IMPROVING ROBOT LEARNING USING AN ACTIVE LEARNING APPROACH IN A LEARNING FROM DEMONSTRATION FRAMEWORK

By

PREETI LEKHA

Bachelor of Engineering in Information Science and Engineering

Visvesvaraya Technological University

Belagavi, Karnataka, India

2011

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 2017

IMPROVING ROBOT'S LEARNING USING ACTIVE LEARNING APPROACH IN LEARNING FROM DEMONSTRATION FRAMEWORK

Thesis Approved:

Dr. Christopher John Crick

Thesis Adviser

Dr. David Cline

Dr. Johnson P Thomas

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude towards my Adviser Dr. Christopher Crick for his guidance, valuable advice at every phase of the problem formulation, and constant support. I would also like to thank my other committee members, Dr. David Cline and Dr. Johnson Thomas for their valuable comments and advices. I would also like to thank my lab mates for their advices and help, any time I needed. Lastly, I would like to thank my parents for their love and support. I dedicate this thesis to my family.

Name: PREETI LEKHA

Date of Degree: MAY, 2017

Title of Study: IMPROVING ROBOT LEARNING USING AN ACTIVE LEARNING APPROACH IN A LEARNING FROM DEMONSTRATION FRAMEWORK

Major Field: COMPUTER SCIENCE

Abstract: Programming new abilities on a robot ought to take negligible time and exertion. One way to accomplish this objective is to permit the robot to ask questions. This idea, called Active Learning, has recently caught a lot of attention in the robotics community. We are interested in the potential of active learning to improve learned skills from human demonstrations in an HRI setting. In this thesis, I explore different types of queries proposed in the Active Learning(AL) Literature and apply them to Learning from Demonstration(LfD) problems. The central part of this work is to design a strategy for data selection for the query in order to avoid unnecessary and redundant queries, select different types of query that will help the robot learn better and explore how the incorporation of AL methods in LfD impacts a robot's learning and performance.

TABLE OF CONTENTS

Chapter Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In existing and future human-robot environments, robots may play various roles. For example, some robots may primarily interact with robots, while others may interact directly with people. Designing controllers for various types of tasks performed by robots is usually done by people specialized in programming robots. Most often this is a complicated process, as it essentially requires creating a new and different controller for each particular task. The variety of situations in which a robot may have to perform makes the job of the robot programmer more difficult. Rather than pre-programming a robot for all the tasks its users might want it to perform, it would be more useful if the robot could learn such tasks from the user, through flexible and natural interaction. However, users may not have the necessary skills to re-set, re-task and program the robots. Because of this, a good goal is to develop an approach to designing the robot controllers that is accessible to all users. Various frameworks have been proposed in the past that aim to provide robots with task learning capabilities that address the above problems, and thus reduce the amount of time and expertise required for the development of an autonomous, intelligent robot systems. A natural approach to this problem is to have the robots learn a particular task from a teacher's demonstration, thus increasing the ability of robots to interact with people, and relieving the user from writing controllers by hand [1].

Robot Learning from Demonstration (LfD) or Robot Programming by Demonstration (PbD) is a paradigm for enabling robots to autonomously perform new tasks [1]. LfD allows the end-user to program the robot simply by showing it how to perform the task - no coding required. To this end, LfD techniques try to maximize the generalizability of the learned skill to unseen situations with a minimal number of demonstrations provided by humans. Recently, Active Learning (AL) methods have been explored to achieve this goal.

For all supervised and unsupervised learning tasks, more often than not we first accumulate a huge amount of information that is randomly sampled from the underlying population distribution and we then initiate a classifier or model. This approach is called passive learning. A *passive learner* (*Figure 1*) gets a random data set from the world and after that yields a classifier or model.

An active learner (*Figure 2*) assembles data about the world by asking questions and receiving information. It then yields a classifier or model contingent on the undertaking that it is being utilized for. An active learner varies from a passive learner which just gets a random data set from the world and after that yields a classifier or model. One analogy is that a standard passive learner is a student that assembles information by sitting and listening to an instructor while an active learner is a student that asks the instructor questions, listens to the answers and asks additional questions based upon the teacher's response. It is conceivable that this additional capacity to adaptively question the world in view of past reactions would permit an active learner to perform superior to a passive learner, and indeed we shall later demonstrate that, in many situations, this is indeed the case.

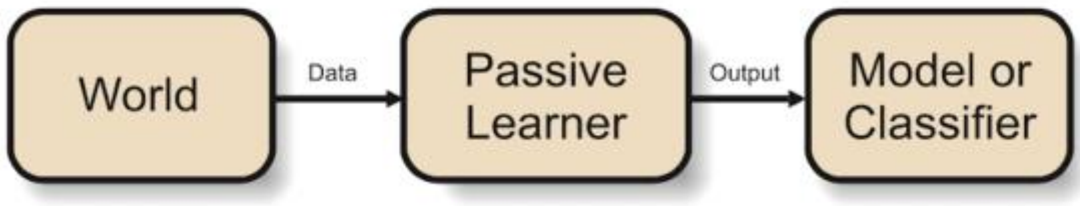

Figure 1: General schema for a passive learner *[2]*.

Figure 2: General schema for an active learner *[2]*.

The idea in AL is to improve learning rates by giving the learner more control over what examples it receives. The core contrast between an active learner and a passive learner is the capacity to get some information about the world in light of the past inquiries and reactions. The thought of what precisely a question is and what reaction it gets will rely upon the exact task at hand. Recent work in robotics involves the robots, making a query [4], which is a request for certain information, and the teacher provides this information. The requested information can take various forms, although most commonly it is a category label for an unlabeled data instance [4].

The assessment of robotic systems that make inquiries to people or request information to people has been limited, particularly in LfD settings. We are interested in estimating the performance improvement by incorporating AL in learning from demonstration setting. We also intend to explore the ease with which AL can be incorporated in Lfd framework. There is some research conducted in this area such as, addressing the question of when to ask questions in a mixed-initiative AL setting [3], types of questions that can be asked, feasibility of novel query types [4] or cost of different query types [5]. In this thesis, we propose that a robot using active learning could achieve more accurate and faster learning as compared to passive learner in HRI setting.

CHAPTER II

LITERATURE REVIEW

2.1 Active Learning

Active learning (sometimes called “query learning”) is a subfield of machine learning and, more generally, artificial intelligence. Active learning is a special case of semi-supervised machine learning in which a learning algorithm is able to interactively query the user (or some other information source) to obtain the desired outputs at new data points [1] [2]. In statistics literature, it is sometimes also called optimal experimental design [3].

The key hypothesis of active learning is that if the learning algorithm is allowed to choose the data by which it learns, it will perform better with less training. Why is this a desirable property for learning algorithms to have? Consider that, for any supervised learning system to perform well, it must often be trained on hundreds (even thousands) of labeled instances. Sometimes labels come at little or no cost, such as five-star rating you might give to films, or likes to articles on a social networking website. Learning systems may use these likes and ratings to suggest movies you might enjoy. In these cases, you are provided labels for free, but for many other supervised learning tasks, labeled instances are very difficult, time-consuming, or expensive to obtain. Here are a few examples:

- *Speech recognition*. Accurate labeling of speech utterances is extremely time consuming and requires trained linguists. [4] reports that annotation at the word level can take ten times longer than the actual audio (e.g., one minute of speech takes ten minutes to label), and annotating phonemes can take 400 times as long (e.g., nearly seven hours). The problem is compounded for rare languages or dialects.

- *Information extraction*. Good information extraction systems must be trained using labeled documents with detailed annotations. Users highlight entities or relations of interest in text, such as person and organization names, or whether a person works for a particular organization. Locating entities and relations can take a half-hour or more for even simple newswire stories [6]. Annotations for other knowledge domains may require additional expertise, e.g., annotating gene and disease mentions for biomedical information extraction usually requires PhD-level biologists.

- *Classification and filtering*. Learning to classify documents (e.g., articles or web pages) or any other kind of media (e.g., image, audio, and video files) requires that users label each document or media file with particular labels, like “relevant” or “not relevant.” Having to annotate thousands of these instances can be tedious and even redundant.

Active learning systems attempt to overcome the labeling bottleneck by making queries in the form of unlabeled instances to be labeled by an oracle as shown in *Figure 3*, making it iterative supervised learning. Since the learner chooses the examples, the number of examples to learn a concept can often be much lower than the number required in normal supervised learning. In this way, the active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data. Active learning is well-motivated in many modern machine learning problems where data may be abundant but labels are scarce or expensive to obtain.

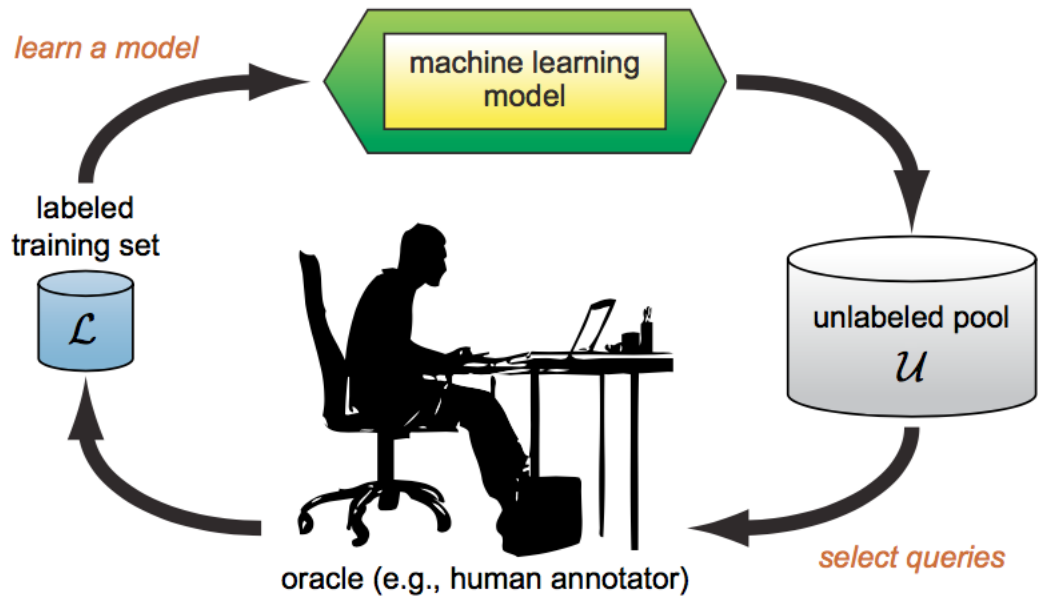

Figure 3: Active Learning process *[8]*

Active learning systems attempt to overcome the labeling bottleneck by making queries in the form of unlabeled instances to be labeled by an oracle as shown in *Figure 3*, making it iterative supervised learning. Since the learner chooses the examples, the number of examples to learn a concept can often be much lower than the number required in normal supervised learning. In this way, the active learner aims to achieve high accuracy using as few labeled instances as possible, thereby minimizing the cost of obtaining labeled data. Active learning is well-motivated in many modern machine learning problems where data may be abundant but labels are scarce or expensive to obtain.

2.1.1 Active Learning Heuristic

Typical active learning strategies use a simple heuristic to decide which data point to label next [9];

- Start with a pool of unlabeled data
- Pick a few point at random and get their labels
- Repeat the following
    1. Fit a classifier to the labels seen so far
    2. Pick the BEST unlabeled point to get a label for

2.1.2 How does the learner ask queries?

There are several different problem scenarios (*Figure 4*) in which the learner may be able to ask questions [8]:

- *Membership Query Synthesis*: The learner may request labels for any unlabeled instance in the input space.
- *Stream-Based Selective* Sampling: The learner obtains an unlabeled instance and uses an "informative measure" or "query strategy" (threshold) to decide whether to request its label or not.
- *Pool-Based Sampling*: The learner obtains large collections of unlabeled data at once and uses an "informative measure" or "query strategy" (threshold) to evaluate and classify the entire collection.

