January 2013

Human Intention Recognition Based Assisted Telerobotic Grasping of Objects in an Unstructured Environment

Karan Hariharan Khokar
University of South Florida, kkhokar@mail.usf.edu

Human Intention Recognition Based Assisted Telerobotic Grasping of Objects in an Unstructured Environment

by

Karan Khokar

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Mechanical Engineering
College of Engineering
University of South Florida

Major Professor: Rajiv Dubey, Ph.D.
Redwan Alqasemi, Ph.D.
Sudeep Sarkar, Ph.D.
Kyle B. Reed, Ph.D.
Kandethody Ramachandran, Ph.D.

Date of Approval:
November 21, 2013

Keywords: robotics, machine learning, teleoperation, Hidden Markov Model, human robot interaction

Dedication

To mummy, papa, Ishan and Guruji

Acknowledgments

I would like to thank all the members of the Rehabilitation Robotics Lab for helping me in this project including Paul Mitzlaff, Mustafa Mashali, Daniel Ashley, Kester Duncan, Shyam Ramnath, and Indika Pathirage. I would also like to thank Dr. Derek Lura from Rehabilitation and Robotics Testbed lab. I extend my thanks to all the previous lab members including Fabian Farelo, Dr. Eduardo Veras, Peter Schrock and Garrett Pence.

I would like to thank my major professor, Dr. Rajiv Dubey and committee members, Dr. Redwan Alqasemi, Dr. Sudeep Sarkar, Dr. Kyle Reed and Dr. Kandethody Ramchandran for guiding me throughout this project. I would especially like to thank Dr. Rajiv Dubey and Dr. Redwan Alqasemi for supporting me and for being so patient with my research. This has been a great learning experience for me at the University of South Florida both professionally and personally. I would like to thank Merry Lynn Morris for giving me experience with dance at Dance Department.

I would like to thank my mom, dad and brother for their constant support, encouragement and motivation. Their sacrifices have enabled me to reach this stage of my education and professional development. I would also like to thank my cousins and friends back home. A big thanks also goes to my friends in Tampa and in US who are like family to me.

Last but not the least, I would like to thank the University of South Florida for the immense opportunity it has given me to reach my goals both academically and recreationally. I especially thank Student Affairs and the Center for Student Involvement for their support in enabling me to develop leadership and teaching qualities.

Table of Contents

List of Tables

List of Figures

Abstract

In this dissertation work, a methodology is proposed to enable a robot to identify an object to be grasped and its intended grasp configuration while a human is teleoperating a robot towards the desired object. Based on the detected object and grasp configuration, the human is assisted in the teleoperation task. The environment is unstructured and consists of a number of objects, each with various possible grasp configurations. The identification of the object and the grasp configuration is carried out in real time, by recognizing the intention of the human motion. Simultaneously, the human user is assisted to preshape over the desired grasp configuration. This is done by scaling the components of the remote arm end-effector motion that lead to the desired grasp configuration and simultaneously attenuating the components that are in perpendicular directions. The complete process occurs while manipulating the master device and without having to interact with another interface.

Intention recognition from motion is carried out by using Hidden Markov Model (HMM) theory. First, the objects are classified based on their shapes. Then, the grasp configurations are preselected for each object class. The selection of grasp configurations is based on the human knowledge of robust grasps for the various shapes. Next, an HMM for each object class is trained by having a skilled teleoperator perform repeated preshape trials over each grasp configuration of the object class in consideration. The grasp configurations are modeled as the states of each HMM whereas the projections of translation and orientation vectors, over each reference vector, are modeled as observations. The reference vectors are the ideal translation and rotation trajectories that lead the remote arm end-effector towards a grasp configuration. During an actual grasping task performed by a novice or a skilled user, the trained model is used to detect their intention. The output probability of the HMM associated with each object in the environment is computed as the user is teleoperating towards the desired object. The object that is associated with the HMM which has the highest output probability, is taken as the desired object. The most likely Viterbi state sequence of the selected HMM gives the desired grasp configuration. Since

an HMM is associated with every object, objects can be shuffled around, added or removed from the environment without the need to retrain the models. In other words, the HMM for each object class needs to be trained only once by a skilled teleoperator.

The intention recognition algorithm was validated by having novice users, as well as the skilled teleoperator, grasp objects with different grasp configurations from a dishwasher rack. Each object had various possible grasp configurations. The proposed algorithm was able to successfully detect the operator’s intention and identify the object and the grasp configuration of interest. This methodology of grasping was also compared with unassisted mode and maximum-projection mode. In the unassisted mode, the operator teleoperated the arm without any assistance or intention recognition. In the maximum-projection mode, the maximum projection of the motion vectors was used to determine the intended object and the grasp configuration of interest. Six healthy and one wheelchair-bound individuals, each executed twelve pick-and-place trials in intention-based assisted mode and unassisted mode. In these trials, they picked up utensils from the dishwasher and laid them on a table located next to it. The relative positions and orientations of the utensils were changed at the end of every third trial. It was observed that the subjects were able to pick-and-place the objects 51% faster and with less number of movements, using the proposed method compared to the unassisted method. They found it much easier to execute the task using the proposed method and experienced less mental and overall workloads. Two able-bodied subjects also executed three preshape trials over three objects in intention-based assisted and maximum projection mode. For one of the subjects, the objects were shuffled at the end of the six trials and she was asked to carry out three more preshape trials in the two modes. This time, however, the subject was made to change their intention when she was about to preshape to the grasp configurations. It was observed that intention recognition was consistently accurate through the trajectory in the intention-based assisted method except at a few points. However, in the maximum-projection method the intention recognition was consistently inaccurate and fluctuated. This often caused to subject to be assisted in the wring directions and led to extreme frustration. The intention-based assisted method was faster and had less hand movements. The accuracy of the intention based method did not change when the objects were shuffled. It was also shown that the model for intention recognition can be trained by a skilled teleoperator and be used by a novice user to efficiently execute a grasping task in teleoperation.

Chapter 1: Introduction

In this section, the problem statement and the motivation behind solving the problem are introduced. This is followed by a brief overview of the project which includes the objectives of the project and the unique contributions. After this, a description of the previous related work is presented.

1.1 Introduction to the Problem

In general, an indoor environment is unstructured and has a number of objects, each with a number of different grasp configurations. Wheelchair-bound individuals are unable to reach out to areas that are too high, too low or otherwise unreachable, due to their disability. Elderly individuals face similar problems. A robotic arm mounted on a wheelchair or a mobile platform can assist them in grasping and manipulating objects. However, autonomous grasping is not robust and teleoperating a robotic arm is a physically and mentally challenging activity. On the other hand, humans have cognitive abilities and knowledge from experience that even the most advanced robots lack. A grasp with a configuration that is selected by a human is more likely to succeed than the one selected by a robot. Moreover, they are needed in the loop of robotic control to inform the robot about the object and the grasp configuration they are interested in, for grasping. The configuration for grasping is dependent on the task that the human would like the robot to perform after the object has been grasped. Elderly and individuals with disabilities still possess good decision making capabilities which can be used in a shared supervisory control framework to work collaboratively with the robot and grasp objects. The human controlling the robot must be able to command the robot about the object and grasp configuration of interest with ease and minimal involvement.

1.1.1 Problem Statement

A methodology for controlling the robot is needed, that identifies the object and grasp configuration of interest for the robot so that either the object can be grasped autonomously at the desired configuration or the human can be assisted to preshape and grasp the object. The human should be able to indicate this to the robot easily without any additional interface burden. Thus, the system should be able to detect the human intention and assist the human to grasp the object. The method should also be able to detect the changes in intention on the fly and assist the human accordingly. The method should be able to work accurately in unstructured environments i.e. shuffling the objects in the environment should not require a new model to be trained and should not affect the accuracy of the intention recognition algorithm. The method also needs to be scalable i.e. adding or removing objects from the environment should not affect the accuracy.

1.2 Motivation

Over the past few years, research has increased in the area of service robots for carrying out Activities of Daily Living (ADL) in indoor environments, especially for persons with disabilities and the elderly [1]-[4]. These service robots are either in the form of a mobile manipulators [3]-[4] or wheelchair mounted robotic arms [1],[2]. Grasping is an important task since it is involved in most ADLs.

Autonomous grasping is a widely researched area [5-25]. Early research in grasping [5-14] mainly focused on (a) developing mathematical models of grasping, (b) geometric and force analysis of grasping 2D and 3D polyhedral shapes and, (c) determining optimum number of finger contacts for grasping. The aim was to determine the necessary and sufficient conditions for a stable grasp. The approach to the grasping problem was from a theoretical standpoint rather than a practical standpoint. Grasping commonly encountered objects was not considered and ideal models of the hand and the object were assumed. With the advances in computing, it became possible to model shapes of commonly encountered objects and the field of grasp planning emerged. In grasp planning [15-17, 19, 20], all the stable grasps that could grasp an object were run through an optimization step, based on a quality criteria, to determine the best grasp. The quality criteria could be force-closureness [20, 21], based on task requirements [19], based on clutter [15] or a combination of some or all of the above criteria. Although, the various methods of grasp planning were successful in determining stable grasps, the main drawback was the large computing time needed for determining the grasp. This was due to the high dimensionality of the search space over which an optimization of the grasp criterion was carried out [18, 26]. The high dimensionality was due to the large end-effector Cartesian space around the object where grasps would occur. It was also due to the high configuration space of the multi-DOF hands. This made grasp planning a time consuming and computationally intensive process. To circumvent this problem, programming-by-demonstration (PbD) or learning-from-demonstration (LfD) techniques have been developed [22-25, 27-30]. These methods were of two kinds, the human hand centered [22, 23] and the object centered [24, 25, 27-30]. In the human hand centered methods, the human would first demonstrate a grasp to the robot, the robot would then create a learned model of the grasp from the sensor data and then the grasp would be reproduced. Human hand centered LfD methods determined the grasp quicker but had the following drawbacks: (a) They depend on a reliable vision system to identify the object, (b) grasp determination was only applicable to trained objects, (c) mapping from human hand to robot hand was needed for the method to be practicable and, (d) inaccuracies were introduced as humans have different ways of grasping the same object at the same configuration. In the object centered LfD methods, the learning algorithm would learn features of the object that resulted in good grasps and would determine grasps for novel objects by finding similar features. The main drawback of these methods was that, in some cases, the grasps returned were good grasps for objects used for training but were not that good for on objects used in testing. This was true if the trained and tested objects were different. For example, a pencil would be grasped from its mid-point and so would be the bottle, although this grasp configuration for the bottle may not be the best one. Moreover, these methods would return a grasp or grasps that may be difficult or impossible to achieve due to kinematic or environment constraints. The human operating the robot was unable to change the grasp. Thus, although LfD methods have shown great amount of success, a method that allows the human to choose a grasp configuration, out of the various LfD trained grasp configurations is needed. The human may choose the grasp depending on the task requirements and/or based on the feasibility of the grasp.

Ciocarlie and Allen [18] have demonstrated that a human input can reduce the time it takes for an autonomous planner to compute a robust grasp. In their work, the human preshaped a gripper at a point around the object and their grasp planner computed a locally optimized grasp. This reduced the problem from a global to a local optimization one. Their method took less than 2 seconds to compute a force closure grasp, which was far quicker than what autonomous planners usually take. Their work demonstrates that a human selected initial pose leads to a robust grasp faster than the grasp planners and that, the grasp is more likely to be successful. Human's cognitive abilities and their knowledge gathered from experience are better than the most advanced robots. Humans can plan, analyze, act, react and in general process information quicker, more efficiently and in a more robust manner than the state-of-the-art computing systems. In indoor tasks involving a number of objects in cluttered unstructured environments, a human is needed in the loop for commanding the robot on the object to be grasped and its grasping configuration.

Teleoperation of a remote arm offers a direct way of involving human in the loop but it has been largely unused for service robot applications. This is because teleoperation is a mentally and physically challenging activity [31, 32]. Teleoperation is the remote control of any mechatronic device e.g. a robot, whereas telemanipulation is specifically controlling a robotic manipulator or a robotic arm for manipulating a remote environment [33]. In an indoor setting, the arm can be mounted on a mobile base or a wheelchair. It could also be mounted on a stationary base such as a kitchen platform or an office table. The user could be controlling the remote arm with an input device such as a joystick. Input could also be acquired from a smaller low-impedance robotic arm, a data glove or it could be gesture-based control. This device is called the master device and the remote arm is also known as a slave.

Transferring motions from the master to the slave manipulator leads to errors due to mapping and scaling. It makes the process time consuming and leads to frustration on the part of the user. Errors are produced even in translating the remote arm end-effector; a task that might seem simple to execute. Preshaping for grasping an object involves rotations of the gripper, which are more challenging to execute. Orienting the remote arm gripper, so that it is at a convenient configuration for grasping an object of interest, is one of the most difficult sub-activities [34]. Robots, on the other hand, are good at following trajectories accurately, quickly and in a more stable manner, once the trajectory is known. This applies to translational as well as rotational motions, and for linear as well as curved trajectories. Robots do not get tired or frustrated, neither do they experience fatigue. Thus, human abilities and robot capabilities must be combined in such a way, that they complement each other to collaboratively enable task execution in an efficient manner.

To make teleoperation easier, various computer mediated techniques such as virtual fixtures [35], potential fields [36] etc. have been developed. These techniques assist the operator by guiding the motion of the robot towards the target. In this manner, these techniques reduce the operator's workload and increase their task performance. In order to determine the appropriate trajectory and hence the direction of guidance, researchers have made use of motion intention recognition [37]. The tasks executed in [37] were simple linear trajectory following tasks.

Other than the LfD techniques mentioned previously, there are several works in which humans have been involved in the loop for grasping. In these works, unlike the LfD methods, the involvement of the human is during the task execution rather than in the offline stage. Human input has enabled robotic vision systems to interpret a scene better by helping the system to segment and/or recognize objects [38], [39]. Once an object is recognized, an autonomous planner would evaluate the grasp on the object and grasp would ensue. Better grasp success rates were obtained when the human was involved in the grasping process than when the grasp was executed autonomously. In [3] and [40], the human points to the object of interest using a laser pointer and the robot autonomously completes the grasp. Only one grasp configuration and objects on flat surfaces were considered. In [41], a human was involved in the loop in scenarios when the robot failed to perform a grasp autonomously. In these cases, the human would interact with a virtual world, which was a replica of the real world, using a mouse. The human would locate and orient the virtual gripper at a desired grasp configuration over the virtual object of interest, by operating one DOF at a time. The physical robot would then complete the grasp autonomously. The main drawback of this system was that the user could operate only one DOF at a time, which would be time consuming and tiring for the user.

In this work, a novel human-in-the-loop (HitL) grasping strategy, in which the human's intention from motion in teleoperation is used to predict the desired object and grasp configuration, is presented. Based on the prediction, the human is assisted to preshape the remote arm gripper over the detected object and grasp configuration using scaled teleoperation [42]. The motion of the remote arm is amplified in the directions that lead to the desired object and grasp configuration and, attenuated in the directions that do not. Motion intention recognition is carried out by processing user motion data through an HMM based model. The model is trained by a skilled teleoperator since they have the least errors and tend to follow the shortest translation and rotation path. The proposed is intended for use by wheelchair-bound individuals who are unable to reach out to surfaces in indoor environments for picking up objects due to their disability. They are not necessarily skilled teleoperators.

1.3 Project Overview

In order to detect the desired object and grasping pose that the user is interested in, a probabilistic model comprising of an exponential distribution model and Hidden Markov Model (HMM) theory has been used. [43] provides an excellent reference to the HMM theory. Although the application they have discussed is on speech recognition, HMM theory can be applied for any application that can be modeled as a doubly embedded stochastic process. We have defined and explained Hidden Markov Models (HMM) in section 3.3 of chapter 3. For motion intention recognition using HMMs we first classify objects based on shapes and preselect orthogonal grasp configurations for each object class.

An HMM is associated with each object class and the states of that HMM are the various preselected grasp configurations for that shape. The parameters of each HMM are trained by having a skilled teleoperator repeatedly preshape from random starting poses to various preselected grasp configurations and, by recording and analyzing their teleoperation data.

Once trained, the intended object and grasping pose is determined by solving the problems 1 and 3 of the HMM theory [43]. Problem 1 is that of determining the likelihood that a given set of observations are from an HMM. Problem 2 is that of determining an optimal state sequence given the HMM and the observations. Thus, the likelihood of the HMM for each object in the environment is determined on the fly as the user is teleoperating towards an object. As per the problem 1, the HMM with the highest likelihood is the shape and hence the object that is most likely to be the intended object. In other words, that is the object of interest based on motion data up to that time instant. Once the object has been determined, solve problem 2 is solved to determine the most likely state sequence for that object. The most likely state sequence gives the grasp pose that the user intends to grasp the object from. The details of the HMM models, the training procedure, the intention recognition algorithms etc. are provided in chapter 3.

Once the object and grasp configuration are detected, assistance is provided using a concept called scaled teleoperation [42]. In scaled teleoperation, the user's input is broken down into components along the desired translation and orientation vectors and, along vectors orthogonal to the desired. The components along the desired vectors are scaled up and those along orthogonal directions are scaled down. This results in a visible increase in motion of the arm along the desired directions. The increase in motion is proportional to how accurately the desired vectors are being followed by the user. The larger the deviation, the lower is the motion seen at the remote arm in terms of magnitude. However, the proportion of motion along the desired direction is still more than that in the orthogonal directions. This not only reduces errors due to deviation from the desired trajectory but also gives the user an intuitive feedback as to what direction he/she should be teleoperating towards in order to reach the target faster. This concept assists the user to follow a trajectory more accurately and helps them in executing it faster. The amount of assistance depends on the magnitude of the scaling factor. If its value is the maximum possible, there will be no motion output at the remote arm when the user is moving along a trajectory perfectly orthogonal to the desired and any motion in any other direction will produce a motion output proportional to how close the input is to the desired. The vector algebra of how this is achieved is explained in section 2.4 of chapter 2.

Moreover, since the grasp poses represent the mental states, they are hidden and hence are justified being modeled as HMM states. On the other hand, translation and orientation vectors are visible and are justified as being modeled as HMM observations. Thus, the mental states of the user are decoded by processing their observed motions through the HMM as the user is teleoperating. When the intention (either object or grasp pose intention) of the user changes, the likelihood values computed by the algorithms change and the system detects the new intention.

In this work, the various grasp configurations of the different objects have been manually measured. However, a vision based algorithm that automatically determines the grasp configurations is necessary to be integrated with the system.

The scenario where the user is able to directly observe the environment that needs to be manipulated i.e. without any feedback from a camera has been specifically considered. The person teleoperating is expected to have decent degree of functionality in at least one of the upper limbs.

1.3.1 Objectives

The following are the objectives of this research:

a) Formulate a methodology for intention recognition based identification of object and grasp configuration in a cluttered unstructured environment, with each object having various possible grasp configurations.

b) Implement the devised method on a physical robot that is part of a teleoperation set-up.

c) Validate the methodology by determining the accuracy of intention recognition.

d) Compare the developed methodology with other methods and determine the benefits.

1.3.2 Contributions

The novel contributions of the proposed work are listed as follows:

a) The unsolved problem of object and grasp configuration identification, from motion data in teleoperation, in a multi-object and multi-grasp configuration set-up, has been solved.

b) The Hidden Markov Model (HMM) theory has been utilized in a novel manner by modeling objects as HMMs and grasp configurations as states.

c) Objects can be shuffled around, added or removed from the environment without the need for training the system.

d) A novice user need not train a model for intention recognition i.e. they can use the model trained by a skilled teleoperator novice and skilled users to quickly and easily execute a grasping task in teleoperation.

e) Novice and skilled users are quickly and easily able to execute a grasping task in teleoperation using the model trained by a skilled teleoperator.

f) Scaling has been implemented to assist a user teleoperating the robot in rotation. Thus, the user is assisted in all the 6 degrees of freedom (DOF) in space.

1.4 Previous Work

1.4.1 Human-in-the-Loop Grasping

There has been some research on grasping in teleoperation where the human is constantly involved. Utsumi et al. [44] assisted the operator, driving an underwater backhoe system, in grasping objects by using augmented reality. Objects underwater are not as clearly visible with camera feedback as on the ground due to turbidity. As the user touches the partially observable object, it is reconstructed virtually and rendered for the user. They call this rendering a haptic image. They then use this model to grasp the object. Results for tests in simulation showed good results for haptic image assisted grasping of invisible objects.

In [3] and [45], Kemp et al. have experimented with various interfaces for shared autonomy such as button-based teleoperation and, point and click method for reaching tasks. This work involved human in grasping tasks depending on the autonomous grasp execution ability of the robot. There was no human involvement when objects could be segmented and the grasp planner could grasp them autonomously. When an object could not be segmented, the human specified the grasp pose by manually positioning and orienting a virtual model of the robot's hand on a screen. When the grasp planner failed to plan a path to the object, the human would be involved in manually positioning and orienting the hand at via points to guide the robot arm. This method of commanding grasps can work well but, if the user decides to grasp a different object or the same object from a different configuration mid-way during a grasping task, then the arm will need to be stopped and new grasp pose will need to be entered. Thus, it might be cumbersome to change grasp poses on the fly. With our method this change is quicker as the user only need teleoperate in a different direction until intention is detected. In their work, the user might have to change the view point of the virtual environment to get the mouse pointer to a grasp pose for commanding. In our case, the user need not make such adjustment. This saves time and makes control easier for the user.

Srinivasa et al. have presented an example of human in the loop grasping [38] in which activities such as object image acquisition, 3D object reconstruction, surface reconstruction, model evaluation etc. are divided between the human and the robot for object recognition and for grasping an object. Inexpensive human help from Amazon's crowdsourcing platform, Mechanical Turk, is utilized. Their challenges included quality control, latency, constructing correct interface for workers and, limited technical background and attention spans of humans online. They use human input only in object recognition. Grasping is autonomous. No assistance in grasping or grasp pose recognition is implemented in their work.

1.4.2 Human Motion Intention Based Assistance in Teleoperation

The work in the first decade of the 21st century saw new work, not only automatic segmentation and recognition of human motion but also using this recognition to assist the human to accomplish the task. Motion intention recognition is used to determine the category of sub-task the user intends to execute in teleoperation and assist the use in executing the identified sub-task easily and/or quickly. In some cases, the sub-tasks are mutually exclusive or independent of each other and the user, at a point in time, may be executing one of the several sub-tasks. In some cases all the individual sub-tasks may make up the total task and the user has to execute each in some sequence to complete the task in teleoperation. The values of a certain physical quantity usually help in determining the sub-task the user is executing at a point in time. These quantities may be discrete or continuous. They may have distinct values for each sub-task or the values may overlap in the space of the physical quantity. In the former case deterministic models can determine human intention and in the latter case a statistical model needs to be used. HMM is the model choice in almost all of these works because of the sequential nature of the sub-tasks in most of the works. In cases where the sub-tasks are not sequential, the HMM still gives a good classification. The physical quantity that distinguishes the sub-tasks becomes the feature data that trains the HMM. Other methods may perform better than HMM or on par with HMM but none of these works use any other method to give a comparison with HMM. In some of these works where the sub-tasks are mutually exclusive, a simpler binary model can be used instead of the HMM. Our work has taken a lot of inspiration from these works. We have also used human motion intention recognition to identify the sub-task the user is interested in executing and accordingly assist the user in executing it in teleoperation. However, our sub-tasks are the various grasping poses that can grasp an object; something which has never been implemented using intention based assisted teleoperation. Moreover, we are trying to solve the problem of grasping an object with a specific pose of interest using assisted teleoperation in a cluttered unstructured environment. The environment has a number of objects of similar or different shapes and each with a number of grasping poses that the user can select from. Our methodology determines the object and grasping pose of interest as the user is teleoperating and assists the user to preshape towards that pose.

Kragic et al. [37] implemented HMM by using the HMM Toolkit. Used Baum-Welch (B-W) for training with initialization from the toolkit and made use of problem 1 of HMM i.e. finding the output probability of the model given the partial set of observations or the likelihood that the given observations came from a particular model. They did not make use of the Viterbi [43] at all for segmentation or recognition. They were the first ones to provide assistance to the operator operating a robotic manipulator based on the output from HMM. The assistance was in the form of a virtual fixture that guided the operator through the trajectory. They found an improvement in the operator performance as a result of action recognition based assistance.

Their central idea was to train individual HMMs, one for each sub-task. This contrasted with previous implementations by Hannaford and Blake in which only one HMM was trained for the whole task and individual states were the sub-tasks. It was, however, similar to the structuring of HMMs in speech recognition where each phoneme was modeled as an HMM and a network of such phonemes was used to model a word. They sub-tasks that they modeled HMMs as were simple motion primitives, such as approach, insert, withdraw, remove etc. (think in the context of a peg-in-hole task). By modeling an HMM as a sub-task or motion primitives (they called them gestemes), they were able to define a vocabulary of primitives that could be used as a task language (similar to language in speech). Any task using a combination of gestemes could then be identified. To identify the sub-tasks they ran all the HMMs in parallel and the one with the highest output probability based on partial observations gave the most-likely HMM and hence the most likely sub-task or gesteme that the user is executing. This way tasks and operator actions could be identified and then appropriate assistance could be provided. Using a set of HMMs for a set of corresponding gestemes and being able to identify any task made up of a few or all of such gestemes, provided the authors with good scaling or extensibility to the human action identification using HMMs.

In one of their implementations, they modeled peg-in-hole and painting tasks with a subset of gestemes from the set that consisted of human actions such as approach, position, insert, withdraw, remove and paint. As mentioned, one HMM was designed for each gesteme. Each HMM was a 5-state SLR model. The observations were vectors of 7 values viz. 6 force/torque values in the end-effector frame (measured from a 6-axis force/torque sensor mounted at the end-effector) and magnitude of translation from previous end-effector frame to the current. They trained each HMM individually by repeating actions for that gesteme. Then solved problem 1 of HMM, as described earlier, to determine the most likely gesteme as the user was executing a task. They used user-segmentation of task, obtained by pressing a keyboard spacebar, to compare the automatic segmentation and thus validate the recognition algorithm. They obtained a recognition accuracy of 84% for peg-in-hole task and 90% for paint task. Then the gestemes trained for the peg-in-hole task were used in the recognition of the paint task and vice-versa, the recognition accuracy dropped only by 1.08%. Recognition accuracy is defined as the percentage of samples out of the total number of samples, that recognize the same gesteme as that provided by the user via spacebar presses.

The recognition via output probability had latency. This may have caused a delayed detection in the change in gesteme and it may have affected the recognition accuracy but, no such point was mentioned. The authors did not specify the output probability distribution i.e. discrete or continuous (Gaussian or some other kind).

In the second implementation [46], they used HMM to switch the virtual fixture on or off depending on whether the operator is following a pre-specified curved trajectory or leaving it to avoid an obstacle. The gestemes were silence (no action), follow the curve or leave the curve. The task was to follow a curve while avoiding obstacles on it. The operator followed a virtual 2D curved trajectory and avoided obstacles that were displayed on the graphical interface.

Similar to the first implementation, they used 5-state SLR HMMs to model individual gestemes and the HMM with highest output probability value, based on partial observations, is used to determine the most likely gesteme. The operator applies forces on the end-effector while executing a gesteme. The observations used are absolute values of the projections of the force on the tangent to the closest point on the curve and perpendicular to it, the absolute magnitude of the dot product of the force vector and the unit vector along the tangent to the closest point on the curve. It is not understood as to why the dot product was used as an observation when it is the same as the magnitude of projection on the tangent. They obtained an average recognition accuracy of more than 90% when the trained models were tested on the same curves and on different similarly shaped curves. They also compared accuracy and times for completing a curve following task with one obstacle mid-way suing HMM+VF mode, VF mode and NG (no gain) mode. In HMM+VF mode, the VF was switched on or off depending on the HMM recognition of curve-following gesteme or leaving the curve gesteme. A hard VF is applied in follow-curve case. In the VF mode, a constant soft virtual fixture was applied irrespective of the obstacle, and in NG mode, no assistance was provided. The HMM+VF mode was more accurate and faster than the other two modes, although statistical significance could not be proven for all cases.

In the implementation just described, a simpler binary method could have very well identified the user action. This is because the projection magnitudes on the tangent and its perpendicular differ by large amounts for each gesteme and a simple comparison between these values would have given the gesteme intended by the operator. In the curve-following task, the projection magnitude on the tangent was substantially greater than that on its perpendicular and vice-versa when the gesteme is leaving the curve. This is true as long as the operator leaves the curve in a direction perpendicular to it. The authors mention the idea of changing the compliance of the VF depending on the certainty of the recognized action or output probability value.

Aarno et al. [47] used a combination of K-means clustering, Support Vector Machines (SVMs) and HMMs to implement adaptive virtual fixtures based assistance in trajectory following during teleoperation. The direction of the virtual fixture changed with the direction of motion in teleoperation so that the operator was provided with appropriate assistance. In their implementation, a task consisted of a number of sub-tasks or trajectories that are executed in a certain order to accomplish the task. For developing the model, the input data is filtered to reduce noise. The input data is in the form of unit vectors along the directions of motion and is generated from 3D coordinates as the arm is moving. K-means clustering estimates the number of sub-tasks or linear trajectories within a task. Each sub-task differs from others only in terms of the direction of motion. K-means precluded the need for manual labeling of the trajectories. Depending on the number of clusters generated for a task, an SVM is trained for each task which gives the probabilities of unit vectors of motion belonging to a certain linear trajectory. They justified the use of SVMs by not needing to convert input data to its discrete form and that no assumption of a probability distribution will be needed. These SVMs form the observation probabilities for each linear trajectory, which is a state in the HMM. Again, a separate HMM is created for each task. The HMM, also referred to as a state sequence analyzer, is then trained to learn the most likely linear trajectory to be followed for a given task. When executing a task, the HMM is used as an online state estimator to determine the state or the linear trajectory that the user is following in teleoperation. The

HMM uses the state sequence probability information as well as the SVM modeled state-observation probability information to infer the states at each time instance as the arm is moving. Once the linear trajectory is known, a virtual fixture along that direction is automatically applied to assist the user. Baum-Welch algorithm was used for training the state-transition probabilities and initial state probabilities. The state with the highest value of δ [43] at every instant was used as the recognized state. As we will see, δ for a state is the probability of being in that state over all states at that time instant given the observations up to that time instant. For validation of the methodology, each user repeated the task 5 times for training the models. When tested on the same task, good segmentation of the linear trajectories was obtained and depending on the recognized state, the virtual fixtures were applied automatically to assist the user. The accuracy of segmentation was not quantified. The method adapted well to changes in the environment, such as increasing the height of the obstacle or adding a new obstacle. In the latter case, if the system was unsure about the trajectory the user undertook, to avoid the obstacle, the virtual fixture was removed so that the user could avoid the obstacle using unaided teleoperation. The main drawback of this method is that the user will only be assisted in the environments that the model is trained on, and it is not suitable for unstructured environments. The motivation behind this work was to assist the user in various assembly tasks in the manufacturing industry, where the environment is structured. Our method is designed to segment the task in unstructured environments. The only information system has is about the shape and the pose of the objects in the environment and depending on operator motion, our method provides assistance in appropriate direction. We consider translation as well as rotation for assistance and segmentation.

Yu et al. [48] applied a VF, a repulsive potential field or an attractive potential field depending on the sub-task viz. following a trajectory, avoiding an obstacle or aligning with a target. The modeling of HMM for the task, training and recognition was exactly the same as that in [46]. Baum-Welch (B-W) along with forward-backward algorithm is used for training and the HMM with the highest output probability at an instant gives the recognized sub-task based on partial observations. Multi-dimensional 5-state Bakis HMMs were trained for each sub-task. The components of the input velocity along the trajectory and along its perpendicular were used as observations. These continuous values are converted into discrete vectors by using FFT and VQ. This pre-processing of the data is similar to that carried out by Yang et al.

[49].The observation probabilities are in the form of two matrices of size 256X2, with one column for each state. There are two matrices since the HMM is two dimensional. The two dimensions of observations are considered independent of each other.

The concept was tested in a 2D structured virtual environment consisting of predefined trajectories, targets and obstacles. The user manipulated a Phantom Omni and received visual feedback from the environment displayed on a screen Force-feedback was given via the haptic device. The recognition accuracy was increased from 86% to over 90% when the training samples were increased from 200 to 500. Accuracy and time for task execution was also evaluated using no assistance and motion intention recognition based assistance. When the recognition algorithm detected that the trajectory was being followed, a hard VF was applied so that the user can be guided along the trajectory. When alignment with a target was detected, attractive forces via the Omni forced the user towards the target center. When a direction of motion perpendicular to the trajectory was detected, repulsive forces around the obstacle center helped the user avoid the obstacle. An increase in motion accuracy and a decrease in execution time were thus observed. Since the difference between the two components of velocity is substantial when a trajectory is followed and when the end-effector is moved away from it, a simpler binary model could have been used as well. The method could only be applied to a structured environment.

1.4.3 Human Action Segmentation and Recognition Using Hidden Markov Models

In this section, we provide a review of the works in which a Hidden Markov Model was used to either segment a teleoperation task into sub-tasks or develop a model via learning of how a human executed a teleoperation task. These works occurred in the 90’s and contribution from two groups can be identified viz. the work by Hannaford et al. [50, 51] at Jet Propulsion Lab in California, USA and that by Yang et al. [49, 52] at Carnegie Mellon University in Pittsburg, USA. Some of them did identify the sub-task from a task using HMM theory but none of them made use of the developed model to increase the performance of the teleoperation task or assist the human user in executing the task in teleoperation.

However, these works were the first ever to use the HMM theory on teleoperation tasks or on human action learning and identification.

Hannaford and Lee [50] were the first to borrow the idea of using HMM theory to segment and identify human initiated tasks from its successful application to speech recognition in the 70’s and 80’s. They used HMM for automatically segmenting a task executed in teleoperation based on sensor data. They wanted the sub-tasks to be tested for performance once they are automatically segmented. According to them, even though the force/torque sensor data is produced from a deterministic process, randomness was introduced due to exploratory and task identification movements by the human operator. They found that the probability densities that encoded the sub-task based on sensor values were overlapping. HMM formalism provided robustness in state estimation and allowed uncertainty in sensor signals. They took inspiration from techniques in speech recognition that used HMM for recognizing the mental states of the speaker. In their case, the force signal was similar to the speech signal in that it was a noise-like signal modulated by actions corresponding to mental states. The mental states in their case were the sub-tasks the operator was trying to achieve. The task that they segmented and analyzed was inserting peg-in-hole and the sub-tasks were moving, inserting, extracting and tapping. The forces were measured by a 6-axis force/torque sensor mounted at the end-effector. Only forces along the x-axis were considered since these produced a distinct force pattern for each sub-task i.e. zero forces (moving), brief positive forces (tapping), predominantly positive forces (inserting) and predominantly negative forces (extracting). Thus the HMM was one-dimensional since only a single parameter was considered as an observation. 4 Gaussian probability density functions (PDFs) were used for the 4 sub-tasks. They used 11-states for the HMM and developed a simple left-right (SLR), an augmented left-right (ALR) and ergodic model. For definition of various HMM parameters including states and observations and for the description of HMM, please refer to Section 2.3. The SLR and ALR models follow the constraints given by Equations (1) and (2).

$$a_{ij} = 0 \qquad j < i \text{ or } j > \ i + 1 \tag{1}$$

$$a_{ij} = 0 \qquad j < i \text{ or } j > \ i + 2 \tag{2}$$

where i and j refer to the states. Constraint (1) means that the state can transition to itself or to the next higher numbered state but to no other state. Constraint (2) means that the state can transition to itself, the next two higher numbered states and to no other. In ergodic model, a state can transition to any state. The following figures demonstrate the SLR and ALR models diagrammatically.

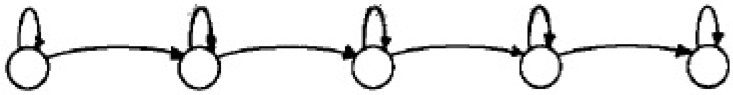


Figure 1: Simple left-right (SLR) model of HMM

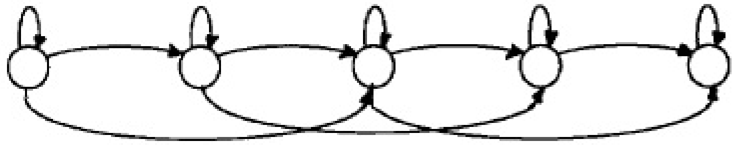


Figure 2:   Augmented left-right (SLR) model of HMM

The 11-states are a series of moves, taps, one insertion and one extraction state. Using 11 states instead of 4 for the 4 sub—tasks has the advantage of better-modeled task structure although the state transition matrix becomes larger and the more computations are involved.

