, 2008 9:39:41 AM
Cc: x9wu
Subject: Ethics Clearance of Modifications, no comments (ORE # 15219)

Dear Researcher:

A Request for ethics review of a modification or admendment (ORE 104) to your ORE application:

Title: Human-Inspired Service Robot Task Teaching and Learning
ORE #: 15219
Faculty Supervisor: Jonathan Kofman
Student Investigator: Xianghai Wu

----
together with a copy of relevant materials, was received in the Office of Research Ethics on: 18 December 2008 -- also recruit directly using email script; minor changes to script for easier reading

The proposed modification request has been reviewed and has received full ethics clearance.

A signed copy of the 'Request for Ethics Clearance of a Modification to an Ongoing Application to Conduct Research with Human Participants' will be provided through regular mail. In the case of student research, the signed copy will be sent to the Faculty Supervisor.

****
Note 1: This project must be conducted in accordance with the description in the application and modification for which ethics clearance has been granted. All subsequent modifications to the protocol must receive prior ethics clearance through the Office of Research Ethics.

Note 2: Researchers must submit a Progress Report on Continuing Human Research Projects (ORE Form 105) annually for all ongoing research projects. In addition, researchers must submit a Form 105 at the conclusion of the project if it continues for less than a year.

Note 3: Any events related to the procedures used that adversely affect participants must be reported immediately to the ORE using ORE Form 106.

Susanne Santi, M. Math.,
Senior Manager
Office of Research Ethics
NH 1027
519.888.4567 x 37163

B.2 Recruitment Script

Recruitment Script

Hi,

We are recruiting some volunteers to try out a new human-inspired robot teaching system. The goal is to enable general users to teach different object-manipulation tasks to a robot, and allow the robot to adapt its learned tasks to new setups. The session will take approximately two hours. You will use natural language (English) to interact with the robot. You will demonstrate some object-manipulation tasks to the robot by tele-operating (controlling) the robot. To do this, you will use natural hand motion as if you were completing a task. Your hand motion will be observed by multiple cameras and the position information will be sent to the robot to imitate your motion. The robot learns the tasks from the demonstrations and then generalizes the taught tasks and organizes the task knowledge. The robot teaching experience should be quite interesting.

The experiments will be conducted at Intelligent Human-Machine Systems Laboratory. To ensure your safety, you will always be kept out of the robot's reach. A coloured tape has been placed to separate the robot working area from your working area, to prohibit you from entering the robot area.

This project was reviewed and received ethics clearance through the Office of Research Ethics, University of Waterloo.

If you are interested in the study or want to know more information regarding this study, please contact Mr. Xianghai Wu (Tel. 888-4567 ext 37839;
Email: x9wu

Thank you very much,

Xianghai Wu
Tel. 519-888-4567 ext 37839
Email: x9wu

B.3 Information Letter

ORE #_____15219______

Information letter

University of Waterloo

November 14, 2008

Title of Project: Testing of a Service-Robot Teaching and Learning System

Faculty Supervisor: Professor Jonathan Kofman
University of Waterloo, Department of Systems Design Engineering,
519-884-4567 Ext. 35185

Student Investigator: Xianghai Wu
University of Waterloo, Department of Systems Design Engineering,
519-884-4567 Ext. 37839

Purpose of this Study

The goal of this study is to test the effectiveness of a human-robot teaching and learning system, which allows you to teach a robot new tasks, and the robot to learn the tasks and adapt its learned task to new task setups. The study will specifically assess:

(a) how easy it is to teach a task, including how intuitive and effective the system is in allowing you to introduce the task structure, demonstrate the task to the robot, give vocal task-segmentation cues during task demonstration, and offer timely feedback during the robot's task practice;

(b) how well the robot is able to understand the taught task, including how well the robot constructs the task structure from your overall task introduction and how well the robot segments the taught task into subtasks;

(c) how well the robot can perform the taught task, including how well the robot grasps objects of interest and places them onto a target by measuring the alignment between the placed objects and target, how well the robot adapts its learned tasks to new task setups, and how well the robot refines its task knowledge responding to the teacher's timely feedback.

(d) how much you accept the system as a method of teaching a robot tasks, i.e. the degree of intuitiveness, effectiveness, and frustration.