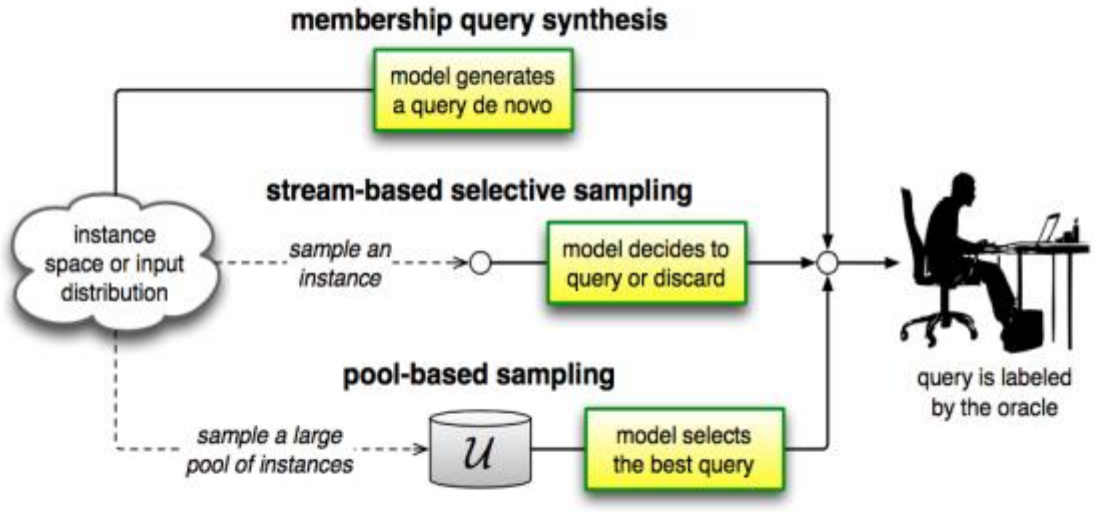

Figure 4: Three main Active Learning Scenarios *[1]*

2.1.3 Query Strategies

Algorithms for determining which data points should be labeled can be organized into a number of categories [8] :

- *Uncertainty sampling*: label those instances for which the current model is least certain as to what the correct output should be
- *Query by committee*: a variety of models are trained on the current labeled instances, and vote on the output for unlabeled instances; label those instances for which the "committee" disagrees the most
- *Expected model change*: label those instances that would most change the current model
- *Expected error reduction*: label those instances that would most reduce the model's generalization error
- *Variance reduction*: label those instances that would minimize output variance, which is one of the components of error

- *Balance exploration and exploitation:* the choice of examples to label is seen as a dilemma between the exploration and the exploitation over the data space representation. This strategy manages this compromise by modelling the active learning problem as a contextual bandit problem. For example, Bouneffouf et al. [6] propose a sequential algorithm named Active Thompson Sampling (ATS), which, in each round, assigns a sampling distribution on the pool, samples one point from this distribution, and queries the oracle for this sample point label.
- *Exponentiated Gradient Exploration for Active Learning:* [7] In this paper, the author proposes a sequential algorithm named Exponentiated gradient (EG)-active learning that can improve any active learning algorithm by employing an optimal random exploration.

A wide variety of algorithms have been studied that fall into these categories. [1] [3]

2.2 Related Work

Leveraging human input has received considerable attention in both the machine learning and robotics communities. Active Learning has been applied to a number of LfD frameworks. For example, confidence based autonomy [8] and dogged learning [9], use the principle of uncertainty sampling to select states in which the learning agent requests a demonstration while learning a policy. In [10], the robot actively selects points outside the region of stability of a learned policy, and requests demonstrations from these states.

Much earlier work deals with the scenario in which a machine learns by watching human conduct [11] [12]. Other work has concentrated on how a machine can learn tasks from human direction [13] [14], with human advice [15] [16] or through interruption on the learner's activities amid the learning procedure [17].

The assessment of robotic systems that make inquiries to people has been limited, particularly in LfD settings. Prior work deals with comparing passive and active learning [18], addressing the question of when to ask questions in a mixed-initiative AL setting [19]. One framework demonstrated certainty based active learning with (nonsocial) human labelers [20]. Similarly, Lopes et al. utilized active learning to figuring out how to choose the states in which a specialist human ought to be questioned for an appropriate action [21]. Rosenthal et al. investigate how augmenting robot's questions with different types of additional information could improves the accuracy of human teacher's [22]. Considerable work has been done to explore the idea of learning actions for robots through dialogue and demonstrations [23] [24]. Many studies investigated different HCI issues such as feasibility of novel query types (e.g. feature queries) [25] or cost of different query types [5]. Cakmak et al. and Gervasio et al. have investigated question asking in procedure learning with a range of question types [26] [27].

Our work focuses on utilizing various types of label queries using an active learning approach in a learning from demonstration framework. Querying is crucial to many robotic tasks, allowing the robot to precisely localize objects, build maps, perform manipulation tasks and achieve many other goals. There has not been enough work on annotations or queries on motion primitives to learn the target concept.

The active learning setting in the review introduced here is fundamentally unique in relation to different assessments of active learning on the grounds that the active learner receives labels through a dynamic teaching–learning collaboration instead of through monotone query labeling or annotation. In our situation, the individual is attempting to educate the robot independent of whether the robot makes questions. The learner can therefore, pick up data without making a question, and the learner is ensured that the educator will answer its inquiry.

CHAPTER III

METHODOLOGY

3.1. Problem Statement

As outlined in previous sections, if we just pick the RIGHT examples to label, we can learn the concept from only a few labeled examples. This thesis proposes that a robot using active learning could achieve more accurate and faster learning as compared to passive learner in HRI setting.

In an active learning approach the learning process may iteratively query unlabeled samples to select the most informative samples to annotate and update its learned models. The central part of our work is (i) to design a strategy for data selection for query in order to avoid unnecessary and redundant queries, (ii) select different types of query that will help the robot learn better and (iii) test how the incorporation of AL methods in LfD impacts a robot's learning and performance. One way the robot could make a query is to execute a motion and ask a teacher whether the skill was performed correctly. This is called a label query [29]. Second, the robot could request demonstrations. This is called a demonstration query [29]. Other queries involve asking whether a feature is important or relevant for the target concept that is being learned [5] [31].

We use various types of queries described in chapter II with an HRI experiment that involves teaching skills to a robot and answering its questions.

3.2. Preliminaries

We present here, the basic definitions that are used throughout the further description of the methodology. We use the terms (making a) query and (asking a) question interchangeably throughout the discussion. The tasks or skills involve a fixed set of objects. Four demonstrations, varied in certain aspects, were recorded for each skill.

3.3. Platform

The robot platform used in our experiments is Baxter, an upper torso humanoid robot with 7 Degree of Freedom (DoF) compliant arms and state-of-the-art sensing technologies, including force, position, and torque sensing and control at every joint, cameras in support of computer vision applications, integrated user input and output elements such as a head-mounted display, buttons, knobs and more.

Interfaces for Demonstration:

1) Directly recording human motions: Various methods used for this approach are existing motion tracking systems, based on vision, exoskeleton or other wearable motion sensors. These external means of tracking human motion return a precise measurement of the angular displacement of the joints. They have been used in various works for LfD of full body motion [32] [33] [34]. These methods are advantageous in that they allow the human to move freely, but require good solutions to the correspondence problem. Typically, this is accomplished by an explicitly mapping between human and robot joints, but can be quite difficult if the robot (e.g., a hexapod) differs greatly from the human.

2) Kinesthetic teaching: The robot is physically guided through the task by the humans. One main drawback of kinesthetic teaching is that the human must often use more of their own degrees of freedom to move the robot than the number of degrees of freedom they are trying to control. A possibility is to proceed incrementally [35].

3) Immersive teleoperation scenarios: A human operator is limited to using the robot's own sensors and effectors to perform the task. The teleoperation itself may be done using joysticks or other remote control devices, including haptic devices [36] [37]. Teleoperation is, however, more frequently used solely to transmit the kinematic of motion [14] [38]. In our experiments, teaching robot involves both Kinesthetic teaching and Immersive teleoperation. In the first two experiments, the robot is guided through the task by human using teleoperation. The teleoperation is done using keyboard control. In the third task, the robot is physically guided through the task by a human. In our task scenarios, the robot works alongside or across from a human partner at a tabletop workshop.

Baxter's joint names are mentioned a number of times in this documentation. A labeled diagram can be found below (S=Shoulder, E=Elbow, W=Wrist) in Figure 5.

Figure 5: Baxter's joint names [39]

3.4. Experimental Design

The experiment involves the teaching of three different skills to Baxter using multiple demonstrations. From the given demonstration a learned motion is obtained and evaluated on Baxter for successful task completion. If Baxter fails to perform the task correctly, it communicates to the human using various Active Learning queries with the goal of improving its skills with the help of human interaction.

3.4.1 Skill Representation

The robot state can be represented in a number of different ways. Among the commonly used in LfD are joints of the robot, end-effector configuration (position and orientation) relative to the robot, or relative to a reference/goal coordinate frame in the world. In this paper, we consider the example skill of pick and place of a box (Figure 6) and writing numbers (Figure 7).We represent the state ‘x’ of the robot as the end-effector configuration relative to the Baxter’s torso.

