

Assistive Robotic Grasping

Jonathan D. Weisz

Submitted in partial fulfillment of the
requirements for the degree
of Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2015

©2015

Jonathan D. Weisz

All Rights Reserved

ABSTRACT

Assistive Robotic Grasping

Jonathan D. Weisz

This thesis describes some contributions towards the implementation of Human-in-the-Loop(HitL) grasping for assistive robotics, with a particular focus on low throughput, high noise interfaces such as electroencephalography(EEG) or electromyography(EMG) brain-computer interface(BCI) devices in natural environments. Although progress in the robotics field has been swift, it is unlikely that truly independent operation of robots in situations where they will interact closely with objects, obstacles, and perhaps even other people in their environment will evolve in the immediate future. However, with the help of a human operator, it is possible to achieve robust, safe operation in complex environments. This work describes a system that can accomplish this with minimal interfaces that are accessible even to individuals with impairments, which will enable the development of more capable assistive devices for these individuals.

Grasping an object generally requires contextual knowledge of the object and the intent of the user. We have developed a user interface for an on-line grasp planner that allows the user to effectively express their intent. Grasping in natural environments requires grasp planning algorithms that are robust to target localization errors. This work describes grasp quality measurements that generate more robust grasps by considering the local geometry of the object as well as how uncertainty will effect the proposed grasp. These new measures are integrated into an augmented reality interface that allows a user to plan a grasp online that matches their intent for using the object that is to be grasped. This interface is validated by testing with real users, both healthy and impaired, using a variety of input devices suitable for impaired subjects, such as low cost EEG and EMG devices. This work forms the foundation for a flexible, fully featured HitL system that will allow users to grasp objects in cluttered spaces using novel, practical BCI devices that have the potential to

bring HiTL assistive devices out of the research environment and into the lives of those that need them.

Table of Contents

1	Introduction	1
1.1	The Promise of Assistive Robotics	1
1.2	The Challenge of Robotic Grasping	2
1.3	The Eigengrasp Planner	2
1.4	An Integrated Assistive Robotic Grasping Platform	3
1.5	The Contribution Of This Thesis	4
2	Related Work	6
2.1	Brain Computer Interfaces with Robotics	6
2.1.1	Overview	6
2.1.2	Direct Demonstration	7
2.1.3	Joint Level Control	7
2.1.4	End Effector Cartesian Control	8
2.1.5	Discrete Mode Control	9
2.1.6	Task Level Control	10
2.2	Robust Grasping	10
2.2.1	Wrench Space Metrics	11
2.2.2	Wrench Space Metrics Robust to Contact Location Uncertainty . . .	11
2.2.3	Simplified Analytical models of Robustness	12
2.2.4	Sampling Based Heuristics	12
3	Robust Grasping	14
3.1	Grasp Planning with Contact Points	14

3.2	Grasp Wrench Spaces	15
3.3	The ϵ_{GWS} Quality Metric	16
3.4	Optimization Based Grasp Planning	16
3.4.1	Potential Contact Quality Function	17
3.4.2	Implications for pose error robustness	17
3.5	Analyzing The Effects of Pose Error In Simulation	18
3.5.1	The Columbia Grasp Database	18
3.5.2	Grasping Pipeline	19
3.5.3	Model of Pose Uncertainty	20
3.5.4	Simulation Results	21
3.5.5	Chosing optimal grasps: ϵ_{GWS} vs $P(fc)$	23
3.6	Physical Experiments	24
3.7	Discussion	29
3.7.1	Online Extensions	29
4	A Grasp Planning UI for Facial Gestures	31
4.1	A BCI input device prototyping	31
4.1.1	Mindwave	31
4.1.2	The Emotiv Epoc	32
4.1.3	UI Design	33
4.1.4	Grasping Platform	39
4.1.5	Validation	39
4.1.6	Conclusions	40
4.2	Handling Novel Objects	42
4.2.1	Encorporating a grasp database	42
4.2.2	Experiments	44
4.2.3	Discussion	51
4.3	Integrating a novel sEMG Device	51
4.3.1	Surface EMG Recording	51
4.3.2	sEMG GUI	53
4.4	Handling Cluttered Scenes	55

4.4.1	GUI Modifications for Clutter and sEMG Interface	57
4.4.2	Pipeline Version 3	60
4.4.3	Validation	61
4.5	Further Refinements	65
4.5.1	Adaptations for the Mico Manipulator	65
4.5.2	Improved Online Reachability Checking	66
4.5.3	Further UI Improvements	67
4.5.4	Updated pipeline	68
4.5.5	Validation	70
I	Bibliography	79
	Bibliography	80