The state transition matrix, A, values and the observation PDF parameters were first initialized using heuristics and then determined by finding a local maximum near the heuristic. This was done by using maximum likelihood estimate or the Baum-Welch (B-W) algorithm. The initial value for matrix A was based on the duration of time spent in a state. It was not clear how the mean and variance for the PDF were determined. This way of initializing A based on duration was not a good representation of state transition, the authors mentioned. For optimization, they ran the B-W using 5 experimental datasets with the heuristic model as the initialization model. The model generated from these 5 datasets was averaged and this averaged model was set as the initialization for the next iteration of another 5 datasets. There were 5 iterations done and a convergence was achieved by the 3rd iteration. An interesting property of Baum-Welch they mentioned was that the values of elements in matrix A that are zero at initialization remain zero throughout the optimization process.

Using their developed model, they generated 4 model datasets in the form of force data. They found these to be qualitatively similar to the peg-in-hole task. They ran the Viterbi algorithm over these model datasets and 4 experimental datasets obtained from actual task executions. The Viterbi algorithm was able to segment the task successfully in both cases. Task segmentation on ergodic model was inaccurate. The authors said that the sequential structure built from task knowledge was lost in this model and was essential for sub-goal tracking. Viterbi on complete datasets resulted in 100% correct segmentation for SLR and ALR models whereas the numbers were 0% for SLR and 67% for ALR for partial datasets. Partial datasets were created by removing one of the taps which were purposefully inserted so that the data could be segmented manually.

They introduced Normalized Probability Value (NPV) as the average total probability (NPV-alpha) and as the average maximum probability (NPV-delta) over time. They were computed from the alpha (summation) and delta (maximum probability state argument) measures of probability. Logarithm values of NPV were used. An interesting observation was that the values of both NPVs converged with time and they were found to be almost equal throughout the time series. The latter meant that the probability of the highest likely state is substantially higher than the next highest likely state. Transient values of NPVs were observed at the points of state change. Examples of application of this method include fault detection, sensor-fusion based task segmentation, time-delay teleoperation failure prevention, and telerobotic control.

Since the experimental datasets used for Viterbi evaluation were chosen to be qualitatively similar to the model datasets, the robustness of their model cannot be confirmed. Moreover, they only used one-dimensional HMM i.e. only the x-axis force data as observations. All the testing was done offline after the data from teleoperation was collected and there was no online segmentation or segmentation as the task was being performed. Intention of the user was not determined as the user executed the task and task segmentation was not used for any performance improvement.

The same authors extended their work [51] to include multi-dimensional HMM. In this, the observation data from more than one sensor was used. They also modeled and segmented a task in which a sub-task could branch into multiple sub-tasks, using Viterbi algorithm and HMM. The development of the HMM was similar to that in their previous work including heuristic initialization and B-W usage. They used an SLR model. Each sensor data was modeled as a Gaussian PDF and each distribution was independent of the other. The joint Gaussian distribution had a diagonal covariance matrix. They used x-axis force, z-axis translation and roll torque data as the observations from sensors.

As part of their results, they compared one-dimensional HMM and multi-dimensional HMM for sequence correctness and path correctness. Three multi-dimensional HMMs were compared, one employing x-axis force and z-axis translation, one with x-axis force and roll torque and the third with all three sensor data. Sequence correctness is the ability of the segmentation algorithm to identify each state correctly as the operator is executing it. Path correctness is the ability to detect the right sub-task at the branch. The results of segmentation by Viterbi algorithm were the best for multi-dimensional HMM employing all three sensors. 100% of tests had both path and sequence correctness in one type of task (tested with 6 datasets) and 80% on the second task (tested with 10 datasets).

They mentioned that the Gaussian PDF for observation probability distribution is not a good model for modalities such as z-axis position. That is because its value changes smoothly as the arm moves from one point to the next. As a result, average parameter of Gaussian distribution does not represent the observation well or will give a higher probability value later in the sub-task. On the other hand, an average represents a fast changing modality, such as force, better. They also mentioned an interesting technique to reduce the influence of a certain sensor value in determining a state. They said that if the variance was increased to a large value for a state-observation relationship, it would result in small probability values for that sensor-state combination thus de-weighing the selected sensor signal for that state. They added that this technique might also be useful in cases where it is difficult for a sensor to discriminate between two states. Inducing a constraint in the observation probability density would make it easier to identify one state out of the two.

Since, the sensors gave a clear indication of the sub-task or state that the user was in, a deterministic model could have very well segmented the task. No online segmentation was done, and the segmentation was not used to improve the task performance.

Yang et al. [52] used HMM theory to model a human skill for a teleoperation task. The task was to move the end-effector vertically down, unscrew a threaded component and then move the end-effector vertically up. Thus, the motion was predominantly in the z-axis direction. They created three learning models or three HMMs for three different representations of the same task. One was based on 7 joint angles' positions through the trajectory as the observations, one on Cartesian z-axis position and one on Cartesian z-axis velocity. The observations were multi-dimensional for the joint angles based HMM but the dimensions were assumed to be independent of each other. For the other two, the observations were one dimensional. Feature vectors were generated from the observations using Short Time Fourier Transformation (STFT) and then discretized into symbols using Vector Quantization (VQ) technique. The observation probability matrix was of size 256X5, where each column represented the distribution for a state. For the multi-dimensional model, there were as many such matrices as the number of dimensions. They used a 5-state Bakis model (same as ALR model) HMM and mentioned that the 5 states represented the sub-tasks. They did not mention the sub-tasks the states represented or why they used Bakis model instead of the SLR or Ergodic model.

They used a combination of B-W and forward-backward algorithm to determine a maximum likelihood estimate of the model. The model was recursively built from 100 operator trials of the task and 15 iterations per trial. Matrix A was initialized such that non-zero probabilities were uniformly distributed and matrix B had all its probabilities uniformly distributed. Initialization of π was not mentioned. The forward scores or output probability seemed to converge after six iterations. The converged probability increased and decreased with increasing trials in a zigzag manner but the general trend was that the probability was increasing as the number of trials increased. Thus the model parameters were constantly modified with each trial so that the model tended towards a more likely representation of the human skills for the particular teleoperation task.

It is not clear how they would use the learned skill. Segmentation of the teleoperation task to the 5 states will not give any information about the task. Thus, the segmentation cannot be used for anything beneficial, such as task analysis or performance improvement. They did briefly mention skill transfer to a robot using Bayes' rule but did not produce any experimental results. They mentioned an interesting point related to using the logarithm of the values in order to avoid underflow. They claimed that the precision of the probability values increase if the base of the logarithm is selected closer to 1. Also, the Cartesian position profile for their task was approximately the same for all trials but the joint angle trajectory profile was different for many joints. This demonstrated that the inverse kinematics computed different solutions each time and it could be attributed to the redundancy of their arm.

In [49], Yang et al. presented their previous work just described but in addition implemented gesture recognition using HMM. They mentioned that human mental states are immeasurable similar to human actions, and that human actions via teleoperation are stochastic. Thus, this scheme fitted very well to an HMM. Although underlying stochastic process is immeasurable, it can be measured indirectly through another stochastic process. They represented gestures as a trajectory of points in a 2D plane and the gestures were numbers from 1 to 9. The points were time sampled to a common time series using interpolation because the X-Windows interface had irregular time intervals. Implementation of learning algorithm was exactly the same as in their earlier work just described. The model for gesture recognition was trained on individual, connected characters (those with no inter-character space) and series of characters. Either Viterbi algorithm or the forward-backward algorithm was used to recognize. They claimed that accurate gesture recognition was obtained. Accuracy of 99.78% was obtained i.e. 449 out 450 gestures were recognized correctly when the training size was 100 samples. Accuracy was 91.56% for training a sample size of 10.

1.4.4 Grasp Intention Recognition Using Hidden Markov Models

There has been a considerable amount of research for determining grasping postures and gestures based on human motion intention recognition. These researches have focused on learning multi-DOF human hand motion while grasping various objects using standard grasping postures. A data glove was worn to record the human hand motion. Once learned, human hand motion intention was determined by running test grasps through the model and the accuracy of the recognition was measured using the output grasp produced in simulation. The main difference of our work from these works is that they are interested in determining hand posture for grasping whereas we are interested in determining the end-effector configuration for grasping. All of these research works are in the Programming by Demonstration (PbD) realm where human movement demonstrates a grasp to the robot, which learns and reproduces. Our implementation is in the teleoperation realm where the replication of motion is almost instant. None of these works were concerned with how soon the intention was determined. They were also not concerned about the change (increase) in accuracy along the trajectory as the desired posture was approached. For assisting the user to grasp an object, rate of intention recognition needs to be considered. We are interested in how soon is the system able to determine the intention so that the user can be assisted towards the desired grasp pose. These implementations do not provide any assistance to the user. In fact assistance is difficult to achieve unless the gripper for replication is kinematically similar to the one using which the model was trained. A kinematic mapping between the two hands is needed but this is not proposed or tested in any of these implementations. None demonstrate any replication of grasping posture on a robotic hand or arm. Our goal, on the other hand, is to provide assistance based on the intended grasping pose the user is interested in. Moreover, these works dealt with a multi-DOF hand which is a complicated system. Their focus is on the grasping posture of the hand and not on the objects. We, on the other hand, deal with grasping pose and not the posture making our system simpler as far as the volume of data is concerned. Also, out methodology is object centric and the grasping pose comes into play when the object class is determined. Thus, the comparison of the two methods has little relevance. Even though the two methods are for determining grasp intention, our method is focused on quicker determination of the end-effector configuration for assistance in teleoperation whereas those in the literature are focused on determining finger and palm postures for replicating grasps by observing human grasps. An account of such works from the literature is given next.

Kragic et al. [53] used HMM to determine human hand grasping postures for grasping objects. The HMM was developed by recording human fingertip pose data as the hand was grasping objects with hand postures based on Cutkosky's taxonomy. The data was recorded by using Flock of Birds sensor mounted on three fingertips and one mounted on the palm for reference. An HMM was developed for each posture and each grasping task had to be conducted in a fixed sequence viz. open hand and prepare, approach and, close hand. Five states were used for each HMM, Baum-Welch was used for training and output probability was used for determining the grasping posture. Another method created a model by recording the end-effector pose and velocity for each grasp posture and made the trajectories time, space and scale invariant. This model was not HMM based. They found out that the HMM method performed better grasp recognition than the other method and a hybrid method was the best. They obtained a grasp recognition accuracy rate of about 100% when the same user trained and tested and that of about 65% when a user trained and a different one tested for the 10 grasp posture types. Grasp intention recognition rate and accuracy were also determined for five of the grasp postures and for the first part of the grasp sequence. Grasp recognition accuracy of 95% was achieved at 60% grasp completion for a single user case and 90% to 95% was achieved at 80% grasp completion for a multiple user case. Grasp recognition accuracy is the fraction of grasps correctly determined by the algorithm. Even though their states have a physical meaning i.e. states are grasp postures, the method of selecting the number of states per HMM is not convincing. They selected five states per HMM since they found fifty clusters of observations and there were ten postures. The methodology of grasp recognition was not tested on a robot and no explanation of how it can be implemented was given. In order to implement on a robotic hand the DOF of the human and the robotic hand need to be mapped and the accuracy of the method will depend on that. Accuracy was found to depend on the mounting locations of the sensors and the variations in grasping the same object by the same user. Sensor errors of 1 cm. to 2 cm. were reported. Though tests on part of the grasping sequence were done to determine the rate of intention recognition, the focus of this work is on grasp recognition and not the speed of recognition.

Ferguson and Dunlop [54] used EMG signals generated from the forearm and hand movements to identify grasp by training a statistical model. They used statistical clustering based on Mahalanobis distance to identify grasps postures from electromyography (EMG) signals generated from the forearm. The signals were measured from the electrodes mounted on the forearm. They used Vuskovic's grasp taxonomy. Noise errors and errors due to mounting variations of the EMG electrodes on the forearm were resolved. EMG signals in the frequency domain were used as feature data due to high volume of data. Fast Fourier transform (FFT) was used for feature extraction. They reported a success rate of 75 to 80%. However, the authors did not mention any specific test plans or methods on which they based their experiments. The authors also mentioned that that the method was replicated on a robotic hand. But, the type of robotic hand used for replicating the method was not mentioned.

Bernardin [55] used an HMM based method to determine 12 grasp postures from Kakamura's grasp taxonomy. The feature vector for the HMM comprised of thirty values viz. 16 joint angles from the cyber glove worn on the hand, thirteen pressure values from tactile sensors mounted on the glove and the highest pressure value from the tactile sensor. The tactile sensors gave the contact points for grasping and their values were passed through a low pass filter to reduce the noise. There was an HMM for each grasping posture with additional ones for garbage movements, release operation and silence. Each HMM had 9 states. Baum-Welch algorithm was used for training and Viterbi algorithm was used for intention recognition. Training and testing was done by asking the users to execute a sequence of grasps, each grasp having a different grasping posture. They called this a composite HMM. Classification accuracy was measured by testing grasp sequences on 4 subjects on a variety of objects. They achieved an accuracy of about 85% for the case in which the same user's data was used to train and test the model and about 92% for the case in which a user trained model was tested on another user. A manual post-processing step was used to improve the classification accuracy. They listed a number of factors that they thought might be responsible for poor accuracy viz. lack of calibration for all users, hand size, manner of putting the glove, and inconsistencies in motion such as hesitations and so on. Again for this method to be used for grasping using a robotic hand, kinematic mapping is necessary. No results on a robotic hand were demonstrated.

Kang and Ikeuchi [56] used ink marks on the user's hands and their impressions on objects from grasping them for geometrically measuring grasp parameters. Static range and intensity images were used to measure parameters and classify grasps. They obtained good recognition rates. No demonstration on a robotic hand for replicating the grasp was reported.

Palm and Iliev [57] compared three methods for grasp recognition viz. time clustering based model, fuzzy based, and a hybrid HMM and fuzzy based. Feature data for each of the methods was in the form of finger joint angle trajectories captured using a data glove. The hand model consisted of 5 fingers with 3 joints per finger. Cutkosy's and Iberall's taxonomy was used for grasp postures. Their time cluster based model was not time and space invariant thereby requiring a fixed environment set-up w.r.t. object pose, hand initial and final pose, and a number of via-points in the trajectory. Their fuzzy model only works for a particular sequence of grasps and needs to be retrained for a different sequence. For their HMM, they have discretized the continuous observations by picking up intermediate poses from the trajectory lending the HMM inaccurate at points along the trajectory not part of discretization. Simulation results in a structured set-up show that time clustering performed better than other methods.

Ju et al. [58] have done work very similar to Palm and Iliev. Instead of fuzzy clustering they have used a Gaussian Mixture Model (GMM). Finger joint angle trajectory, captured using a data glove with 19 sensor values, is used as input features. Their model is again not time invariant. As per results in simulation, HMM performs the poorest and according to the authors, this is due to lack of sufficient training data.

Chapter 2: Arm Kinematics, Control and Teleoperation

In this chapter, a brief introduction to the arm kinematics and control is given. For a detailed description on the theory of forward kinematics and transformation matrix algebra, the reader is requested to refer to the text by John Craig [59]. For a detailed description on the remote arm design, construction, inverse kinematics and optimization, the reader is requested to refer to [60]. We will follow the convention from the text by John Craig to describe any kinematic equations and mathematical quantities. Next we mention the improvements in joint limit avoidance implementation. We also briefly describe the master device used in our experiments and present the ensuing telemanipulation control. The last section includes the theory on scaled teleoperation concept that we have implemented for assistance in translation and rotation.

2.1 Remote Arm Frames and Forward Kinematics

Kinematics of an object is the mathematical description of its position and the higher derivatives of position, like velocity and acceleration, without regards to the forces causing the motion. The location of an object in the Euclidean space is described by six independent parameters – three for position and three for orientation. These are represented in the form of Cartesian x-y-z frames and are collectively known as pose (position and orientation). Each parameter also represents a degree of freedom (DOF). The position is generally represented by x, y and z symbols whereas the orientation by (α,β,γ) or roll, pitch and yaw. Robotic arms are generally in the form of open-ended kinematic chains and comprise of rotary or prismatic joints. These joints are connected to each other via rigid links. A Cartesian frame is assigned to each joint and its origin is located at the center of the joint. A certain convention is followed for assigning the frames [59], but a frame can be assigned in more than one ways. The relation between two consecutive joints is described by D-H parameters [61], which are a set of standard linear and angular distance parameters. Depending on the assigned frame, there can be more than one D-H parameter sets. The relation between two consecutive joints can be encoded is in the form of a transformation matrix, which can be generated from D-H parameters [59]. In this way, a transformation matrix describes the pose of a joint frame with respect to the previous. In the case of a robotic arm with all rotary joints, the only variables are the angular distance parameters from the D-H set. The last frame in the kinematic chain is known as the end-effector and the angular distance parameter is known as joint angle. Forward kinematics of an arm is the process of determining the pose of the end-effector from the joint angles of all the joints. Forward kinematics is carried out by successive multiplication of the transformation matrices, starting from the base joint of the arm and working towards the last joint in the kinematic chain.

Forward kinematics can be also used to determine the pose of all the joints and links of the arm but generally the description of the pose of the end-effector is sufficient. This is because the end-effector is the most important joint for the execution of a task. A tool or a hand is mounted on the end-effector and these are the main elements used to execute any task. The task can be holding a brush for brushing the teeth or grasping an object with the hand. For moving the tool along a trajectory or grasping with the hand, it is important to know where the end-effector is in the workspace. Knowing the pose of the end-effector also helps to avoid collisions. Determining the pose of the remaining joints and links of the arm can be useful for obstacle avoidance. Forward kinematics also helps in determining the pose of the objects in the environment by combining information from a depth sensor like a laser range finder or an RGBD vision system. This is because the sensor can determine the pose of the object with respect to itself, and since the pose of the end-effector is already known with respect to the base, the pose of the object can be computed with respect to the base from transformation matrix equations. When the arm is in motion, forward kinematic can be carried out by querying the encoder angles at each time instant as the arm is moving. Forward kinematics is also used to generate a more accurate trajectory. This is because, often while traversing a trajectory, the arm does not move to the commanded points at each instant. Arm modeling errors and motor errors are responsible for this inaccuracy. Thus, the pose from forward kinematics determined from encoder feedback can be used to find out the true pose of the arm and then move to the next commanded point. The result is a more accurate tracking of the generated trajectory [34].The D-H parameters of the7 DOF arm we have used in our test-bed are given in Table 1. Figure 3 shows the frames assigned to each joint and the corresponding D-H parameters. All the joints in the robot are revolute joints. It was designed at University of South Florida's Center for Assistive, Rehabilitation and Robotics Technologies [62]. Complete details of the arm are included in [60].

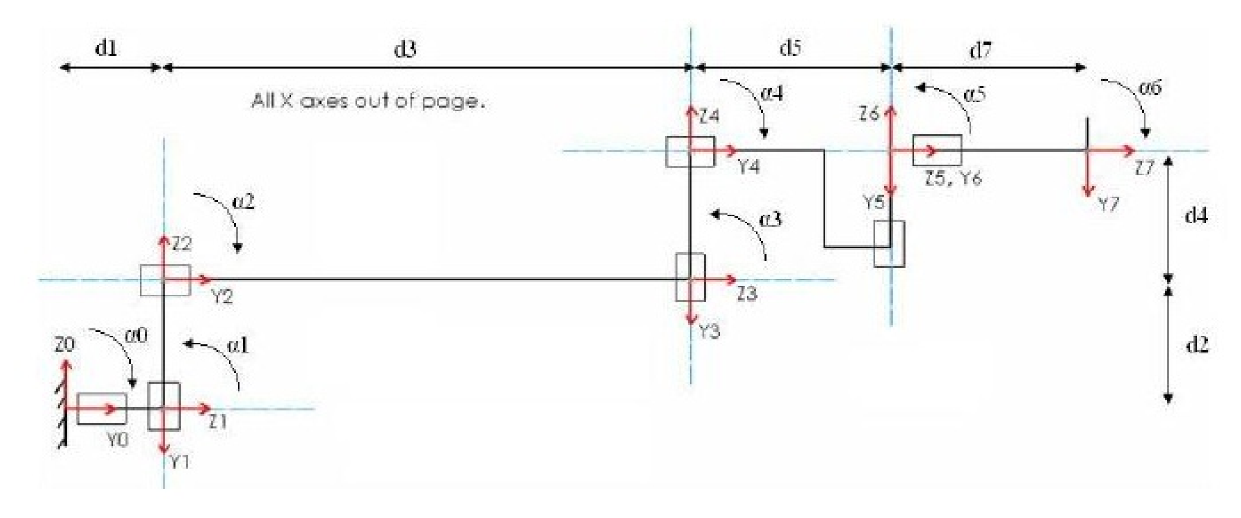


Figure 3: Cartesian frame assignments and DH parameters for the 7 DOF robotic arm

Table 1: DH parameters of the 7 DOF robotic arm

Link i	$\propto_{i-1}$	a_{i-1}	d_i (mm)	θ_i
1	-90°	0	102	θ_1
2	90°	0	133	θ_2
3	-90°	0	502	θ_3
4	90°	0	130	θ_4
5	-90°	0	387.7	θ_5
6	90°	0	-11.8	θ_6
7	-90°	0	361	θ_7

Link d_7 actually measures to 161 mm. However, there is an electronic gripper attached to this link. The gripper's frame has the same orientation as joint 7 and it does not change as the arm is being teleoperated. When a user is teleoperates, she imagines the end-effector reference as the center of the gripper. Therefore, the distance between the end point of joint 7 and gripper center is added to d_7 to yield 361 as the total length of d_7. Thus, the inverse kinematics gives solutions compatible with user's perception of the arm. Moreover d_5 and d_6 have been modified to account for the new 90 deg. gear-box that was installed on joint 6. With the old gearbox (currently on joint 4) they were 375 and 0 respectively.

2.2 Inverse Kinematics

As mentioned earlier, inverse kinematics solves the problem of determining the joint angles given the end-effector pose. Thus, it is the inverse of the forward kinematics problem. Given the start and the end pose of a trajectory, inverse kinematics helps to determine the joint angles that the motors at each joint need to be commanded, to complete the trajectory. Trajectories are specified in terms of end-effector poses rather than joint angles because it is intuitive for a human to specify a trajectory in Cartesian space rather than joint space. However, drive motors need joint angles' information to move the joints. Hence, inverse kinematics is very important for trajectory generation and general motion control of a robotic arm.

Inverse kinematics can be solved in a number of ways viz. closed form solutions, geometric solutions, numerical methods etc. Closed-form solutions are specific to specific arms and can only be used for that particular arm design. This is because their solution is based on the kinematics of the arm i.e. frame assignment and DH parameters and every arm has a different kinematic structure. A more general method to solve inverse kinematics is the Resolved Rate method [63], which we use in our implementation. This method employs the inverse of the Jacobian for solving the inverse kinematics. The inputs to the algorithm are the differential end-effector pose between two trajectory points and the Jacobian. The outputs are the joint angle values needed by motors to move the end-effector to the next point in the trajectory. Every arm model has a different Jacobian but once the Jacobian is obtained, inverse kinematics using the resolved rate method is the same for any arm. The inverse of the Jacobian does not exist for redundant arms since inverse of only square matrices exists and, the Jacobian of redundant arms is not a square matrix. Pseudo-inverse of the Jacobian is used in these cases. Because the solution using the inverse and the pseudo-inverse of the Jacobian fails at certain points, approximate solutions are used for trajectories near those points. The points are called singular points and at these points the determinant of the Jacobian evaluates to zero. As a result the inverse of the Jacobian is infinity and the joint rates assume very high values. The arm motion becomes unstable, uncontrollable and dangerous. To avoid singularities singularity-robust inverse [67] of the Jacobian are used for solving inverse kinematics. The joint position error is increased at the expense of reducing joint velocities at and near singularities. Thus if singularity is approached while following a trajectory, the arm will minimally deviate from the commanded trajectory instead of attaining a singular configuration. It gets back on to the trajectory once the point of singularity has been passed.

Both pseudo-inverse and SR-inverse solve for redundancy by optimizing the motion based on least norm criterion i.e. minimization of the end-effector velocity errors. Even after least-norm optimization, a null space may still exist in case of redundant arms. This redundancy can be resolved by using the weighted least norm solution of the SR-inverse of the Jacobian. This is done by adding a weight term to the SR-inverse of the Jacobian solution. Using this weight term, motion preference can be given to some joints over others to achieve the required motion based on a criterion. One such criterion is joint limit avoidance. In this case, weights are added to the joints depending on whether they are approaching their joint limits. This prevents the joints from physically hitting their limits and causing damage due to collision.

2.2.1 Improvements in Joint Limit Avoidance

The block diagram for the information flow for SR-inverse and joint limit avoidance based weight least norm solution for inverse kinematics, as it was implemented before the start of this project, is shown in the Figure 4.

A problem with this implementation is that the joint will get locked once it crosses joint limit. We now explain how this happens. Imagine that the joint limit is crossed as the arm is moving. This is not the actual joint limit but a software limit that is very close to the actual limit i.e. a little less than actual $q_{i,\max}$ and a little more than the actual $q_{i,\min}$. Now as the arm moves to the next via point, the joint weights are calculated based on certain heuristics [67]. The weight on the joint that just crossed the limit will be high and so it will not move. Because the joint is out of its limit, and because it did not move, the weight on that joint will again evaluate to a high value resulting in it not moving. This goes on repeating for every iteration and the joint is locked. To counter this condition, we modify the inverse kinematics computation as depicted in the Figure 5.

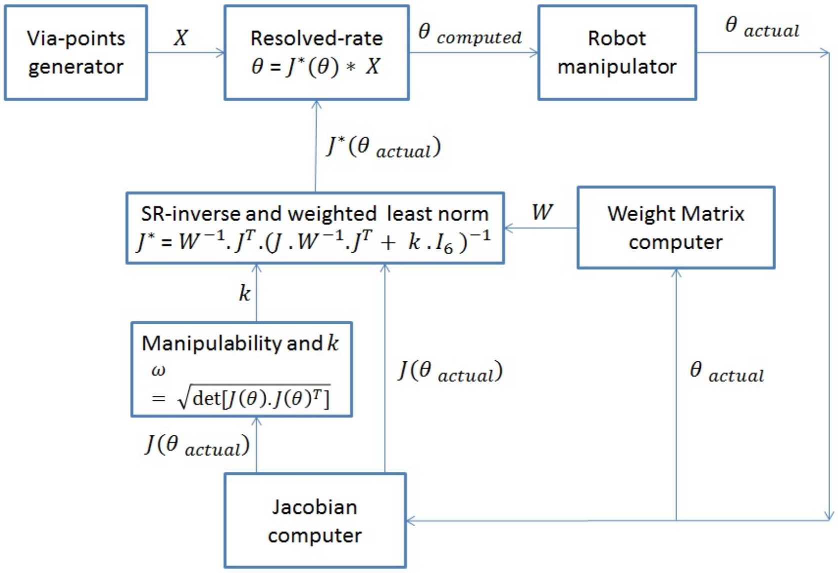
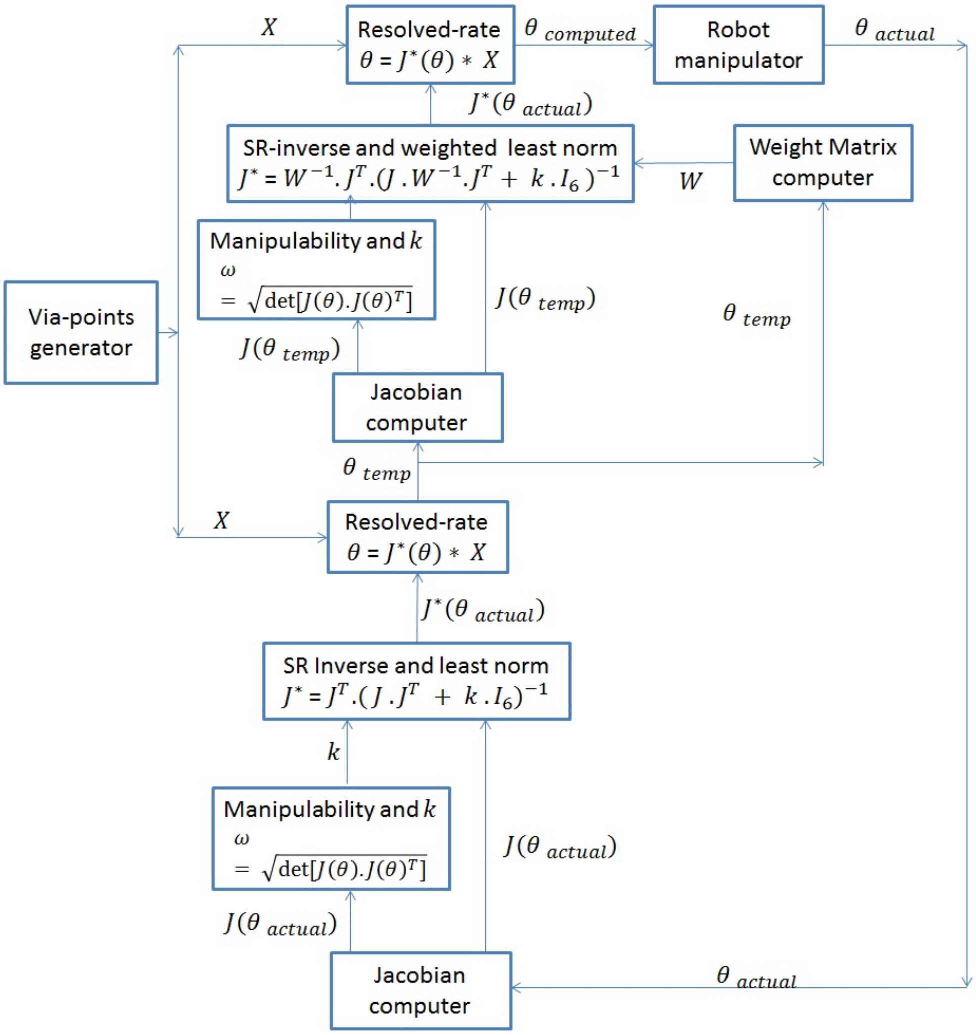


Figure 4: Block diagram showing information flow for inverse kinematics on the 7 DOF arm using SR-inverse of the Jacobian and joint-limit avoidance based weighted least norm

Here, we compute the joint angle rates without the weighted least norm solution. We use just the least norm or set the weight matrix to unity. However, we do not use these joint rates to move the arm. In fact, based on these we calculate the weights and then compute joint rates using weighted least norm. It is these joint rates that are used to move the arm. We must note that SR inverse is used in both the steps. Calculating joint rates without weights in the first step lets the algorithm know the direction that each joint is moving in, in terms of the joint limit. Using these as the commanded angles (when we actually do not command using these), and then assigning weights depending on the heuristics, puts weights on the joints that are still trying to exceed the limit. The resulting joint rates will move the arm without allowing any of the joints to exceed their limits and without locking any of the joints, if they have already crossed the limit.

A problem was observed when two of the joints crossed their limits and the arm trajectory (in teleoperation) wanted to move in a direction that would have caused these joints' limits exceeding further. Due to the implementation of the above method, it was expected that the arm will move in an erroneous direction but continue to move without exceeding any of these joint limits further. Or that the arm will not move at all. However, one of these joints' limits continued to be exceeded progressively. The joints in question are joint 2 and 6 and 6 continued to move.



X
Resolved-rate $\theta = J^*(\theta) * X$
$\theta_{computed}$
Robot manipulator
θ_{actual}
$J^*(\theta_{actual})$
SR-inverse and weighted least norm $J^* = W^{-1}.J^T.(J.W^{-1}.J^T + k.I_6)^{-1}$
W
Weight Matrix computer
Manipulability and k $\omega = \sqrt{\det[J(\theta).J(\theta)^T]}$
Via-points generator
$J(\theta_{temp})$
$J(\theta_{temp})$
θ_{temp}
Jacobian computer
θ_{temp}
X
Resolved-rate $\theta = J^*(\theta) * X$
$J^*(\theta_{actual})$
SR Inverse and least norm $J^* = J^T.(J.J^T + k.I_6)^{-1}$
k
Manipulability and k $\omega = \sqrt{\det[J(\theta).J(\theta)^T]}$
$J(\theta_{actual})$
$J(\theta_{actual})$
Jacobian computer
θ_{actual}

Figure 5: Block diagram of inverse kinematics of the 7 DOF arm using SR-inverse of the Jacobian and the modified joint-limit avoidance based weighted least norm, the modified version prevents the joint from locking once it crosses the limit.

2.3 Master Device Control and Telemanipulation

The master device that we used for telemanipulation is a 6 DOF manipulator called Phantom Omni from Sensable Technologies [65]. This device is basically used to teleoperate the remote manipulator. This is done by manipulating the Omni with hands of the user. Thus translating Omni in its Cartesian space along x, y and z directions and orienting it about the x, y and z axes results in the movement of the remote arm in the corresponding directions in its own Cartesian space. The details of this telemanipulation motion control are explained in the next sub-section. In order to transfer the master device motion to the remote arm, the motion of the master device needs to be read and processed. Thus, we need to determine the kinematics of the master device in order to obtain, say, its end -effector pose. However, in the case of the master device we do not develop the kinematics, as we did in the case of the remote manipulator. This is because we use the readily available functions called Application Programming Interface (API) that give results such as end-effector poses, when queried. APIs come with a library called OpenHaptics library that comes along with the device. The APIs are compatible with C/C++ programming environments. For more details on the APIs and the device, please refer [66].

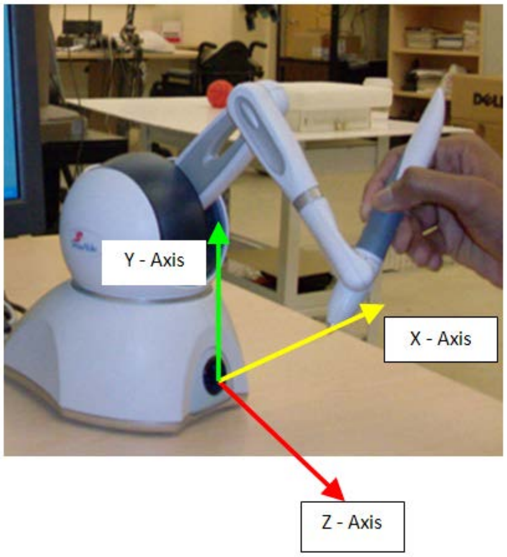


Figure 6: Phantom Omni manipulator used as a master device and the principal axes of the right-handed Cartesian base frame

This device has a very small footprint and very low inertia which makes it very easy to manipulate with the hands. This device requires no finger movement, only sufficient finger strength to hold its stylus (refer Figure 6). For manipulating it, the user mostly needs wrist movement, some elbow movement and rare shoulder movement. So, the user must have good functionality and strength, at least in one of the upper limbs. We must also mention that we use this device as an impedance type of device i.e. the user input force through their hands and arms and the device outputs pose (through values retrieved from a program). It can also be used as an admittance type of device i.e. the user inputs pose values for the end-effector to move (through the program) and the device outputs force (realized through its movement). The device also has two buttons on its stylus that can be used for preprogramming any functions like opening and closing a gripper or activating / deactivating control modes of motion. The master device also has haptic feedback capabilities for translation i.e. it provides force feedback in translation by using the appropriate APIs. This is because the motors in its first three joints provide resistance torque depending on the task and control program. However, we do not use any of the haptic feedback capabilities and only use kinematic capabilities. Even the usage of kinematics from the Omni is limited to forward kinematics. The Omni also has its own real-time thread that runs real-time control loops even on non-real-time operating systems like Windows. The loop frequency can be set as 250 Hz, 500Hz or 1000 Hz in the program. For haptic feedback, a minimum of 1000 Hz is needed. We have selected 500 Hz since we do not use haptic feedback.