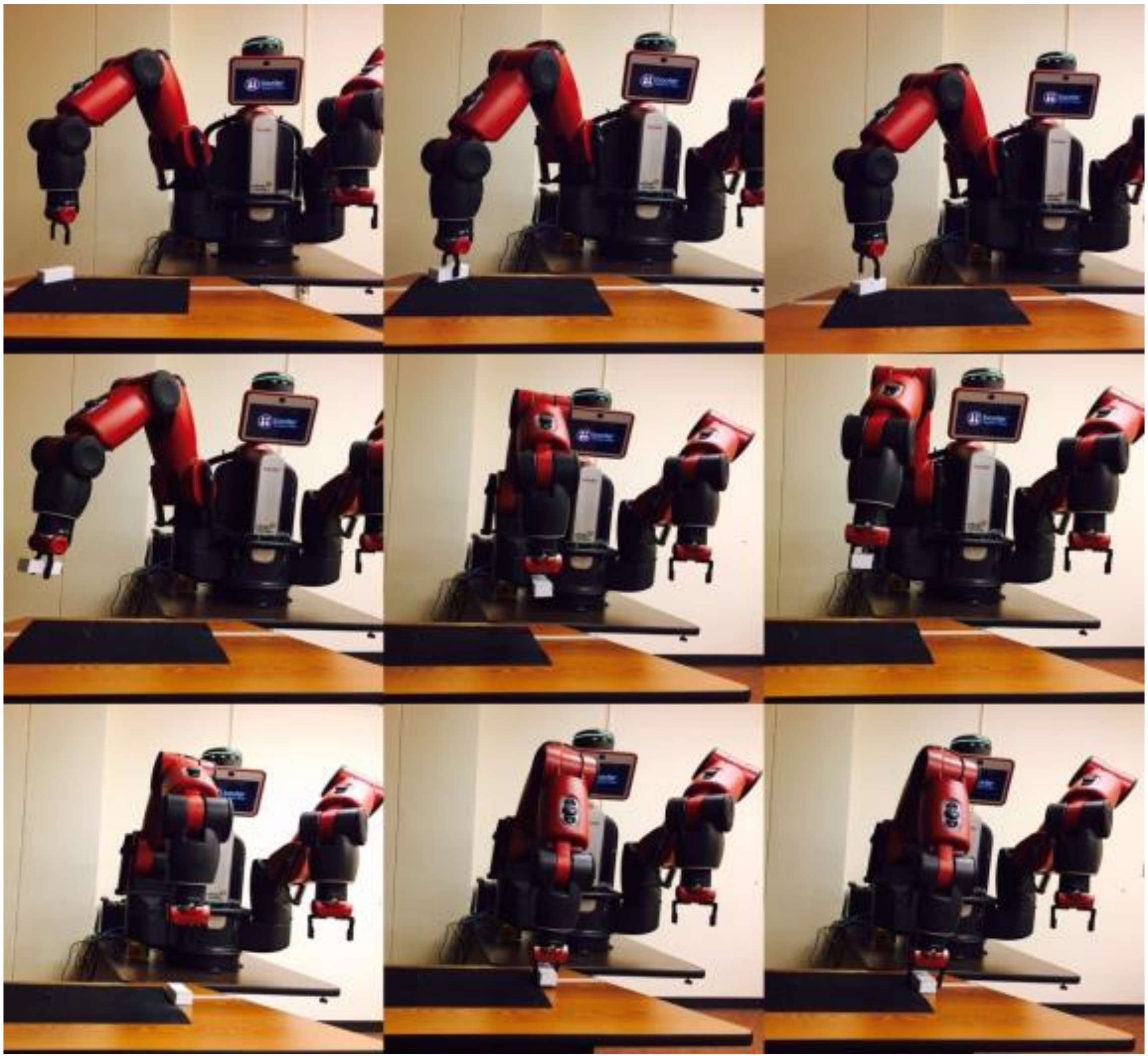

Figure 6: Pick and Place Task

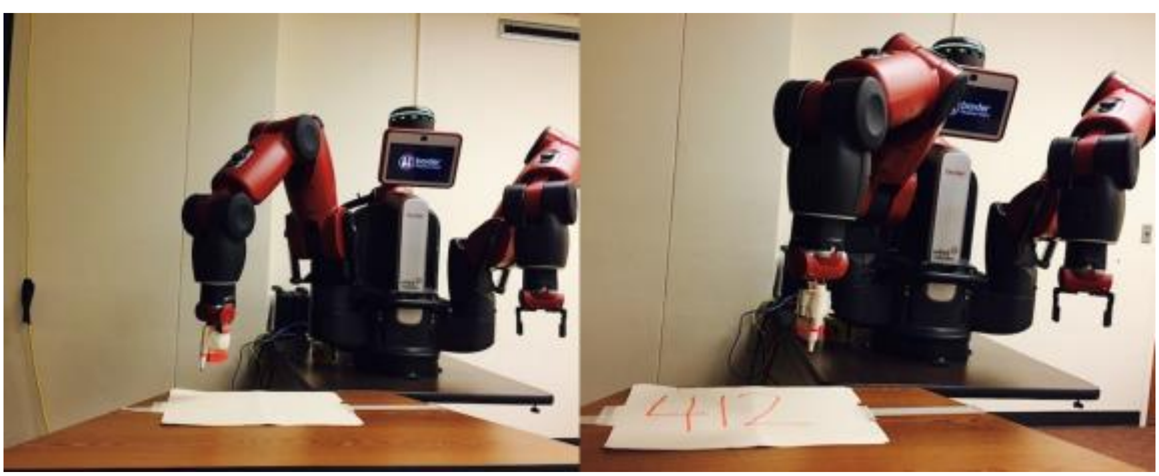

Figure 7: Writing Task

3.4.2 Description of Tasks

We considered three tasks as described in Figure 6 and Figure 7. The tasks are chosen to cover two types of LfD problems in robotics. The first two tasks are focused on learning a task goal (desired end state) whereas the latter one emphasizes learning skills to perform the task (desired actions or movements to accomplish the task). Each of these is designed to be an abstraction of a real-world task (e.g. moving an object). To access the effectiveness of our approach, following tasks are performed using Baxter in the real world. We perform these tasks using the right arm of the robot. The end-effector is Baxter's right arm gripper.

Task 1: Baxter learns to pick up and place a box with fixed start and end positions.

Task 2: Baxter learns to pick up and place a box with varying start and end positions.

Task 3: Baxter learns to write numbers e.g. '412'.

3.5. Teaching Skills

A demonstration consists of a sequence of state-action pairs; $D_i = \{(x_{i0}, \dot{x}_{i0}), \ldots, (x_{iN_i}, \dot{x}_{iN_i})\}$. A skill is learned from a set of m demonstrations $\{D_i\}_{i=1}^{m}$. Demonstration data in the form of joint positions x_t are sampled from a demonstration of a task at time instants $t = 1, 2, \ldots, T$, and is used as an input for the experiment. For learning tasks, Baxter is provided with four demonstrations using keyboard control and manual demonstration.

Task 1: pick and place a box with fixed start and end positions

Demonstration data in the form of joint positions x_t are sampled from four demonstrations of a pick and place task performed using keyboard control at time instants $t = 1, 2, \ldots, T$. We divide the pick and place operation into two main sub-tasks and we break them into five states as shown in *Figure 8*. The main states of a pick action are as follows (the description of place states is similar):

- Start: At the start of the experiment, Baxter's right arm is at a specific 3D end-effector position ($x_{initial}$, $y_{initial}$, $z_{initial}$).
- Approach: The end-effector moves to the vicinity of the box to be picked up, located at a fixed position (x'_{box}, y'_{box}, z'_{box}).
- Grasp: The end-effector moves to the box position.
- Close Gripper: The end-effector closes the gripper attached to the right arm to grab the box.
- Retract: The end-effector moves (z'_{box} + h) in z direction.

After the box has been picked up the end-effector position is offset by (x'_{box}, y'_{box} + l, z'_{box} + h). When the arm has been moved to the desired destination, the box is placed at (x''_{box}, y''_{box}, z''_{box}) following similar steps as in the pick state.

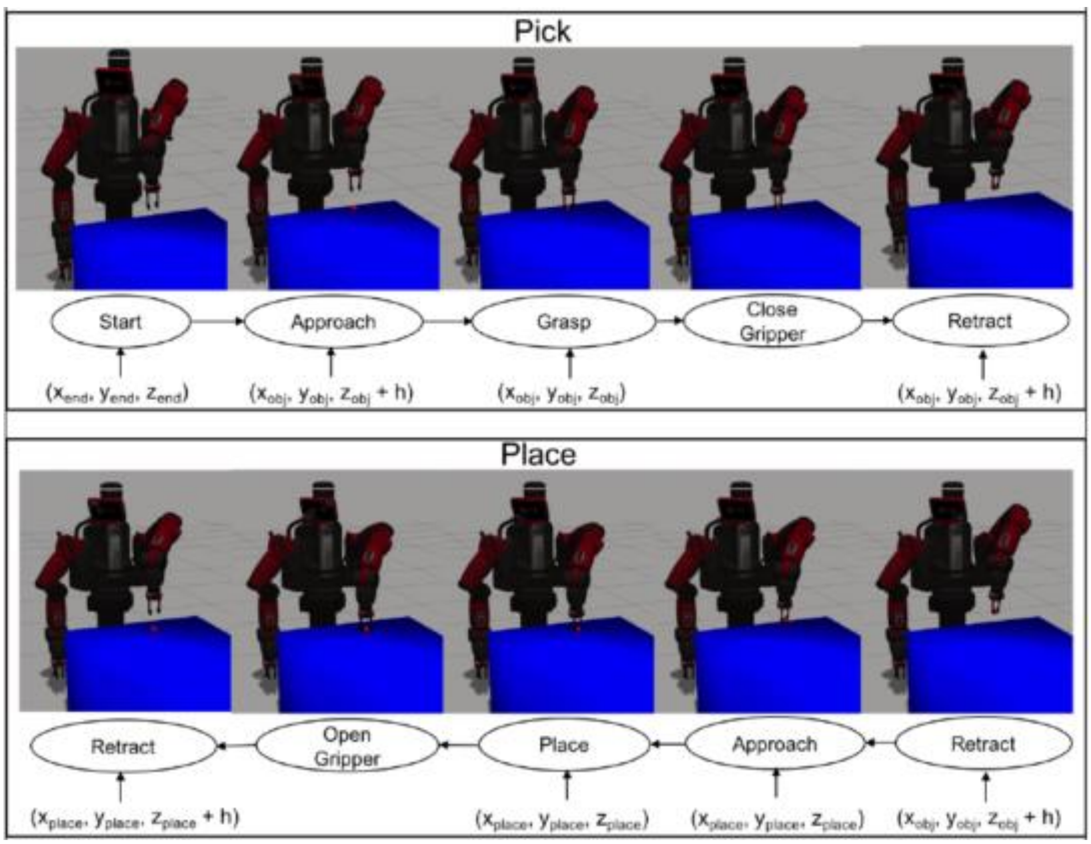

Figure 8: The states involved in a pick and place task [37]

Task 2: Baxter learns to pick and place a box with varying start and end positions

Demonstration data in the form joint positions x_t are sampled from four demonstrations of a pick and place task using keyboard control. We divide the pick and place into two main sub-tasks and break them into five states as shown in Figure 8 same as Task1. In this task, the initial position of the box varies in each demonstration.

Task 3: Baxter learns to write numbers say '412'

Demonstration data in the form of joint positions x_t are sampled from four demonstrations of a writing task by manual control of Baxter's right arm. The writing task in our case involved drawing numbers '4', '1', '2' on a sheet of paper.

3.6. Deriving a policy

After acquiring the dataset of state-action example, the next step involves deriving a policy using the data. The approach for deriving the policy is initially split between the three core derivation techniques:

Mapping Function: The mapping function approach to policy learning calculates a function that approximates the state to action mapping. In general, mapping approximation techniques fall into two categories depending on whether the prediction output of the algorithm is discrete or continuous. Classification techniques produce discrete output and Regression techniques produce continuous output [2]. Many techniques for performing classification and regression have been developed outside of LfD [41].

System Models: This approach uses a state transition model of the world, T (s'|s, a), and from this derives a policy. This approach is typically formulated within the structure of Reinforcement Learning (RL). Demonstration data, and any additional autonomous exploration the robot may do, generally determines the transition function T(s'|s, a). To derive a policy from this transition model, a reward function R([17]), which associates reward value r with world state 's', is either learned from demonstrations or defined by the user. The goal of RL is to maximize cumulative reward over time. Inverse Reinforcement Learning(IRL) is one of the techniques concerned with finding the reward function of a human demonstrator from the demonstrated state and action samples [17] [42] [43].

Plans: The planning framework, represents the policy as a sequence of actions that lead from the initial state to the final goal state. Actions are often defined in terms of pre-conditions, the state that must be established before the action can be performed, and post-conditions, the state resulting from the actions execution. Unlike other LfD approaches, planning techniques frequently rely not only on state-action demonstrations, but also on additional information in the form of annotations or intentions from the teacher.

In our case study, we have used the mapping function to derive the policy. In our experiments, demonstrations are first collected and synthesized into learned motion using dynamic time warping (DTW) and barycenter averaging [44] [45].

Dynamic time warping: DTW is based on the Levenshtein distance (also called edit distance) and was introduced in [46] and [47], with applications in speech recognition. It finds the optimal alignment (or coupling) between two sequences of numerical values, and captures flexible similarities by aligning the coordinates inside both sequences. The cost of the optimal alignment can be recursively computed by

$$D(A_{i,}B_j) = \delta(a_{i,}b_j) + \min \begin{cases} D(A_{i-1,}B_{j-1}) \\ D(A_{i,}B_{j-1}) \\ D(A_{i-1,}B_j) \end{cases}$$

where $A_{i,}$ is the subsequence $\langle a_1, \ldots a_i \rangle$. The overall similarity is given by $D(A_{|A|,}B_{|B|}) = D(A_{T,}B_T)$. This measure has thus a time and a space complexity of $O(|A|.|B|)$. DTW is able to find an optimal global alignment between sequences and is probably the most commonly used measure to quantify the dissimilarity between sequences. It also provides an overall real number that quantifies similarity.

DBA: To create pairwise averaging, a global averaging strategy called D_{TW} Barycenter Averaging (DBA) is used. The aim is to minimize the sum of squared DTW distances from the average sequence to the set of sequences. Technically, for each refinement i.e., for each iteration, DBA works in two steps:

Step 1: Computing DTW between each individual sequence and the temporary average sequence to be refined, in order to find associations between coordinates of the average sequence and coordinates of the set of sequences.

Step 2: Updating each coordinate of the average sequence as the barycenter of coordinates associated to it during the first step.