Procedures Involved in this Study

You will use natural language (English) to interact with the robot and utilize your natural hand motion to tele-operate (control) the robot. You will demonstrate some object-manipulation tasks to the robot by moving your hand as if you were completing the tasks yourself. Your natural hand motion will be observed by multiple cameras and the hand position information will be sent to the robot to imitate your motion. The robot learns the tasks from your demonstrations, and then generalizes the taught tasks, organizes the task knowledge, and practices the taught tasks. During the study, to ensure your safety, you will always be kept out of the robot's reach. A coloured tape has been placed to separate the robot working area from your working area, to prohibit you from entering the robot area.

You will be asked to complete the following procedures:

1) complete a standard training session of the Microsoft Speech Recognition on Windows XP. A training profile, characterizing your speech, will be generated as a result, and it will be used during your participation in the study. This profile will be deleted immediately after your participation in this study.

2) receive a brief introduction to the system as well as scripts to use for the human-robot vocal interaction. The student investigator will explain the scripts and how they will be used during the session.

3) watch a video of a demonstration of this experiment, conducted by Mr. Wu.

4) pre-analyze an object-manipulation task, including its task structure and its approximate task path in the space.

5) check and teach the robot's needed background knowledge via voice dialogue, i.e. ask the robot information regarding the objects and tasks that it knows.

6) introduce the overall task to be taught to the robot via voice dialogue, i.e. explain to the robot the task structure and involved robot hand actions. The robot will then build the task structure.

7) teach the task to the robot step by step, i.e. demonstrate subtasks via teleoperation while giving some vocal subtask-segmentation cues in the subtask transitions, such as using the utterance "first subtask" and "next step", if needed.

8) demonstrate the task in whole while giving some vocal cues in the subtask transitions.

9) give feedback and guidance to the robot on its motion in a timely manner, by uttering phrases to the robot such as "move faster" and "move more left/right", while the robot is practising the tasks.

10) request the robot to perform the newly learned task with new task setups.

11) complete a short questionnaire about your experience in interacting with the robot.

Risks to Participation and Associated Safeguards

There will be no known or anticipated risks since you will use natural language to interact with the computer, your natural hand motion will be observed by a non-invasive vision (camera-based) system and will be used to teleoperate the robot. The robot could potentially harm you if you enter the robot area. To ensure your safety, you are prohibited from entering the robot working area. A coloured tape is placed to separate the robot working area from your working area and to prohibit and warn you from accessing the robot. In your working area, you will always be out of the robot's reach. Furthermore, the student Investigator, Mr. Xianghai Wu, will be monitoring all of the experiments, making sure that you do not cross the tape-marked area. You will therefore have no difficulty remaining in the safe area. As an additional safety measure, the student investigator will have his hand placed over an emergency stop button throughout the experiment. Pressing this emergency stop button at any time during the experiment will instantly stop all motion of the robot. You will also have access to an emergency stop button that you can also press, and it is in your working area, out of the reach of the robot arm.

You may experience some fatigue in your arms. To ensure that there is no discomfort or injury due to overuse, you will be able to: stop the demonstration at any time with a hand gesture, take a resting break for as long as you need, and then continue the demonstration from where you paused when you are ready. You will be instructed to rest as much as needed. There are no repetitive motions involved. The student Investigator will be monitoring all of the experiments, and reminding you to rest at any time you desire.

You do not need to be worried or stressed about possibly damaging the robot because the student Investigator will be monitoring all of the experiments and he can press the emergency stop button at any time to stop all robot motion.

Time Commitment

Participation in this study will take approximately 2 hours of your time.

Changing Your Mind about Participation

Even if you do choose to take part now, you are still free to withdraw from this study at any time and for any reason without penalty. You can do this by simply telling Mr. Xianghai Wu, "I no longer wish to participate in this study".

Your decision – either to take part in the study or to leave the study – will have no effect on your study and research at the University of Waterloo, now or in the future.

Benefits of Participation

This study will likely be very interesting for you and it will give you an opportunity to participate in exciting leading edge technology in human-robot interaction and robot learning from human demonstration. The result of this study will aid in developing a multiply functional service or personal robot to improve the quality of life of people who require assistance, especially in the home.