2.3.1 Cartesian Mapping and Communication for Teleoperation

We have mentioned that the Cartesian motion of the Omni in its space is transferred to the remote arm space to teleoperate the remote arm. In this section we look at the details of how this is implemented.

One way to teleoperate is by using joint based mapping. The remote arm is a 7 DOF manipulator whereas Omni is a 6 DOF manipulator. Joint based mapping, in which one joint on the remote arm is moved corresponding to the joint on the master device, is not possible unless one of the joints on the remote arm is not used at all. Moreover, due to the different link sizes of the master and remote manipulators, the motion of the joints on the remote arm will have to be scaled down. This is especially true for the shoulder joints since their link lengths are longer than those of the elbow or the wrist. This will need a lot of trial and error and may result in non-intuitive motion at the remote arm. Also, different scaling may be needed for gross motion execution compared to fine motion. On the contrary the motion will be very intuitive if the end-effector of the remote arm is moved by the same amounts and in the same direction as the master. This is because for human users imagining and representing the motion in Cartesian coordinates is more natural and intuitive than in the joint space. Thus, we have implemented a Cartesian mapping of motion control for telemanipulation. This method is also scale invariant since the differential joint angles of the remote arm motion will be computed based on its end-effector movements i.e. no scaling is required to be implemented.

In order to move the remote arm using Cartesian based mapping, the differential end-effector poses from the master device are retrieved as it is manipulated by the user. These differentials are mapped to the remote arm's frame of motion, which could be base frame, end-effector frame or any other frame. Inverse kinematics on the remote arm then yields the joint angles that need to be commanded to the joint motors so that it moves corresponding to the master device. This concept is explained in the Figure 7.

The mathematical implementation is now presented. Referring to Section 2.4, the translational and rotational velocities of the master end-effector are determined using Equations (8) to (13). These are clubbed into the Cartesian velocity vector as in Equation (15) and mapped to the remote arm frame as in Equation (16). The velocity vector in the remote arm space V_s thus computed is the same vector that is the input to the SR-inverse of the Jacobian. Inverse kinematics using SR-inverse of the Jacobian thus computes the joint angles that need to be commanded to the remote arm motors so that it moves in accordance with the master device. Thus, joint angles for the remote arm are computed at every instant as the master device is being manipulated.

The master and remote arm control threads run on the same machine and on separate threads. A mutex based queue has been implemented to securely transfer data from master to remote arm. It prevents data loss and data corruption. It may not have been necessary to implement a queue if the control loop frequencies on the master and remote arm were the same. However, due to the difference in the control loop frequencies, secure communication is necessary.

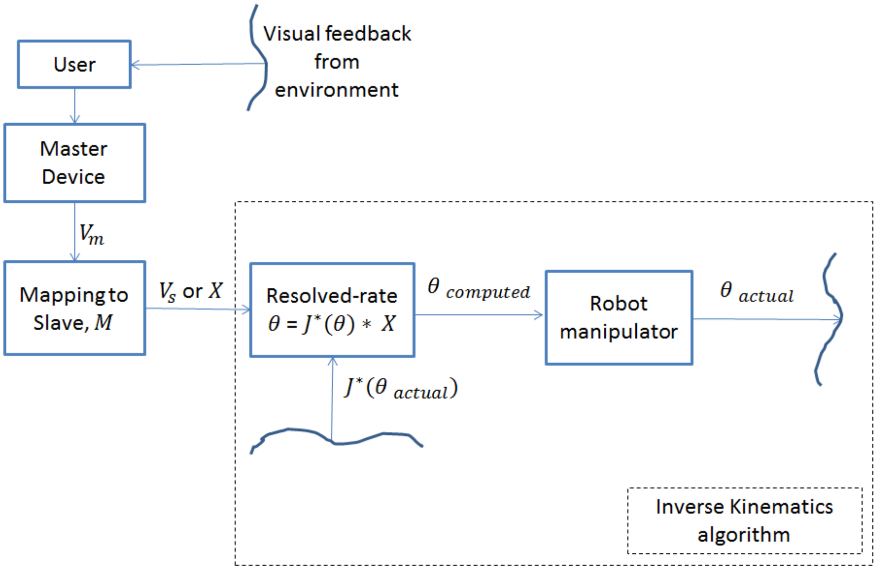


Figure 7: Block diagram showing information flow for Cartesian based mapping for teleoperation and its connection with inverse kinematics and environment

We now give details of the mapping matrix M from Equation (16) that is used to map the Cartesian velocity from the master arm frame to the remote arm frame.

2.3.2 Control Frames for Mapping Motion

Mapping is important because it gives the user an idea as to what motion on the master will produce a corresponding motion at the slave. While teleoperating a remote robot, it is important that the direction of motion input corresponds to the motion direction output that is intuitive to the operator. The base frame control and the end-effector frame control are the two frames that are intuitive for teleoperating a remote arm.

2.3.2.1 Base Frame Control

In base frame control, the motion of the master device is transformed from its base to that of the remote arm. The concept is depicted in two dimensions in the following figure. From the figure we see that when the robot is at an initial configuration and is given a command to move forward, it moves from

Position 1 to Position 2. When it is oriented at a later time and is commanded to move in the same direction, it moves in the same direction. We see that the base frame does not orient with the robot.

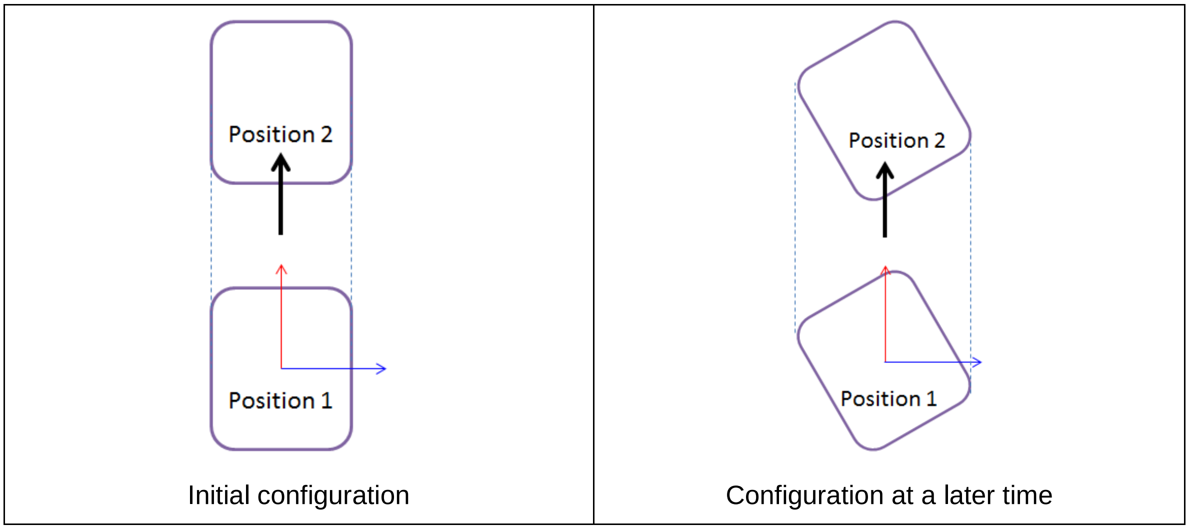


Figure 8: Conceptual figure depicting base frame concept.

The mapping matrix R_m^s in Equation (16) for frame mapping for the arms used in our test-bed is given by the following equation and the frames are pictorially represented in Figure 9,

$$R_m^s = \begin{bmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{3}$$

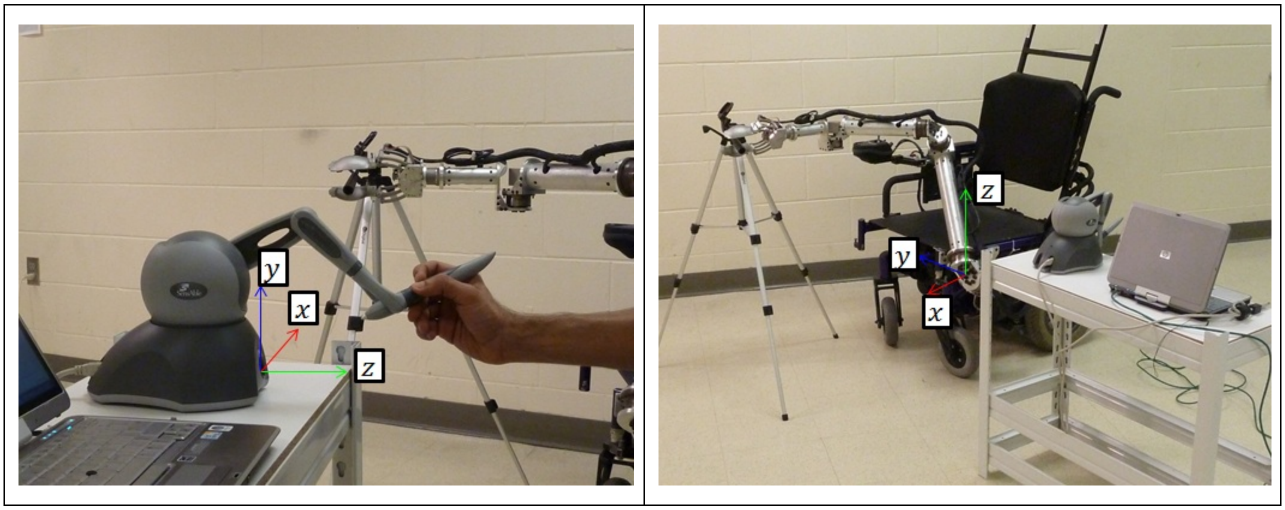


Figure 9: (Left) Omni base frame (Right) 7 DOF arm base frame

2.3.2.2 End-effector Frame Control

At times, it is desirable to control the motion in teleoperation in terms of the end-effector frame of the remote arm while the motion at the master is still in the base frame. End-effector frame is important when the motion at the remote arm needs to be generated with the objects in the environment as reference. The concept is depicted in the figures below in two dimensions. We see that the reference frame orients with the robot.

The Figure 11 shows the master device base frame and the remote arm end-effector frame. The mapping matrix R_m^s for end-effector frame mapping for the arms we have used in our test-bed is given below.

$$R_S^M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (4)$$

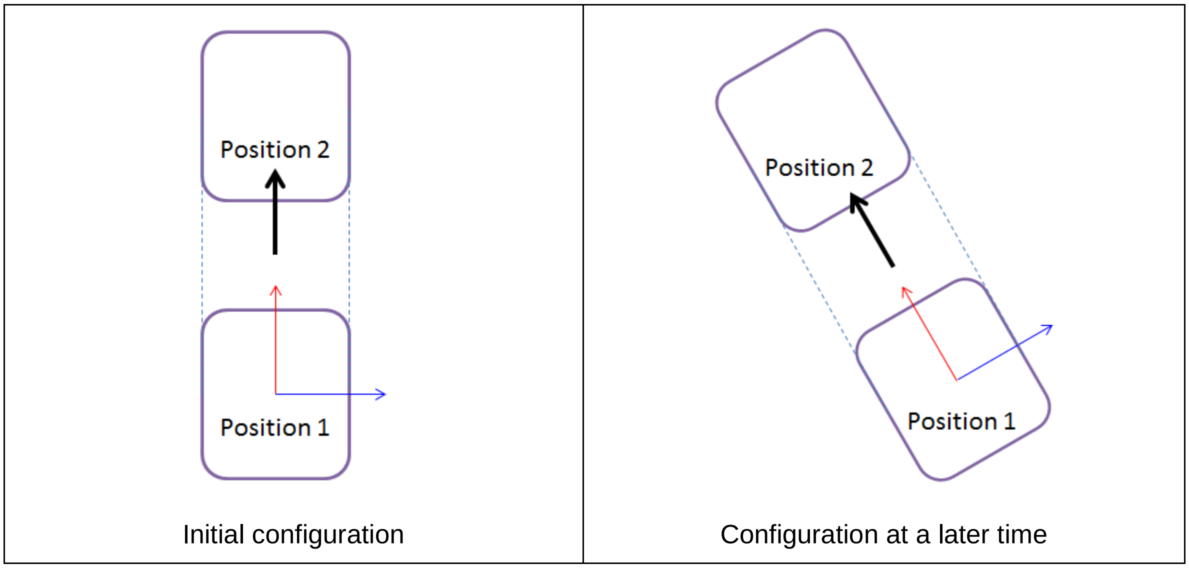


Figure 10: Conceptual figure depicting end-effector frame concept

We must note that not only the Cartesian velocity vectors are transformed, but the Jacobian J also needs to be transformed to generate the required joint angles to move the remote arm in its end-effector frame.

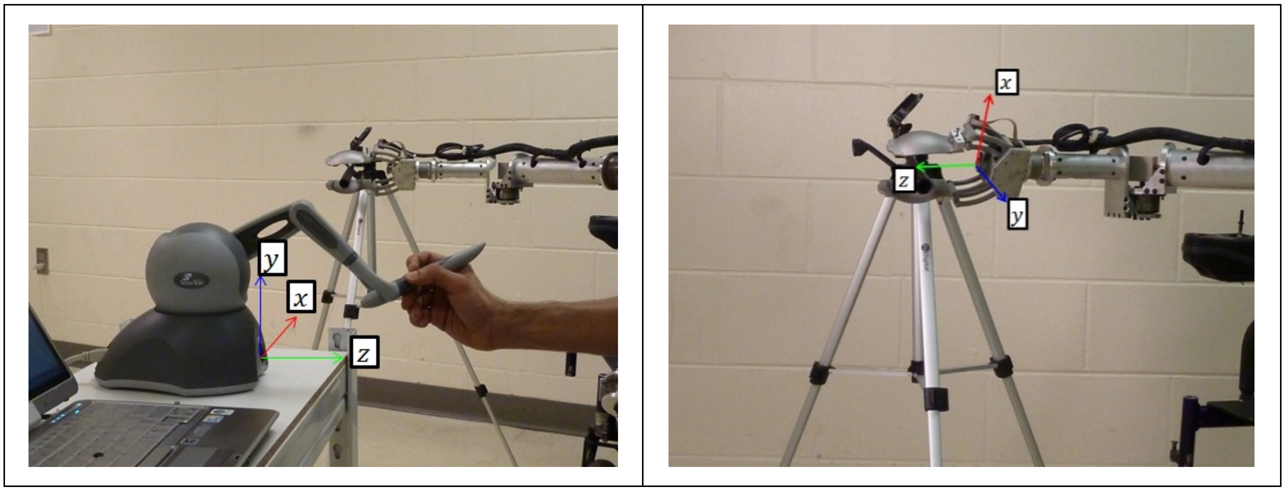


Figure 11: Omni base frame (left) and 7 DOF end-effector frame (right)

2.4 Assistance Using Scaled Teleoperation

Once the intended preshape pose has been determined, the next step is to assist the user to traverse and orient the gripper to align with the desired pose. The assistance is provided using scaled teleoperation [42] in which the components of motion along the desired directions are scaled up whereas those along the direction perpendicular to the desired are scaled down. In this way any deviations from the desired path or trajectory are reduced and movements along the desired direction are amplified. This way, the errors due to deviation from the desired path are reduced considerably and the users observe an appreciable movement along the desired path. This provides an intuitive visual feedback to the user and gives them a cue of the direction along which they should teleoperate to reach the target by making less number of movements. The method helps the users to teleoperate towards the desired pose quicker and with much ease. This will be confirmed from the results that we will generate from the experiments. The motion scaling is carried out along translation as well as rotation directions. The convention for describing manipulator kinematics is used from the text by John Craig and the frames used for the calculations are the Euclidean frames or Cartesian coordinate frames.

Let T_i^O and T_f^O represent the initial and final transformation matrices of the end-effector frame with respect to the manipulator base frame i.e. the transformation matrices of the end-effector as it moves from an initial pose to a final pose. T_i^O and T_f^O are of the order 4X4 and consist of a 3X3 rotation matrix and a 3X1 translation vector, both combined into a 4X4 homogenous transformation matrix.

$$R_i^O = \begin{bmatrix} n_{x_i} & o_{x_i} & a_{x_i} & p_{x_i} \\ n_{y_i} & o_{y_i} & a_{y_i} & p_{y_i} \\ n_{z_i} & o_{z_i} & a_{z_i} & p_{z_i} \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{and } R_f^O = \begin{bmatrix} n_{x_f} & o_{x_f} & a_{x_f} & p_{x_f} \\ n_{y_f} & o_{y_f} & a_{y_f} & p_{y_f} \\ n_{z_f} & o_{z_f} & a_{z_f} & p_{z_f} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Let, v be a 3X1 vector representing linear velocity of the end-effector frame as it moves from initial point to final point and let ω be the angular velocity. Let v_x, v_y, v_z be the components of v and let ω_x, ω_y, ω_z be the components of ω. Thus,

$$v = (v_x, v_y, v_z) \tag{6}$$

$$\omega = (\omega_x, \omega_y, \omega_z) \tag{7}$$

v and ω are computed as follows. v is the Euclidean distance between each of the x, y and z components of the initial and final frames. ω is computed by taking the cross-product of each of the x, y and z principal axis of the initial frame with the corresponding axis of the final frame and summing up the three computed vectors into a single vector. Thus,

$$v_x = p_{x_f} - p_{x_i} \tag{8}$$

$$v_y = p_{y_f} - p_{y_i} \tag{9}$$

$$v_z = p_{z_f} - p_{z_i} \tag{10}$$

$$\omega_x = 0.5 * (o_{y_i} * o_{z_f} - o_{z_i} * o_{y_f} + n_{y_i} * n_{z_f} - n_{z_i} * n_{y_f} + a_{y_i} * a_{z_f} - a_{z_i} * a_{y_f}) \tag{11}$$

$$\omega_y = 0.5 * (o_{z_i} * o_{x_f} - o_{x_i} * o_{z_f} + n_{z_i} * n_{x_f} - n_{x_i} * n_{z_f} + a_{z_i} * a_{x_f} - a_{x_i} * a_{z_f}) \tag{12}$$

$$\omega_z = 0.5 * (o_{x_i} * o_{y_f} - o_{y_i} * o_{x_f} + n_{x_i} * n_{y_f} - n_{y_i} * n_{x_f} + a_{x_i} * a_{y_f} - a_{y_i} * a_{x_f}) \tag{13}$$

Equations (11) to (13) above for the computation of angular velocity ω can also be written in a concise form as,

$$\omega = \frac{1}{2} (n_i \times n_f + o_i \times o_f + a_i \times a_f) \tag{14}$$

Let V be the velocity vector that represents linear and angular velocity as a single unit. Thus,

$$V = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{15}$$

Let V_m be the velocity generated every time the master manipulator is moved from one point to another by the user and let V_s be the corresponding velocity of the slave manipulator which moves as a result of teleoperation. Let v_m, ω_m, v_s and ω_s be the corresponding linear and angular velocities. These vectors are generated at every time instant as the slave arm (remote arm) is being teleoperated with the master device. V_m and V_s are related by a mapping matrix that relates the base frame orientations of the master and slave manipulators. Thus,

$$V_s = M\ V_m \tag{16}$$

where,

$$M = \left[\begin{array}{cc} & \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} \\ \hline R_m^s & R_m^s \\ \hline \begin{matrix}1&0&0\\0&1&0\\0&0&1\end{matrix} & \end{array}\right] \tag{17}$$

R_m^s is the 3X3 rotation matrix that specifies the orientation of the base frame of the slave manipulator with respect to the base frame of the master manipulator. Therefore the mapping matrix M is a 6X6 matrix.

Ordinarily, V_s is computed at every time instant as the master device is being manipulated by the user. Inverse kinematics on the 6X1 velocity vector V_s gives the joint angles that the slave arm needs to be commanded by in order to move by the same amount and in the same direction as the master device. (The details of inverse kinematics and mapping matrix values are covered in the chapter on Hardware implementation). However, in order to provide assistance in scaling V_s needs to be modified so that its components in the desired directions are amplified. Let this modified form of $V_s{}'$. Inverse kinematics on $V_s{}'$ will lead to scaled motion. We now show how $V_s{}'$ is calculated.

2.4.1 Scaled Teleoperation for Translation

We first explain how we scale the motion in translation. We know from intention recognition algorithm the particular preshape pose that the user is interested in. In other words the system knows the location to which the end-effector should be traversing in order to align with the desired preshape pose. Let the unit vector along the linear velocity vector that defines this desired linear trajectory be v_k. Let $v_l$and $v_m$be the unit vectors perpendicular to v_k so that v_k, v_l and v_m are orthonormal i.e. mutually orthogonal unit vectors. In other words, v_k, v_l and v_m form an x, y and z Cartesian triad. We already have v_s which is the linear velocity of the slave due to teleoperation from master device. We must keep in mind that v_s is not a unit vector. Let, t_k, t_l and t_m be vectors generated by projecting v_s on v_k, v_l and v_m and having the direction along v_k, v_l and v_m. Therefore,

$$t_k = (v_s \,.\, v_k)\, v_k \tag{18}$$

$$t_l = (v_s \,.\, v_l)\, v_l \tag{19}$$

$$t_m = (v_s \,.\, v_m)\, v_m \tag{20}$$

Since, v_k is the desired direction that the user intends to move the remote arm along, we scale the motion along v_k and scale down along v_l and v_m. Let $s_u$be the scaling factor for scaling up and $s_d$be the scaling factor for scaling down. We have,

$$s_u \geq 1 \tag{21}$$

$$1 \geq s_d \geq 0 \tag{22}$$

Let, t_k', t_l' and t_m' be the scaled versions of t_k, t_l and t_m, such that,

$$t_k' = s_u\, t_k \tag{23}$$

$$t_l' = s_d\, t_l \tag{24}$$

$$t_m' = s_d\, t_m \tag{25}$$

If, however, t_k is in the direction opposite to that of v_k i.e. the user is deviating away from the desired, then all the three components of projection are scaled down. In this case, we get,

$$t_k{}' \; = s_d \; t_k \tag{26}$$

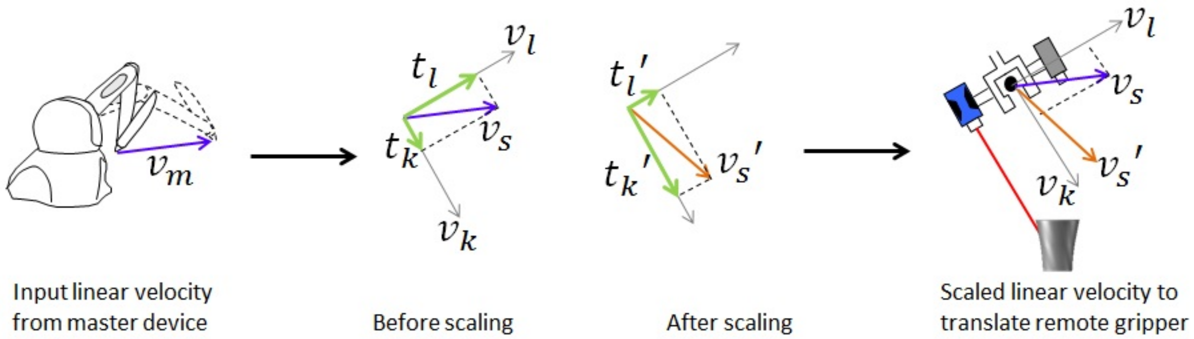
$$t_l{}' = s_d \; t_l \tag{27}$$

$$t_m{}' \; = s_d \; t_m \tag{28}$$

Thus, the modified version of v_s is given by the vector sum of $t_k{}'$, $t_l{}'$ and $t_m{}'$. Let us call it $v_s{}'$. Thus,

$$v_s{}' = t_k{}' + t_l{}' + t_m{}' \tag{29}$$

$v_s{}'$ is thus the scaled form of v_s, It is scaled up in the direction of v_k which is the desired direction for translation of the slave arm. The users teleoperating the arm will not only see a change in the direction of motion but also a change in the magnitude. The closer the user teleoperates the slave arm along the desired direction, the larger is the magnitude of motion. The more the deviation from the desired trajectory, the smaller is the magnitude of motion. In this way the user is assisted along the desired direction of motion. The concept of scaled teleoperation along a linear trajectory is shown in the figure below. For ease of appearance and understanding, the concept is shown in two dimensions only.



Input linear velocity from master device

Before scaling

After scaling

Scaled linear velocity to translate remote gripper

Figure 12: Scaled translation concept

2.4.2 Scaled Teleoperation for Rotation

The scaled rotation concept is similar to the scaled translation concept. This is because even though a rotation in Euclidean space has to be represented by 3X3 matrix of 9 elements, equivalent angle-axis form reduces these elements to just 3. Equations (11) to (13), which give a vectorial representation of the differential rotation matrices, are derived from equivalent angle-axis form [19]. Thus, when the rotation matrix can be represented in the form of a 3X1 angular velocity vector, given by

Equations (11) to (13), the calculations for scaling it become similar to those for translation. All we need to do is to replace the linear velocity vectors by angular velocities. The motion scaling along orientation is explained next.

We know that ω_s is the angular velocity of the slave end-effector generated after mapping that of the master device end-effector ω_m when the master moves from one point to another under user control. ω_s and ω_m are generated at every time instant as the user is teleoperating and are related by the mapping matrix in a manner similar to the linear velocity vectors given by Equation (16). Based on the intention recognition, the desired direction of orientation, that the user should be teleoperating in order to align the slave arm with the desired preshape pose, is already known. Let ω_k denote a unit vector in that direction and it can be calculated using Equations (11) to (13). Again, ω_k is a 3X1 vector in Euclidean space and can be treated like any other vector in the space. Let ω_l and ω_m be unit vectors perpendicular to ω_k such that the three vectors are orthonormal and form an x, y and z Cartesian triad.

Let, r_k, r_l and r_m be the vectors generated by projecting ω_s over ω_k, ω_l and ω_m respectively. We must keep in mind that r_k, r_l and r_m are not unit vectors. Thus,

$$r_k = (\omega_s \,.\, \omega_k)\, \omega_k \tag{30}$$

$$r_l = (\omega_s \,.\, \omega_l)\, \omega_l \tag{31}$$

$$r_m = (\omega_s \,.\, \omega_m)\, \omega_m \tag{32}$$

Now, since the desired direction of rotation is along ω_k, the component along that direction should be increased while those in the perpendicular directions should be decreased. In other words, we have to scale r_k up and scale r_l and r_m down. Let s_u and s_d be the scale factors for scaling the motion up and down. The amount of scaling depends on the amount of assistance that needs to be provided to the user. A higher value of s_u will result in the slave arm having a higher magnitude of motion along the desired orientation direction for the same amount of movement of the master. It will also result in the slave arm having lesser magnitude of orientation motion when the arm deviates from the desired orientation direction. Here too, s_u and s_d follow the constraints given by Equation (21) and (22).

Let , r_k', r_l' and r_m' be the scaled versions of r_k, r_l and r_m so that,

$$r_k' = s_u\, r_k \tag{33}$$

$$r_l' = s_d\, r_l \tag{34}$$

$$r_m' = s_d\, r_m \tag{35}$$

Just like the case in translation, if $r_k$is in a direction opposite to that of ω_k, i.e. if the user is deviating away from the desired rotation path, all the components are scaled down. In this case, we get,

$$r_k' = s_d\, r_k \tag{36}$$

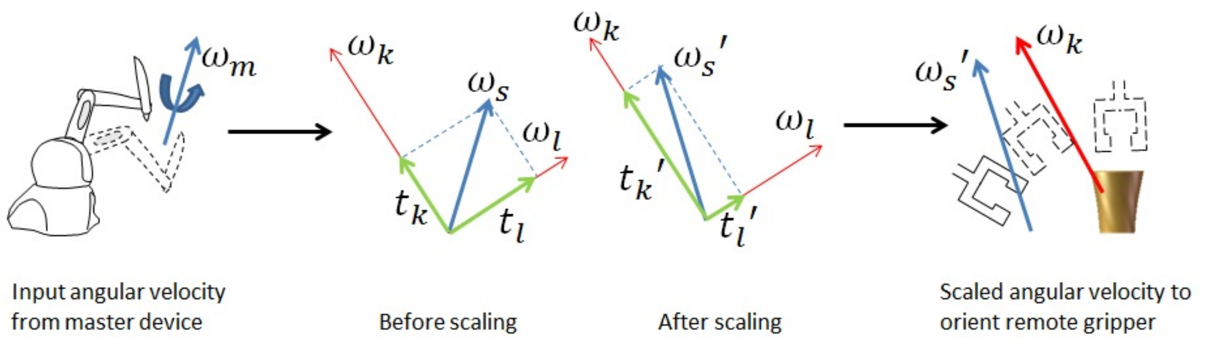
$$r_l' = s_d\, r_l \tag{37}$$

$$r_m' = s_d\, r_m \tag{38}$$

Let, ω_s' be the vector sum of r_k', r_l' and r_m'. Thus,

$$\omega_s' = r_k' + r_l' + r_m' \tag{39}$$

ω_s' is the modified angular velocity vector that will result in the angular motion of the end-effector scaled up in the desired directions. Thus, the users will see the end-effector rotating with a higher angle towards the desired pose with scaled teleoperation. The angle is higher when the user orients the slave end-effector closer to the desired rotation angle. It also results in the end-effector aligning quicker with the desired pose. The concept of scaled orientation is shown in the figure below in two dimensions only, for the sake of simplicity and understanding.



Input angular velocity from master device

Before scaling

After scaling

Scaled angular velocity to orient remote gripper

Figure 13: Scaled rotation concept

Chapter 3: Hidden Markov Model - Theory, Design and Implementation

In this chapter, first the multi-object multi-grasp-configuration identification problem is defined. An explanation of the use of Hidden Markov Model (HMM) to model such a problem and determine the intended object and grasp configuration is presented. An HMM is formally defined and the various parameters that make-up an HMM are listed. A description of the various quantities that make-up the feature vector of an HMM and how its various parameters are estimated from training data is presented. Next a mention of the use of output probability of an HMM and the Viterbi state sequence, to probabilistically determine the desired object and grasp configuration, is made. Finally, the design of the HMM used in the project is presented. This includes the various objects we have modeled and the specifics of the parameter estimation process. Notes on the implementation of our HMM, which includes the user interface developed for testing our algorithm, are presented.

3.1 Multi-object and Multi-grasp-pose Identification Problem and Hidden Markov Model for Motion Intention Recognition

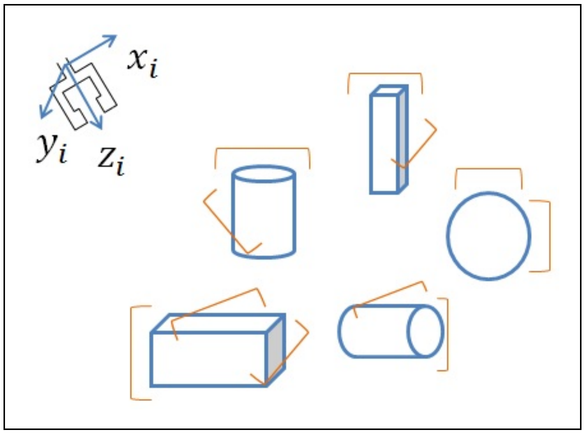


Figure 14: End-effector of a robot and objects from various shape categories with pre-defined grasp poses

In this sub-section, we give a high-level explanation of identifying the object of interest and the grasp pose of interest using Hidden Markov Model (HMM) theory. Consider the environment shown in Figure 14. In the figure we see a gripper at an initial pose with its Cartesian frame defined. There are objects of general shapes like cylinder, sphere and box. The possible grasping poses for each of those objects is shown with gripper jaws surrounding each object. The problem is to identify the object of interest and the grasp pose for grasping the object, using motion data, as the user is teleoperating towards the object.

As mentioned in chapter 1, in order to identify an object of interest in a cluttered environment, the first step we take is to categorize the objects into classes based on their shapes. Let us assume, for the purpose of developing the theory, that we select three basic shapes, a cylinder, a sphere and a box, as the object classes. Thus, any object that approximates one of these shapes can be an object of interest. We also define pre-set grasp poses for each object shape. Thus, from Figure 14, the cylinder can be grasped in certain pre-defined grasp configurations. The same is true for the other shapes. We then develop a model for each object shape by training it based on human motion data. Thus a model is developed, one for each, the cylinder, the sphere and the box. The human motion data used for training is based on the translation and orientation vectors that are generated as the user is teleoperating. For training, a skilled teleoperator is asked to repeatedly preshape in teleoperation to each of the grasping poses for a particular shape. Each time the teleoperator starts from a random starting pose of the end-effector and from different approach directions to the object. Thus, a model for that particular shape is developed. The states of the model are the various grasp configurations of the object and the observations are the projections of incremental master translation and rotation vectors onto each of the remote arm reference vectors. Reference vectors are the ideal translation and rotation vectors from the current remote arm end-effector pose to each of the grasp configurations for the object for which the HMM is being trained. We have observed that the distribution of the projections on the desired reference vectors is approximately exponential in nature. Thus, we have used approximated exponential distribution as our observation probability distribution for each HMM. This process of developing the model is repeated for all shapes. The models for the shapes differ by the parameters of the exponential distribution and by the states. For intention recognition, the likelihood of the HMMs for each object, as the user is teleoperating towards the desired object, is determined. From Figure 14, the likelihood of the 5 HMMs (since there are 5 objects) is computed as the user is teleoperating. We must note that although the two cylindrical objects in the Figure 14 have the same model viz. the one trained for a cylinder, they are still two different models. These two models for the same object shape differ based on the pose of the two cylindrical objects in the environment. Because the poses of the two cylindrical objects are different, the observation probability is higher for the HMM associated with the cylinder that is being approached. This provides a clear indication of the intended cylinder and hence its grasp configurations. If the pose of the cylinders is changed, then the HMM probability computations will change depending on the pose and will compute intentions based on the new poses. Also, a model for each shape needs to be trained only once i.e. the two HMMs for the two cylindrical objects need not be trained twice. The same motion data, based on translation and orientation vectors, is used for intention recognition. The object whose HMM gives the highest likelihood based on the motion data up to that point in time is the one that the user is interested in traversing towards.

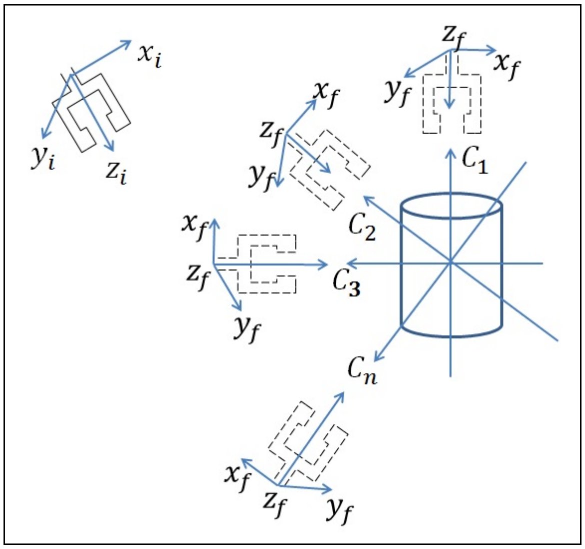


Figure 15**:** Desired grasp configurations along various axes for an object

Once the object of interest has been determined, the next step is to determine the grasping pose that the user is interested in grasping the object with in teleoperation. If the user is only interested in indicating the object of interest to the robot, then the next step is not needed to be computed. It has been already mentioned that the various grasp poses are modeled as the states of the HMM for the selected object shape. States will be explained later in the chapter. For now, the readers should understand that states are those quantities in the HMM that are not visible or measurable using direct means. For example, human mental states are invisible and cannot be measured. The beauty of the HMM lies in determining the invisible states from observable or measurable data. The observable data are called observations and in our method, the motion data based on translation and orientation vectors are the observations. As mentioned in the last paragraph, the observations are the projections of the translation and orientation vectors onto the reference vectors. Assume that a cylindrical object is the object of interest. Let the various pre-defined grasp poses for this shape be represented as in Figure 15. The figure shows the initial end-effector pose with subscript i and the possible final end-effector poses with subscript f. For each of these possible grasp poses, the frame of the end-effector needs to be aligned with that of the object frame at the grasp point. For simpler cases, aligning the z-axis of the end-effector frame, z_i, with the alignment vectors, C_1, C_2, C_3, … C_N, is sufficient. Thus, at each grasp point, there is an alignment frame. The z-axis of this frame is called the alignment vector.