The resulting learned motion L obtained as a result of D_{TW} Barycenter Averaging performed on a set of m demonstrations $\{D_i\}_{i=1}^{m}$ is then tested on Baxter for successful task completion. If the learned motion fails to successfully complete the task, an interactive correction is performed to improve the learned skill as covered in following section.

3.7. Interactive Corrections/ Active Skill Learning

Interactive correction utilizes the concept of Active Learning for correction and improvement of the learned skill M_{LfD}. The conventional query in the AL literature involves choosing an unlabeled instance and requesting a label for it. The instance can be chosen from a pool of unlabeled instances or instantiated by the learner in some way. Such queries have been used in learning skills on a robot, where a skill is represented as a policy that maps a state to a discrete action. In this context, a query consists of asking for an action in a chosen state [46] [21] [9]. This is useful in an HRI setting when the robot's actions are discrete and the human has a way to refer to each action. However, robot skills often involve continuous actions and the input from the teacher to the learner is a sequence of state-action pairs (i.e. trajectories). In these cases, it is impractical to ask for an isolated state-action pair (e.g. asking for the motor commands of a given arm configuration). Thus we need to re-think the way that a query is made. We consider the following alternatives.

3.7.1 Types of Queries

Label queries: Robot skill learning involves modeling/encoding a skill from a set of demonstrations, such that it can be reproduced correctly. The demonstrations provided by the teacher are inherently labelled as positive. One way the robot could make queries is to execute a motion and ask whether the skill was performed correctly. We call this a *label query*. Methods for generating label queries depend on the particular framework used for representing the skill. However, a general approach applicable to most frameworks is to sample trajectories from the learned skill and evaluate them with a certain criterion. For instance, the robot can choose to query the trajectory that it is least certain about or the trajectory that is most likely to increase the applicability of the skill.

Demonstration queries: The second type of query, which we call demonstration or demo queries, involves requesting a demonstration from the teacher. Demo queries give less control over what information is acquired from a query, as compared to label queries. In label queries, the learner specifies the whole trajectory and the teacher only provides a label. In demonstration queries, the learner only specifies certain constraints, while the trajectory is still produced by the teacher. Nevertheless, this gives some control to the learner such that useful demonstrations can be acquired. Demo queries are analogous to a method known as active class selection [47], which consist of requesting an example from a certain class.

One way to constrain trajectories provided by the teacher is to specify the starting state. Trajectories are often represented with a sequence of end-effector configurations relative to a goal object. Thus the robot can configure its end-effector in a certain way relative to the goal and request the demonstration to start in this configuration. A larger range of queries can be made by manipulating the target object.

3.7.2 Data Selection Strategy for Query

Given a task's learned motion M_{LfD} obtained as a result of D_{TW} Barycenter Averaging performed on a set of m demonstrations $\{D_i\}_{i=1}^{m}$, we collect critical joint-position instances that are representative of a target concept. Steps to generate the change-points in the Learned Skill M_{LfD} are as follows:

Step 1: Compute second derivatives of the joint-position data of M_{LfD}.

Step 2: Compute the standard deviation on each segment of the second derivative data that are half a second apart (half a second time intervals were selected for our task. These can be changed based on what task is being demonstrated to Baxter and how frequently the changes need to be observed).

Step 3: Compute the Mean of all the standard deviations across all the segments.

Step 4: Select segments with standard deviation greater than the Mean-Standard-Deviation.

3.7.3 Approach for Querying

Once the change-points are collected, the following approach is used for querying label as shown in Fig.4:

Step 1: Let the robot perform the learned task M_{LfD}.

Step 2: While the learned task is being performed by robot, map the running instances of the joint positions to the set of critical points collected in section 3.6.2.

Step 3: If a match is not found, goto step 2 and continue. If a match is found, goto step 4.

Step 4: The robot queries the correctness of the subtask (*label query*) performed, from a teacher. If a teacher confirms the correctness of subtask, goto step 2. If the subtask was performed incorrectly, goto step 5.

Step 5: The robot queries for a correction of subtask performed, from a teacher (*demonstration query*).

Step 6: Demonstration query response are introduced on the same learned motion to improve the sequence.

Step 7: Goto step 2 and continue until the learned skill is improved and the robot is able to perform the task successfully.

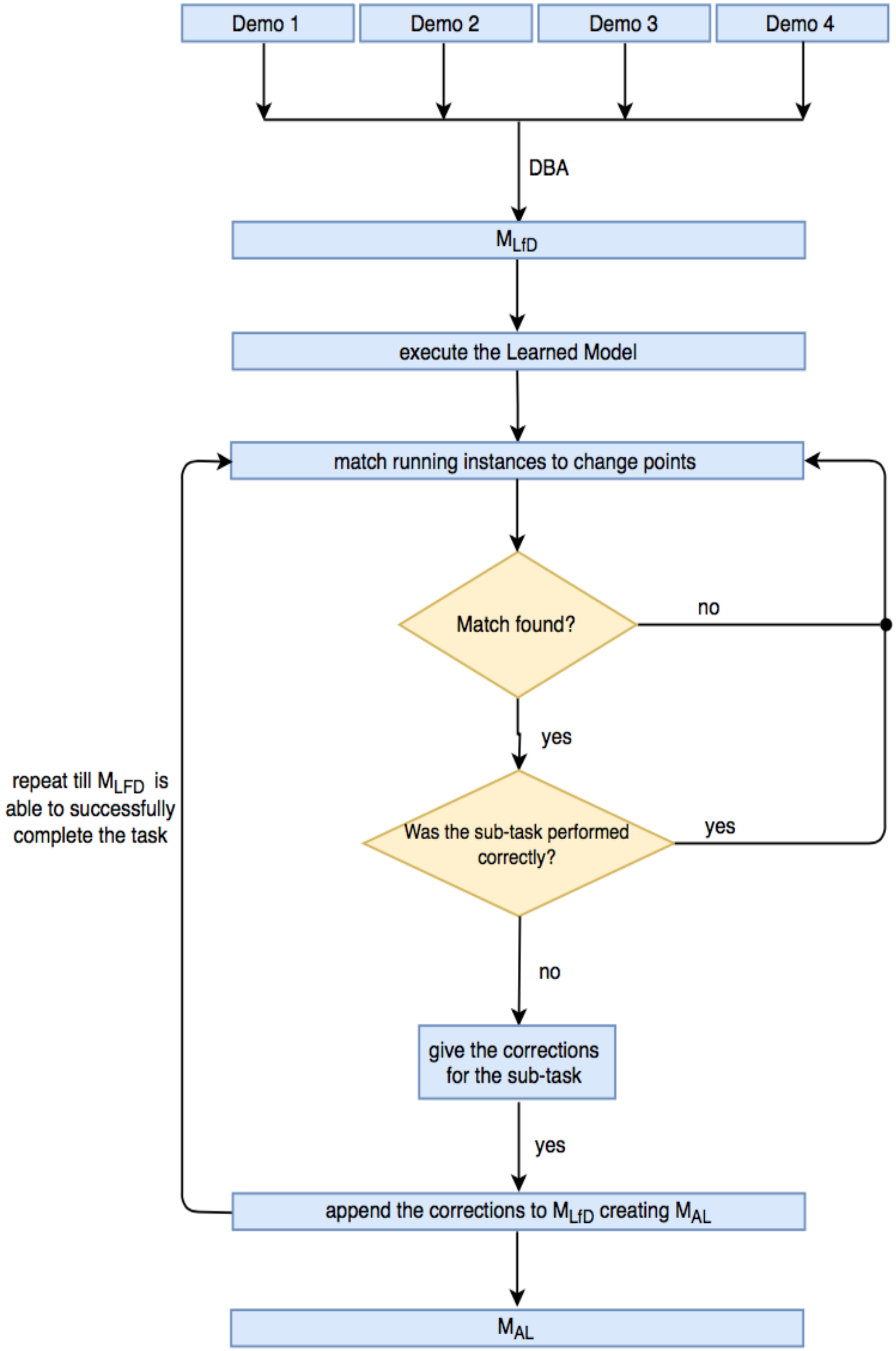

Figure 9: Approach for Interactive Corrections

3.8. Evaluation Measure

The model obtained using the Active learning approach i.e. *demonstration queries,* in conjunction with *label queries* (M_{AL}), is expected to perform better than the robot's learned model obtained using plain LfD (M_{LfD}) when compared with an idealized, hard-coded, perfect task (M_I). To evaluate the performance of the models, we use two parameters:

1. Successful task completion
2. RMSE (Root Mean Square Error) of M_{LfD} vs RMSE of M_{AL}

CHAPTER IV

FINDINGS

4.1 Experiments

4.1.1 Experiment: Pick and place – TYPE I

Task Details: Baxter learns to pick and place a box with fixed start and end positions.

We have collected four sample demonstrations of pick and place. In this experiment, Baxter is trained with demonstrations that are very precise and close to each other in terms of trajectory used for pick and place. Demonstration samples are shown in Figure 1. Each data set sample consists of joint position data for joints at right arm of Baxter, named right_s0, right_s1, right_e0, right_e1, right_w0, right_w1 and right_w2.

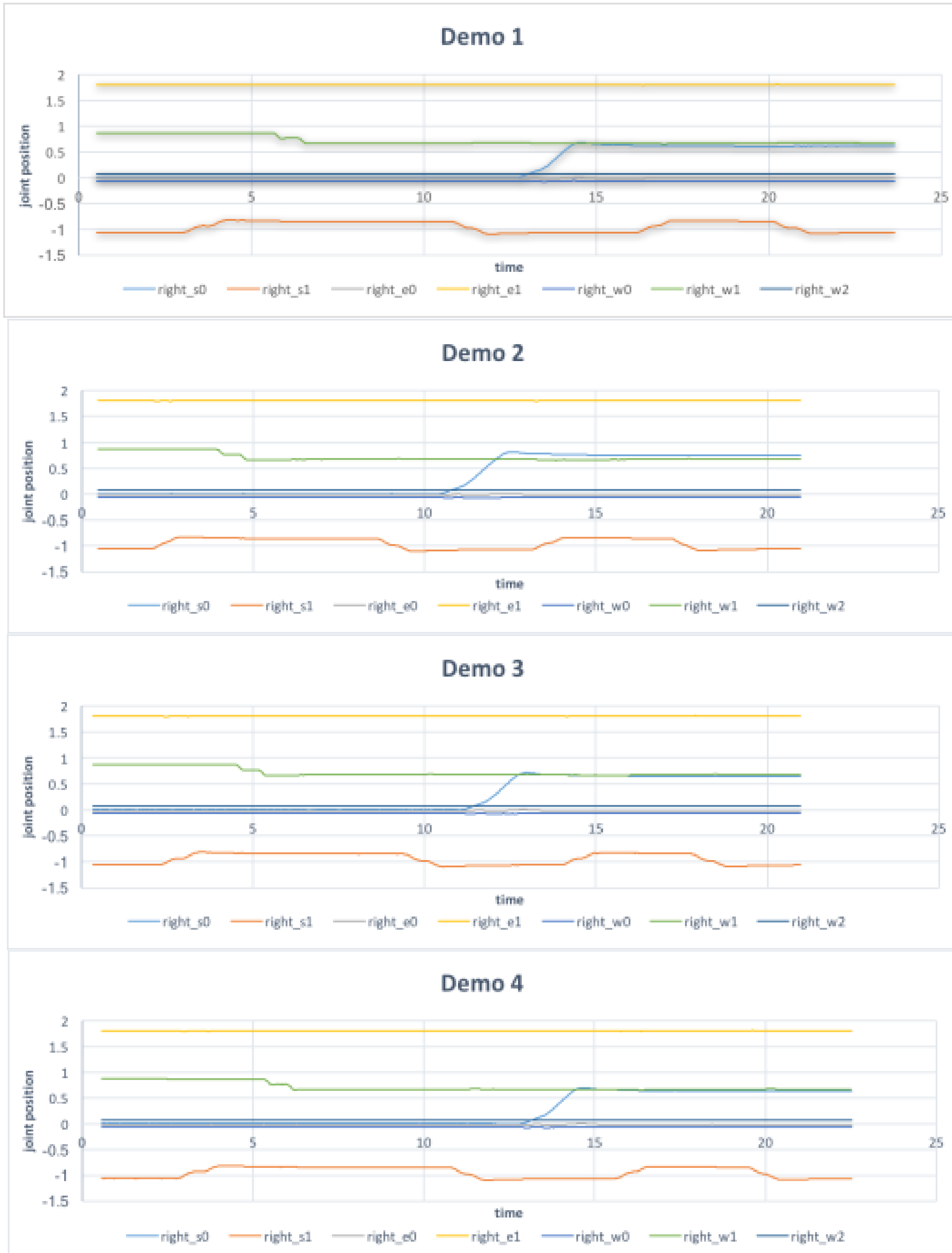

Figure 10: Task 1 - Plot for joint position data in demonstration samples

The learned model M_{LfD} is obtained by processing sample demonstrations using DBA in plain learning from demonstration framework. As seen in *Figure 11*, the resulting learned model from DBA is extremely noisy. To smooth the noisy learned model, we used Gaussian smoothing. The result of Gaussian smoothening is as shown in *Figure 12*.