Confidentiality

To ensure the confidentiality of individuals' data, each participant will be identified by a participant identification code known only to the faculty supervisor and his research assistants. Paper records and electronic data regarding your participation in this study will be securely kept in a secure cabinet at Intelligent Human-Machine System Laboratory, and will be confidentially shredded after 10 years. Videotapes and/or photographs will be stored indefinitely in the secure cabinet. A separate consent will be requested in order to use the videotapes and/or photographs for teaching, for scientific presentations, or in publications of this work.

Participant Feedback

Any scientific articles prepared for presentation and/or publication based on this research will be sent to you if you provide an e-mail address to the student investigator or faculty supervisor.

Concerns about Your Participation
I would like to assure you that this study has been reviewed and received ethics clearance through the Office of Research Ethics, University of Waterloo. However, the final decision about participation is yours. If you have any comments or concerns resulting from your participation in this study, you may contact Dr. Susan Sykes, Director ORE, at (-519-888-4567 ext. 36005.

Questions About the Study

If you have additional questions later or want any other information regarding this study, please contact Professor Jonathan Kofman at 519-888-4567 ext. 35185.

B.4 Consent Form

ORE #____15219______

CONSENT TO PARTICIPATE

I agree to take part in a research study being conducted by Dr. Jonathan Kofman and Mr. Xianghai Wu of the Department of Systems Design Engineering, University of Waterloo.

I have made this decision based on the information I have read in the Information letter. All the procedures, any risks and benefits have been explained to me. I have had the opportunity to ask any questions and to receive any additional details I wanted about the study. If I have questions later about the study, I can ask one of the researchers (Dr. Jonathan Kofman, 519-888-4567 ext. 35185; and Mr. Xianghai Wu, 519-888-4567 ext. 37839).

I understand that I may withdraw from the study at any time without penalty by telling the researcher.

This project has been reviewed by, and received ethics clearance through, the Office of Research Ethics at the University of Waterloo. I may contact this office (519-888-4567, ext. 36005) if I have any concerns or questions resulting from my involvement in this study.

______________________________ ______________________________

Printed Name of Participant Signature of Participant

______________________________ ______________________________

Dated at Waterloo, Ontario Witnessed

Assigned number (used internally) _______________

ORE #____15219______

Consent to Use Video and/or Photographs

Sometimes a certain photograph and/or part of a video-tape clearly shows a particular feature or detail that would be helpful in teaching or when presenting the study results in a scientific presentation or publication. If you grant permission for photographs or videotapes in which you appear to be used in this manner, please complete the following section.

I agree to allow video and/or photographs to be used in teaching or scientific presentations, or published in scientific journals or professional publications of this work without identifying me by name.

______________________________ ______________________________

Printed Name of Participant Signature of Participant

______________________________ ______________________________

Dated at Waterloo, Ontario Witnessed

Assigned number (used internally) _______________

B.5 Questionnaire

ORE #_____15219______

User Questionnaire

Robotics Experience:
1. Have you ever tele-operated a robot? ___ yes ___ no

2. Have you ever programmed a robot? ___ yes ___ no

If yes, please describe the robot and its task:

Subjective Feedbacks, scale 1 (do not agree) to 5 (strongly agree)

1. The provided training (including video and instruction) is adequate. ______

2. The system is intuitive to use. ______

3. The system is effective overall. ______

4. The demonstrated tasks were segmented (broken into steps) and assigned as you expected. _____

5. The robot responded well to your feedback overall. ________

   a) robot responded **promptly** (i.e. without delay). ________

   b) robot responded with appropriate **magnitudes.** _____

   c) robot responded with motion in the **direction** you expected. ________

   d) the given vocal utterances (such as "move more up" and "turn more right") for your feedback were intuitive to you. ________

6. The robot refined its task knowledge and applied the new modification later on during a robot task repeated practice and execution. ______

7. The system adapted its learned task to new setups (i.e. different locations and orientations of the objects, targets, and the robot's starting position and orientation). _____

8. How frustrated were you in using the system overall? ______
(1: very frustrated; 5: not frustrated)