As the user is teleoperating towards a grasp pose for the selected object, the most likely state sequence is determined using motion data. Since the grasp poses represent state of the HMM, the most likely state sequence gives the grasp pose that the user has a likelihood of grasping the object from. In this way we determine the grasp pose of interest for the user. The details of this method will be explained later in the chapter.

The HMM for a shape needs to be trained only once by a skilled teleoperator. Any number of objects of the trained shapes can be added to our environment or object can be removed from the test environment. Re-training is not needed in either of these cases. This makes the method scalable and suitable for unstructured environments. The shape and pose of the objects in the environment needs to be recognized automatically. An RGB-D based vision system is currently being developed in our lab that will serve this purpose. In this implementation, it is assumed that this information is already available to the system before starting a grasping task. A limitation of the proposed method is that it will only work for objects whose shapes closely match to one in our shape set. If an object is encountered whose shape does not match closely to any shape in the set, then the method will return grasp pose that may not give a robust grasp. However, that shape can be trained and then included in our method. We now give details of the Hidden Markov Models (HMM) and explain how we solve our problems using HMM theory.

3.2 Why a Hidden Markov Model?

A Hidden Markov Model is a type of statistical model that models a stochastic process. They have been applied to a number of real-world processes like speech recognition. The process of teleoperating to a desired orientation is stochastic in nature. The randomness of our process comes from the fact that a human will produce errors in motion when teleoperating to traverse the end-effector along a linear trajectory or when orienting it to a desired configuration. During teleoperation, the user deviates from their path unless guided by a virtual fixture. The errors in the case of orientation are more pronounced than those in the case of translation. So even if the user is teleoperating the end-effector along a trajectory or aligning it with a preshape configuration, at some points during the task the user might seem to be teleoperating along a different trajectory or preshape configuration due to errors in motion. Thus, at any point during teleoperation, there is a probability or likelihood associated with the user trying to guide the end-effector to a certain target. The motion from teleoperation has been modeled as an exponential distribution. This means that the histogram of projections of incremental translational vectors on the reference vector over the length of the trajectory is exponential. The reference trajectory is the ideal (shortest straight line) trajectory that would lead to the target. The same is true for orientation vectors. We need a system that models the errors in motion and generates a likelihood (or probability) of the desired preshape configuration out of the several possible ones that the user is trying to teleoperate to. HMM with exponential distribution model is the stochastic model of our choice because of its suitability to the application to our problem. This is explained as follows:

a) Since the motion from teleoperation follows an exponential distribution, we could use an exponential probability distribution function (PDF) to find the probability of translating to a point on an object and orienting to preshape with a specific configuration on it. There is an exponential PDF for each of the preshape configurations. An HMM combines all the PDFs into one model. Instead of comparing the various PDFs with each other to determine the preshape configuration of interest, all the PDFs are combined into a single structure. If we consider the motion data from teleoperation as observations of the HMM, then the observation probability distribution can be comprised of each of the exponential PDFs for each preshape. Each preshape will then lend itself as a state of the HMM. HMM theory will be introduced in the next section and it will define states and observations and will also explain the structure of an HMM.

b) Only using the output of the exponential PDF may not give the right intention if the user deviates from the trajectory by a large amount. Thus, a method is needed that will detect the change in intention not by erroneous movements but by the user trying to deviate from the trajectory on purpose. It should be able to gradually change the intention by steadily increasing the confidence level of the new intention. Viterbi decoding lends to this requirement. Viterbi decoding will be explained in a later section in detail.

c) In order to identify the object of interest from a number of them in an environment, a method is needed that will compute the likelihood of the user teleoperating to all objects present and determine the one which has the highest likelihood.. By associating an HMM with every object, the HMM with the highest likelihood of occurrence gives the most likely object that the user is teleoperating towards. Determining the HMM that is most likely to occur out of a number of them, is a well-defined problem that can be solved using the HMM theory.

d) Human intention or mental states are invisible and cannot be measured. A method is needed to determine the invisible states from observable or measurable data. If we consider the motion data as observations and grasp poses as states, then HMM can be used to determine grasp poses or human intention from motion data.

Moreover, our process obeys the following constraints of a discrete state first-order Markov chain. The states are not dependent on time and the observations at any point in time are also dependent on the state at that instant only.

3.3 Hidden Markov Model Theory

As mentioned previously, a Hidden Markov Model (HMM) models a doubly embedded stochastic process. The states are hidden whereas the observations are visible. It has also been mentioned that, in our model, the grasp configurations are modeled as states and the projections of translation and rotation vectors, as observations. The main elements that make up a Hidden Markov Model and describe its structure are its initial probability distribution, state transition probability distribution and observation probability distribution [43].

a) Initial State Distribution: Let as S = $[S_1, S_2, S_3, \ldots S_N]$ represent the set of states of an HMM. The initial state distribution gives the probability of the system being in a certain state, for all states, at the instant the process begins. It is denoted by π = { π_i} where,

$$\pi_i = P[q_1 = S_i], \qquad 1 \leq i \leq N \tag{40}$$

b) State Transition Probability Distribution: As mentioned earlier, the state transition probability distribution gives the probabilistic relation or interdependence among the various states. Generally, these are discrete probability values as the states are discrete in most applications. They are denoted by A={ a_{ij}} where

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \qquad 1 \leq i, j \leq N \tag{41}$$

For those cases, where it is not possible to transition from state i to state j, a_{ij}=0, otherwise $a_{ij} \geq$ 0. Also, as stated earlier ${}_{j=1}^{N}\Sigma\, a_{ij}$= 1 i.e the sum of the probabilities of transitioning from one state to all the other states is 1.

c) Observation Probability Distribution: The observation probability distribution for a state gives the probabilistic relationship between that state and all the observations. It exists for each state. It can be discrete or it can be a continuous probability distribution depending on whether the observations are discrete or continuous. For discrete set of observation symbols V = [v_1, v_2, v_3, … v_M], the observation probability distribution at state j, B = {$b_j(k)$}, where

$$b_j(k) = P[v_k \text{ at } t|q_t{=}S_i] \qquad 1\leq\ j \leq N$$

$$1\leq k \leq M \qquad (42)$$

The observations at various time instants are represented as O_1, O_2, … O_T, where at a certain time instant t the observation O_t can have any value from the set V. For the case in which the observations are continuous, B is computed as a probability density function (PDF),

$$b_j(\mathrm{O}) = \mathscr{N}(\mathrm{O}, \mu_j, \Sigma_j) \qquad (43)$$

where O is the observation vector being modeled, $\mathscr{N}$ is a PDF usually a Gaussian [43], μ_j is the mean and Σ_j is the covariance for the distribution for jth state. $\mathscr{N}$ can represent any form of probability distribution. This is decided based on the distribution of the observations over time. Thus, to completely specify an HMM, one needs to specify the number of states and observations N and M, observation symbols and state symbols V and S, and the probability measures π, A and B. The probability measures are often written in a compact form,

$$\lambda = (\pi, A, B) \qquad (44)$$

Once an HMM is defined, it can be used to solve three basic problems [43]. The first problem that the HMM solves is that of determining the probability that a certain sequence of observation are produced from a certain HMM. The second problem is that of estimating the most likely state sequence given a set of observations and the HMM. The third problem is that of determining the parameters of the HMM from a given observation and state sequence. For more details on HMM definition, design and implementation, the reader is requested to refer to the tutorial on HMM from Rabiner [43].

3.4 Adapting the Hidden Markov Model to Multi-Grasp Pose Identification Problem

As mentioned previously, a model comprising of exponential distribution and HMM will be trained for each object class from the motion data generated by teleoperating to various grasp preshape poses for that object class. It was also mentioned that the motion data from teleoperation could be used as observations and each preshape pose as a state. In this section, we explore the various ways by which we make our problem fit the three problems that HMM solves.

The problem 1 that HMM solves is that, what is the likelihood that the given sequence of observations is produced by the given model. In other words, the solution to problem 1 gives a score of how well a given model matches the given sequence of observations. This viewpoint is extremely useful in our case as it can be used to determine as to which of the HMMs out of the various available ones most closely matches the observations. In our case, an HMM is associated with every object and motion data are the observations. Thus, based on the motion data, problem 1 can be used to determine the most likely object in a cluttered environment that the user is teleoperating towards.

Problem 2 that the HMM solves gives the optimal state sequence given the observations and the model i.e. it gives the state sequence that is most likely to be produced. An HMM has been associated with every object and the grasp configurations and motion data have been modeled as states and observations respectively. Thus, once the system determines the intended object by solving the problem 1, the most likely sequence of grasp configurations, as the user is teleoperating, can be determined by solving problem 2. The mean or mode value of this sequence of grasp configurations can be used to determine the intended grasp configuration. Viterbi decoding has been used to solve problem 2 [43].

Problem 3 that the HMM solves estimates the model parameters given the state and observation sequences. We can use the solution to this problem to train our model for each object class. This can be achieved by teleoperating to a specific grasp configuration several times starting from random poses. This will be repeated for all the possible grasp configurations for an object class. As a result, observation sequences will be generated and since we are teleoperating to specific known grasp configuration, the state sequence is known too. Thus, solution to problem 3 can be used, to estimate the HMM parameters for each object class. Baum-Welch algorithm is a well-known algorithm that gives the solution to problem 3. However, such a training procedure will be costly due to the number of trials needed for training. If n trials are needed to train a state transition matrix element and if there are k grasp configurations for an object, the total number of trials needed for training the object's state transition matrix will be n^k. For 10 trials and an object with 4 grasp configurations, this number is 10000. Therefore, we assume the values of state transition matrix and initial probability distribution. Since no reasonable assumption can be used for the parameters of the observation probability distribution, these are estimated by conducting repeated trials of teleoperating to each grasp configuration of an object. The process will be discussed later.

3.5 Hidden Markov Model Design for an Object and Feature Vector Computation

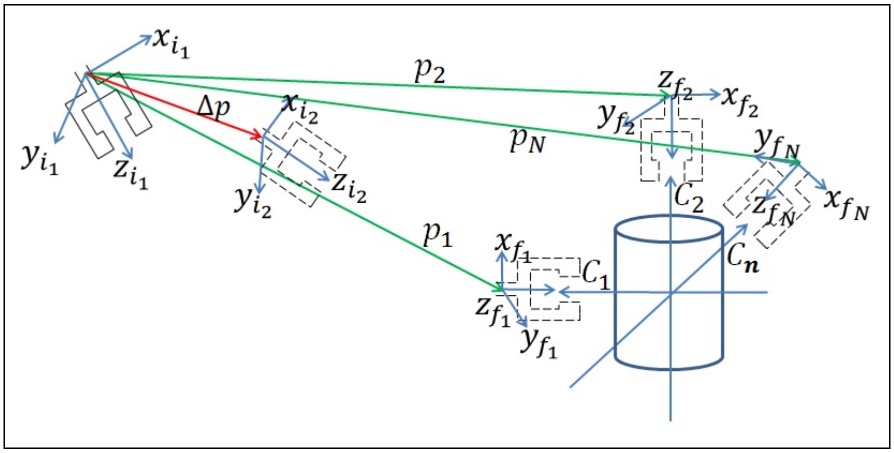


Figure 16: Feature vector representation for a cylindrical object

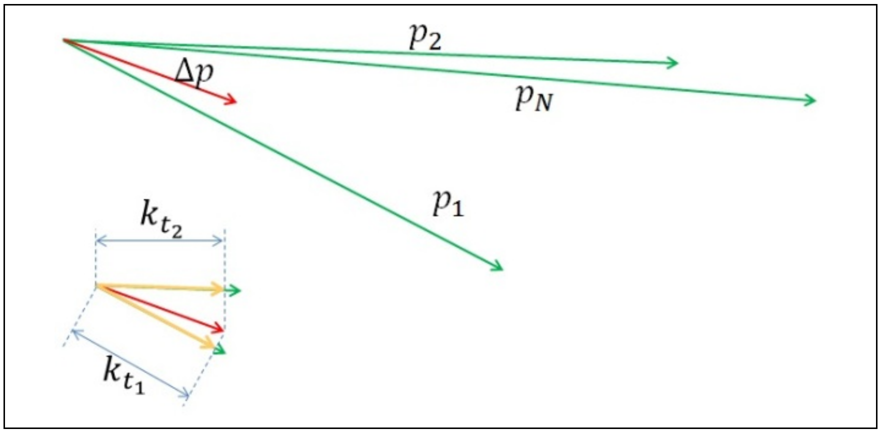


Figure 17: Translational projection elements of the feature vector for a cylinder

For the purpose of presenting the theory, we will develop an HMM for a cylindrical object. Consider a cylindrical object with N possible preshape configurations as shown in the Figure 16. Each of these configurations is a state of the HMM. Thus, the HMM has N states. Let us now see how the observations are calculated. The observation data is in the form of a $2N$X1 vector and it is called a feature vector. The feature vectors are generated at every time instant as the master arm and remote arm are moving towards the preshape pose under teleoperation.

As shown in the Figure 16, let the remote arm gripper frame at a certain time instant be defined by $(x_{i_1}, y_{i_1}, z_{i_1})$ and let the gripper frame change to $(x_{i_2}, y_{i_2}, z_{i_2})$, at the next time instant, as a result of teleoperation. The pose of the gripper also changes from one time instant to the next. The aim of the user teleoperating the arm is to align the gripper to one of the preshape configurations at $C_1, C_2, \dots\ C_N$. Let, (x_f, y_f, z_f) be the gripper frame at these preshape configurations. Let Δp be the incremental translational vector (shown by the red arrow in Figure 16) due to the movement of the gripper frame. Let, $p_1, p_2, \dots$ $p_N$be unit vectors having the direction from the current gripper frame location to the desired location of the gripper frames at various preshape configurations. Let, $k_{t_1}, k_{t_2}, \dots\ k_{t_N}$be the magnitude of projection of Δp on each of $p_1, p_2, \dots\ p_N$. These magnitudes are not absolute values and take into consideration the sign of projections also. These magnitudes of projections are diagrammatically shown in Figure 17.

Also, as the gripper is moving, the direction cosines of the gripper z-axis $z_{i_2}$is computed with all the z-axes of the likely preshape frames viz. direction cosines of $z_{i_2}$and $z_{f_1}, z_{f_2}, \dots\ z_{f_N}$. Let these direction cosines be called as $k_{r_1}, k_{r_2}, \dots\ k_{r_N}$. These projection magnitudes are shown in Figure 18. Again, these are not absolute values and consider the sign of the projection also.

The magnitudes of the projections of translational vectors as well as those of the gripper z-axis together make up the feature vector of observations for the HMM of the cylinder. Thus, the observations generated at every time instant are $O = [k_{t_1}, k_{r_1}, k_{t_2}, k_{r_2}, \dots,\ k_{t_N}, k_{r_N}]^T$. Since there are two members for every preshape configuration and since there are N preshape configurations, the length of the feature vector generated at every instant is $2N$X1. Since these values are continuous, we will be using exponential probability density function for determining the observation probabilities.

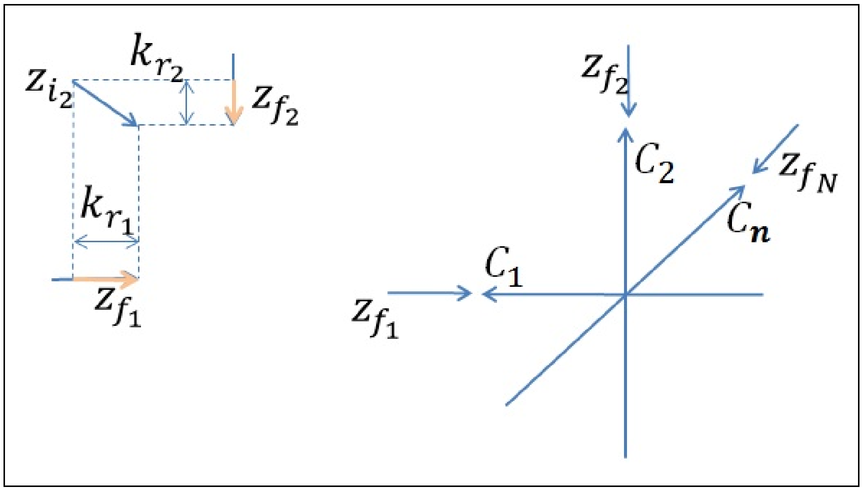


Figure 18: Gripper z-axis projections for the feature vector for a cylinder

3.6 Estimating Hidden Markov Model Parameters and Computing Observation Probability

We have seen how observations are calculated for objects which can be approximated to a cylinder. In order to develop an HMM for these shapes, we need to train the model using these observations and states as inputs. Training the model is determining the parameters that completely define the model viz. the parameters π, A and B. As mentioned earlier, Baum Welch algorithm is a popular algorithm that can be used to estimate the parameters. In our case, we assume the initial probability π and state transition probability distribution A. Since we are using exponential PDF to compute observation probability distribution B, the rate parameter of the Gaussian PDF for each state becomes the parameter that needs to be estimated. The estimation process and the assumptions are explained next. The assumptions and the estimation process of the parameters are the same for the models for all shapes.

We assume that the user can start orienting towards any of the available desired preshape configurations. Thus, there is an equal likelihood to start teleoperating towards any of the preshape configurations. Or in other words, all states have equal initial probability values. Thus, for N states, our initial state probability distribution is π = [1/ N, 1/ N …1/ N] and it is of the order Nx1. In certain cases when the initial gripper pose is always the same, the probability for that case can be made high or equal to 1 and 0 for the rest of the preshape poses. In some cases, it may be impossible to start at certain gripper poses. Here the probability for those poses can be made zero and can be equally divided among the rest of the poses as long as the likelihood of starting at rest of the poses is equal. In our experiments, we start at random poses and hence all the elements of the initial probability distribution have equal value.

In our experiments, a user teleoperating for grasping an object from a desired preshape pose will be assisted once the system determines their intention as to which preshape pose the user is interested in. We assume that the user normally adheres to a particular desired preshape configuration throughout a grasping task. This is a natural assumption because even in our daily activities, we normally adhere to our chosen grasping configuration while translating and orienting our hands for grasping a particular object. Thus, our state transition probability distribution has the following form:

$$\begin{aligned} A_{ij} &= 0.9, \qquad\quad i = j \\ A_{ij} &= 0.1/(N-1), \; i \neq j \end{aligned} \tag{45}$$

In tasks where the users change their intention of grasping from one pose to the other, the state transition probability distribution can be trained accordingly. This can happen in cases where after grasping an object from a certain pose, the user is highly likely to go to grasp another object with a certain preferred pose e.g. a kitchen based environment where a spoon is usually grasped after a bowl is picked up. Baum Welch algorithm can be used for training in these cases.

Our observations are continuous in nature and we have found out that the observations follow an exponential distribution. Thus, there is an exponential distribution associated with each state and this makes up our observation probability distribution B. In order to estimate B we need to estimate the rate parameter λ_j of the exponential PDF for each state j. Several trials of teleoperating to preshape each desired pose starting from random poses are conducted. Observations are recorded for each trial and λ_j for each desired pose j are computed as explained below.

We have found that the distribution of projection magnitudes of the incremental translation and orientation vectors on the desired translation and orientation vectors, over the length of the trajectory, is positive exponential in nature. The projection magnitudes of translation and orientation vectors on the reference vectors belonging to undesired grasp configurations is not exponential for all cases. In the case certain objects (viz. the plate and the bowl) the distribution of projection of orientation vectors on undesired grasp configurations is negative exponential but the same is not true for the configurations 3 and 4 of the cup. The distribution of projections of translation vectors seems to follow an asymmetric Gaussian distribution. Thus, due to lack of a pattern for projection magnitudes on undesired configurations, these are not used in the model.

Let, O_1, O_2, … O_τ, be the feature vectors generated by teleoperating the end-effector to a certain preshape pose j. Let $[k_{t_j}, k_{r_j}]$ be the projection magnitudes of the translation and orientation vectors on the reference vectors for pose j. $[{k_{t_j}}^1, {k_{r_j}}^1]$, $[{k_{t_j}}^2, {k_{r_j}}^2]$, … $[{k_{t_j}}^\tau, {k_{r_j}}^\tau]$ are generated at each time instant as the observations , O_1, O_2, … O_τ, are generated. The generation of $[k_{t_j}, k_{r_j}]$ was explained in the previous sub-section. Let K_t be a 1 x τ matrix of translation projection observations such that,

$$K_t = [{k_{t_j}}^1, {k_{t_j}}^2, \ldots {k_{t_j}}^\tau] \tag{46}$$

where τ is the number of feature vectors recorded for state j. Let K_r be a 1 x τ matrix of translation projection observations such that,

$$K_r = [{k_{r_j}}^1, {k_{r_j}}^2, \ldots {k_{r_j}}^\tau] \tag{47}$$

where τ is the number of feature vectors recorded for state j. Let I be a τx1 vector of ones, such that,

$$I = [1, 1, 1, \ldots\ 1]^T \tag{48}$$

Then, the mean vector for the translation projection exponential distribution for state j can be computed as,

$$\mu_{t_j} = \frac{1}{\tau}\ K_t\ I \tag{49}$$

μ_{t_j} is a vector the order 1x1. The mean vector for the translation projection exponential distribution for state j can be computed as,

$$\mu_{r_j} = \frac{1}{\tau}\ K_r\ I \tag{50}$$

μ_{r_j} is a vector the order 1x1. The maximum likelihood estimate of the rate parameter of an exponential distribution is given by the inverse of the mean. Thus, the rate parameter for the exponential distribution of translation and orientation projections on desired vector is given by,

$$\lambda_{t_j} = \frac{1}{\mu_{t_j}} \text{ and } \lambda_{r_j} = \frac{1}{\mu_{r_j}} \tag{51}$$

In this way, the rate parameters for translation and orientation projections i.e. λ_{t_j} and λ_{r_j} for a particular state j are computed from the observations. Similarly, we can compute λ_{t_j} and λ_{r_j} for all the remaining states. In order to ensure that the distribution is probabilistic or the area under the distribution curve is unity, we evaluate co-efficient c_{t_j} as,

$$\int_{-1}^{1} c_{t_j}\ e^{(\lambda_{t_j})(x)} dx = 1 \tag{52}$$

The limits of integration are the maximum and minimum value of projection scalars on reference vectors. Then, the observation probability for a state j, due to translation and rotation projection scalars k_{t_j} and . k_{r_j}, is given by,

$$p_{t_j} = c_{t_j}\ e^{\lambda_{t_j}\, k_{t_j}} \text{ and } p_{r_j} = c_{r_j}\ e^{\lambda_{r_j}\ k_{r_j}} \tag{53}$$

We also determine the normalized distance of each grasp point from the end-effector position. This is used to weight the probability value, determined from the exponential distribution, according to the nearness of a grasp point. This quantity helps to prevent fluctuations in intention determined when the arm is making insignificant translational movements or is simply rotating without any translations. When the hand is not translating and only rotating, all the objects in the environment are virtually located at the same point. This is because all the projection magnitudes k_{t_j} are zero and there is no way to distinguish among different grasp points based on position. In such a scenario, grasp points with similar grasp configurations are alike and the intention recognition algorithm can output any of these grasp points as the desired grasp points. We have observed that motions involving no translations and only rotations occur near an object of interest. Thus, the nearness measure will help to not consider grasp points far away with similar grasp configurations, increasing the accuracy. Moreover, once the hand has been preshaped, the user usually engages in small movements to fine-align the gripper with the grasp configuration as imagined by the user. This might result in deviations of the gripper frame from the recorded grasp pose. Again, if we only consider the probability due to differential translations and orientations, the algorithm might give a wrong value as there might be other objects along the same direction of motion and with similar grasp configurations. Considering a weighted measure based on object nearness will prevent these unwanted fluctuations in intention determination.

Weighted probability is determined as inverse of twice the normalized distance value of the particular grasp point. The normalization is with respect to all other grasp points of all the objects in the environment. The factor 2 is considered so that the area under the probability distribution curve integrates to 1. Let D be sum of distance of all grasp points from the hand at the current time instant.

$$D = \sum_{i=1}^{P} \sum_{j=1}^{N} d_{ij} \tag{54}$$

where N = total no. of grasp configurations for object i and P = total no. of objects.

Then, the normalized distance value of the grasp point with grasp configuration j, belonging to object i is given by,

$$n_{d_i} = \frac{d_{ij}}{D} \tag{55}$$

Computing weighted probability based on distance makes it difficult for a user to move away from an object if they are close to it. This is because the probability is much higher for the object that is near than for other objects. The user experiences resistance while teleoperating the end-effector away from the object that is near, if the user decides to change her intention and move towards another object. The end-effector seems stuck in the vicinity of the object. The skilled teleoperator was able to retrieve the end-effector after about eight to twelve repeated movements whereas unskilled teleoperators took more movements to achieve the same result. This is because a skilled teleoperator is able to teleoperate the end-effector in such a way that their motions result in higher translation and orientation projections, thus causing an increase in the translation and rotation probabilities sooner than that caused by an unskilled teleoperator. It was a major cause of frustration for unskilled teleoperators. To circumvent this problem, a weight factor based on the direction of translation motion was introduced. This parameter was computed by adding 1.005 to translation projection magnitude k_t. Since, k_{t_j} varies between -1 and 1, the weight factor due to direction of translation will vary between 0.005 and 2.005. The small number 0.005 prevents the total probability from becoming zero at any time. As a result, wrong object detection due to rotation-only movements is prevented by adding the weight due to proximity of an object, to the probability, and the end-effector getting stuck near an object is prevented by adding the weight due to direction of translation.

Once the rate parameters are estimated, they can be used to calculate the likelihood of the occurrence of the observations, given the state. In other words they can tell as to what is the likelihood that a feature vector was generated while teleoperating given that the user is intending to grasp from a certain preshape configuration.

Thus, if an observation O is generated as a result of teleoperation, then the likelihood that this observation occurred assuming that the user was intending to preshape to pose j is given by,

$$b_j(O) = p_{t_j}\ p_{r_j}\ p_n\ \ p_d \tag{56}$$

where:

a) p_n= nearness probability due to normalized distance = 1/(2 n_{d_i})

b) p_d = weight factor due to direction of translation = 1.005 + k_{t_j}

As the user is teleoperating towards a particular grasp configuration, the observation probability $b_j(O)$ is computed for all the grasp configurations. The state for which the observation probability value is the highest, for a particular feature vector, is the most likely state at that instant. Although, the score of the exponential PDF is a good measure of the intention at that instant, it however does not take the history of movements and state transitions into consideration. Hence we need to determine the intention by using a method that uses the score computed by Equation (56) and also gives the most likely states over a period of time. For including these considerations, we make use of the HMM theory. This method is explained in the next section.

3.7 Object Identification by Determining Output Probability

We have already shown how we develop a model for each shape. Thus every object in the environment has a model assigned to it depending on its shape. In order to determine the intended object of interest as the user teleoperates, we use the solution to problem 1 of HMM theory. The problem 1 of HMM theory determines the likelihood that a given set of observations are generated by a certain model. As the user teleoperates, the likelihood of the observations being generated by each model (one for each object) is computed. The model that gives the highest likelihood value for the observations generated up that point is the one that is most likely to occur. Thus, determining the most likely model gives us the most likely object that the user intends to traverse towards. This computation of output probabilities and their comparison takes place online as the user is teleoperating. The mathematical process of determining the most likely model using HMM theory is explained in the tutorial by Rabiner [43] and the reader is requested to refer to that text. Here we give a summary of the process.

The likelihood of a model, given the observations up to a certain time instant, needs to be determined over all the possible state combinations. This process is computationally expensive but a procedure called Forward algorithm [43] makes it efficient. The process starts with determining the probability of the model given the observation at time instant one, for each state. Then, as the observations are continuously generated, the probability of the model being in a certain state at a time instant, over all states, is computed by taking the summation of the product of the probabilities of being at a state at the previous time instant, the state transition to the state under consideration and the observation probability given the state. The total probability of the model at a time instant is then the sum of the probabilities at each state just computed.

3.8 Multi-grasp-configuration Identification Using Viterbi Decoding

After the intended object has been identified, the next step for the system is to identify the grasp configuration of interest. . Since, the intention or the desired poses are the states of the HMM, we need to find the most likely states as the user is teleoperating, given the model and the observations. Thus, we use the solution to the problem 2 of the HMM viz. to find the most likely state sequence given the model and the observations.

There are a number of ways by which the optimal state sequence can be chosen depending on the optimality criteria. One way is to select the single best state at every instant. This method will not work for those cases where the state transition probability for transitioning from one state to the next may be 0 and those two states are the best states for two consecutive time instants. Other solutions find the best state pairs (q_t, q_{t+1}) or triplets (q_t, q_{t+1}, q_{t+2}). However, the most widely used algorithm for determining the most likely state sequence is the Viterbi algorithm which finds the single best state sequence for all observations given as input i.e. it maximizes $P(Q \mid O , \lambda)$.The Viterbi algorithm is explained in the tutorial by Rabiner [43] and we briefly describe it here. For complete details and mathematical equations, the reader is requested to refer to [43].

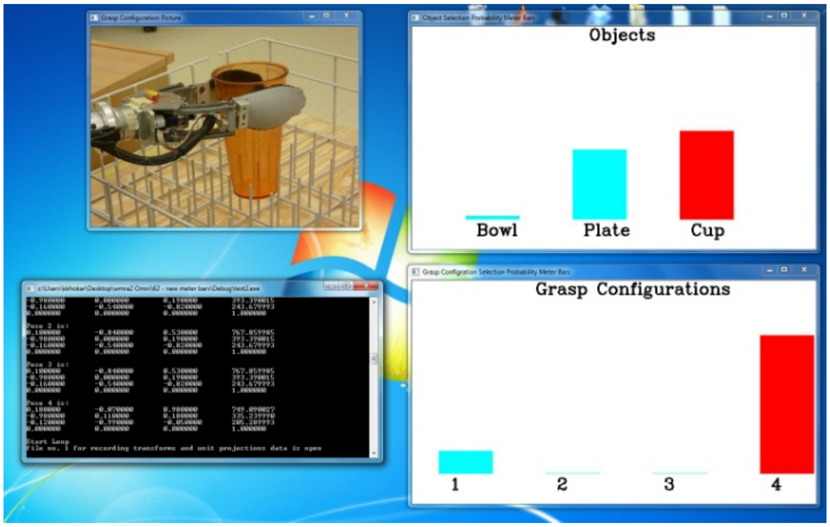
The procedure to determine the most likely state sequence is similar to that of determining the probability of the model except that the summation is replaced by maximization. The state from the previous time instant that leads to the maximum transition probability is stored in a separate data structure and so is the probability value for the state at the current time instant. This way the best states are determined at every time instant Once the last observation is encountered, the state with the maximum probability at that time instant is taken as the best state and then backtracking the stored maximum states up to the first time instant gives the optimal state sequence.

Thus, as the arm is being teleoperated and observations are generated, Viterbi decoding will give the most likely state sequences or the most likely preshape poses that the user intends to align the gripper frame with. By backtracking to the last T time steps every time a new feature vector is encountered, the most likely state sequence can be updated. If the states are denoted by whole numbers, then either the average of the states over the last T time steps or the maximum mode state over the last T time steps can be used to determine the preshape pose at that instant. Refreshing the path of best state sequence this way when a new feature vector is encountered takes into account the trend of changing intention because the values of best state paths are retained and are not reinitialized every time a new feature vector comes in and the first one in the queue is removed.

3.9 Hidden Markov Model Design and Implementation

In our implementation of the HMM theory presented above, we have used a bowl shape, a plate shape and a cup shape as the object classes. The objects and the preselected grasp configurations are shown in the Figure 20. Note the acronyms used to describe grasp configurations for each object in Figure 20 caption. Thus, we have selected objects in a kitchen environment. The three object classes are clearly distinguishable based on their shapes. Moreover, the bowl can be grasped from three configurations, the place from two and the cup from four. These grasp configurations have been selected by the skilled teleoperator according to his knowledge of what grasp will lead to a robust grasp. Other grasp configurations are possible; for example, the bowl can be grasped from its fourth edge and the plate from the other side with a horizontal grasp. But, without the mobility of the wheelchair or the platform, these configurations are difficult to reach or unreachable by the manipulator approaching objects from one side.

Figure 19 shows the graphical user interface developed to test our object and grasp configuration identification algorithm. It is shown on the computer screen in front of the user as the user is teleoperating towards the desired object and grasp configuration. It also acts as a feedback for the users executing a grasping task in teleoperation, using our method. It is a passive display i.e. the user can only be feedback visual information and the user cannot send any commands using this interface. It consists of dynamically changing images of the predicted grasp configuration and dynamically changing probability meter bars. The same images shown in Figure 20 are the images produced on the screen. The top window on the right half of the screen shows the normalized output probability of each object's HMM. The tallest column gives the object predicted. The bottom window shows the normalized count of the states from the Viterbi sequence of the selected object. The tallest column in this window represents the predicted grasp configuration.


Grasp Configuration Picture
Object Selection Probability Meter Bars
Objects
Bowl
Plate
Cup
Grasp Configuration Selection Probability Meter Bars
Grasp Configurations
1
2
3
4

```
c:\Users\kkhokar\Desktop\wmra2 Omni\62 - new meter bars\Debug\test2.exe
-0.980000       0.000000        0.190000        393.390015
-0.160000       -0.540000       -0.820000       243.679993
0.000000        0.000000        0.000000        1.000000

Pose 2 is:
0.100000        -0.840000       0.530000        767.859985
-0.980000       0.000000        0.190000        393.390015
-0.160000       -0.540000       -0.820000       243.679993
0.000000        0.000000        0.000000        1.000000

Pose 3 is:
0.100000        -0.840000       0.530000        767.859985
-0.980000       0.000000        0.190000        393.390015
-0.160000       -0.540000       -0.820000       243.679993
0.000000        0.000000        0.000000        1.000000

Pose 4 is:
0.180000        -0.070000       0.980000        749.090027
-0.980000       0.110000        0.180000        335.239990
-0.120000       -0.990000       -0.050000       205.289993
0.000000        0.000000        0.000000        1.000000

Start Loop
File no. 1 for recording transforms and unit projections data is open
```

Figure 19: Static image of predicted grasp configuration and dynamically changing probability meter bars

The interface consisting of meter bars was developed using OpenCV library. The rate at which the meters were displayed was a fourth of the WMRA control loop, which was 70 to 870 Hz. This was done because it was observed that arm motion becomes intermittent and much delayed if the OpenCV image display functions ran at 70 to 80 Hz. This results in non-intuitive motion. Dropping the frequency of OpenCV image processing functions to a fourth of WMRA control gave normal motion of the arm and an intuitively changing feedback. There was still a delay in the in WMRA executing master device movements. However, the arm moved continuously once it began moving. This delayed response was still intuitive and persists even when OpenCV image processing is not used.