Figure 11: Task 1 - Plot for joint position data in Learned Model M_{LfD}

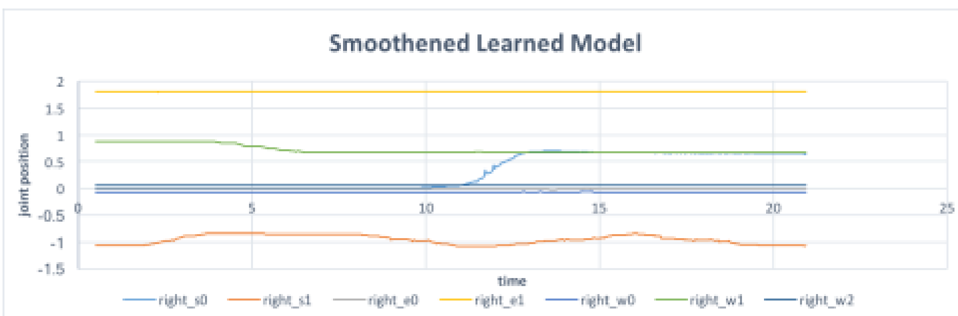

Figure 12: Task 1 - Plot for joint position data in Smoothened Learned Model M_{LfD}

The smoothened-learned-model the change point is computed using the approach described in the section “Data selection strategy for query” of the methodology chapter. We found fifteen change points in the learned model as highlighted in *Figure 13*. The learned model is tested on Baxter to pick up and place a box. While the learned model is being executed, the running instances of the model are mapped to change points. On encountering a change point, a label query “Is the subtask performed correctly?” is generated. In this task we found that, at every change point, the answer to label query was “yes”.

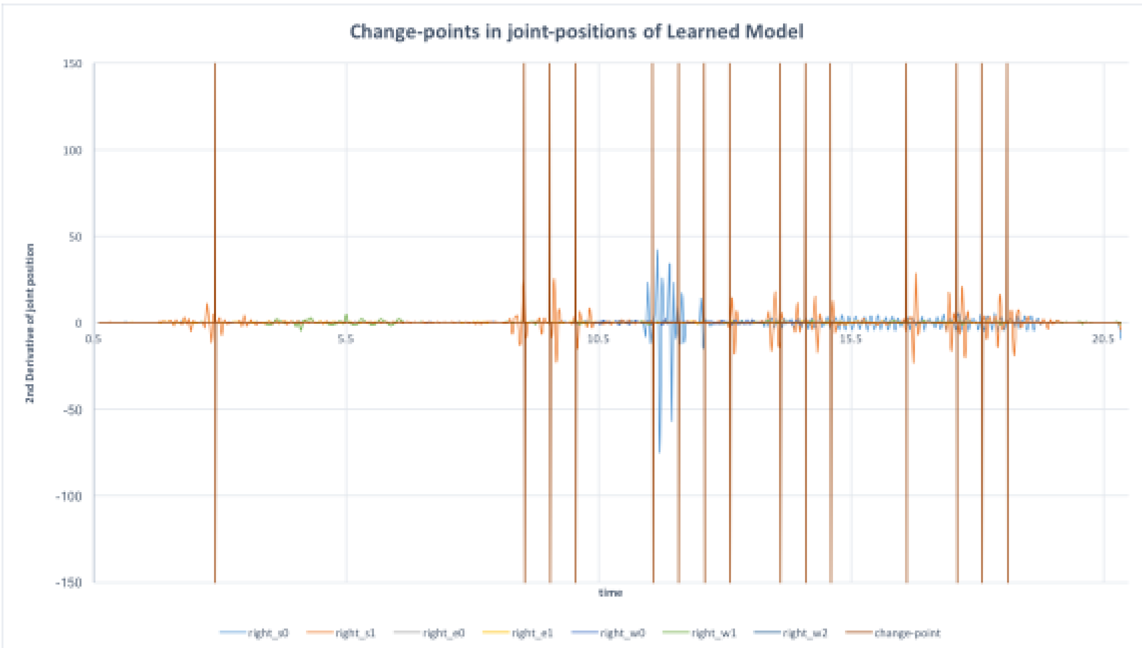

Figure 13: Task 1 – Plot for change points in joint positions in Learned Model M_{LfD}

Analysis: When tested on Baxter, the learned model MLfD gave positive results for the task completion and therefore did not require any interactive corrections. This shows us that for a simple task, with very similar sample demonstrations, the MLfD obtained was quite reliable in terms of successful task completion. But in a true real world setting, it is improbable to give indistinguishable demonstrations considering human factors, which is one of the major issues in LfD. Therefore, we conducted another experiment considering real world setting and human errors.

4.1.2 Experiment: Pick and place – TYPE II

Task Details: Baxter learns to pick and place a box with varying start and end positions.

We have collected four sample demonstrations of pick and place. For every sample demonstration, the x coordinate for the box was offset by 5 centimeters. Demonstration samples are shown in *Figure 14*. Each data set sample consists of joint position data for joints of Baxter's right arm, named right_s0, right_s1, right_e0, right_e1, right_w0, right_w1 and right_w2.

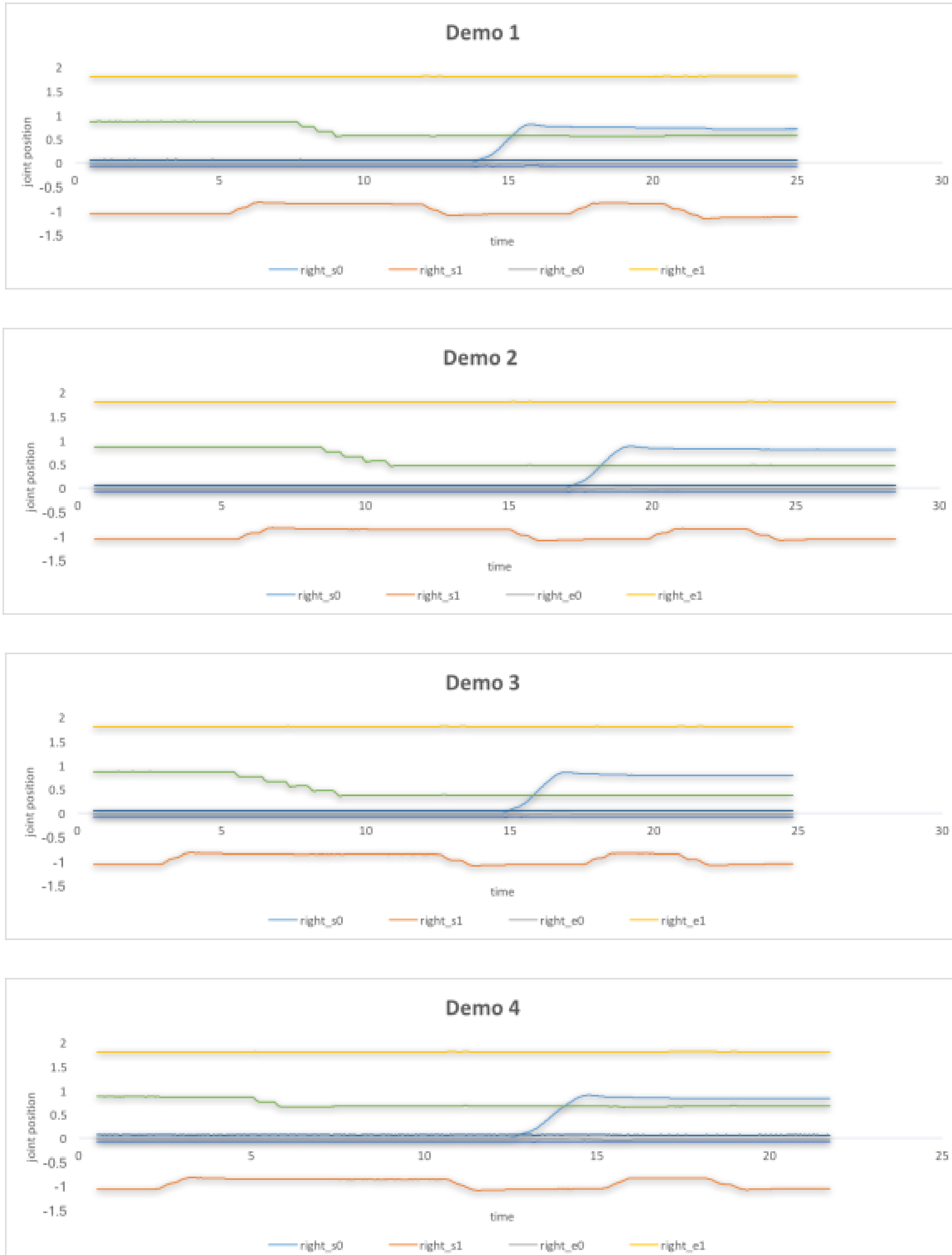

Figure 14: Task 2 - Plot for joint position data in demonstration samples

The learned model M_{LfD} is obtained by processing sample demonstrations using DBA in plain learning from demonstration framework. As seen in

Figure 15, the resulting learned model from DBA is extremely noisy. To smoothen the noisy learned model, we used Gaussian smoothing. The result of Gaussian smoothening is as shown in

Figure 16.