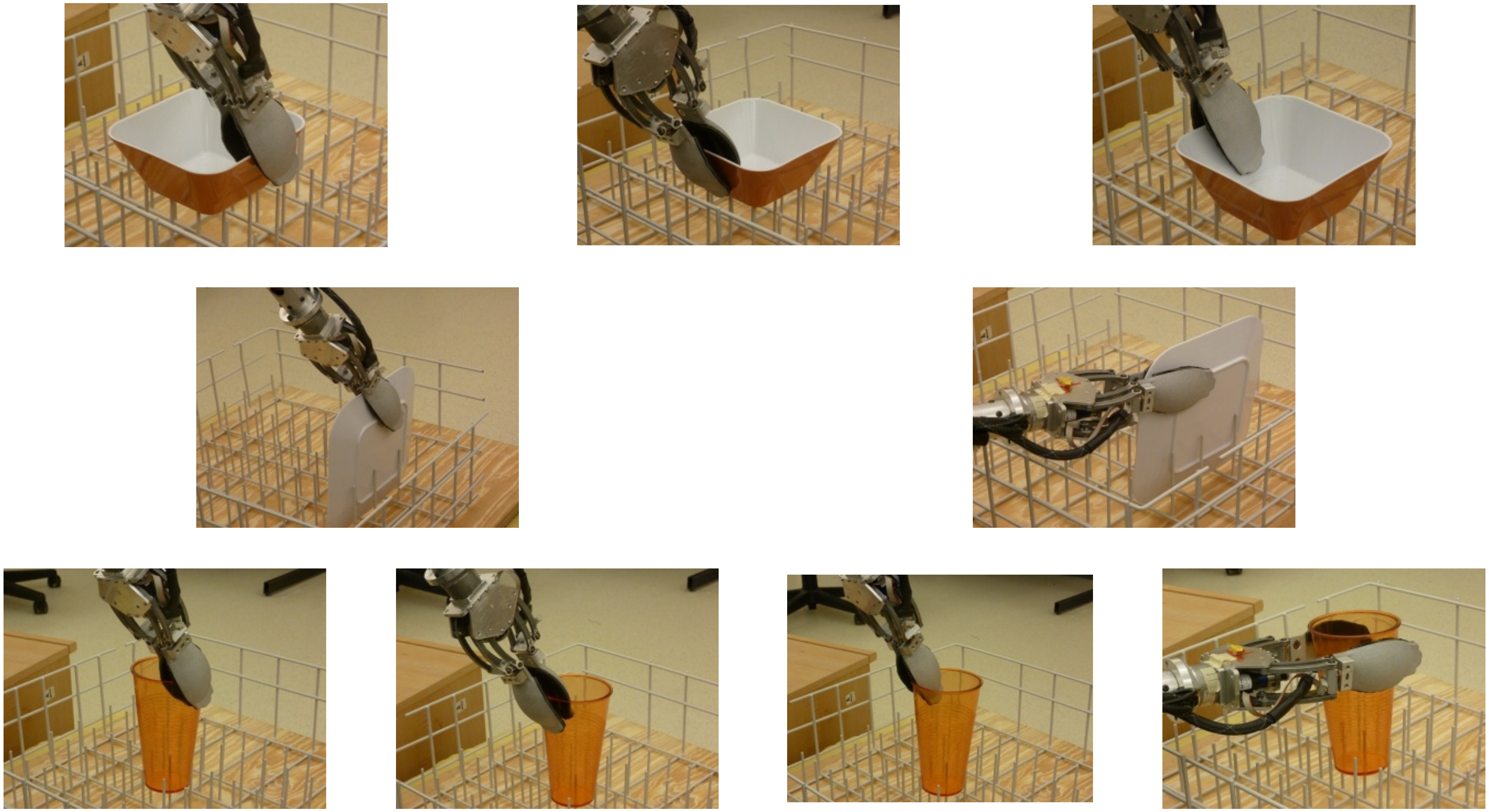
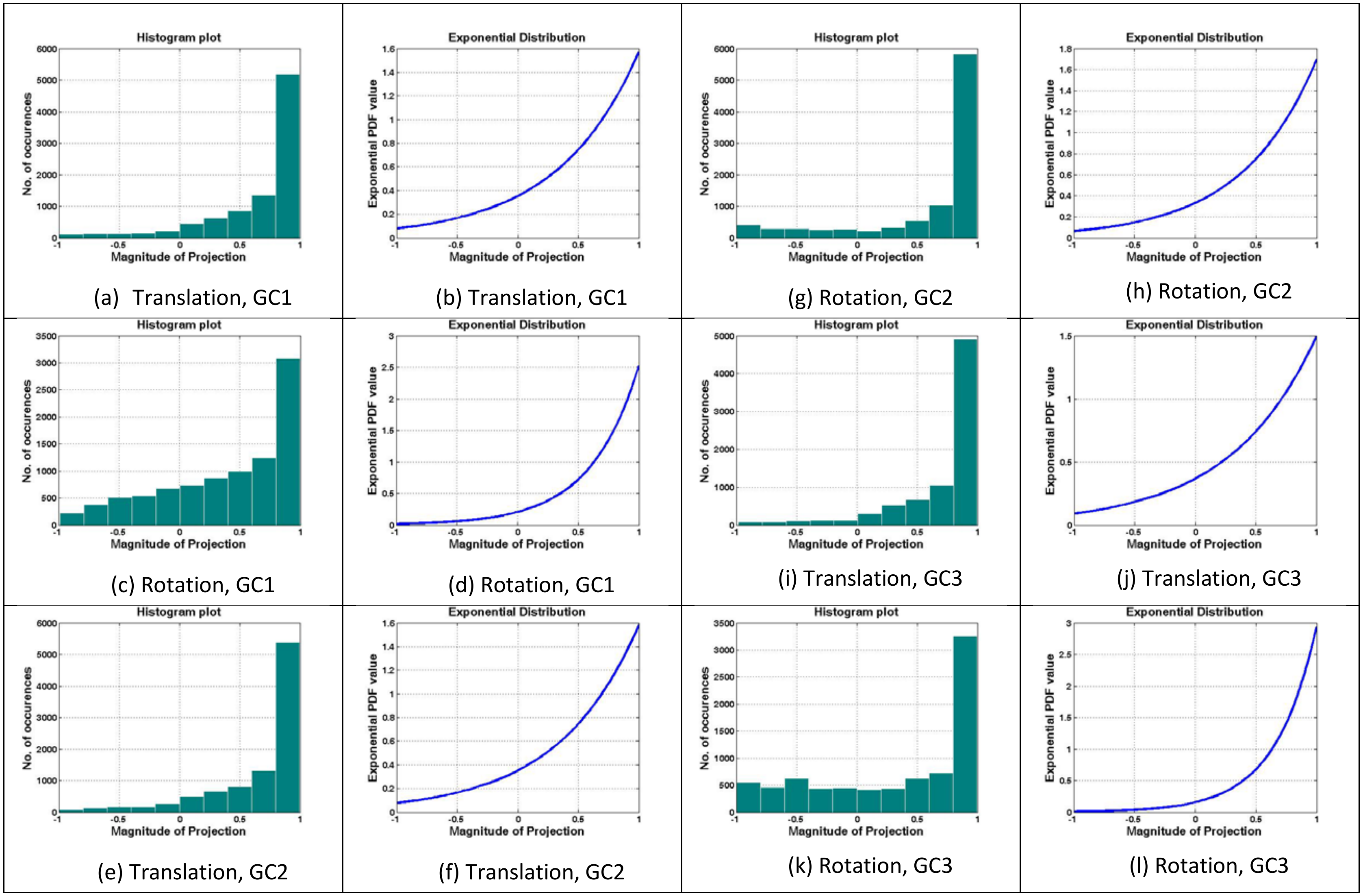


Figure 20: Objects and grasp configurations used in our implementation. First row shows three configurations for the bowl (B1, B2 and B3). Second row shows two configurations for the plate (P1 and P2). Third row shows four configurations for the cup (C1, C2, C3 and C4)

Since, we have three distinctly shaped objects, viz. bowl, plate and cup, we have three object classes and hence, three HMMs. The HMM for bowl has three states since bowl has three grasp configurations. Similarly, the HMM for plate and cup have two and four states respectively. We know that, two exponential distributions, one for translation and one for rotation, are associated with each state and they are used for observation probability computation. For computing the rate parameters and coefficients for each exponential distribution, as explained in Section IV, a skilled teleoperator repeatedly preshaped over each grasp configuration of each object ten times, starting from random remote arm end-effector poses each time. Ten trials were chosen since convergence of rate parameter values was obtained after 10 trials for each configuration. In all, the skilled teleoperator performed 90 trials. Figure 21 shows the histograms and the approximated exponential distributions for translation and rotation, obtained from training trials on all the grasp configurations of the bowl. Figure 22 shows the histograms and the exponential distributions for the plate. Figure 23 shows the histograms and exponential distribution for the first two grasp configurations of the cup whereas Figure 24 shows the ones for the remaining two configurations. Table 2 lists the values of the rate parameters and the coefficients for translation for all the grasp configurations of all the objects. Table 3 lists the same values for the rotation. These values were calculated by using the training data from the skilled teleoperator and using Equations (51) and (52) described earlier.

Table 2: Rate parameters and coefficients for translation

Object	Grasp Configuration	λ_{t_j}	c_{t_j}
Bowl	1	1.4946	0.353
	2	1.5077	0.351
	3	1.4046	0.366
Plate	1	1.2459	0.391
	2	1.3488	0.375
Cup	1	1.6076	0.336
	2	1.6961	0.322
	3	1.4833	0.355
	4	1.7564	0.313



(a) Translation, GC1

(b) Translation, GC1

(g) Rotation, GC2

(h) Rotation, GC2

(c) Rotation, GC1

(d) Rotation, GC1

(i) Translation, GC3

(j) Translation, GC3

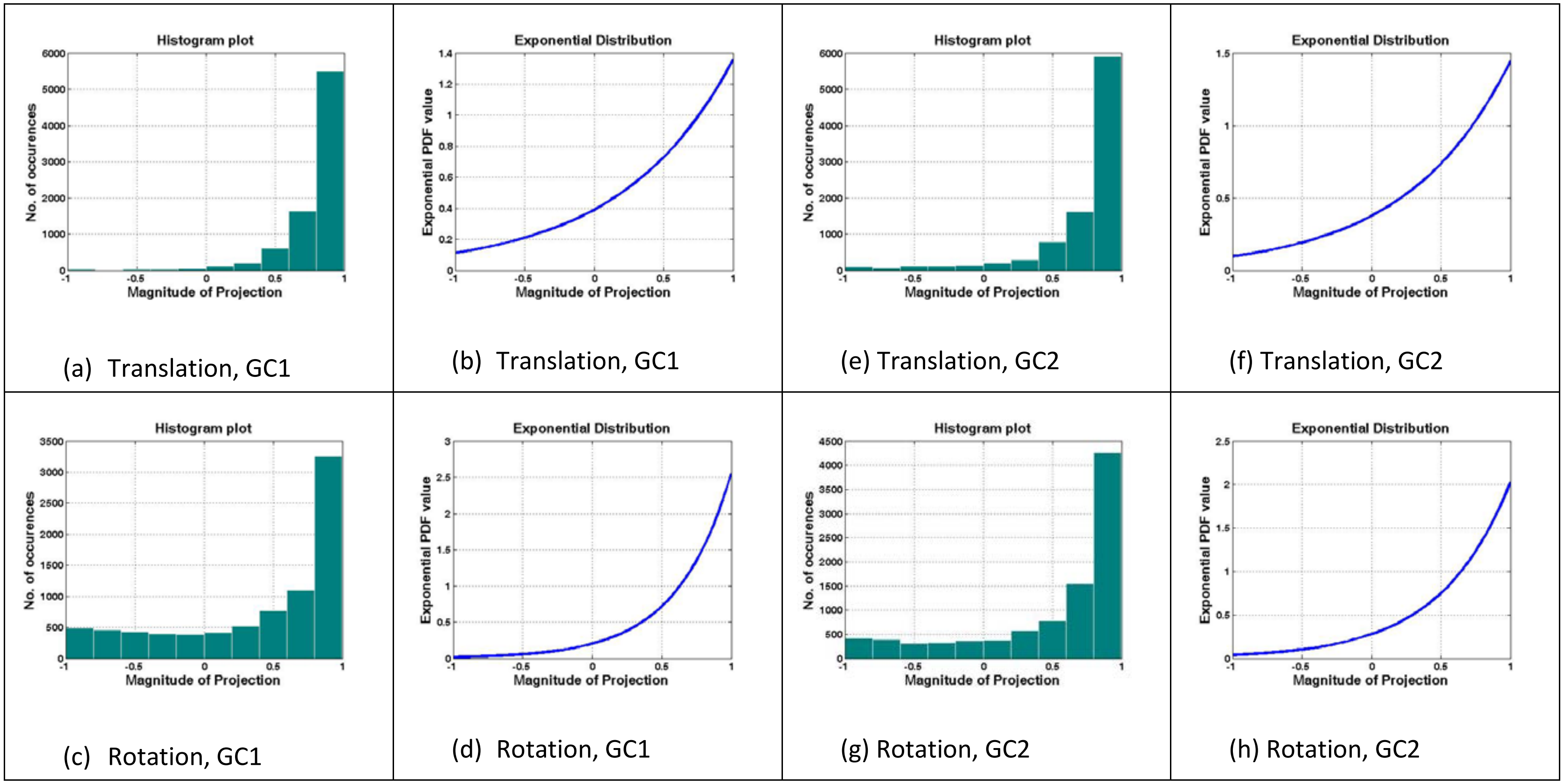
(e) Translation, GC2

(f) Translation, GC2

(k) Rotation, GC3

(l) Rotation, GC3

Figure 21: Histograms and approximated exponential distributions of projection scalars for the bowl. GC: grasp configuration; x-axis (all figures): projection of translation and rotation vectors on reference vectors, y-axis (figures a, c, e, g,I, k): number of values in histogram, y-axis (figures b, d, f, h, j, l): value from approximated exponential distribution (Equation 53)



(a) Translation, GC1

(b) Translation, GC1

(e) Translation, GC2

(f) Translation, GC2

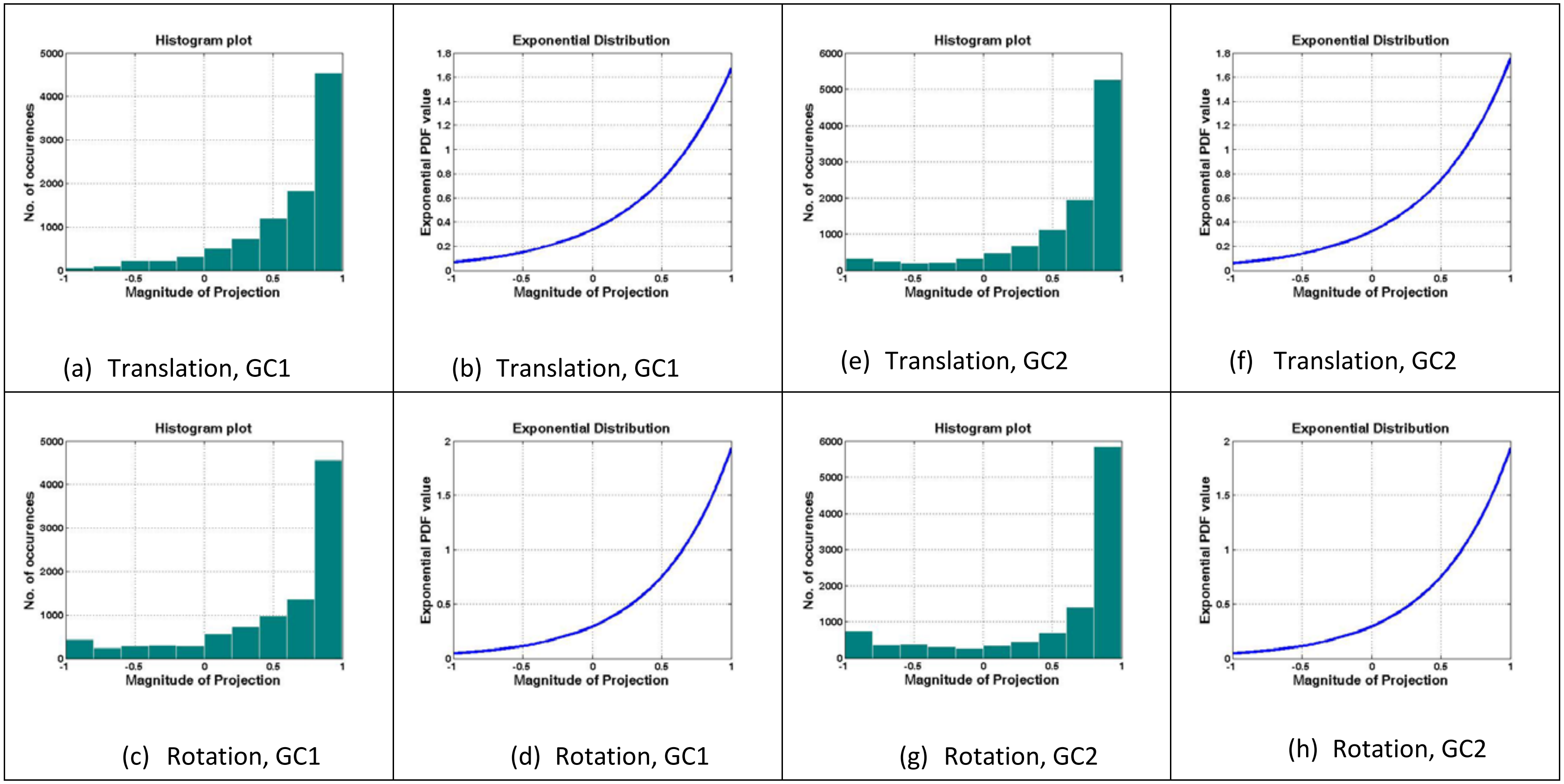
(c) Rotation, GC1

(d) Rotation, GC1

(g) Rotation, GC2

(h) Rotation, GC2

Figure 22: Histograms and approximated exponential distributions of projection scalars for the plate. GC: grasp configuration; x-axis (all figures): projection of translation and rotation vectors on reference vectors, y-axis (figures a, c, e, g): number of values in histogram, y-axis (figures b, d, f, h): value from approximated exponential distribution (Equation 53)



(a) Translation, GC1

(b) Translation, GC1

(e) Translation, GC2

(f) Translation, GC2

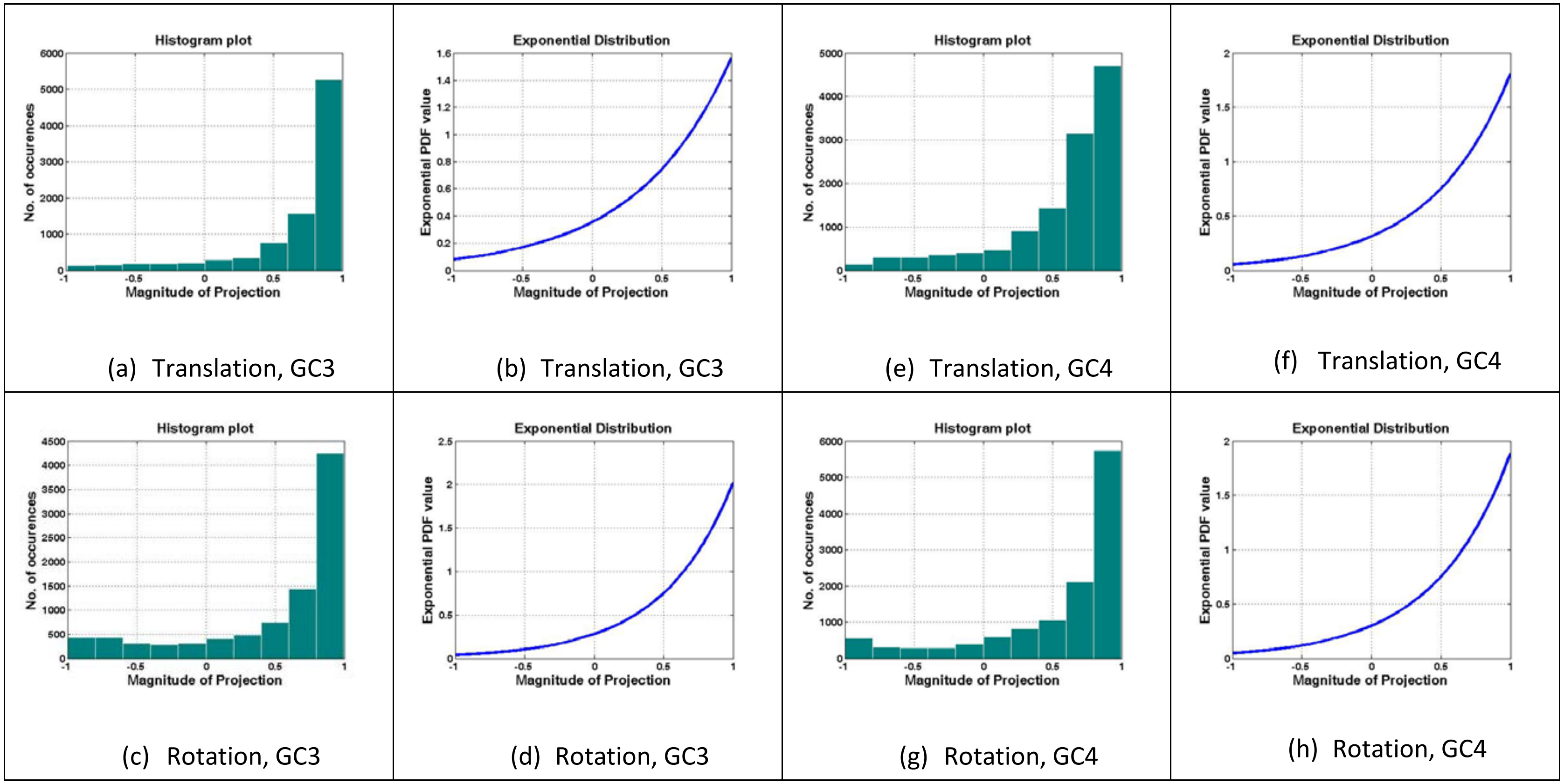
(c) Rotation, GC1

(d) Rotation, GC1

(g) Rotation, GC2

(h) Rotation, GC2

Figure 23: Histograms and approximated exponential distributions of projection scalars for GC1 and GC2 of the cup.GC: grasp configuration; x-axis (all figures): projection of translation and rotation vectors on reference vectors, y-axis (figures a, c, e, g): number of values in histogram, y-axis (figures b, d, f, h): value from approximated exponential distribution (Equation 53)



(a) Translation, GC3

(b) Translation, GC3

(e) Translation, GC4

(f) Translation, GC4

(c) Rotation, GC3

(d) Rotation, GC3

(g) Rotation, GC4

(h) Rotation, GC4

Figure 24: Histograms and approximated exponential distributions of projection scalars for GC3 and GC4 of the cup. GC: grasp configuration; x-axis (all figures): projection of translation and rotation vectors on reference vectors, y-axis (figures a, c, e, g): number of values in histogram, y-axis (figures b, d, f, h): value from approximated exponential distribution (Equation 53)

Table 3: Rate parameters and coefficients for rotation

Object	Grasp Configuration	λ_{r_j}	c_{r_j}
Bowl	1	2.5055	0.21
	2	1.6355	0.331
	3	2.9345	0.156
Plate	1	2.5344	0.202
	2	1.9924	0.277
Cup	1	1.8900	0.292
	2	1.8888	0.292
	3	1.9851	0.278
	4	1.8410	0.3

For implementing HMM, we have used logarithm of probabilities in order to avoid overflow problems in digital computers. The difference in the output probabilities of detected object and all other objects in the environment had an upper bound of 10^{200}. If no upper bound was placed, it would take very long for the algorithm to register an intention change. The backtracking window for Viterbi algorithm was 50 Viterbi steps. Higher values would slow down the intention change detection. Lower values would result in sudden changes in intention recognition of grasp configurations even due to erroneous movements from the user. Mode value of most likely state sequence was used as the detected grasp configuration. For assistance, the motion of the remote arm was amplified by two times in the desired direction and attenuated by five times in the undesired direction. In this proof-of-concept, we have manually measured the object locations and their preshape and grasp configurations for the purpose of testing. In an actual setting, we will be using a depth based vision system, which we are in the process of integrating. The complete implementation was in C++ using object oriented programming paradigm. The code was made general enough to dynamically account for variable number of objects, their poses, trained model parameters, drop-off poses, visual interface feedback components etc. All the components or implementing HMM are developed in the lab i.e. third party software tools have not been used. A schematic representation of the complete algorithm is shown in the Figure 25.

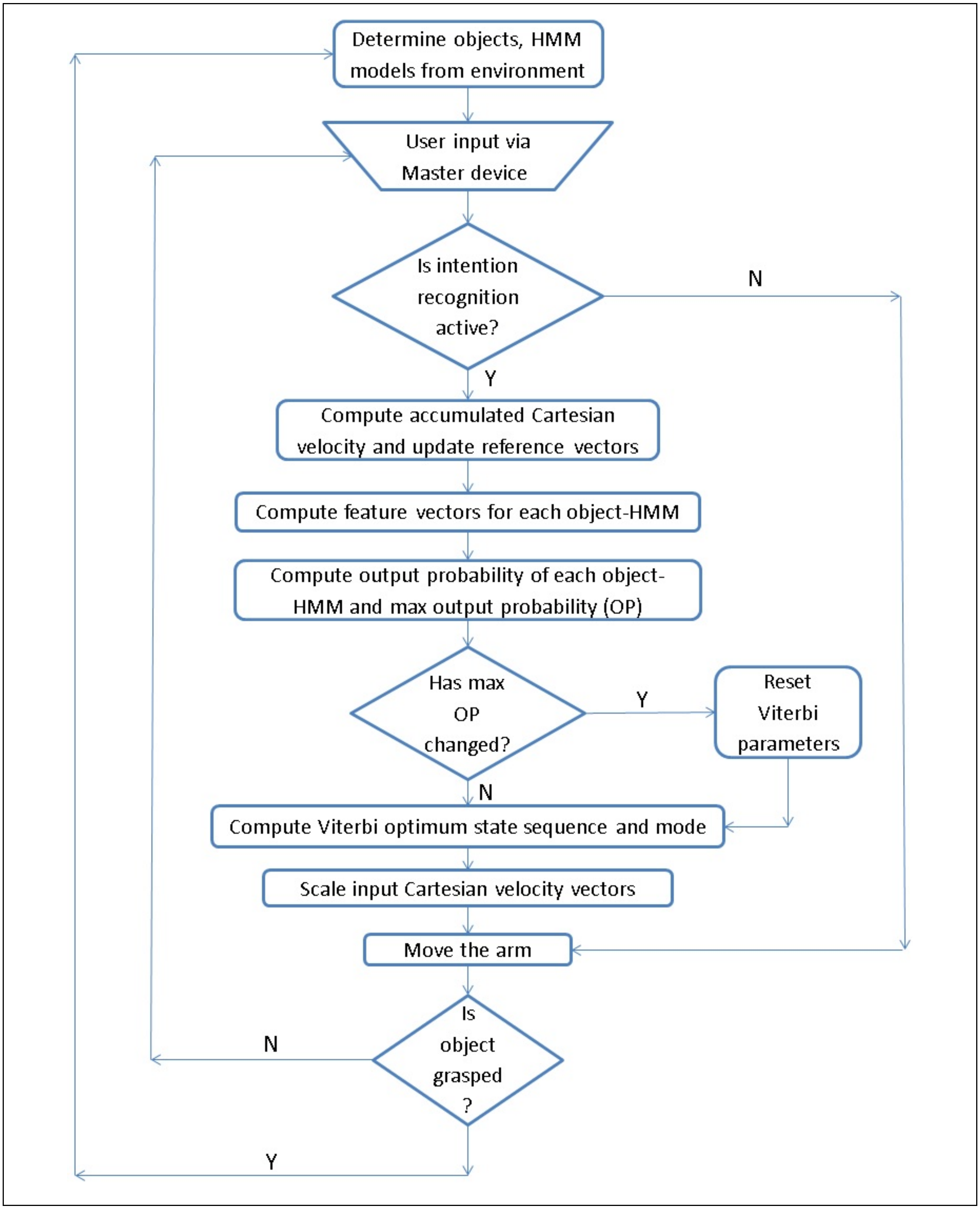
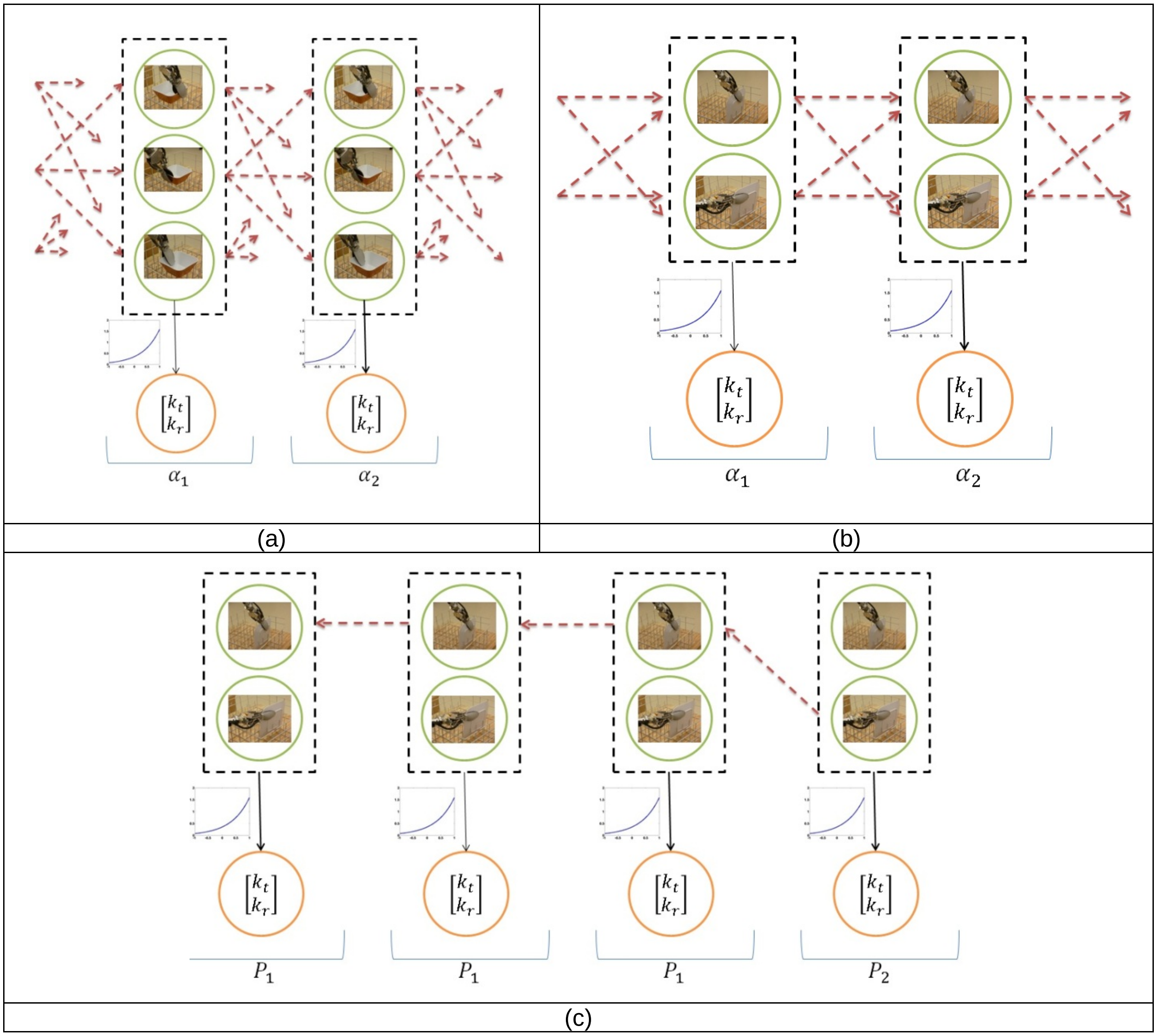

Determine objects, HMM models from environment
User input via Master device
Is intention recognition active?
N
Y
Compute accumulated Cartesian velocity and update reference vectors
Compute feature vectors for each object-HMM
Compute output probability of each object-HMM and max output probability (OP)
Has max OP changed?
Y
Reset Viterbi parameters
N
Compute Viterbi optimum state sequence and mode
Scale input Cartesian velocity vectors
Move the arm
Is object grasped ?
N
Y

Figure 25: Schematic representation of the intention recognition process



(a)

(b)

(c)

Figure 26: Pictorial representation of output probability and Viterbi algorithm in our implementation

A pictorial view of the maximum output probability and Viterbi algorithm is presented next. Figure 26(a) gives an example of the processing of the HMM for the bowl whereas Figure 26(b) gives the example of the plate. The set of circles with embedded images represent the states or the grasp configurations whereas the lowermost circles in each figure represent the observations. Currently, we have two feature vectors for each grasp configuration viz. k_t, projection magnitude due to translation and k_r, projection magnitude due to rotation. The probabilistic relationship between the states and the observations is depicted by the approximated exponential probability distribution function, shown next to the arrows connecting the states to the observations. The crisscross arrows mean summation over all the states while computing the output probability. The same arrows signify maximum transition probability while computing Viterbi optimal state sequence. $\propto_1$ and $\propto_2$represent the value of the output probability computed at every time instant. The pictorial representation for the cup, with four states, is similar. During teleoperation, Viterbi decoding is carried out on the HMM which has the maximum output probability to determine the intended grasp configuration. This is shown in the Figure 26(c). Let’s say that the plate is the object with the maximum output probability. Then, P_2, P_1, P_1, P_1etc. represent the best states at each time instant according to the Viterbi algorithm. The arrows run in the direction opposite to time, and represent backtracking.

Chapter 4: Experiments and Results

This chapter gives details of the experiments conducted and the results obtained that validate our grasp configuration identification algorithm and also validate our hypothesis, that the intention based assisted method makes grasping easier and faster. In the first section, the details of the test set-up, the subjects and the test objectives are given. The second section gives validation of intention recognition algorithm. The following section presents a comparison of intention-based assisted method with unassisted method whereas the last section gives a comparison with maximum-projection method. Each section describes experimental methodology, results and a discussion on results.

4.1 Introduction

4.1.1 Experimental Test-bed

The complete test-bed is shown in the Figure 27. The teleoperation test-bed consists of an in-house developed 7 DOF wheelchair mounted robotic arm (WMRA) [67] teleoperated using a 6 DOF Phantom Omni device. Detailed description of the manipulators was presented in Chapter 2. A parallel jaw gripper is mounted on the WMRA. The wheelchair is stationary throughout the experiments. The workspace for grasping objects is directly visible to the user. Joint velocity vectors from Omni space are mapped to the WMRA space and then Singularity-robust inverse of the Jacobian is used to compute WMRA joint velocities. Optimization criteria based on weighted least norm of joint velocities and joint limit avoidance is used to control the redundancy of WMRA, as mentioned in Chapter 2. The Omni and the WMRA control loop ran at 500 HZ and 70~80 Hz respectively but this delay does not affect task performance.

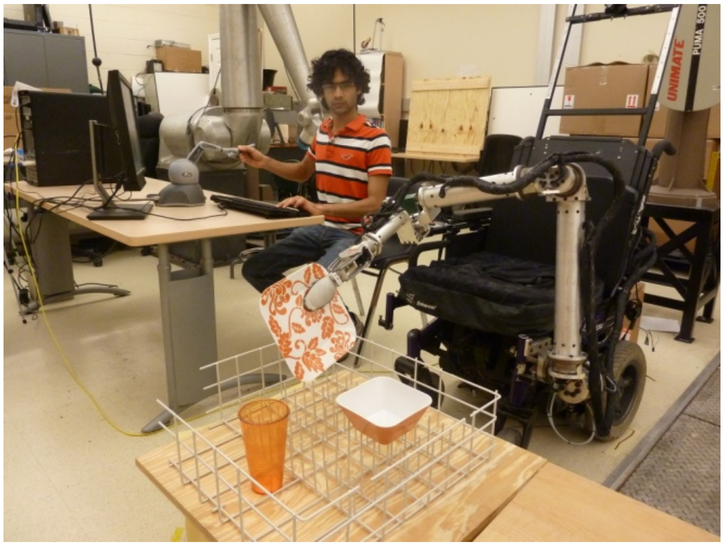


Figure 27: A subject teleoperating to grasp utensils from a dishwasher rack and lay the table (partially visible) on the right

4.1.2 Human Subjects Information

The experiments were conducted on six able-bodied human subjects and one wheelchair bound individual. All the subjects gave their informed consent through an Institutional Review Board (IRB) approved protocol. The six able-bodied subjects were all males, aged 22 to 29 years. The wheelchair-bound individual was a 24 year old female. One out of the six able-bodied individuals was a skilled robot teleoperator, having more than 5 years of experience in teleoperation. The HMM models were developed by using the training data from this subject. The other five able-bodied subjects and the wheelchair-bound individual had no experience in using a telerobotic system. These trained HMM models were used by all subjects, including the skilled teleoperator, while testing using our method.

4.1.3 Experimental Objectives

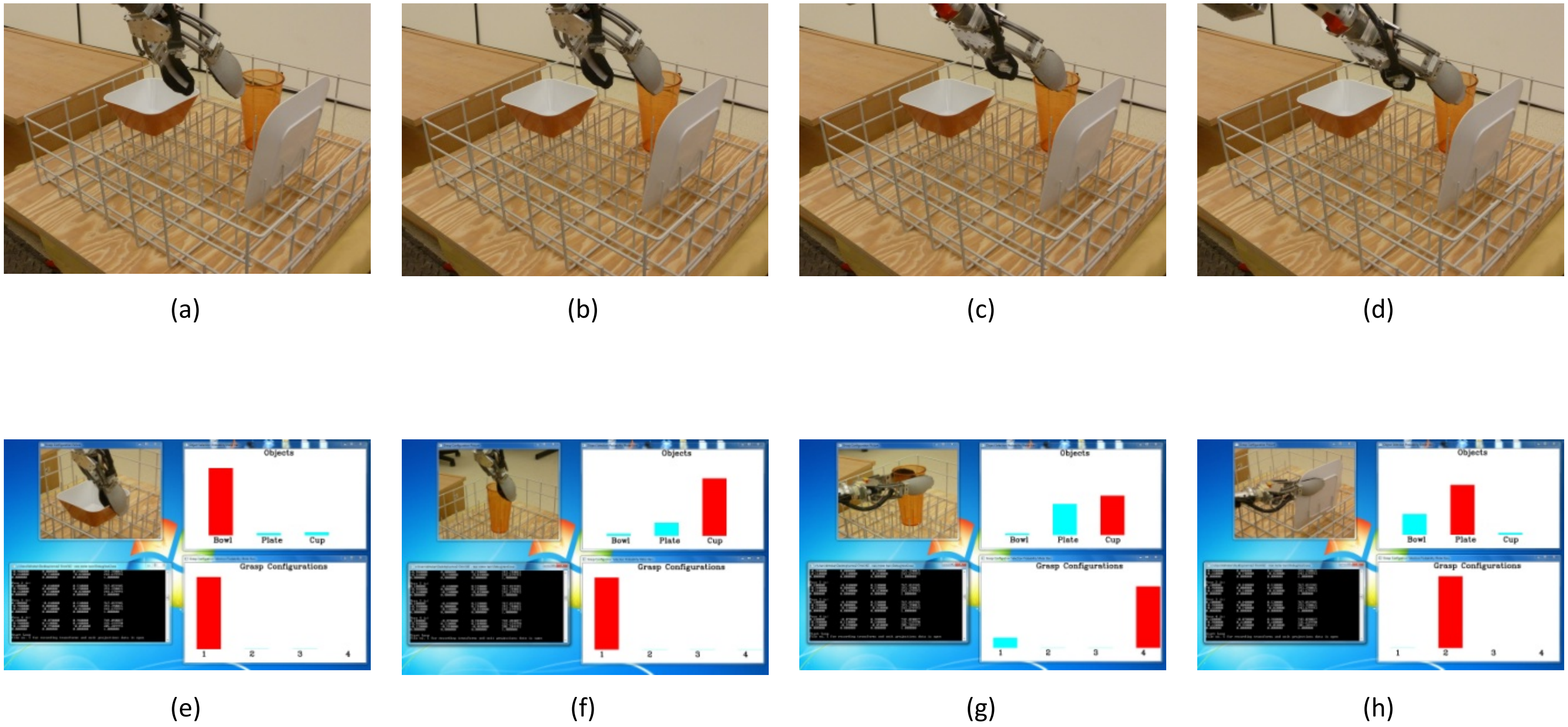
The main objectives of the experiments were to (i) validate the intention recognition algorithm (ii) compare the implemented method with unassisted teleoperation method (iii) compare the implemented method with maximum-projection method (iv) determine the accuracy of the method when objects are shuffled and (v) determine the robustness of the method with respect to intention changes. Shuffling of objects was carried out while comparison with the unassisted and the maximum-projection methods. Accuracy has been measured in a comparative form with maximum-projection method.

4.2 Validation of Intention Recognition

In order to validate our intention recognition algorithm, the subjects were asked to grasp from various configurations. Different arrangements, in terms of the pose, of the three objects on the dishwasher rack was used i.e. the objects were shuffled around and oriented differently each time. The output on the computer screen in front of the subjects (refer Figure 19) was used to determine if the algorithm was successfully able to detect the intention and predict the desired object and the grasp configuration. Observing the assistance function scale the motion of the arm gave another form of verification. Videos of the subjects executing the grasping trials were also recorded. The results from testing with the skilled operator i.e. subject 6, are presented next.

4.2.1 Results from Validation of Intention Recognition

The images in the first row of Figure 28 are snapshots of the remote arm workspace from a single run of teleoperation performed by a subject. Those in the second row are the corresponding snapshots of outputs as seen on the screen in front of the subject. As described previously (refer Figure 19), the image on the left shows the predicted object and grasp configuration. The meter bars on the right side top window represent normalized output probability whereas ones on the bottom right side represent normalized count of Viterbi sequence. Thus, the tallest meters represent the intended object and grasp configuration, as determined by the algorithm. They are shown in red so that they can be distinguished quickly. Initially, the subject was teleoperating the remote arm gripper with the intention of grasping from B1 i.e. bowl at configuration one. This state is shown in Figure 28 (a). As seen from the Figure 28 (b), the algorithm predicted the grasp configuration correctly. The meter bar for the bowl is the tallest among objects and the one for the configuration 1 of the bowl is the tallest. As the subject made a slight movement in the forward direction, the algorithm predicted C1 i.e. cup with grasp configuration one, as the desired grasp configuration. These states are shown in Figure 28 (b) and (f). C1 has a grasp configuration similar to B1. This shows how probability due to translation p_{t_j} plays a role in Equation (56) in determining the intention. Movement in the forward direction increased the p_{t_j} associated with the observation PDF of cup and reduced the one associated with the bowl. The subject then changed her intention and desired to grasp the plate from P2 or configuration two. As the subject started teleoperating by pitching the gripper to get it to a horizontal orientation, the algorithm predicted C4. This is seen from the Figure 28 (c) and (g). We must note that C4 has a configuration similar to P2; both have a horizontal configuration for the gripper. The algorithm thus predicted the configuration C4 instead of P2 because the gripper is still closer to the cup than it is to the plate. With hardly any translation, predominant rotational movement and nearness to the cup, the observation probability associate with C4 achieves a higher value than that associated with P2. Through this snapshot, we see the role played by probability due to rotation p_{r_j} and the weight factor due to nearness p_n . We must note here that an incorrect intention was detected as the user had barely started to teleoperate toward the new target P2. Consistent movements, as we will see, will ultimately make the algorithm detect the right intention. We must also note that, the intention change as detected by the algorithm is not abrupt but takes place gradually over a period of time as the user teleoperates. Moreover the user’s motion is scaled up toward the grasp configuration leading to the detected intention and scaled down in other directions. So, when the user decides to change their intention and teleoperate to a new target, they might experience resistance in the direction that lead to the new target. Only when the correct intention change is detected by the algorithm, is the user able to align easily with the new target. Moving on with the sequence of movements in in Figure 28, a slight translation and rotation from the state in Figure 28 (c), towards P2 then makes the algorithm predict the right intention. This can be confirmed from the snapshot of the screen as shown in Figure 28 (h). This happens because the observation probability associated with P2 is higher than that associated with C4 or any other configuration, over the duration of teleoperation from Figure 28 (c) to Figure 28 (d). This is due to the high values of p_{t_j}, p_{r_j} and p_n associated with P2 observation PDF. Thus, the observation probability values and hence output probability values gradually change to give the correct prediction for the object as plate, at the instant shown in Figure 28 (d). The Viterbi sequence for the plate is started when plate is detected and it returns the configuration 2.



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 28: Results from the motion intention recognition of a teleoperation run. The figure shows how the algorithm correctly predicts the subject's intention as her intention changes

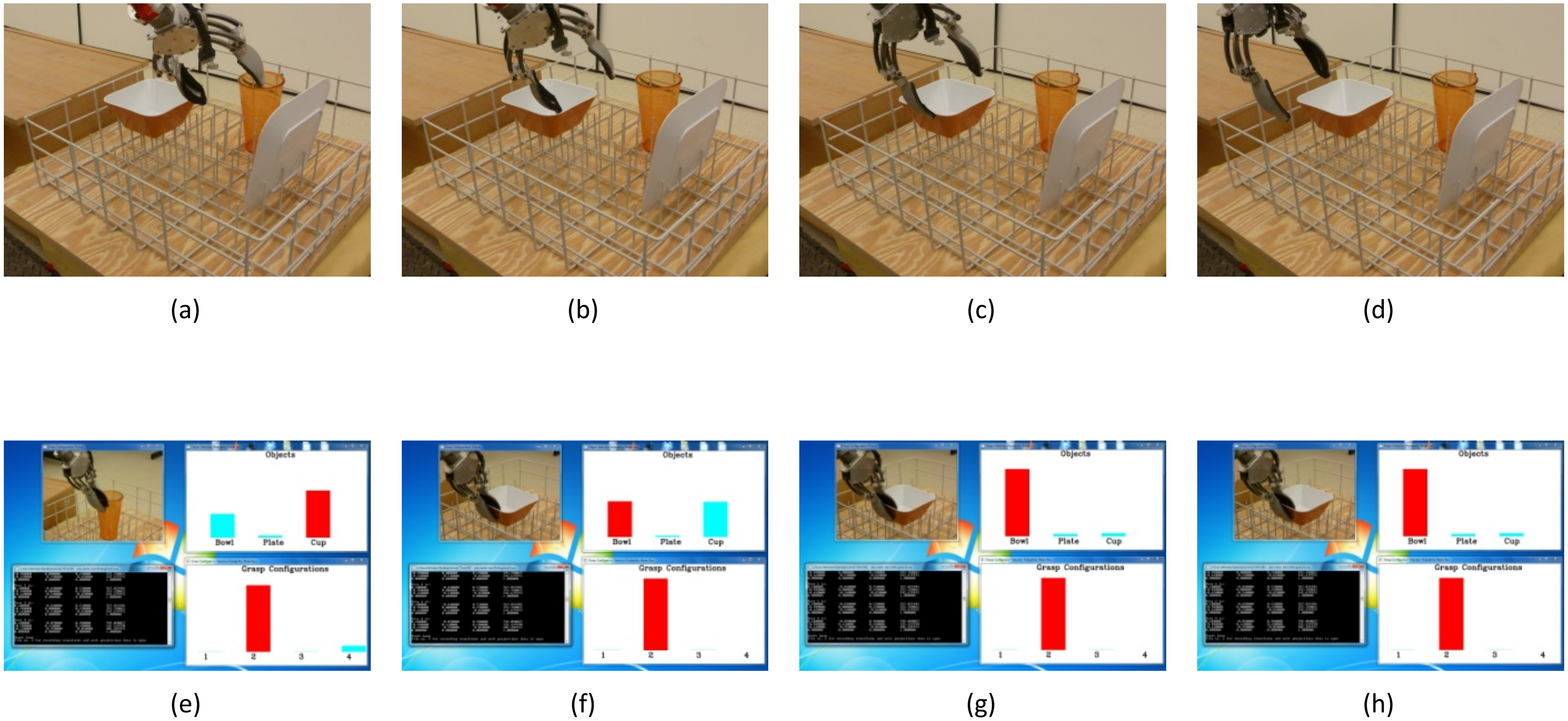


Figure 29: Results from the motion intention recognition of another teleoperation run. The figure shows how the algorithm is robust to fluctuations and at the same time correctly predicts the subject's intention

Now, we move our attention to Figure 29. The first row in Figure 29 consists of snapshots from various points in time of the remote arm and its workspace as the same subject was performing another run of teleoperation. The bottom row images are the snapshots of the computer screen as seen by the subject, at the same points in time as the images directly above them. The subject initially desired to grasp the bowl from configuration two, or B2, and began teleoperating from a point when the gripper was very close to the cup. This state is shown in Figure 29 (a). The algorithm predicted cup configuration two, or C2, as seen in Figure 29 (e). This is because the object is very near the cup, so probability due to nearness p_n is higher for the observation probability of the HMM associated with the cup than it is for the HMM associated with any other object. The probability due to rotation projections, p_{r_j}, are almost the same for the cup and the bowl and, are higher than that for the plate, the plate not having a grasp configuration similar to B2 or C2. The probability due to projection of translation vectors, p_{t_j}, are low for the HMM associated with bowl or the p_n of the bowl outweighs it, thus resulting in an overall higher observation probability for the cup. In the duration between Figure 29 (a) and (b), the subject has translated and rotated the gripper in the direction of B2. We can see from the Figure 29 (f), that according to the intention recognition algorithm, the likelihood of the user grasping the bowl has increased and that of the user grasping the cup has gone down. This is evident as the meter for the bowl is taller than the cup. The grasp configuration that the algorithm predicts is two. This is happening because the probabilities due to translation, rotation and nearness are continuously increasing the B2 and decreasing for all other configurations as the remote arm is teleoperated between Figure 29 (a) and (b). Thus, the output probability of the HMM associated with bowl gradually increases while those associated with other objects decrease. The Viterbi sequence predicts configuration two for the bowl from among configurations one, two and three, again because of high p_{t_j}, p_{r_j} and p_n for configuration two. The image of the hand grasping the bowl at B2, seen on the output screen in Figure 29 (f), is another representation of the predicted object and grasp configuration. Also, the subject faced resistance in motion when she had started to teleoperate from the state in Figure 29 (a) since the robot was assisting the user to preshape to C2. But persistent motion towards B2 led the algorithm to ultimately detect B2 as the configuration of interest.

From the state in Figure 29 (b) to (c), the user continued to rotate and translate the gripper towards B2. We can see from the output of the intention recognition algorithm in Figure 29 (g) that the likelihood that the subject will grasp the bowl has increased while that for the cup has decreased. This is because the probabilities due to p_{t_j}, p_{r_j} and p_n continue to increase for the bowl. The same is true for the Viterbi sequence. However, we must note that the time difference between Figure 29 (b) and (c) is much smaller than that between Figure 29 (a) and (b). This is because, the correct intention had been detected by the time the subject was at state Figure 29 (b) and hence the user was assisted while teleoperating from Figure 29 (b) to (c). This is in contrast to the initial resistance that the subject experienced while at the state of Figure 29 (a). The intention detected continues to predict B2 as the grasp configuration of interest as the subject teleoperates from the state at Figure 29 (c) to that at Figure 29 (d). When teleoperating through this phase, translations were predominant and there were hardly any rotational movements. Thus, the contribution of p_{r_j} to the probability computation was much less compared to p_{t_j}. As a result, the Viterbi did fluctuate in predicting the intention correctly due to errors in human translational movement. There were no fluctuations in predicting the object since the value of p_n for the bowl was much higher compared to that for other objects due to nearness to the bowl. However, the fluctuations in predicting the grasp configuration were far less than that the subject experienced when executing the task in the maximum-projection mode. In the maximum-projection mode, the intended object was determined by taking the maximum translation projection and the grasp configuration was determined by taking the maximum rotational projection. Correct object and grasp configuration was detected as the arm was being teleoperated towards B2. But, after the gripper was oriented to align with B2, and the arm was still traversing to B2, the rotational projections were negligible as the gripper had already oriented. With no definite rotational projections, the maximum-projection algorithm randomly selected grasp configurations of the bowl as the desired. This resulted in rapid fluctuations in prediction of grasp configuration and often wrong prediction of grasp configuration. Our algorithm, on the other hand, was more robust to these fluctuations. By conducting similar tests with more subjects, we confirmed that the algorithm was able to determine the correct intention every time. We tested with shuffling objects and placing them in the dishwasher at different poses each time. In all the runs, we were able to obtain correct intention recognition.

4.3 Comparison of Intention based Assistance with Unassisted Teleoperation

4.3.1 Experimental Methodology

In order to validate our hypothesis, the subjects executed pick-and-place tasks involving the three objects shown in Figure 20, in intention based assisted mode and unassisted teleoperation mode. The set-up used for the experiments is shown in Figure 27. The objects are to be picked up from the dishwasher rack and placed on a table next to it. In the intention-based assisted mode, the intention recognition algorithm identifies the object and grasp configuration of interest and assists the subject to preshape and grasp it. The assistance until preshaping is provided based on intention recognition. After this point, the subject is assisted by scaling the motion of the gripper along its forward motion axis so that the subject easily envelopes the paddles around the object to grasp it. Assistance is also provided in the place phase, but no intention is detected in this phase since each object can be placed in only one way. In the unassisted mode, there is no intention recognition and no assistance. Each subject executed a pick-and-place task four times in each mode. This way, all distinct grasp configurations were accounted for twice in each mode. For four of the subjects, the intention-based assisted mode followed the unassisted mode whereas for the other three, it was the other way round. This interchanging of the order of the modes helped nullify any learning effects due to any of the modes. It helped to prevent our results from being biased to any mode. Each assisted and unassisted mode pair is one set and so each subject executed four sets. The order of grasp configurations, in which the various objects were to be grasped, and the pose of the objects in the dishwasher was changed after each set. As a result, randomness was introduced in the tests and any improvement in task performance by a subject due to predictability in the arrangement and order of grasp configuration was prevented. The objects were also shuffled at the end of each set for each subject. Sample object placements for subject number 4 are shown in the Figure 40. In all, each subject performed 24 pick-and-place trials. Before beginning the trials, each subject was given ample time to get familiarized with the telerobotic system. On an average, each subject took around 45 minutes to get familiarized and about 4 hours to complete the tests. The subjects were given a 15 minute break after they completed the first two sets.

As the subjects executed the pick-and-place tasks, we recorded the time for grasping each object, the total time for completing each pick-and-place task and the number of Omni stylus button clicks. The number of clicks gave us an idea of the amount of master device movements and hence the amount of effort expended by each subject. This is explained as follows. Users ordinarily click the button on the Omni stylus button to engage the WMRA and then move the stylus to teleoperate. The clicking of the Omni stylus button acts as a safety check and prevents the movement of the WMRA if only the stylus is moved. However, since the workspace of the Omni is smaller than that of the WMRA, the users would click and drag the stylus until the Omni workspace limit is reached. After that they would release the button, reposition the stylus to the other end of the workspace and start teleoperating again. We call this activity as indexing. Thus, one index is a unit of subject's movement at the master. By counting the total number of Omni button clicks, we could estimate the number of unit hand movements made by the subject during a trial. This measure gave us an idea of the amount of efforts each subject puts in during a task trial.

After completing the first trial set, the subjects were asked to rate on a scale of 1 (easy) to 10 (difficult), how easy was it to execute the task in the two modes. They also were asked to provide a self-rating of the various factors from the NASA-TLX assessment [68] that contribute to operator workload. The various factors were mental demand, physical demand, temporal demand, performance, fatigue and frustration.Effort is a part of NASA-TLX assessment and this factor was replaced with 'fatigue'. This is because fatigue is more relevant to this study and, mental and physical demands are representational of effort. The definition of all the parameters was read out to the participants before they were given the scoring sheets for evaluation.

A sample NASA-TLX scoring sheet is shown in the Appendix A. After rating each factor, the subjects were asked to comparatively weight the factors. In order to weight the factors, the subjects were presented with all possible paired combinations of the factors and were asked to select the factor that they felt was dominant. Let s_i be the score of the ith factor and $w_i$be the number of times it was rated more dominant compared to other factors. The adjusted rating for each factor i is given by,

$$A_i = s_i\, w_i \tag{57}$$

$\sum_{i=1}^{6} A_i$ gives the total weighted rating for the subject. In this manner, a weighted NASA-TLX score was obtained for each subject. The surveys were presented to the subjects after the first trial set. After the ends of subsequent trial sets, the subjects were asked if they needed to modify their feedback, in which case they were given an opportunity to. We observed that all the subjects were satisfied with their original feedback and did not modify it at the end of subsequent trial sets. An Analysis of Variance (ANOVA), on the quantitative as well as the qualitative data, was carried out to statistically compare the two modes.

4.3.2 Results from Comparison with Unassisted Method

As mentioned earlier, we compared our method with the unassisted teleoperation method by comparing the time to grasp, time to complete the pick-and-place task and the number of human hand movements. We also present qualitative results which compare the ease-of-use and the total workload experienced by the subjects while executing the task in the two modes. These results are based on the experimental procedure described In Section 4.1.

Figure 30 shows the average time it took for each subject to grasp an object using our method and the unassisted teleoperation method. The average was taken over all the 24 trials that each subject executed except for subject no. 7, who represents the wheelchair-bound individual. Subject 7 suffered from muscular dystrophy. To prevent any physical stress on the subject, long hours of continuous manipulation with the Omni by the subject were avoided. Subject 7 executed only one set as against 4 sets executed by all the other subjects. Thus, subject 7 executed 3 trials each in the two modes, totaling six pick-and-place trials.

We must also note that the results presented in Figure 30 compare the time taken by each subject to grasp the object i.e. it only takes into account the time from the start of a pick-and-place trial to the point when the subject closes the gripper paddles to complete the grasp on the first object and, for other objects in the pick-and-place trial, it is the time from dropping an object to when the paddles are closed to grasp the next object. A software timer, integrated into the program, was used to measure the time, thereby relieving the experimenter from holding a stop-watch and measuring and allowing him to focus on the subject executing the task. The software timer was started and stopped automatically when the object was placed or grasped, which in turn was detected when the subject pressed the keyboard key to command opening and closing of the gripper.

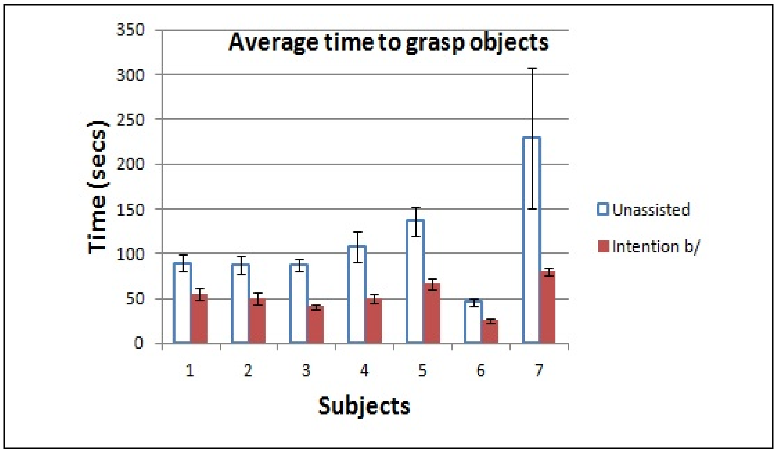


Figure 30: Average time taken by each subjects to grasp an object. 'Intention b/' is a short for intention-based asssited method.

As we can see from the plot, all the subjects were able to complete the grasp much faster using our method than that unassisted method. Subject 6 is the skilled teleoperator, who also trained the HMMs. We can see that subject 6 was able to execute the grasping task quicker in both modes compared to all the other subjects. The error bars on the plots are standard errors, which were computed as follows:

$$\text{Standard error} = \frac{\text{Standard Deviation}}{\sqrt{\boldsymbol{number\ of\ samples}}} \quad (58)$$

We can see that the standard error for the subject 7 for the unassisted mode is much larger compared to other subjects. One reason could be that subject 7 performed only 3 trials in the unassisted mode compared to 12 trials performed by other subjects. The disability of subject 7 could have been a cause for huge variation in time but this is not confirmed. We also see that the same subject had very less variation while performing the task using our method. This shows that our method makes the grasping task not only quicker but also with fewer variations for a person with disability. This, however, needs to be confirmed.

Figure 31 shows the average time taken by each subject to complete a pick-and-place trial, which consisted to picking and placing three objects from the dishwasher rack on to the table. Again, a software timer integrated into the algorithm recorded the time. The average is taken over all the 8 pick-and-place trials for the first six subjects and for the one pick-and-place trial for the subject 7. Error bars represent standard error. There are no error bars for subject 7 since she executed only a single trial. We see that all the subjects were able to execute the pick-and-place task much quicker using our method than the unassisted method. The trend of average times among the subjects is also similar to that seen in Figure 30. Subject 6, the skilled teleoperator was able to complete the task much quicker in both modes compared to all the other subjects whereas subject 7 took the longest time for the unassisted method.

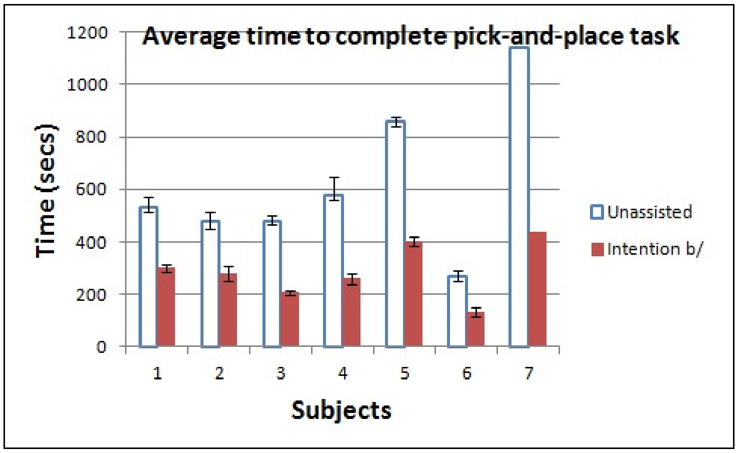


Figure 31: Average time taken by each subjects to complete a pick-and-place task trial. ‘Intention b/’ is a short for intention-based asssited method.

Figure 32 gives the average number of unit hand movements made by the subjects when executing a pick-and-place trial. Again, a single pick-and-place trial consisted of picking and placing three objects from the dishwasher rack onto the table. From the figure, we can see that the subject made lesser hand movements while executing the task using our method than the unassisted method. Error bars are again standard errors and they are not shown for subject 7 since she executed only one trial. We also see that the trend in this figure is not the same as that in Figure 30 and Figure 31 i.e. subjects that took lesser time do not necessarily click the Omni stylus button lesser number of times. This could be due to several factors. One, the amount of unit hand movement may not be the same for all subjects i.e. the amount of movement between the engagement of the stylus button and its release may be different for different subjects. Second, some subjects may take time to think before they actuate a movement. This adds to the time taken by the subjects to grasp or complete the pick-and-place task. Thus, they may be slower than other subjects but may have the same, or more number of stylus button clicks. Whatever factor may be responsible, it is still consistent for the same subject. Thus, the number of stylus button clicks gives a fair comparison of the amount of movements made, and hence the amount of physical effort put in by each subject while executing the task in the two modes. Based on ANOVA, savings in time and human efforts were found to be statistically significant, at 95% confidence level ($p < 0.001$).

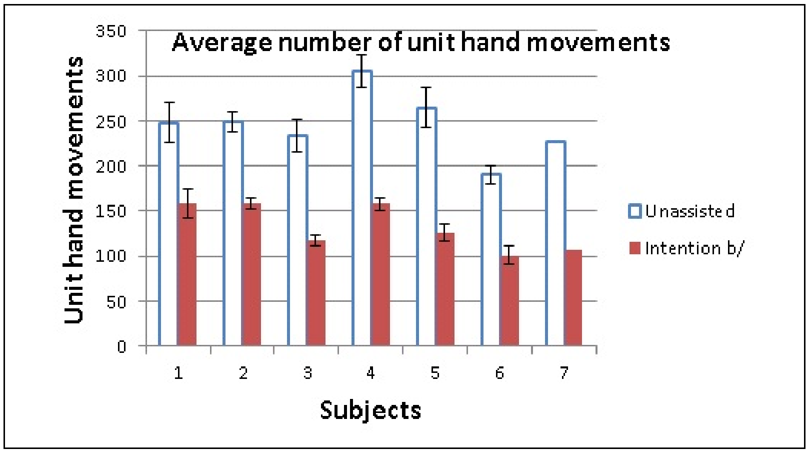


Figure 32: Average number of unit hand movements while executing a pick-and-place trial. ‘Intention b/’ is a short for intention-based asssited method.

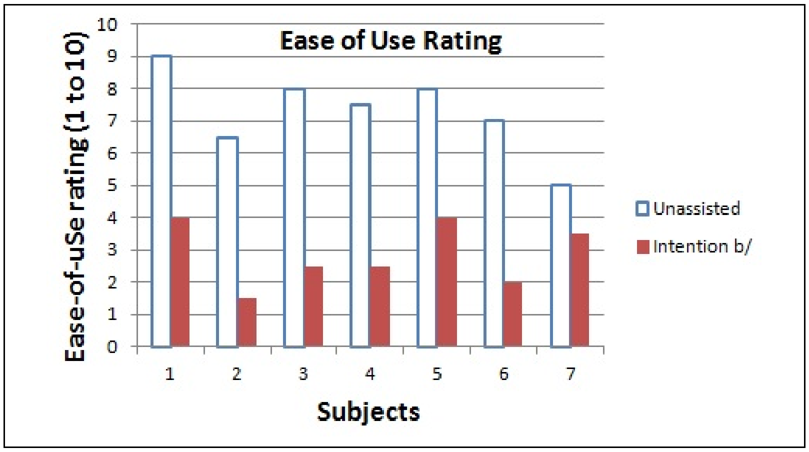


Figure 33: Ease-of-use rating for executing the grasping task from survey. Rating is from 1 (easy) to 10 (difficult). ‘Intention b/’ is a short for intention-based asssited method.

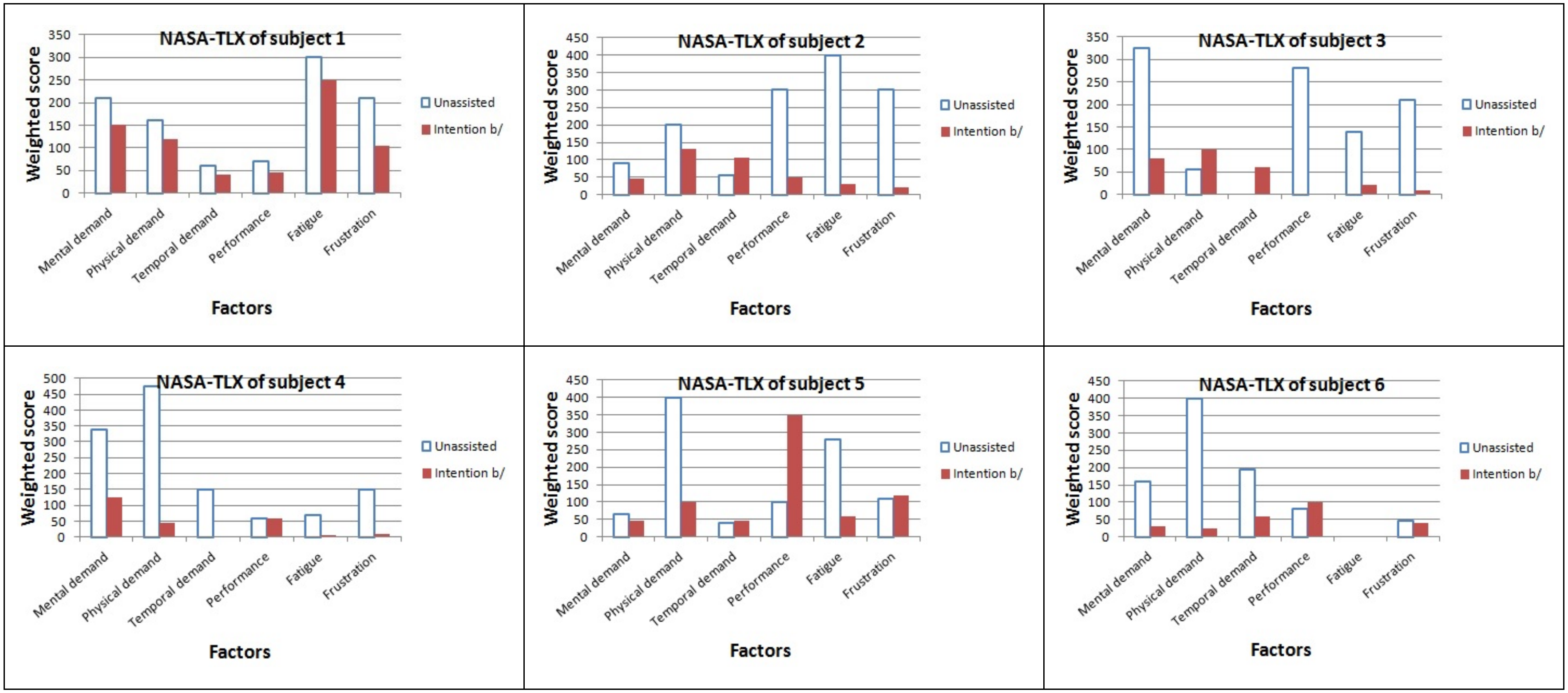


Figure 34: Weighted NASA-TLX ratings for different factors for each subject from survey at the end of grasping task. ‘Intention b/’ is a short for intention-based asssited method.

We now present qualitative results which were obtained by conducting a survey at the end of the first test trial. Figure 33 gives the ease-of-use rating in performing the task in unassisted and intention-based assisted modes. The two extremes of the scale were 1, which was very easy and 10, which was very difficult. We observe that for all the subjects, it was easier to execute the tasks using intention based assisted method than the unassisted method.

Next, we present the NASA-TLX ratings that represent another set of qualitative results, gathered from the subjects through the survey conducted, after they completed the first set of trials. As mentioned earlier, NASA-TLX is a task load index i.e. it represents the task load or the workload experienced by a subject as they are executing a task. Figure 34 represents the weighted ratings for each factor as assigned by the first six subjects whereas Figure 35 gives that assigned by subject 7. We can see from the Figure 34 and Figure 35 that different subjects marked different factors as being dominant in the overall workload they experienced while executing the task. According to some subjects, certain factors did not contribute to their workload at all. For example, according to subject 2, performance, fatigue and frustration were dominant contributors to the workload whereas according to subject 4, physical and mental demands were the dominant factors. It is seen that temporal demand is a minor factor according to all the subjects i.e. all the subjects were of the opinion that the pace of the task, in both the modes, was generally slow and that they did not feel rushed while executing the task. We also see that most of the factors rated high for the unassisted mode compared to the intention based assisted mode for the same subject

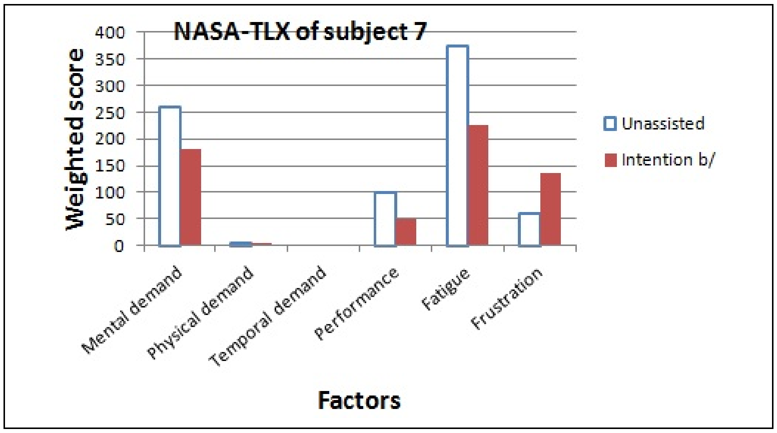


Figure 35: Weighted NASA-TLX rating for all the factors for subject 7 from survey at the end of grasping task. ‘Intention b/’ is a short for intention-based asssited method.

A few exceptions are physical and temporal demands experienced by subject 3, performance and frustration experienced by subject 5 and, frustration experienced by subject 7. In the case of subjects 3 and 5, the scores for all the factors were higher for the unassisted mode but the weights made the weighted score higher for these few factors. However, subject 7 did experience more frustration with the intention based assisted method. The reason for this was the difficulty in teleoperating away from the currently detected object and grasp configuration to the desired one. The subjects need to make repeated movements towards the desired configuration when the detected configuration is not the desired. If the subjects make movements that do not lead to the desired configuration or do not translate and rotate the gripper simultaneously, they experience difficulty in manipulating out of the currently detected configuration. This occurs because the scaling is provided in the direction of detected configuration and the motion is attenuated in other directions. It is similar to a user being trapped in local minima. As we have seen, this does not affect the time taken by the subject to grasp an object but does annoy some subjects, leading to frustration. We must also note that subject 7 has zero weighted score for temporal demand. This does not necessarily mean that subject 7 did not experience any temporal demand. The subject may have assigned a score to the factor but did not give it a higher weight compared to any other factor i.e. the weight for this factor was zero. As a result, the weighted score is zero. Thus, NASA-TLX does not represent the independent influence of a factor but it represents a comparative influence.