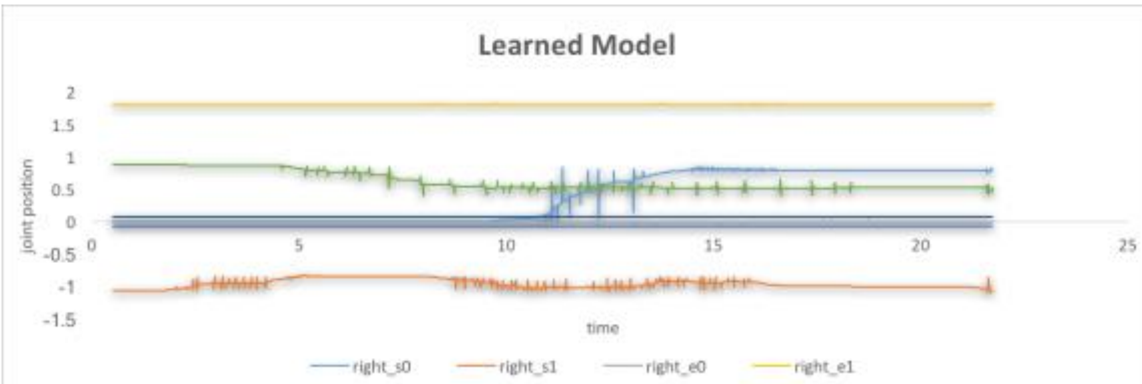

Figure 15: Task 2 - Plot for joint position data in Learned Model M_{LfD}

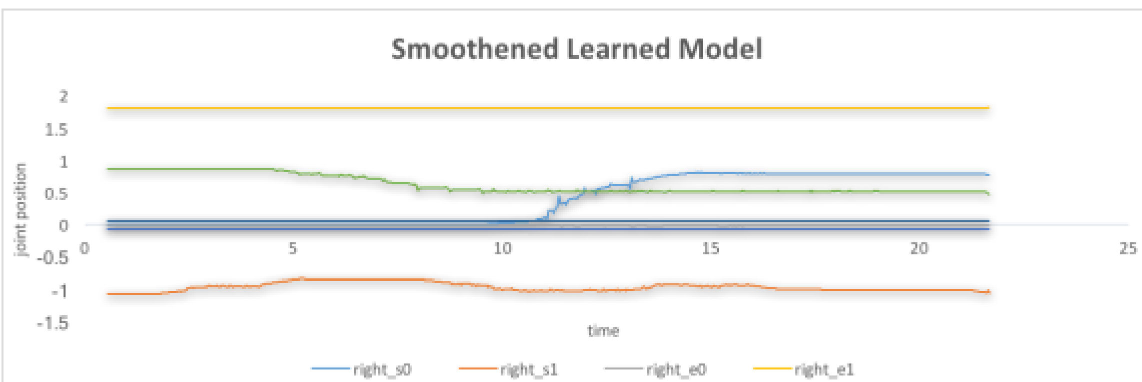

Figure 16: Task 2 - Plot for joint position data in Smoothened Learned Model M_{LfD}

The smoothened-learned-model change point is computed using the approach described in the section “Data selection strategy for query” of the methodology chapter. We found sixteen change points in the learned model as highlighted in

Figure 17. The learned model is tested on Baxter to pick and place a box which is initially located at one of the coordinates used in a sample demonstration. While the learned model is being executed, the running instances of the model are mapped to change points. On encountering a change point, a label query “Is the subtask performed correctly?” is generated.

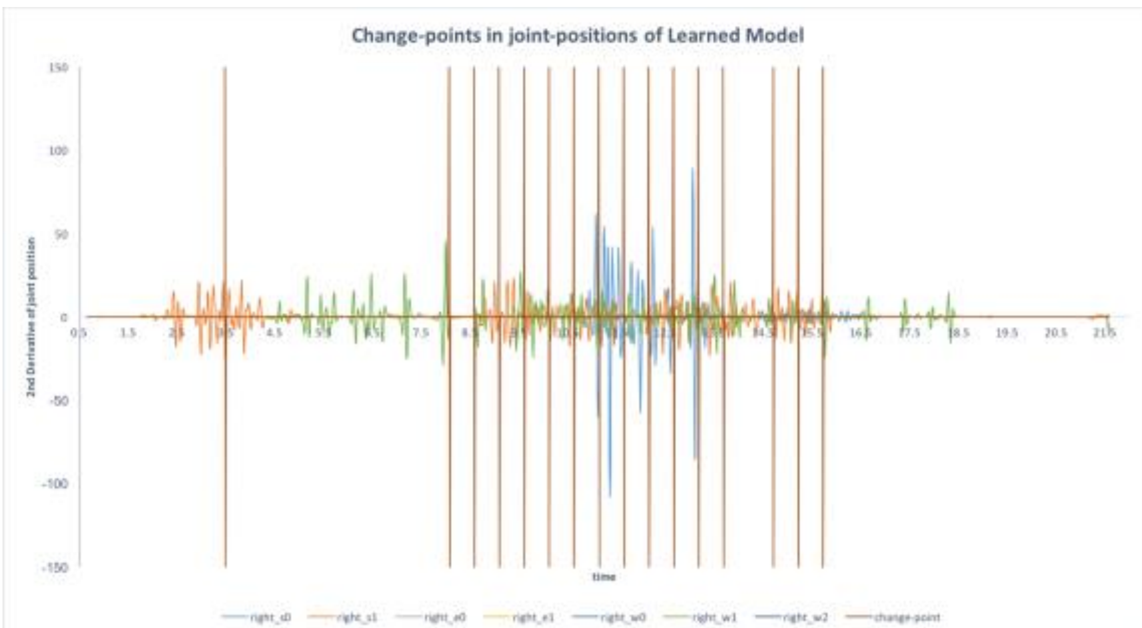

Figure 17: Task 2 - Plot for change points in joint positions in Learned Model M_{LfD}

Analysis on M_{LfD}: In this task we found that, at sixteenth change point in M_{LfD}, the answer to label query was “no” and therefore required interactive correction for placing the object correctly. At this point, Baxter requests corrections using Demonstration query. To provide the corrections, Baxter's arm is moved to the previous change point (i.e. fourteenth change point, the state where the subtasks were performed correctly so that the corrections can be given from that point onwards) for further corrections. After providing interactive corrections and updating the learned model M_{LfD} the resulting corrected model M_{AL} is as shown in *Figure 18*.

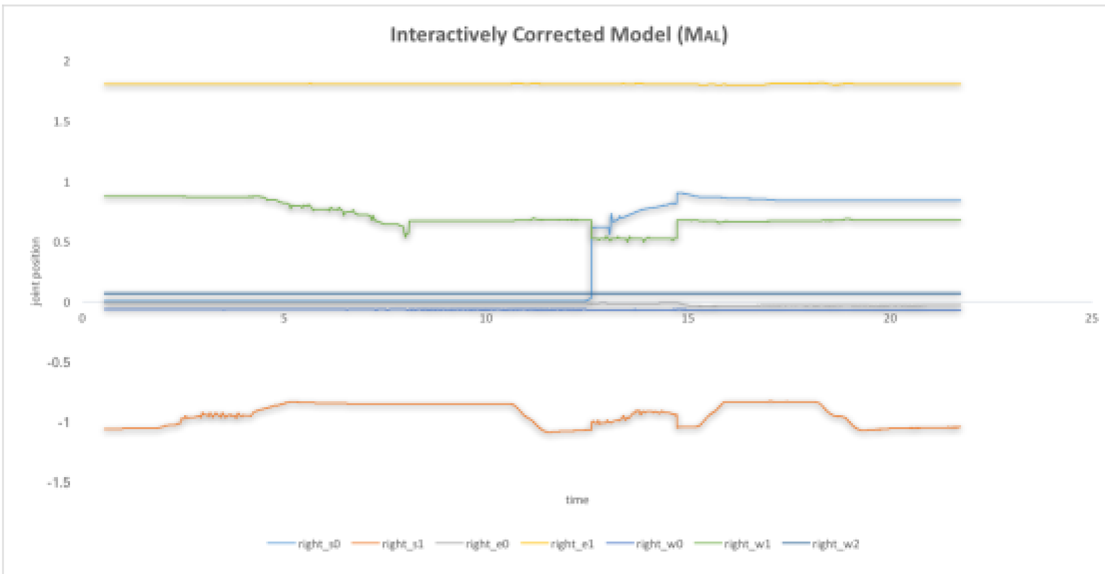

Figure 18: Task 2 - Plot for joint position data in Interactively Corrected Model M_{AL}

Analysis on M_{AL}: When tested on Baxter, the interactively corrected model was able to successfully complete the task.

Evaluation: To evaluate the performance of M_{AL} over M_{LfD}, we computed the RMSE for the end-effector positions of the M_{AL} and M_{LfD} against the standard Model M_{Std} and RMSE for the joint-positions data of the M_{AL} and M_{LfD} against M_{Std}.

RMSE	$\mathbf{M_{LfD}}$	$\mathbf{M_{AL}}$
joint-position data	0.0836	0.0493
end-effector data	0.0608	0.0224

Table 1: Task 2 - RMSE of joint-position data and end-effector data for M_{LfD} vs M_{AL}

Results from *Table 1* and performance of MAL shows us that for a simple task, with distinguishable sample demonstrations, the MLfD is not always completely reliable in terms of successful task completion and therefore required interactive corrections. Next, we conducted the experiment to test the performance of LfD for complex task.

4.1.3 Experiment: Writing Task

Task Details: Baxter learns to write numbers “412”.

We have collected four sample demonstrations for writing “412”. In this experiment, Baxter is trained with demonstrations that are close to each other in terms of similarity in trajectory. Demonstration samples are shown in Figure 19. Each data set sample consists of joint position data for Baxter’s right arm, named right_s0, right_s1, right_e0, right_e1, right_w0, right_w1 and right_w2.

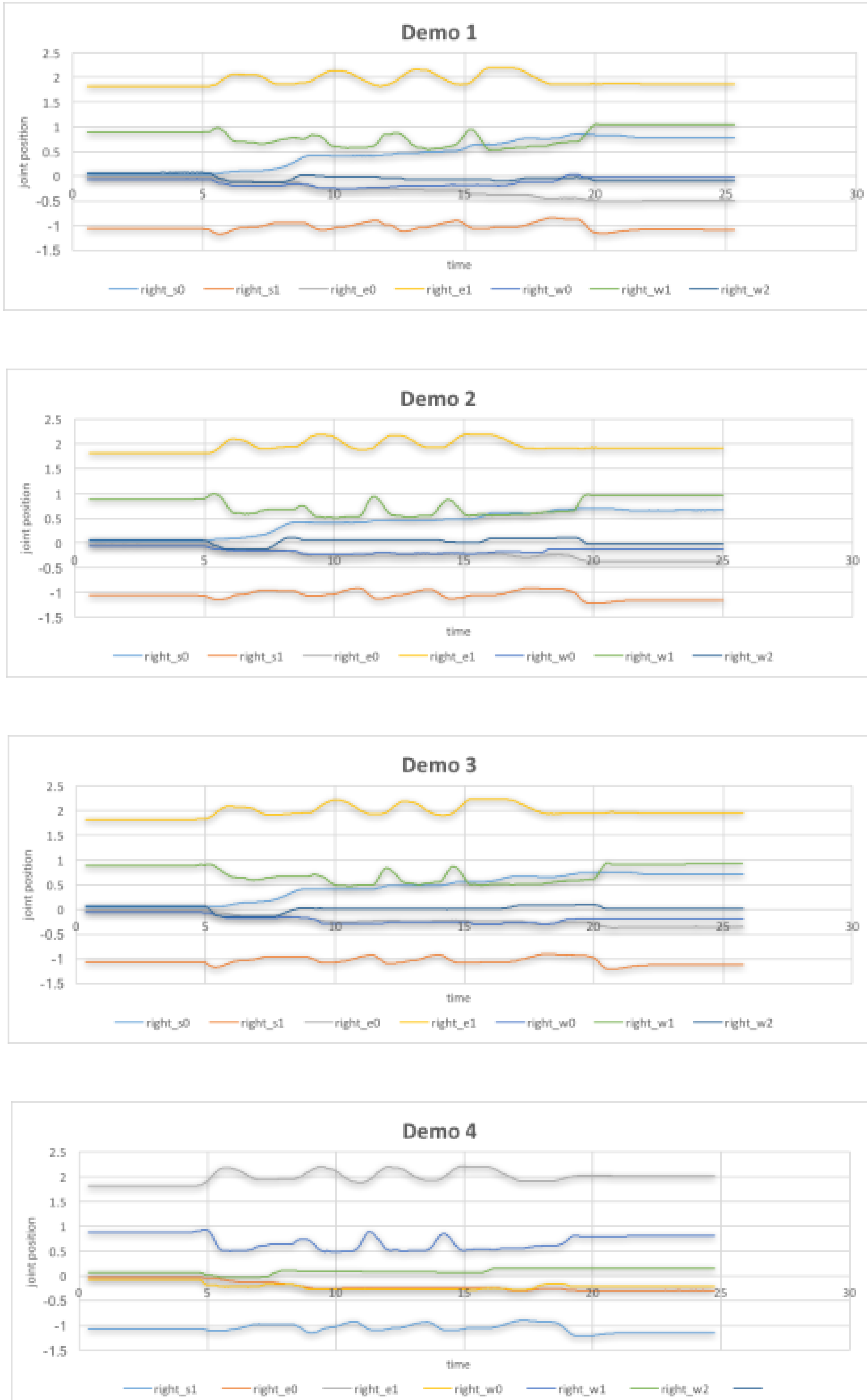

Figure 19: Task 3 - Plot for joint position data in demonstration samples

The learned model M_{LfD} is obtained by processing sample demonstrations using DBA in plain learning from demonstration framework. As seen in *Figure 20*, the resulting learned model from DBA is extremely noisy. To smoothen the noisy learned model, we used Gaussian smoothing. The result of Gaussian smoothening is as shown in *Figure 21*.