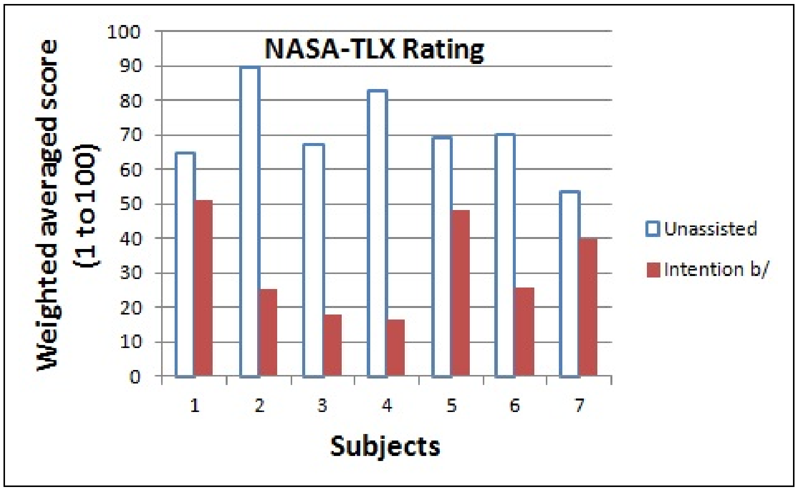


Figure 36: Weighted averaged NASA-TLX rating for all the subjects. ‘Intention b/’ is a short for intention-based asssited method.

Figure 36 shows the weighted averaged NASA-TLX ratings for all the subjects as determined from Equation (57). We see that the ratings are high for all subjects for the unassisted mode compared to the intention based assisted mode. Thus, the odd few ratings for some subjects in which the weighted rating for the intention based assisted mode was higher, does not seem to affect the overall score.

In summary, the subjects found it much easier to execute the task using our intention based assisted method than the unassisted method. All the subjects preferred executing the preshaping and grasping task using our method. This can be confirmed from the ratings that they provided as well as from their feedback that we recorded in the attached video. According to one subject, he had to put in fewer thoughts into the task for preshaping to the desired configuration. The robot performed the movements for him while he made small movements. He added that he did not have to perform complicated maneuvers for orienting the gripper correctly, but only had to translate in the right direction and the robot would help in aligning. The unassisted method was tougher and the subject said that he had to think more while executing the task using this method. This is consistent with the subject's feedback from Figure 34 (subject 3). He also estimated to make more movements to accomplish some preshapes as he thought that his movement was not perfect, but was surprised when the gripper completed the preshape quickly with him having to make very few movements. The subject mentioned that the initial few movements are more difficult when the algorithm is in the process of detecting the correct intention. This is due to the scaling in the direction of detected intention and attenuation of motion in other directions. We must also note that subject 3 performed better in terms of speed compared to all other subjects. This can be seen from Figure 30 and Figure 31. The only subject better in terms of speed than the subject 3 was the skilled teleoperator (subject 6). Subject 3 was quick to lean the ability to translate and orient simultaneously which helps the intention recognition to detect the correct intention faster. Another subject commented that performing the task using the unassisted method involved a lot more thinking and, there was a lot of stress in figuring out angles of movement to pick up objects. He further added that with intention based assisted method, it seemed that the gripper performed most of the orientations on its own and it was very convenient to grasp the object.

The following observations were also made during testing. Subject 1 and 2 often got confused in the positive and negative pitch and yaw movements of the end-effector. Such a behavior was not seen for the remaining subjects. All the novice subjects tend to teleoperate along individual axis i.e. horizontally or vertically and hence, do not follow the shortest path that leads to the desired configuration. This produces a delay in the time to detect the correct intention. Such a behavior was unexpected because both the input and the output manipulators are free-space 6 DOF manipulators. Subject 2 also had a peculiar way of teleoperating to the desired target pose; the subject made the end-effector follow the edges of the plate to switch from grasping from the side/top to grasping from the top/side. The subject made use of the symmetry and edges of the plate. This may be due to the fact that edges and symmetry of objects gives a reference to teleoperate with respect to. This may also because the subject has never teleoperated before and it may reduce with time as the subject gets used to teleoperation. Some subjects preferred to orient the gripper to the desired target first and then translate whereas others preferred to teleoperate the other way round. Generally, most subjects kept switching between only-translation and only-rotation motions until the gripper was aligned to the target pose. Some subjects intentionally deviated from the desired trajectory in order to approach the desired grasping configuration from a different direction. This may be because of an obstacle or because they found approaching in from another direction more convenient. For example, subject 2 teleoperated the gripper vertically up so that rotations can be made without any collisions with the objects. Once fairly satisfied the subject would then start approaching the target. Unintentional deviations were also observed in cases where subjects would deviate in translation when they were focused on getting the orientation close to the desired and the other way round. In all these cases, the algorithm detected incorrect grasp configurations as the desired, when deviations were occurring. None of the user seemed annoyed due to pressing the Omni stylus button for toggling between intention based scaling and gripper forward-axis scaling when approaching to grasp an object. Also, constraining the motion in the gripper forward-axis after preshaping proved inconvenient as the subjects were unable to change the direction of translation and were bound to the direction they selected.

A skilled teleoperator translates and rotates simultaneously and moves along more than a single DOF. A skilled operator tries to execute a trajectory along the shortest path to preshape over the desired configuration. Our method recognizes intention of novice users correctly and quickly in spite of the disparity in their movement pattern and style, compared to the skilled user. This is one of the major benefits of our method.

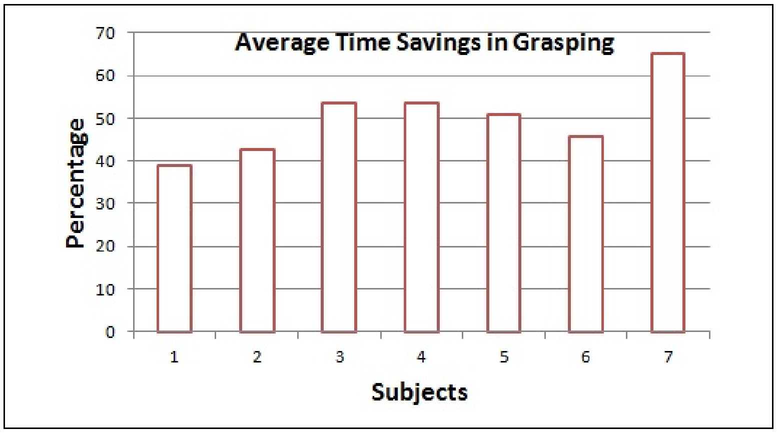


Figure 37: Average savings in time to grasp an object, for each subject

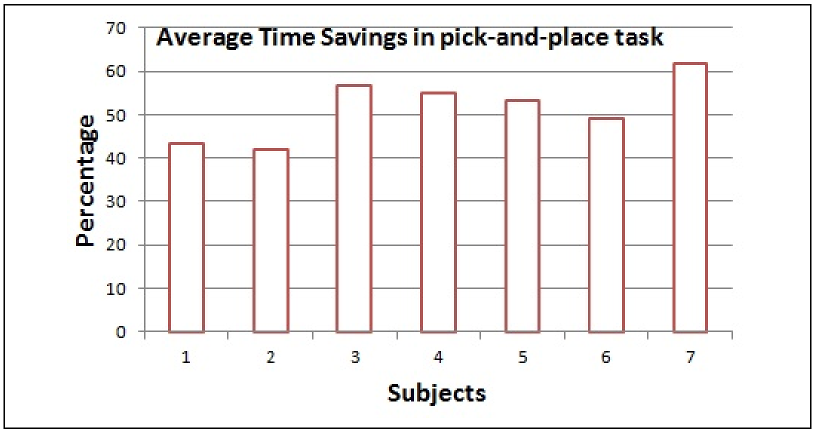


Figure 38: Average savings in time to pick-and-place three objects, for each subject

Figure 37 and Figure 38 show the average savings in time in our method over the unassisted method for each subject, while grasping a single object and while executing a complete pick-and-place task involving three objects. We can see from the figures that the pattern among all the subjects is same for the grasping and the pick-and-place task except for subject 2. It seems that the subject 2 took longer than expected to complete the pick-and-place task. Also, maximum amount of saving obtained is for subject 7. It was found to be approximately 65% for the grasping task and approximately 62% for the pick and place task. This is promising since our method is intended for use by wheelchair-bound individuals. The savings in time for the skilled teleoperator (subject 6) was found to be approximately 46% for the grasping task and approximately 49% for the pick-and-place task. This was not expected since the skilled teleoperator is adept in teleoperating in the unassisted mode.

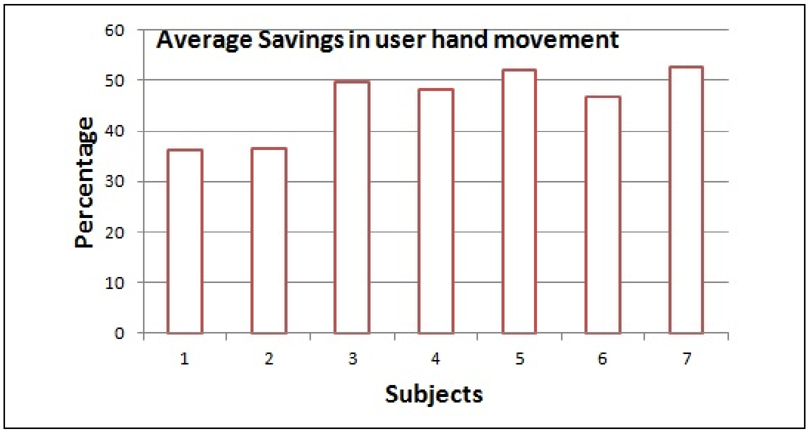
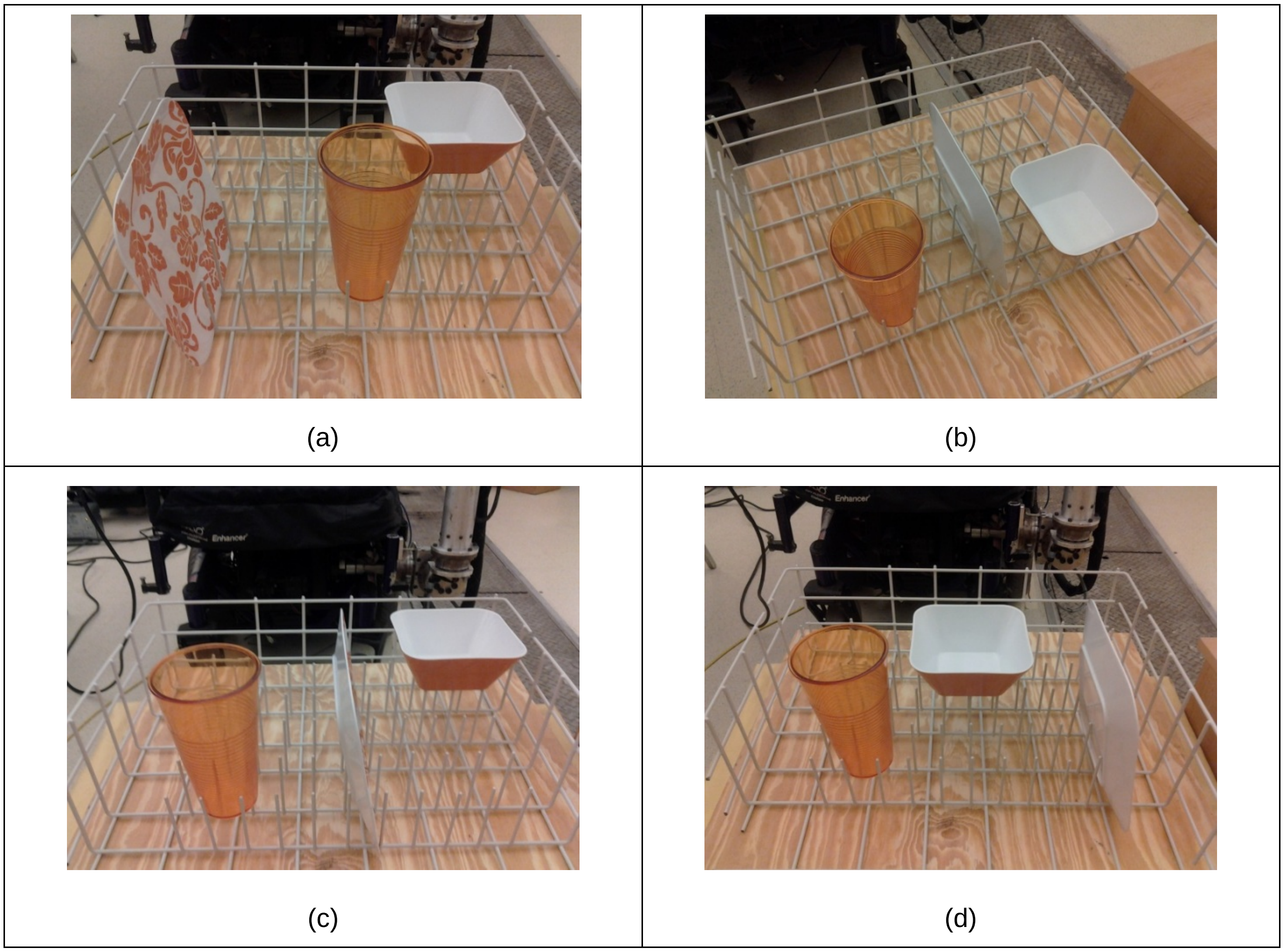


Figure 39: Average savings in number of unit hand movements for each subject for pick-and-place task


(a)

(b)

(c)

(d)

Figure 40: Different arrangements of objects in the dishwasher rack for the four sets of test trials subject 4

The average savings in time over all subjects to grasp an object was found to be approximately 50% and that for the pick-and-place task for three objects was found to be approximately 51%. Thus, the subjects could perform the grasping and pick-and-place task at more than double the speed using our method, compared to the unassisted method. The percentage savings in number of unit hand movements, across all subjects, was found to be approximately 46% for the pick-and-place task.

4.3.3 Results from Shuffling the Objects Around

As mentioned earlier, the arrangement of the objects was changed at the end of each trial set i.e. the objects were shuffled around on the dishwasher rack. This was done to serve two purposes. One was to eliminate any learning effects from repeated grasping of a particular object located at the same point on the dishwasher. The other objective was to test if shuffling of objects had any effect on the accuracy of the object and grasp configuration algorithm. Sample object placements for the four trial sets of subject number 4 are shown in the Figure 40.

It was observed that shuffling the objects around had no effect on the accuracy of the algorithm. This is because an HMM is associated with every object. Changing the pose of an object, changes the reference vectors to that object and the feature vectors of the HMM are computed accordingly. As long as the pose of the object is known fairly accurately, the feature vectors and the ensuing observation probabilities will be accurate. This will result in accurate object and grasp configuration recognition. Thus, the HMM associated with an object moves with it to the new pose. This makes the methodology suitable for unstructured environments. Shuffling of the objects was carried out in a similar way for all the other subjects i.e. the objects were randomly shuffled at the end of a trial set. Objects were also removed from the environment and it was observed that the accuracy of intention recognition did not change.

4.3.4 Time to Detect Intention

Even though the right object and grasp configuration was detected every time, in all the 12 grasping trials the detection was, not immediate i.e. it took certain amount of time for the algorithm to detect the right intention. This time depended on the skill level of the subject and on the arrangement of objects. More the number of objects between the end-effector and the desired object and poorer the skill level of the user, more the time it took to detect the right intention. These times are depicted in Figure 41. They are the times for each subject averaged over all the 12 grasping trials.

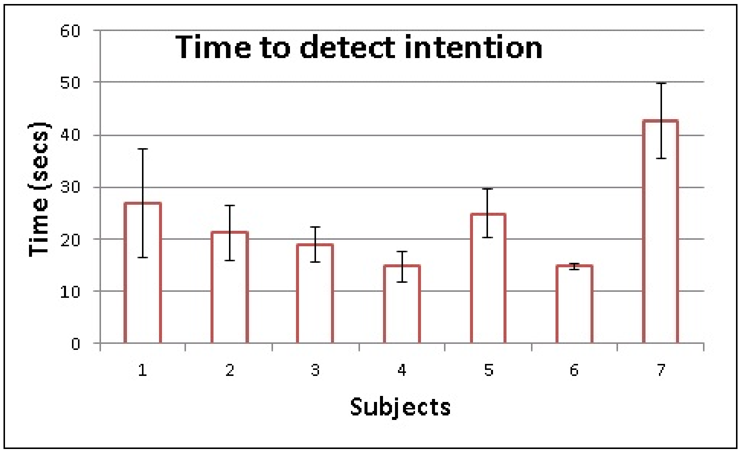


Figure 41: Average time taken by the algorithm to detect the correct intention

4.4 Comparison between Intention Based Assistance and Maximum-projection Method

The maximum-projection method is the one in which the maximum magnitudes of projection are used to determine the object and grasp configuration of interest. The object with the grasp point having the maximum projection of the end-effector translational vector, from among all grasp points of itself and all other objects, gives the object of interest. The grasp point from the selected object having the maximum projection of the end-effector orientation vector, from among all grasp points of that object, gives the grasp configuration of interest. The projection magnitudes are compared after averaging them over the last 100 readings. This is so that errors due to noise or human movement are reduced.

4.4.1 Experimental Methodology

For quantitative comparison, intention recognized by the system, in the two modes, over the length of the trajectory during each trial, was recorded. This would give an idea of accuracy of the two methods since the intention of the subject is known to the experimenter. Total time in the two modes for execution of the complete preshaping task was also recorded to give an idea of the speed of preshaping task execution.

4.4.2 Results

The set-up for subject 2 and subject 6 in the preshape trials over a single object are shown in Figure 42. All three images represent the same set-up from different angles. The subjects were asked to teleoperate to preshape over B1 P2 and C2, starting from the home position (Figure 42 (a) and (b)) each time.

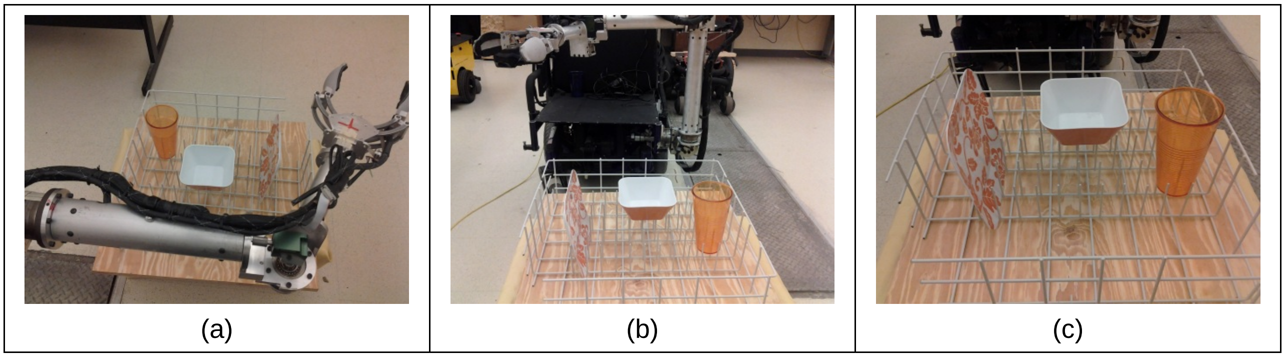
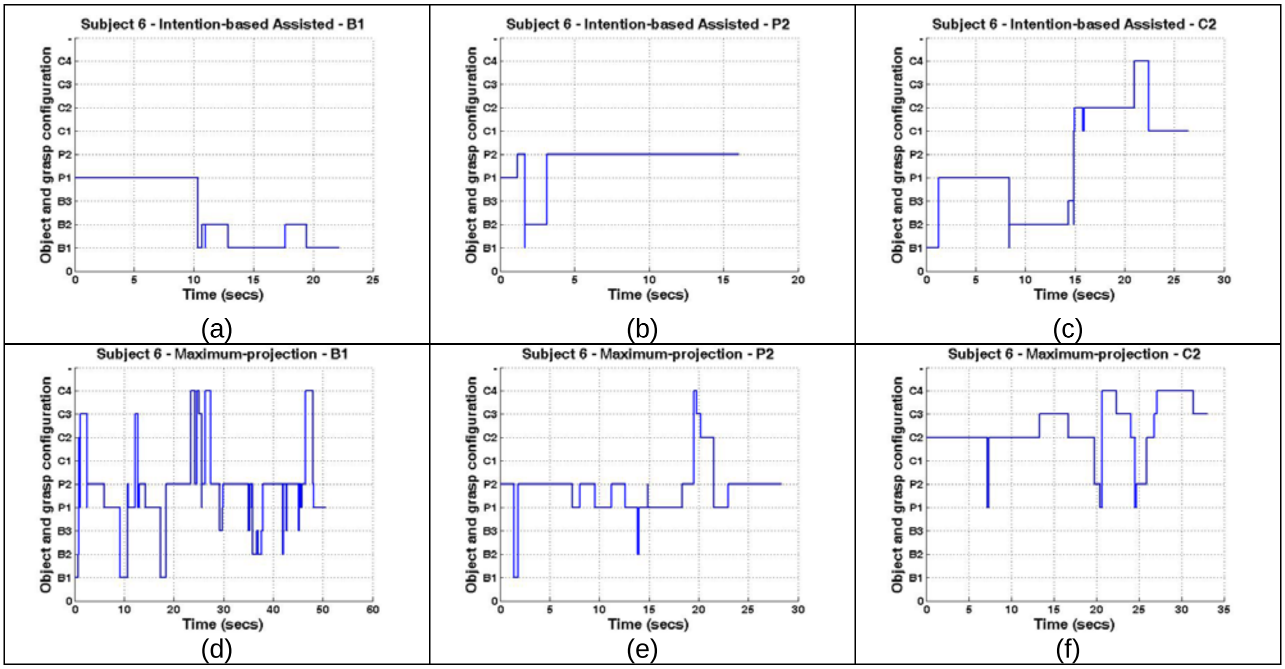

(a)
(b)
(c)

Figure 42: Three views of arrangement of the objects in preshape trials for comparison with maximum-projection method

The results of intention recognition for subject 2 over the length of each trajectory are shown in Figure 43. As seen from the figure the intention recognition in the implemented method is more consistent and accurate than the maximum-projection method. For the trial represented by Figure 43 (a), the subject intends to preshape over configuration 1 of bowl using implemented method. The figure shows that the algorithm detects P1, or configuration 1 of plate as the intention, initially. This is because P1 has a similar grasp configuration as B1 and P1 is nearer to the end-effector. As the subject continues to teleoperate to B1, the intention changes and remains more or less consistent. It needs to be determines as to why the intention jumps to B2 occasionally. Comparing this with Figure 43 (e), we see that there are too many fluctuations and the method determines wrong intention consistently. This could be due to the errors in user motion as the user is teleoperating. We also see that fluctuations for the case of B1 are more than that for P2 (Figure 43 (e)) and C2 (Figure 43 (f)). The reason for this could be that the bowl is placed in between the cup and the plate. This causes the intention recognition to jump in between all the objects. On the other hand, intention recognition for objects that are on the outer periphery may be less since there is no object on at least one side.



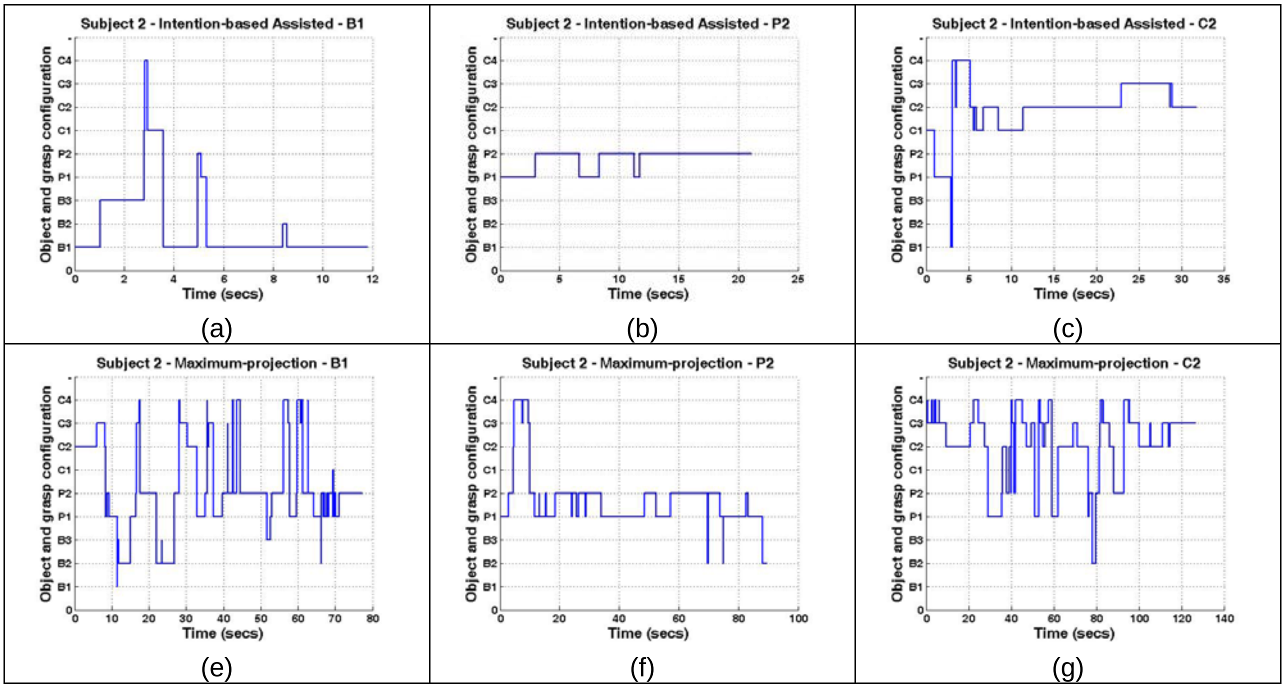
(a) (b) (c)

(d) (e) (f)

Figure 43: Intention recognition over the length of the trajectory for subject 6. (a), (b) and (c) represent our method whereas (d), (e) and (f) represent the maximum-projection method

From Figure 43 (b), we see that the algorithm detects the intention correctly except for a few fluctuations in the beginning of the trial. These could have been caused due to deviations on part of the subject. On comparing with the maximum-probability method, we see that the intention recognition fluctuates between P1 and P2 for the most part of the trajectory. This also goes to the show that the maximum-probability method is able to determine the object somewhat accurately but fails to determine the correct grasp configuration consistently. From Figure 43 (c), we see that the implemented method first detects P1, then B2 and finally C2. This is because plate is near to the starting point of the end-effector and the subject starts with a predominantly pitch motion. Then as the subject teleoperates, B2 is detected since it has configuration very similar to C2. Finally, C2 is detected. In the end, we also see fluctuation to C4 and C1. This was caused as the subject overshot the desired grasp configuration. Comparing with the maximum-projection method (Figure 43 (f)), we see that initially the correct intention is detected consistently since the subject translates and orients towards C2. Here, the intention recognition is even better than our method. However, as the user fine-orients and fine-positions in the end, fluctuations are experienced. This is because rotations are negligible when the subject is fine-positioning and translations are negligible when the subject is fine-orienting. These negligible magnitudes are like noise that fluctuates. The unit vectors in the direction of these magnitudes thus fluctuate rapidly and the maximum could be any of the possibilities. We also see that the time taken by the subject to execute the task in maximum-projection method is almost double than that taken using our method for preshaping over B1 and P2.

Figure 44 represents the intention recognition over the length of the trajectory for subject 2, as the subject was asked to preshape over B1, P2 and C2 in the two modes, our method and maximum-projection method, each time starting from the home position. The arrangement of objects and starting position of the arm is shown in Figure 42.


Subject 2 - Intention-based Assisted - B1

Subject 2 - Intention-based Assisted - P2

Subject 2 - Intention-based Assisted - C2

(a)

(b)

(c)

Subject 2 - Maximum-projection - B1

Subject 2 - Maximum-projection - P2

Subject 2 - Maximum-projection - C2

(e)

(f)

(g)

Figure 44: Intention recognition over the length of the trajectory for subject 2. (a), (b) and (c) represent our method whereas (d), (e) and (f) represent the maximum-projection method

We see that subject 2 experiences more overall fluctuations than subject 6 since subject 2 is unskilled. Also, the fluctuations experienced by subject 2 while executing the task using our method are far less than those experienced in the maximum-projection method. The subject experiences some fluctuations while preshaping to B1 using our method in the first half of the trial (Figure 44 (a)). The second half shows consistent intention detection except occasional fluctuations. Compared to this, the fluctuations are high in number when executing the task in maximum-projection method (Figure 44 (e)). Similar results are seen for C2 (Figure 44 (c) and (f)). While preshaping to P2 (Figure 44 (b)), the intention recognition is more consistent compared to B1 and C2 in our method (Figure 44 (a) and (c)). Figure 44 (e) and (f) show that the maximum-projection method is able to detect the object of interest but fails to detect the grasp configuration of interest. We also see that maximum-projection method takes 4 to 6 times longer than our method. Due to fluctuations, the subject was often guided along the unintended trajectories often. This led to extreme frustration. The most difficult part of the task was fine-aligning in the end. Here magnitudes of motion are negligible and fluctuations are the maximum.

We now present the results of the experiments where the subject was asked to preshape to another grasp configuration while she was in the process of preshaping over another. The set-up is shown in Figure 45. We must note that this arrangement was different than that in the experiment of preshaping over a single grasp configuration, just described. Thus, this experiment not only measured the robustness of our method to intention change but also robustness to shuffling objects around.

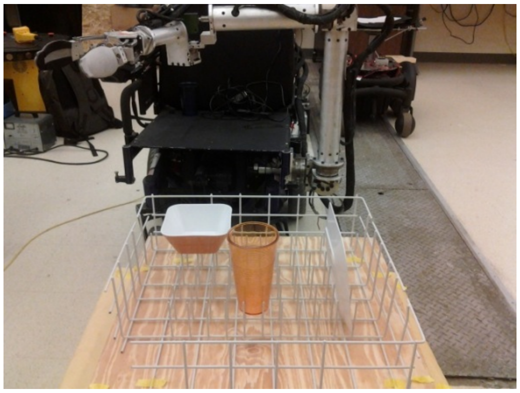
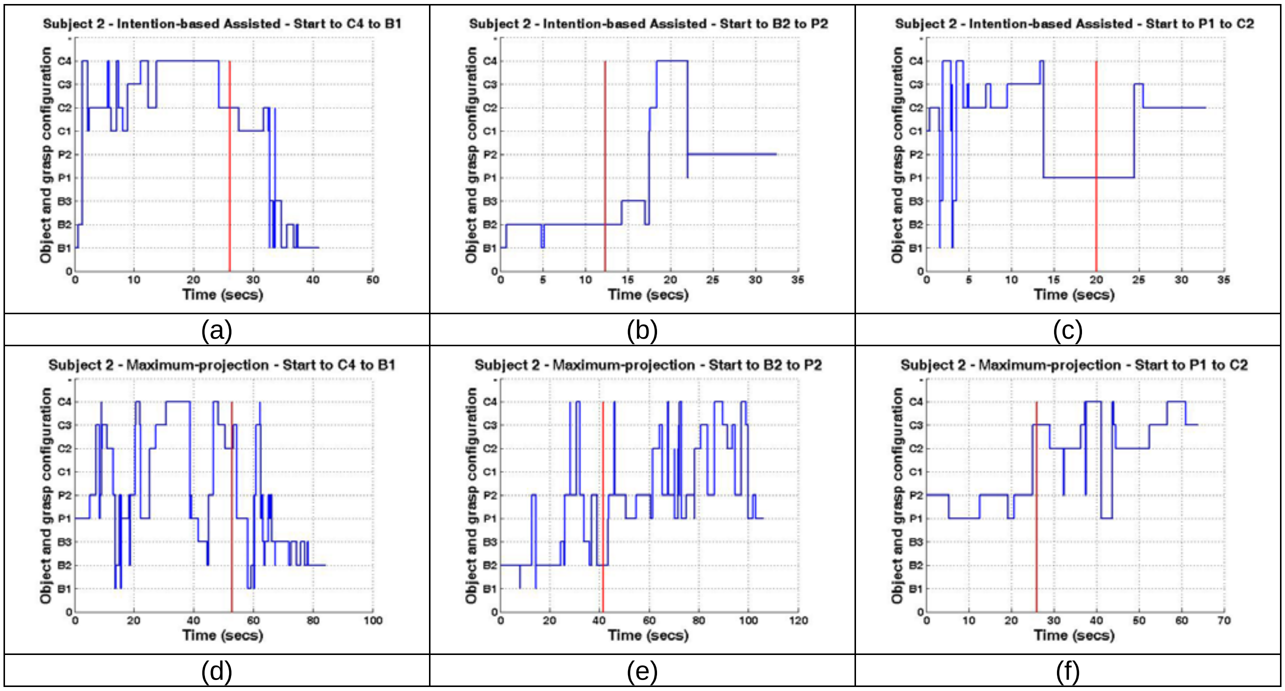


Figure 45: Arrangement of objects for change-in-intention experiment. Objects are shuffled with respect to Figure 42

The results of intention recognition through the trajectory are shown in Figure 46. Red vertical lines show the point when the subject was asked to change their intention. In the first trial (Figure 46 (a) and (d)), the subject is asked to teleoperate to C4 and then as the subject is about to reach the C4, she is asked to preshape over B1. Using our method, fluctuations are experienced by the subject at various grasp configurations of the cup, before intention change. Fluctuations are also experienced after intention change at the bowl. In the maximum-projection method, the cup is not detected consistently in the first half but the bowl is detected in the second half. The grasp configurations are not detected in the maximum-projection method. In the second trial, our method detects intentions consistently correctly. While changing from B2 to P2 (Figure 46 (b)), the algorithm momentarily detects C2 because cup is on the way to plate from bowl and, C2 is similar to B2. In maximum-projection method, the fluctuations are significant. While teleoperating to P1 and C2 (Figure 46 (c) and (f)), our method detects C3 as it is near P1 and is similar to P1. In the maximum-projection method, the object is detected more consistently than in the first two trials. The maximum-projection method takes almost 2 to 3 times more time to execute the task compared to our method. Thus, we can conclude that our method works well when objects are shuffled and intention is changed.



(a) (b) (c)

(d) (e) (f)

Figure 46: Intention recognition over the length of the trajectory for subject 2 when asked to change intention. (a), (b) and (c) represent our method whereas (d), (e) and (f) represent the maximum-projection method

In a nutshell, our method detects intention more accurately because of our usage of HMM. HMM provides cumulative probability and acts as a buffer against fluctuations. It combines various intention indicators into a single feature vector. Due to this, the intention is calculated by total evaluation of all factors and not just a single factor. In contrast, maximum-projection method takes into account only one factor at a time. If one employs heuristics to consider multiple factors, the method will be applicable to only few cases and will fail in other cases.

Chapter 5: Conclusions

5.1 In Summary

The methodology of identifying object and grasp configuration of interest from motion intention recognition makes it much easier for a person to execute a grasping task in teleoperation. They were effortlessly able to align with their desired grasp configuration. This was confirmed from the survey conducted at the end of the trials where the subject had to score the intention based assisted and the unassisted method on a scale of 1 to 10; 1 being easy and 10 being difficult. The intention based assisted method achieved an average score of 2.86 and unassisted method achieved that of 7.29. The users preferred using the implemented method as they experienced less mental load and less overall workload in executing the task. This was confirmed via the NASA-TLX workload ratings. On a scale of 1 to 100, intention based assisted method an average score of 32 whereas the unassisted method averaged to 71.14. Higher values indicate higher workload and hence more difficulty in executing the task. The implemented algorithm is able to determine the intention of the user early in the grasping task and, this combined with assistance, enables users to complete the task quickly. The grasping and the pick-and-place task were found to be twice as quick using the intention based assisted method. In other words, the subjects were able to complete the grasping task 50.05% quicker and the pick-and-place task at 51.47% quicker using the intention based assisted mode compared to the unassisted method. The subject also had to make fewer movements of their hands when executing the task using intention based assisted method. From the trials, an average of 46.04% fewer movements were reported compared to the unassisted method. The subjects also reported that they preferred using the implemented method compared to the maximum-projection method as they had extreme difficulty in completing the task using the maximum-projection method. The fluctuations in the detected intention in the maximum-projection method assisted the subjects in the wrong directions and this added to their frustration. The fluctuations were mainly when the rotation was negligible, near the objects. In many cases of maximum-projection method, the object was detected correctly but the grasp configurations were not. On the other hand, the intention detection using our method had very few fluctuations and our method consistently detected correct intention. This was mainly because of accumulation of output probability, high value of state transition probability over the same object and determining mode of the Viterbi. Thus, an HMM is robust erroneous changes in intention that may be caused due to unintentional deviations and errors in user control of robotic arm. Moreover, an HMM combines various factors, that contribute to intention recognition, into a single feature vector. A simplistic method like maximum-projection method would need thresholds to select the factor and use it depending on the scenario. This restricts its use to certain specific cases.

5.2 Strengths of the Method

a) The major benefit of the implemented method is that novice users are able to preshape and grasp quickly, and with much ease, without them having to train a model i.e. they achieve improvement in task performance by using the model trained by a skilled teleoperator. This saves long hours of training by each subject and frustration that arises from training in teleoperation. It also gives a standard to compare for our tests i.e. the same trained model. Improvement in task performance of the skilled teleoperator was also obtained.

b) Another major benefit of the implemented method is that it is applicable in unstructured environments. In other words, if the object are added, removed or shuffled around in the environment, there is no need to retrain the models and the accuracy of intention recognition does not change. A maximum of three objects in a dishwasher, with different arrangements, were tested.

c) The objects need not be a certain distance apart in translation or orientation. They can be close together. However, the limiting distance, if any, after which the intention recognition fluctuates and is unable to consistently detect the correct intention, has not been determined.

d) Objects that are of shapes similar to ones for which the model is developed, and of different sizes can be grasped using the implemented method. This is because the grasp configurations within an object are not related or constrained by distance or pose.

5.3 Limitations of the Method

a) The method does not work very well for users who have very poor teleoperation skills. How much skill is needed to benefit from the method has not been determined.

b) The method may determine a wrong intention if the user does not teleoperate along the shortest path to a grasp configuration i.e. if the user teleoperates along individual x-y-z or roll-pitch-yaw. This is because the user might be giving an indication to the robot that she may be teleoperating to another object.

c) The method does not determine the right intention if the user momentarily deviates the end-effector from the path leading to the desired object and grasp configuration. The user may do so to configure the pose of the end-effector so that it is in a more convenient configuration to teleoperate or to avoid an obstacle. The correct intonation is detected when the user is again teleoperating towards the desired object.

5.4 Future Work

Although the method that has been developed has proven to be beneficial, there are some important questions that need to be answered and improvements that, if undertaken, may result in a more robust, simpler and an advanced system. The future work has been listed below:

a) Automatic provision of objects and possible grasp configurations: Integrating an RGBD vision system will help in automatic object identification and estimation of the grasp configurations. This will obviate the need for making manual measurements and it will make the method widely usable. One may argue that an autonomous planner can grasp the object after the preshape configurations are known. But, a human will be needed to indicate the object and the grasp configuration. The method we have implemented is a means to achieve this end.

b) Direct intention indication: Is there a need for intention recognition in teleoperation to convey the goal to the robot? The implemented method assumes that the grasp configurations and locations of all the objects are known apriori. An easier method of commanding the robot could be to use an interface that displays the possible options for grasping to the user on a screen and have the user select that option with hands, voice or thought. Then the user could be assisted in teleoperation towards the goal. Although the method is simple and easy, the user will need to interact with multiple interfaces viz. the screen and the joystick. In the case where the user needs to change the intention, they will need to stop teleoperating and query the robot vision for displaying the various grasp possibilities and then select from the display. Using our method, the user simply translates and orients the end-effector towards the new goal and the user is assisted towards the goal based on the intention detected. Hence our method is more intuitive. With the screen-selection method, the user executes more independent steps and this makes it tedious.

c) On enhancing user experience:

   i. Replacing teleoperation with semi-autonomous control: An easier manner of completing the task after intention has been detected would be if the remaining part of the trajectory is executed autonomously.

   ii. Use of haptic feedback: Using of haptic feedback to enable the user to sense when (s)he is teleoperating beyond the desired grasp pose by providing a resisting force and to enable the users to sense that they are nearing a joint limit is expected to enhance the user's experience.

   iii. Velocity control: Velocity control implementation and comparison of it with the current position control method of executing the task will be carried out. The hypothesis is that velocity control will provide a better interface for executing a teleoperation based task as it will be less stressful for the users to teleoperate. This is because the master device will act like a joystick and the number of indexes will drastically reduce.

iv. Automatic detection of phases of task: To further enhance the user experience, automatic detection of the stage of the task viz. preshape, grasp, open grippers etc. is being planned to be implemented and also switching between various modes viz. intention based assistance, gripper-axis assistance etc. This will obviate the need to press the Omni stylus for informing the robot about switching between these stages.

v. End-effector trapped near a wrong object: By addressing the problem of the gripper being trapped in the motion direction, due to the detected intention not being the same as the desired, as a local minima problem, this source of frustration is planned to be eliminated. Part of the problem is due to the object nearness probability and a better modeling of nearness needs to be implemented.

d) On training and the exponential distribution model: Is the training really needed? The approximated exponential distribution models for various grasp configurations look very similar to each other. We should still obtain the same accuracy if the models for different grasp configurations are interchanged. This needs to be tested and if it is true, then can we develop a deterministic model, based on the approximated exponential distribution we have used. One can be developed for translation and one for rotation or the same may be used for both. Can the same deterministic model be used for all users or does it depend on the skill level of the user? It seems that a highly unskilled user will be assisted in the wrong directions if the current models, which are relatively steep, are used. Will using a steep model for a more skilled user and a relatively flat model for a more unskilled user serve better in intention recognition? Will a steep model work better when the object are closer together? It might help to develop deterministic models for different skill levels, scenarios etc. and train the robot to automatically select the right models to provide maximum assistance.

e) On improving accuracy of intention recognition by modifying the feature vectors:

i. Projections over undesired grasp configurations: It needs to be determined if the distribution of projections over reference vectors of undesired grasp configurations follows a pattern that can be modeled mathematically and if it does, then whether including them in the feature vector results in more accurate intention recognition.

ii. Projection of end-effector frame over the desired: It needs to be determined whether including the projection of end-effector frame, over the desired, in the feature vector improves the accuracy of intention recognition. The hypothesis is that the accuracy will improve at the expense of marginal time. This is because the probability will be higher when the frame is near to the desired and low when far away. In the latter cases, the change in translation and orientation will case the probability to increase. Also, the exponential model does not model the projection of end-effector frame very well. Another mathematical model will be needed.

iii. What would be a good way to represent the nearness to a grasp configuration and use that in the feature vector? The probability should increase as a grasp configuration is being approached. Using absolute distances makes the model less general and more specific to training. Or a lot of training will be needed, which will prove costly. Using normalized distances with respect to starting point would work for a trajectory but the reference point needs to change if intention changes.

iv. Weighing features: Some users prefer translating to near the object and then rotating whereas other uses teleoperate the other way round. By learning predominantly translation and rotation motions and weighing the feature vector based on the kind of motion, the accuracy of intention recognition is expected to improve.

f) Distance between objects and accuracy: The limiting distance at which the intention recognition fluctuates and fails to consistently detect the correct intention needs to be determined, if one such exists.

g) Advanced recognition model that considers intentional deviations: An example of intentional deviation is when the user decides to approach an object by traversing around another object, rather than traversing through the shortest distance. This could be because, the shortest path is not possible to achieve due to obstacles or kinematic constraints. How does one develop a model that predicts such an intention? Does it help to use complete trajectory information, say in the form of transformations matrices as features in observations. The model might be restricted to the training examples. In cases where it is not known at training that an alternative route could be taken, learning automata can be used. The algorithm could also not reduce the probability to a small value if it detects that there is a possibility of it being approached from an alternative direction. The obstacle information might trigger this detection. At the same time the probability of other objects being intended objects increases. If the end-effector continues to deviate, the probability may decrease and it starts increasing again if the object is approached again. When the confidence level reaches a certain threshold, the object can be finalized as the detected object. Other peculiarities in the user motion need to be modeled. These are: (a) motion is performed along a single axis, (b) performed in an orthogonal pattern typical of novice users, (c) has deviations and then the gripper returns to the desired trajectory, needs to be developed. Detecting user overall movement progress towards a desired target, rather than intention detection from small range movements, needs to be implemented.

h) Modeling grasp configurations only: Is it sufficient to only determine the most likely state? Is there even a need of modeling objects as HMM? If only modeling as states is sufficient, then the best state sequence will give the desired grasp configuration. In this case a task can be represented as an HMM and the various states can even have state transition probabilities that can be trained, may be by learning automata. E.g. a vegetable cutting task involving vegetables, knife, cutting tray etc. If new objects are added resulting in an increase in grasp configurations, then state transition matrix size can be modified accordingly. It needs to be determined as to why output probability computations are more robust to fluctuations than Viterbi sequence computation.

i) More testing: Tests on wheelchair-bound persons will be carried out in the future to determine if they will be interested in using such a methodology for grasping unreachable objects. The model will be tested with more skilled and novice users. The plan is to extend our system by using a mobile platform and providing visual feedback so that the user can perform tasks remotely.

References

[1] R. M. Alqasemi, "Maximizing manipulation capabilities of persons with disabilities using a smart 9-degree-of-freedom wheelchair-mounted robotic arm system," 2007.

[2] D. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang and A. Behal, "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 42, pp. 2-14, 2012.

[3] A. Jain and C. C. Kemp, "EL-E: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Autonomous Robots,* vol. 28, pp. 45-64, 2010.

[4] T. L. Chen, M. Ciocarlie, S. Cousins, P. Grice, K. Hawkins, K. Hsiao, C. C. Kemp, C. King, D. A. Lazewatsky and A. Leeper, "Robots for Humanity: A Case Study in Assistive Mobile Manipulation," *IEEE Robotics & Automation Magazine Special Issues on Assistive Robotics,* vol. 8, 2012.

[5] M. R. Cutkosky, *Robotic Grasping and Fine Manipulation.* Kluwer Academic Publishers, 1985.

[6] H. Hanafusa and H. Asada, "A robot hand with elastic fingers and its application to assembly process," in *IFAC Symposium on Information and Control Problems in Manufacturing Technology, Tokyo,* 1977, .

[7] H. Hanafusa and H. Asada, "Stable prehension by a robot hand with elastic fingers," in *Proc. 7th Int. Symp. Industrial Robots,* 1977, pp. 361-368.

[8] W. Holzmann and J. McCarthy, "Computing the friction forces associated with a three fingered grasp," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on,* 1985, pp. 594-600.

[9] J. W. Jameson, *Analytic Techniques for Automated Grasp,* 1985.

[10] J. Kerr and B. Roth, "Analysis of multifingered hands," *The International Journal of Robotics Research,* vol. 4, pp. 3-17, 1986.

[11] J. K. Salisbury and B. Roth, "Kinematic and force analysis of articulated mechanical hands,"1983.

[12] J. Barber, R. Volz, R. Desai, R. Rubinfeld, B. Schipper and J. Wolter, "Automatic two-fingered grip selection," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on,* 1986, pp. 890-896.

[13] J. D. Wolter, R. A. Volz and A. C. Woo, "Automatic generation of gripping positions," *Systems, Man and Cybernetics, IEEE Transactions on,* pp. 204-213, 1985.

[14] X. Zhu and J. Wang, "Synthesis of force-closure grasps on 3-D objects based on the Q distance," *Robotics and Automation, IEEE Transactions on,* vol. 19, pp. 669-679, 2003.

[15] D. Berenson and S. S. Srinivasa, "Grasp synthesis in cluttered environments for dexterous hands," in *Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on,* 2008, pp. 189-196.

[16] C. Borst, M. Fischer and G. Hirzinger, "A fast and robust grasp planner for arbitrary 3D objects," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on,* 1999, pp. 1890-1896.

[17] C. Borst, M. Fischer and G. Hirzinger, "Grasping the dice by dicing the grasp," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on,* 2003, pp. 3692-3697.

[18] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *The International Journal of Robotics Research,* vol. 28, pp. 851-867, 2009.

[19] Z. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *Robotics and Automation, IEEE Journal of,* vol. 4, pp. 32-44, 1988.

[20] A. T. Miller and P. K. Allen, "Examples of 3D grasp quality computations," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on,* 1999, pp. 1240-1246.

[21] A. T. Miller, S. Knoop, H. I. Christensen and P. K. Allen, "Automatic grasp planning using shape primitives," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on,* 2003, pp. 1824-1829.

[22] S. Ekvall and D. Kragic, "Grasp recognition for programming by demonstration," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on,* 2005, pp. 748-753.

[23] S. Ekvall and D. Kragic, "Interactive grasp learning based on human demonstration," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on,* 2004, pp. 3519-3524.

[24] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, T. Asfour and S. Schaal, "Template-based learning of grasp selection," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on,* 2012, pp. 2379-2384.

[25] R. Pelossof, A. Miller, P. Allen and T. Jebara, "An SVM learning approach to robotic grasping," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on,* 2004, pp. 3512-3518.

[26] A. Sahbani, S. El-Khoury and P. Bidaud, "An overview of 3D object grasp synthesis algorithms," *Robotics and Autonomous Systems,* vol. 60, pp. 326-336, 2012.

[27] J. Romero, H. Kjellstrm and D. Kragic, "Human-to-robot mapping of grasps," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, WS on Grasp and Task Learning by Imitation,* 2008, .

[28] A. Saxena, J. Driemeyer and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research,* vol. 27, pp. 157-173, 2008.

[29] M. Stark, P. Lies, M. Zillich, J. Wyatt and B. Schiele, "Functional object class detection based on learned affordance cues," in *Computer Vision Systems*Anonymous Springer, 2008, pp. 435-444.

[30] L. Ying, J. L. Fu and N. S. Pollard, "Data-driven grasp synthesis using shape matching and task-based pruning," *Visualization and Computer Graphics, IEEE Transactions on,* vol. 13, pp. 732-747, 2007.

[31] B. P. DeJong, E. L. Faulring, J. E. Colgate, M. A. Peshkin, H. Kang, Y. S. Park and T. F. Ewing, "Lessons learned from a novel teleoperation testbed," *Industrial Robot: An International Journal,* vol. 33, pp. 187-193, 2006.

[32] B. DeJong, J. Colgate and M. Peshkin, "Mental transformations in human-robot interaction," in *Mixed Reality and Human-Robot Interaction*Anonymous Springer, 2011, pp. 35-51.

[33] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control.* Cambridge, Mass.: MIT Press, 1992.

[34] K. Khokar, "Laser assisted telerobotic control for remote manipulation activities," 2009.

[35] L. Joly and C. Andriot. Motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism. Presented at IEEE International Conference on Robotics and Automation. 1995, .

[36] P. Aigner and B. McCarragher. Human integration into robot control utilizing potential fields. Presented at IEEE International Conference on Robotics and Automation. 1997.

[37] D. Kragic, P. Marayong, M. Li, A. Okamura and G. Hager, "Human-Machine Collaborative Systems for Microsurgical Applications," *The International Journal of Robotics Research,* vol. 24, pp. 731-741, 2005.

[38] A. Sorokin, D. Berenson, S. S. Srinivasa and M. Hebert, "People helping robots helping people: Crowdsourcing for grasping novel objects," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on,* 2010, pp. 2117-2122.

[39] B. Pitzer, M. Styer, C. Bersch, C. DuHadway and J. Becker, "Towards perceptual shared autonomy for robotic mobile manipulation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on,* 2011, pp. 6245-6251.

[40] K. Khokar, K. B. Reed, R. Alqasemi and R. Dubey, "Laser-assisted telerobotic control for enhancing manipulation capabilities of persons with disabilities," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on,* 2010, pp. 5139-5144.

[41] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction,* 2012, pp. 1-8.

[42] N. Pernalete, W. Yu, R. Dubey and W. Moreno. Development of a robotic haptic interface to assist the performance of vocational tasks by people with disability. Presented at IEEE International Conference on Robotics and Automation. 2002, .

[43] L. Rabiner, "A Turorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE,* vol. 77, pp. 257-286, 1989.

[44] M. Utsumi, T. Hirabayashi and M. Yoshie, "Development for teleoperation underwater grasping system in unclear environment," in *Underwater Technology, 2002. Proceedings of the 2002 International Symposium on,* 2002, pp. 349-353.

[45] M. Ciocarlie, K. Hsiao, A. Leeper and D. Gossow, "Mobile manipulation through an assistive home robot," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on,* 2012, pp. 5313-5320.

[46] M. Li and A. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in *Proceedings. 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS 2003.* 2003, pp. 125-131.

[47] D. Aarno, S. Ekvall and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *Proceedings of the IEEE International Conference on Robotics and Automation. ICRA 2005.* 2005, pp. 1139-1144.

[48] W. Yu, R. Alqasemi, R. Dubey and N. Pernalete. Telemanipulation assistance based on motion intention recognition. Presented at IEEE International Conference on Robotics and Automation. 2005.

[49] J. Yang, Y. Xu and C. Chen, "Human action learning via hidden Markov model," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans,* vol. 27, pp. 34-44, 1997.

[50] B. Hannaford and P. Lee, "Hidden Markov Model Analysis of Force/Torque Information in Telemanipulation," *The International Journal of Robotics Research,* vol. 10, pp. 528-539, 1991.

[51] B. Hannaford, "Multi-dimensional hidden markov model of telemanipulation tasks with varying outcomes," in *IEEE International Conference on Systems, Man and Cybernetics,* 1990, pp. 127-133.

[52] J. Yang, Y. Xu and C. Chen, "Hidden Markov Model Approach to Skill Learning and its Application to Telerobotics," *IEEE Transactions on Robotics and Automation,* pp. 621-631, 1994.

[53] S. Ekvall and D. Kragic, "Grasp recognition for programming by demonstration," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on,* 2005, pp. 748-753.

[54] S. Ferguson and G. R. Dunlop, "Grasp recognition from myoelectric signals," in *Proceedings of the Australasian Conference on Robotics and Automation, Auckland, New Zealand,* 2002, pp. 83-87.

[55] K. Bernardin, K. Ogawara, K. Ikeuchi and R. Dillmann, "A sensor fusion approach for recognizing continuous human grasping sequences using hidden Markov models," *Robotics, IEEE Transactions on,* vol. 21, pp. 47-57, 2005.

[56] S. B. Kang and K. Ikeuchi, "Toward automatic robot instruction from perception-recognizing a grasp from observation," *Robotics and Automation, IEEE Transactions on,* vol. 9, pp. 432-443, 1993.

[57] R. Palm and B. Iliev, "Grasp recognition by time-clustering, fuzzy modeling, and hidden markov models (HMM)-a comparative study," in *Fuzzy Systems, 2008. FUZZ-IEEE 2008.(IEEE World Congress on Computational Intelligence). IEEE International Conference on,* 2008, pp. 599-605.

[58] Z. Ju, H. Liu, X. Zhu and Y. Xiong, "Dynamic grasp recognition using time clustering, gaussian mixture models and hidden markov models," *Adv. Rob.,* vol. 23, pp. 1359-1371, 2009.

[59] J. Craig, *Introduction to Robotics: Mechanics and Control.* USA: Prentice Hall, 2005.

[60] K. Edwards, R. Alqasemi and R. Dubey, "Design, construction and testing of a wheelchair-mounted robotic arm," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on,* 2006, pp. 3165-3170.

[61] J. Denavit and R. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *ASME Journal of Applied Mechanics,* pp. 215-221, 1955.

[62] *Center for Assistive, Rehabilitation and Robotics Technologies*, http://carrt.eng.usf.edu/

[63] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *Man-Machine Systems, IEEE Transactions on,* vol. 10, pp. 47-53, 1969.

[64] R. Alqasemi, S. Mahler and R. Dubey, "Design and construction of a robotic gripper for activities of daily living for people with disabilities," in *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on,* 2007, pp. 432-437.

[65] Sensable Technologies. *Phantom Omni haptic feedback device*. http://www.sensegraphics.com

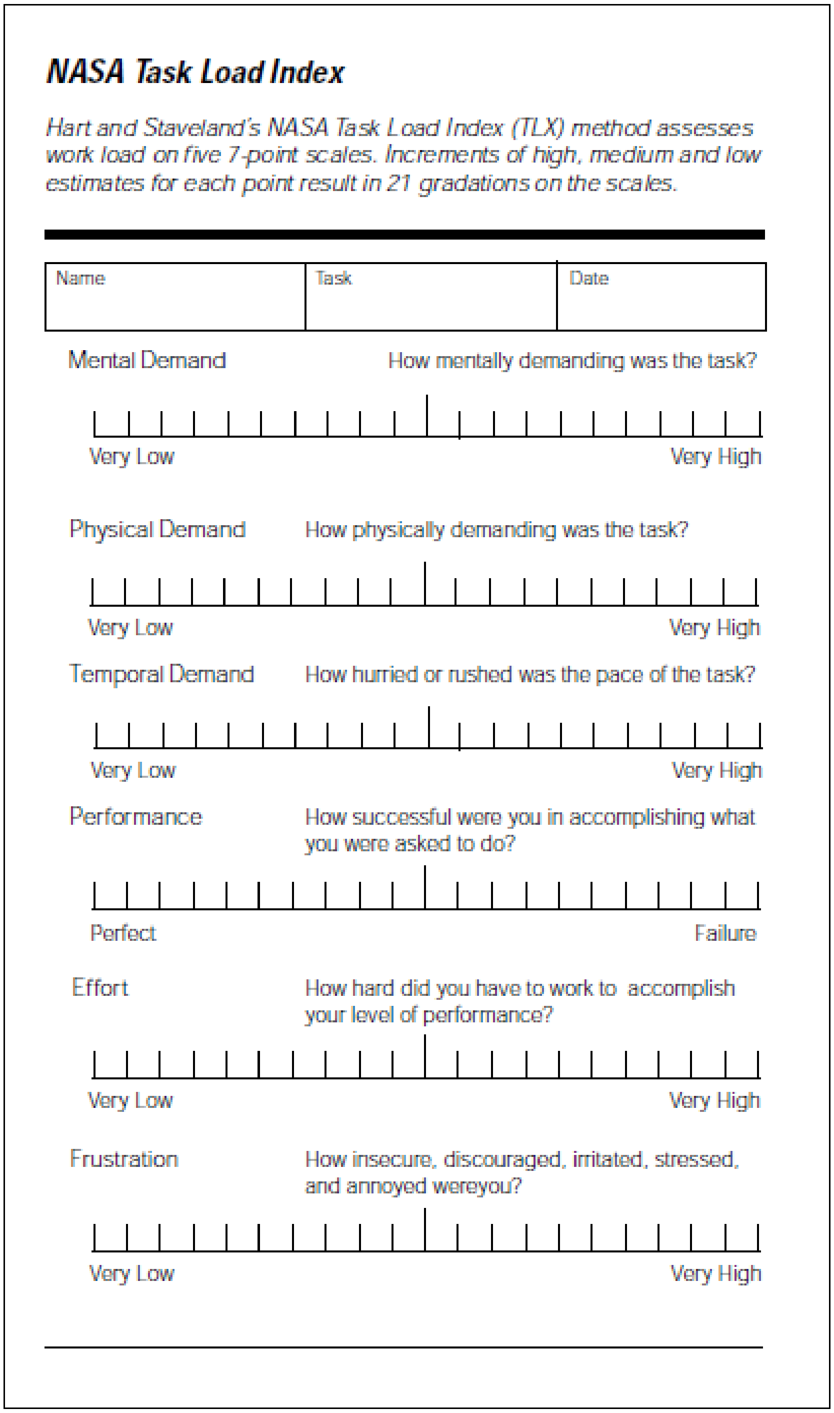
[66] *OpenHaptics Toolkit Reference*, http://www.sensegraphics.com

[67] R. Alqasemi and R. Dubey, "Combined mobility and manipulation control of a newly developed 9-DOF wheelchair-mounted robotic arm system," in *Robotics and Automation, 2007 IEEE International Conference on,* 2007, pp. 4524-4529.

[68] S. G. Hart, "NASA-task load index (NASA-TLX); 20 years later," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* 2006, pp. 904-908.

Appendices

Appendix A NASA-TLX Scoring Sheets of Subjects



NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand — How mentally demanding was the task?

Very Low — Very High

Physical Demand — How physically demanding was the task?

Very Low — Very High

Temporal Demand — How hurried or rushed was the pace of the task?

Very Low — Very High

Performance — How successful were you in accomplishing what you were asked to do?

Perfect — Failure

Effort — How hard did you have to work to accomplish your level of performance?

Very Low — Very High

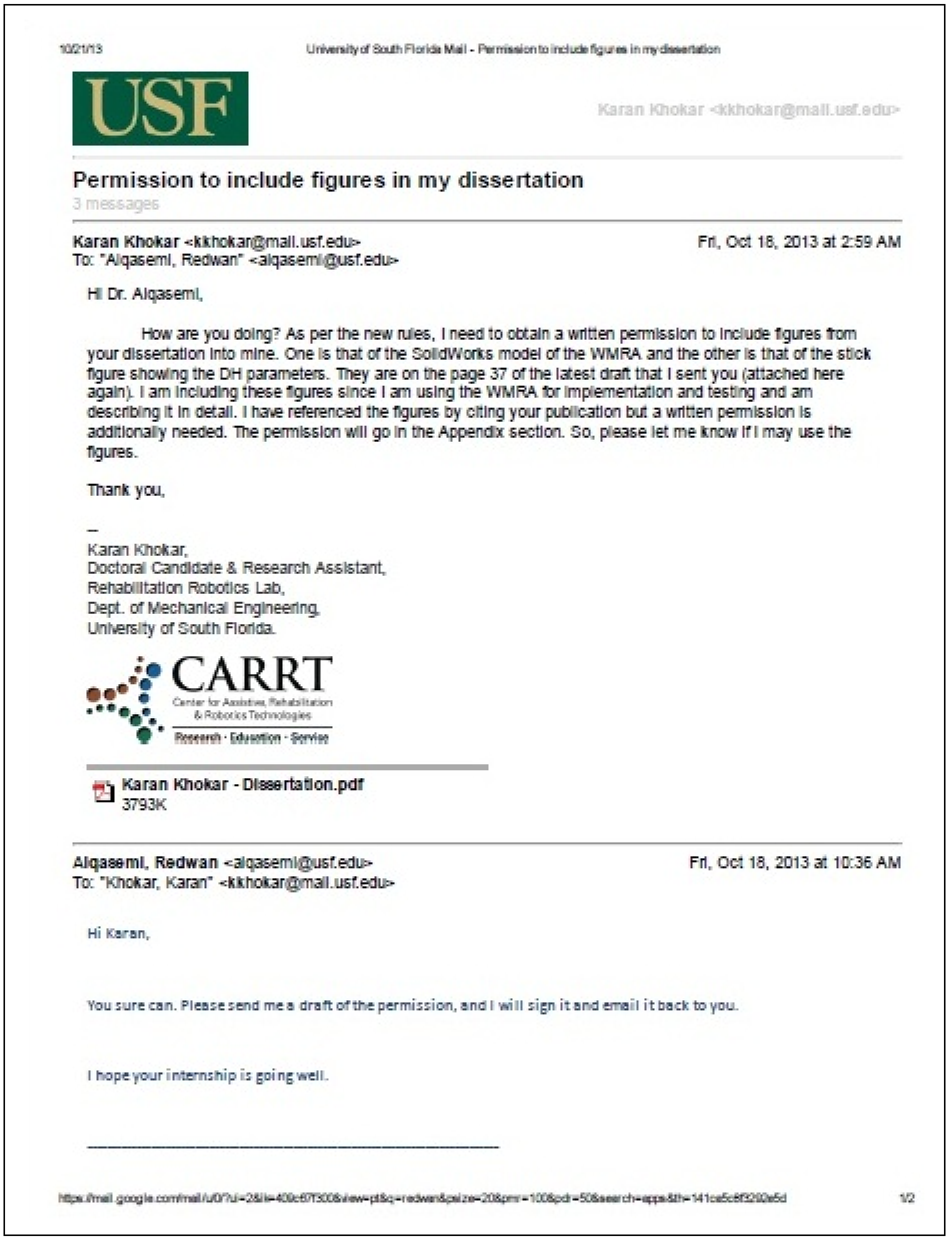
Frustration — How insecure, discouraged, irritated, stressed, and annoyed wereyou?

Very Low — Very High

Figure A1: NASA-TLX scoring sheet

Appendix B Approval to Use Copyrighted Material

B.1 Permission to Reproduce Figure 3



10/21/13 University of South Florida Mail - Permission to include figures in my dissertation

Karan Khokar <kkhokar@mail.usf.edu>

Permission to include figures in my dissertation

3 messages

Karan Khokar <kkhokar@mail.usf.edu> Fri, Oct 18, 2013 at 2:59 AM
To: "Alqasemi, Redwan" <alqasemi@usf.edu>

Hi Dr. Alqasemi,

How are you doing? As per the new rules, I need to obtain a written permission to include figures from your dissertation into mine. One is that of the SolidWorks model of the WMRA and the other is that of the stick figure showing the DH parameters. They are on the page 37 of the latest draft that I sent you (attached here again). I am including these figures since I am using the WMRA for implementation and testing and am describing it in detail. I have referenced the figures by citing your publication but a written permission is additionally needed. The permission will go in the Appendix section. So, please let me know if I may use the figures.

Thank you,

--
Karan Khokar,
Doctoral Candidate & Research Assistant,
Rehabilitation Robotics Lab,
Dept. of Mechanical Engineering,
University of South Florida.

CARRT
Center for Assistive, Rehabilitation & Robotics Technologies
Research · Education · Service

Karan Khokar - Dissertation.pdf
3793K

Alqasemi, Redwan <alqasemi@usf.edu> Fri, Oct 18, 2013 at 10:36 AM
To: "Khokar, Karan" <kkhokar@mail.usf.edu>

Hi Karan,

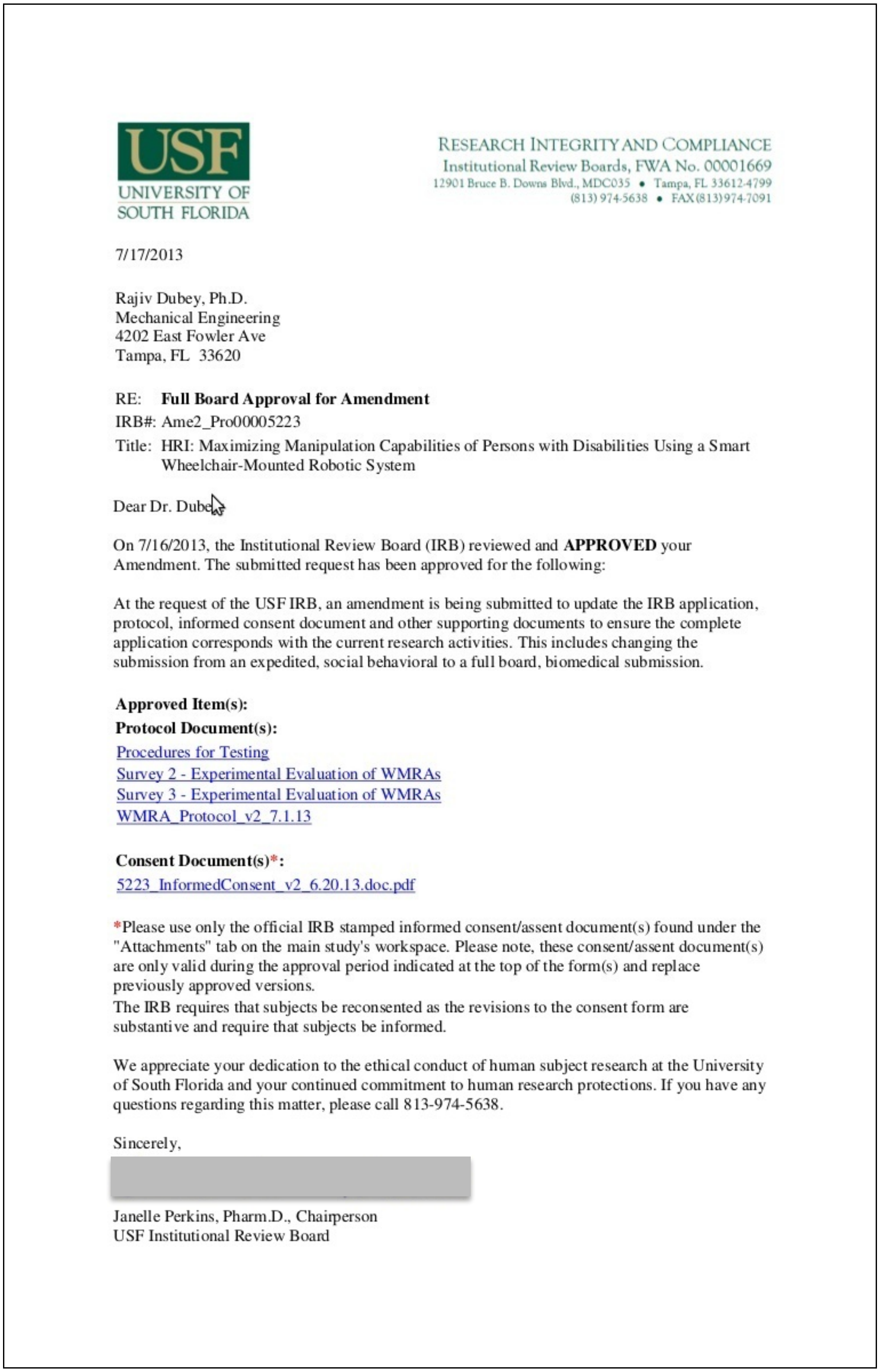
You sure can. Please send me a draft of the permission, and I will sign it and email it back to you.

I hope your internship is going well.

____

https://mail.google.com/mail/u/0/?ui=2&ik=409c67f300&view=pt&q=redwan&psize=20&pmr=100&pdr=50&search=apps&th=141ca5c6f3292e5d 1/2

Appendix C Institutional Review Board (IRB) Approval



USF
UNIVERSITY OF
SOUTH FLORIDA

RESEARCH INTEGRITY AND COMPLIANCE
Institutional Review Boards, FWA No. 00001669
12901 Bruce B. Downs Blvd., MDC035 • Tampa, FL 33612-4799
(813) 974-5638 • FAX(813)974-7091

7/17/2013

Rajiv Dubey, Ph.D.
Mechanical Engineering
4202 East Fowler Ave
Tampa, FL 33620

RE: **Full Board Approval for Amendment**
IRB#: Ame2_Pro00005223
Title: HRI: Maximizing Manipulation Capabilities of Persons with Disabilities Using a Smart Wheelchair-Mounted Robotic System

Dear Dr. Dubey:

On 7/16/2013, the Institutional Review Board (IRB) reviewed and **APPROVED** your Amendment. The submitted request has been approved for the following:

At the request of the USF IRB, an amendment is being submitted to update the IRB application, protocol, informed consent document and other supporting documents to ensure the complete application corresponds with the current research activities. This includes changing the submission from an expedited, social behavioral to a full board, biomedical submission.

Approved Item(s):
Protocol Document(s):
Procedures for Testing
Survey 2 - Experimental Evaluation of WMRAs
Survey 3 - Experimental Evaluation of WMRAs
WMRA_Protocol_v2_7.1.13

Consent Document(s)*:
5223_InformedConsent_v2_6.20.13.doc.pdf

*Please use only the official IRB stamped informed consent/assent document(s) found under the "Attachments" tab on the main study's workspace. Please note, these consent/assent document(s) are only valid during the approval period indicated at the top of the form(s) and replace previously approved versions.
The IRB requires that subjects be reconsented as the revisions to the consent form are substantive and require that subjects be informed.

We appreciate your dedication to the ethical conduct of human subject research at the University of South Florida and your continued commitment to human research protections. If you have any questions regarding this matter, please call 813-974-5638.

Sincerely,

[redacted]

Janelle Perkins, Pharm.D., Chairperson
USF Institutional Review Board

Figure C1: Insitutional Review Board (IRB) approval letter