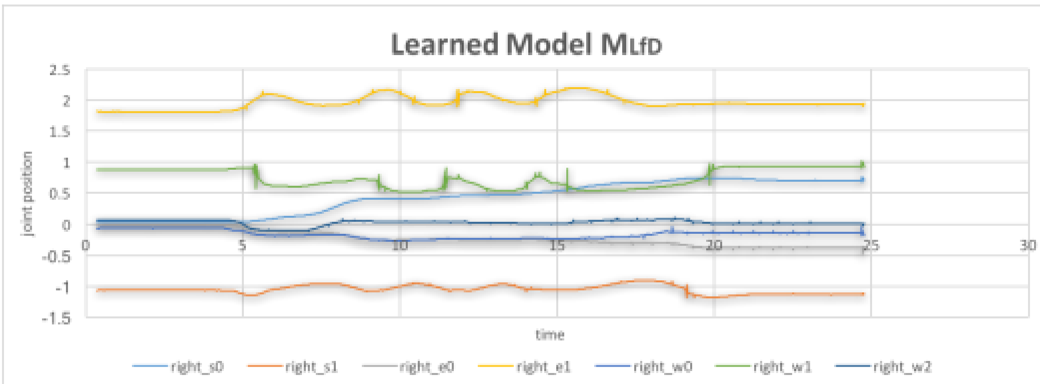

Figure 20: Task 3 - Plot for joint position data in Learned Model M_{LfD}

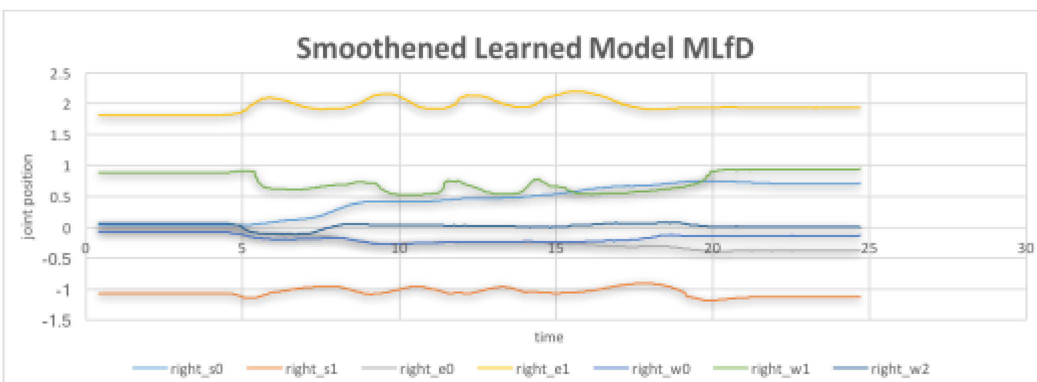

Figure 21: Task 3 - Plot for joint position data in Smoothened Learned Model M_{LfD}

The smoothened-learned-model change points are computed using the approach described in the section “Data selection strategy for query” of the methodology chapter. We found nineteen change points in the learned model as highlighted in

Figure 17. The learned model is tested on Baxter for writing “412”. While the learned model is being executed, the running instances of the model are mapped to change points. On encountering a change point, a label query “Is the subtask performed correctly?” is generated.

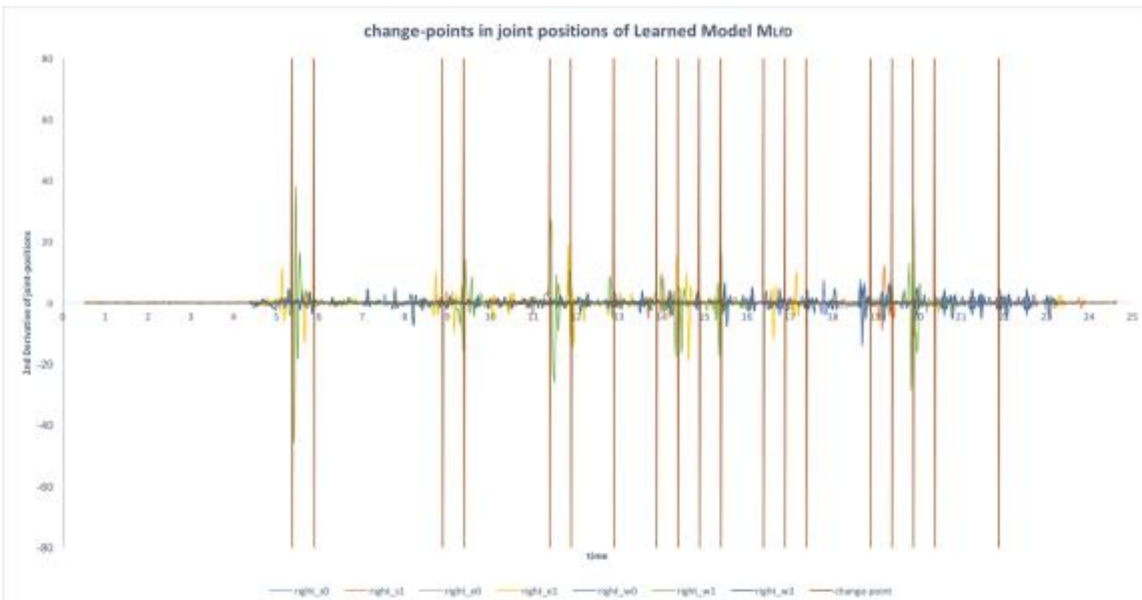

Figure 22: Task 3 - Plot for change points in joint positions in Learned Model M_{LfD}

Analysis on M_{LfD}: When tested on Baxter, the learned model M_{LfD} gave positive outcomes for numbers “4” and “1”; however, M_{LfD} failed to clearly write “2”. Along these lines, an interactive correction was required for writing “2” at the 12th change point in the form of a demonstration query. Subsequent to giving intuitive rectifications and updating the learned model M_{LfD} the resulting model M_{AL} is as shown in

Figure 23.

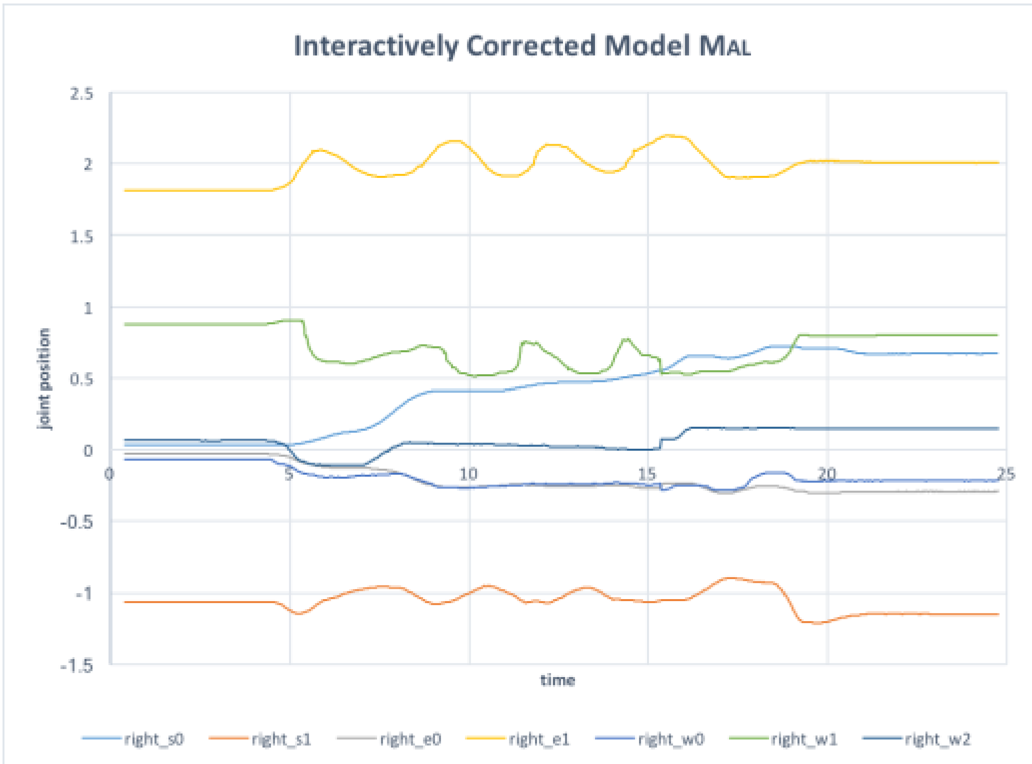

Figure 23: Task 3 - Plot for joint position data in Interactively Corrected Model M_{AL}

Analysis on M_{AL}: When tested on Baxter, the interactively corrected model was able to write all the numbers clearly and was able to complete the task successfully.

Evaluation: To evaluate the performance of M_{AL} over M_{LfD}, we computed the RMSE for the end-effector positions of the M_{AL} and M_{LfD} against standard Model M_{Std} and RMSE for the joint-positions data of the M_{AL} and M_{LfD} against M_{Std}.

RMSE	M_{LfD}	M_{AL}
joint-position data	0.0584	0.0348
end-effector data	0.0173	0.0261

Table 2: Task 3 - RMSE of joint-position data and end-effector data for M_{LfD} vs M_{AL}

Results from *Table 2* and performance of M_{AL} shows us that for a complex task, the M_{LfD} is not always completely reliable in terms of successful task completion and therefore requires interactive corrections.

4.2 Inference:

Experiment 1: Baxter learns to pick and place a box with fix start and end positions.

The learned model M_{LfD}, when tested on Baxter, gave positive results in terms of task completion and therefore did not require any interactive corrections.

Experiment 2: Baxter learns to pick and place a box with varying start and end positions.

The learned model M_{LfD}, when tried on Baxter, gave positive outcomes for pick; however, it neglected to put the box correctly. Therefore, an intelligent revision was given at the change point where the direction was wrong. After the intuitive amendment, RMSE decreased for both joint-position information and end-effector information. Additionally, the model acquired from the active learning M_{AL} gave positive outcomes as far as undertaking task completion contrasted with M_{LfD}.

Experiment 3: Baxter learns to write numbers “412”.

The learned model MLfD, when tested on Baxter, gave positive results for numbers “4” and “1” but failed to clearly write “2”. Therefore, an interactive correction was required for writing “2”. After providing interactive corrections and updating the learned model MLfD, RMSE subsided for only joint position data. There was no improvement in end-effector position data. However, the learned model from interactive correction using active learning MAL gave positive results in terms of task completion compared to MLfD.

Our analysis assumes that (i) human are equally and consistently noisy, and (ii) annotation is a noisy process over some underlying true label because human may not be reliable for several reasons. First, some instances are implicitly difficult for people and machines, and second, people can become distracted or fatigued over time, introducing variability in the quality of their demonstration/annotations. And this may have been one of the reasons for failure in improving the RMSE even after interactive corrections.

In the above three experiments, we have covered three cases:

case 1: simple task, undistinguishable demonstration samples.

case 2: simple task, distinguishable demonstration samples.

case 3: complex task, undistinguishable demonstration samples.

Finally looking at the results of all three experiments, it can be concluded that for a simple task, with very undistinguishable sample demonstrations, the MLfD obtained was quite reliable in terms of successful task completion (case1 covered under experiment 1). But as the complexity of the task being taught increases, even undistinguishable demonstrations do not yield a very reliable Learned model and require rectifications (case3 covered under experiment 3). Also in a true real world setting, it is far-fetched to expect similar and error-free demonstrations considering human factors, which is one of the major issues in LfD. In the second experiment of pick and place which covers a real world scenario (a simple task, varying demonstration samples), the learned model was not completely reliable as expected but only minor corrections were required towards the end to achieve the expected goal. The above results confirm our hypothesis that incorporating an Active Learning approach in a Learning from Demonstration framework outperforms passive learning both in terms of workable results and the ease with which it is incorporated, given a real world setting.

CHAPTER V

CONCLUSION

Active learning is a developing area of research in machine learning, doubtlessly filled with the fact that information is increasingly simple or modest to acquire yet troublesome or expensive to label for training. In the course of recent decades, there has been much work in figuring and understanding the different ways in which queries are chosen from the learner's perspective (CHAPTER III). This has produced a great deal of proof that a number of marked illustrations are important to prepare exact models and can be successfully decreased in a variety of applications.

Drawing on these establishments, in this thesis, we have explored active learning in a human- robot interaction setting. After comparing the task completions in various cases, we found that as the complexity of the task increases, the model obtained from LfD alone is not reliable enough. These M_{LfD} can be easily corrected by incorporating the Active Learning techniques. The performance of the learned models is evaluated on the basis of successful task completion and RMSE against the standard task.

This work provides a foundation for developing active learning systems that can be successfully deployed, with improvements, on robots in everyday human environments. As future work, our approach can also be used for creating semantic labeling for motion primitives which can further be used in the robot-teaching-human setting.

REFERENCES

[1] Monica Nicolette Nicolescu, "A framework for learning from demonstration, generalization and practice in human-robot domains," University of Southern California, 2003.

[2] Brenna D and Chernova, Sonia and Veloso, Manuela and Browning, Brett Argall, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, pp. 469--483, 2009.

[3] Simon Tong, "Active learning: theory and applications," Citeseer, 2001.

[4] Maya and Chao, Crystal and Thomaz, Andrea L Cakmak, "Designing interactions for robot active learners," *IEEE Transactions on Autonomous Mental Development*, vol. 2, pp. 108--118, 2010.

[5] Gregory and Settles, Burr and McCallum, Andrew Druck, "Active learning by labeling features," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, 2009, pp. 81--90.

[6] Burr and Craven, Mark and Friedland, Lewis Settles, "Active learning with real annotation costs," in *Proceedings of the NIPS workshop on cost-sensitive learning*., 2008, pp. 1--10.

[7] Xiaojin and Lafferty, John and Rosenfeld, Ronald Zhu, *Semi-supervised learning with graphs*.: Carnegie Mellon University, language technologies institute, school of computer science, 2005.

[8] Burr Settles, "Active learning literature survey," *University of Wisconsin, Madison*, vol. 52, p. 11, 2010.

[9] cs.northwestern.edu. [Online]. http://www.cs.northwestern.edu/~pardo/courses/mmml/lectures/active-learning-overview-pardo.pdf

[10] Djallel, Romain Laroche, Tanguy Urvoy, Raphael Féraud, and Robin Allesiardo Bouneffouf, "Contextual bandit for active learning: Active thompson sampling," in *International Conference on Neural Information Processing*, 2014, pp. 405--412.

[11] Djalle Bouneffouf, "Exponentiated gradient exploration for active learning," *Computers*, vol. 5, p. 1, 2016.

[12] Fredrik Olsson, "A literature survey of active machine learning in the context of natural language processing," 2009.

[13] Sonia and Veloso, Manuela Chernova, "Multi-thresholded approach to demonstration selection for interactive robot learning," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, 2008, pp. 225--232.

[14] Daniel H and Jenkins, Odest Chadwicke Grollman, "Dogged learning for robots," in *Robotics and Automation, 2007 IEEE International Conference*, 2007, pp. 2483--2488.

[15] E and d'Halluin, Florent and Billard, Aude Gribovskaya, "An active learning interface for bootstrapping robot's generalization abilities in learning from demonstration," in *RSS Workshop Towards Closing the Loop: Active Learning for Robotics*., 2010.

[16] Eric and Breese, Jack and Heckerman, David and Hovel, David and Rommelse, Koos Horvitz, "The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, 1998, pp. 256--265.

[17] Pieter and Ng, Andrew Y Abbeel, "Apprenticeship learning via inverse reinforcement learning," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 1.

[18] Monica N and Mataric, Maja J Nicolescu, "Natural methods for robot task learning: Instructive demonstrations, generalization and practice," in *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, 2003, pp. 241--248.

[19] Cynthia and Brooks, Andrew and Gray, Jesse and Hoffman, Guy and Kidd, Cory and Lee, Hans and Lieberman, Jeff and Lockerd, Andrea and Chilongo, David Breazeal, "Tutelage and collaboration for humanoid robots," *International Journal*

of Humanoid Robotics, vol. 1, pp. 315--348, 2004.

[20] Richard and Shavlik, Jude and Torrey, Lisa and Walker, Trevor and Wild, Edward Maclin, "Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression," in *AAAI*, 2005, pp. 819--824.

[21] Andrea L and Breazeal, Cynthia Thomaz, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," *Artificial Intelligence*, vol. 172, pp. 716--737, 2008.

[22] William D and Kaelbling, L Pack Smart, "Effective reinforcement learning for mobile robots," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4, 2002, pp. 3404--3410.

[23] Crystal and Cakmak, Maya and Thomaz, Andrea L Chao, "Transparent active learning for robots," in *IEEE*, 2010, pp. 317--324.

[24] Sonia and Veloso, Manuela Chernova, "Confidence-based policy learning from demonstration using gaussian mixture models," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 2007, p. 233.

[25] Manuel and Melo, Francisco and Montesano, Luis Lopes, "Active learning for reward estimation in inverse reinforcement learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2009, pp. 31--46.

[26] Stephanie and Dey, Anind K and Veloso, Manuela Rosenthal, "How robots' questions affect the accuracy of the human responses," , 2009, pp. 1137--1142}.

[27] Rehj and Schermerhorn, Paul and Scheutz, Matthias Cantrell, "Learning actions from human-robot dialogues," , 2011, pp. 125--130.

[28] Paul E and Yoon, Kevin and Stolarz, Jeremy and Veloso, Manuela M Rybski, "Interactive robot task training through dialog and demonstration," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, 2007, pp. 49--56.

[29] Maya and Thomaz, Andrea L Cakmak, "Designing robot learners that ask good questions," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, 2012.

[30] Melinda and Yeh, Eric and Myers, Karen Gervasio, "Learning to ask the right

questions to help a learner learn ," in *Proceedings of the 16th international conference on Intelligent user interfaces*, 2011, pp. 135--144.

[31] Hema and Madani, Omid and Jones, Rosie Raghavan, "Active learning with feedback on features and instances," *Journal of Machine Learning Research*, vol. 7, pp. 1655--1686, 2006.

[32] Dana and Takano, Wataru and Nakamura, Yoshihiko Kuli, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains," *The International Journal of Robotics Research*, vol. 27, pp. 761--784, 2008.

[33] Ale, "Programming full-body movements for humanoid robots by observation," *Robotics and autonomous systems*, vol. 47, pp. 93--108, 2004.

[34] Seungsu and Kim, ChangHwan and You, Bumjae and Oh, Sangrok Kim, "Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on," in *IEEE*, 2009, pp. 2518--2524.

[35] Scott and Chitta, Sachin and Barto, Andrew G and Marthi, Bhaskara and Osentoski, Sarah Niekum, "Incremental Semantically Grounded Learning from Demonstration.," in *Robotics: Science and Systems*, 2013.

[36] Luka and Babi, "Humanoid robot posture-control learning in real-time based on human sensorimotor learning ability," in *IEEE*, 2013, pp. 5329--5334.

[37] Jan and Hale, Joshua G and Oztop, Erhan Babi, "Human sensorimotor learning for humanoid robot skill synthesis," *Adaptive Behavior*, vol. 19, pp. 250--263, 2011.

[38] Pieter and Coates, Adam and Ng, Andrew Y Abbeel, "Autonomous helicopter aerobatics through apprenticeship learning," *The International Journal of Robotics Research*, vol. 29, pp. 1608--1639, 2010.

[39] sdk.rethinkrobotics.com. [Online]. http://sdk.rethinkrobotics.com/mediawiki-1.22.2/images/4/4d/Baxter_Joint_Names.png

[40] researchgate.net. [Online]. https://www.researchgate.net/profile/Giovanni_Magistris/publication/311585170/figure/fig2/AS:438646399606789@1481592859735/Figure-2-Images-of-the-the-pick-and-place-states-taken-during-Baxter-performing-the.ppm

[41] ZQ John Lu, "The elements of statistical learning: data mining, inference, and

prediction," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 173, pp. 693--694, 2010.

[42] Andrew Y and Russell, Stuart J and others Ng, "Algorithms for inverse reinforcement learning," in *Icml*, vol. 2000, pp. 663--670.

[43] Andrew Y and Russell, Stuart J and others Ng, "Algorithms for inverse reinforcement learning," , 2000, pp. 663--670.

[44] Fran, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, pp. 678--693, 2011.

[45] Gineke A and Reinders, Marcel JT and Hendriks, EA ten Holt, "Multi-dimensional dynamic time warping for gesture recognition," in *Thirteenth annual conference of the Advanced School for Computing and Imaging*, vol. 300, 2007.

[46] Hiroaki and Chiba, Seibi Sakoe, "A dynamic programming approach to continuous speech recognition," in *Proceedings of the seventh international congress on acoustics*, vol. 3, 1971, pp. 65--69.

[47] Hiroaki and Chiba, Seibi Sakoe, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, pp. 43--49, 1978.

[48] Sonia and Veloso, Manuela Chernova, "Multi-thresholded approach to demonstration selection for interactive robot learning," in *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction*, 2008, pp. 225--232.

[49] Rachel and Brodley, Carla E and Aernecke, Matthew and Walt, David and Friedl, Mark Lomasky, "Active class selection," in *European Conference on Machine Learning*, 2007, pp. 640--647.

[50] Monica Nicolette Nicolescu, "A framework for learning from demonstration, generalization and practice in human-robot domains," University of Southern California, 2003.

[51] Weiwei and Han, Yongkoo and Guan, Donghai and Lee, Sungyoung and Lee, Young-Koo Yuan, "Initial training data selection for active learning," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, 2011.

[52] Dana and Takano, Wataru and Nakamura, Yoshihiko Kulic, "Incremental on-line hierarchical clustering of whole body motion patterns," in *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, 2007, pp. 1016--1021.

[53] Cynthia and Scassellati, Brian Breazeal, "Robots that imitate humans," *Trends in cognitive sciences*, vol. 6, pp. 481--487, 2002.

[54] Stefan and Ijspeert, Auke and Billard, Aude Schaal, "Computational approaches to motor learning by imitation," *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, vol. 358, pp. 537--547, 2003.

[55] Neil and Kaplan, Dain and Sugiyama, Masashi Rubens, *Active learning in recommender systems*.: Springer, 2011.

[56] Chenxia and Lenz, Ian and Saxena, Ashutosh Wu, "Hierarchical Semantic Labeling for Task-Relevant RGB-D Perception.," in *Robotics: science and systems*, 2014.

[57] Cipriano and Fernandez-Madrigal, Juan-Antonio and Gonzalez, Javier and Saffiotti, Alessandro Galindo, "Robot task planning using semantic maps," *Robotics and autonomous systems*, vol. 56, pp. 955--966, 2008.

VITA

PREETI LEKHA

COMPUTER SCIENCE

Master of Science

Thesis: IMPROVING ROBOT LEARNIG USING AN ACTIVE LEARNING IN A LEARNING FROM DEMONSTRATION FRAMEWORK

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science/Arts in your major at Oklahoma State University, Stillwater, Oklahoma in December, May, or July, Year.

Completed the requirements for the Bachelor of Science/Arts in your major at University/College, City, State/Country in Year.

Experience:

Graduate Research Assistant at OSU App Center, Technology Development Center, OSU from March 2015 – May 2017.