List of Figures

3.1	An example of a grasp where the ϵ_{GWS} is consistent with respect to the common localization errors, and one where it is not.	17
3.2	<i>Illustration of the initial pose and posture.</i> The hand is withdrawn along its approach direction and all of the fingers opened at the same constant velocity until each of the fingertips are outside of the knuckles of the hand in its pre-grasp configuration. The hand is then opened an additional ten percent of the remaining joint range. The pre-grasp configuration is shown in green. .	20
3.3	Illustration of the tabletop error model used in this paper. The object is can be translated $[-10, 10]$ mm in both X and Y and rotated $[-20, 20]^\circ$ around θ . The red cones mark the planned contact points. The white wire frame indicates the planned end configuration of the fingers.	21
3.4	ϵ_{GWS} plotted against $P(fc)$, which is equivalent to the probability of attaining a force closed grasp under our error model from a planned starting grasp configuration. 480 grasps over 96 objects are presented in this figure. This figure demonstrates that the ϵ_{GWS} is generally a poor predictor of pose error robustness as measured by the $P(fc)$. The mean ϵ_{GWS} , denoted by the black line, is 0.095. There appears to be a cutoff, denoted by a red vertical line on the graph, around $\epsilon_{GWS} > 0.2$, after which all available grasps are have a high $P(fc)$, but note how few grasps there are in the database with quality values that high. Only five objects have a grasp with a quality that high, which implies that for most objects, ϵ_{GWS} cannot be used to predict $P(fc)$.	25

4.3	Two images captured as a user grasps an object with this system. Top: Clockwise from the top-left are an image of the user wearing the Emotiv headset and operating the system, the computer monitor with user interface, and a screen capture of the simulation. Bottom: User watching the robotic arm implementing the selecting grasp as shown in the simulator window on the lower left.	38
4.4	The results of ten different attempts to grasp five objects using two different user-selected approach directions per object. In each case, the user was able to generate an appropriate grasp using the simulator which was then transferred to the robot.	41
4.5	Results of the object recognition system with known and unknown objects.	45
4.6	The user interface for the semi-autonomous grasp planning in the Online Planner phase. The user interface is comprised of three windows: The main window containing three labeled robot hands and the target object with the aligned point cloud, the pipeline guide window containing hints for the user to guide their interaction with each phase of the planner, and the grasp view window containing rendering of the ten best grasps found by the planner thus far. The object shown in this figure is a novel object that the planner does not have a model of (see Fig. 4.5a). The point cloud allows the user to visualize the fit of the model and act accordingly.	46
4.7	This handle grasp for the all bottle is not a force closure grasp, but when chosen by the subjects in our experiments it succeeded 100% of the time. Adding a grasp database allows such semantically relevant grasps to be used in our system.	47
4.8	The phases modified grasping pipeline incorporating more control over the vision system and the grasp database. The purple diamonds reflect decision points in the pipeline that require input from the user. If the user chooses grasp n from the database, $n+4$ user inputs are required. If none of the grasps are suitable, the online planner can be invoked with a few simple inputs to refine one of the grasps further.	48

4.9 A diagram of the muscles around the ear. The sEMG device has been used on both the Posterior and Superior Auricular muscles. In order to record from the Superior Auricular (SA), it is necessary to remove the hair in the area above the ear. The Posterior Auricular(PA) is generally easier to access, however it is a smaller muscle that is more difficult to learn voluntary control over.	54
4.10 The single sEMG signal is first processed through a 60 Hz noise filter. It is then run through two different band pass Butterworth filters to extract two separate signals. The bands are then linearly combined to compute the x and y cursor positions.	54
4.11 The sEMG Interface: (a) Hitting one particular target changes the color of the cursor to reflect the selection and makes the other targets unavailable. (b) If the user does not return to the rest area after a few seconds, the selection times out and is deselected and all targets become available again for selection. (c) The user interface is composed of 4 targets overlayed on the grasping scene. Target 1 usually signals acceptance of the current option. Target 2 toggles the next option. Targets 3 and 4 provide input to the planner.	55
4.12 The Grasp Planning Interface - Object Selection Phase: The subject is able to see the planning scene in the main UI window. The window on the bottom tells the user the current phase and what the green and red inputs will do in this phase. In this phase, the subject sees the pointcloud and hits the red target until the object they wish to grasp is highlighted in green. Then they hit the green target to proceed to the next phase.	58

4.13 The Grasp Planning Interface - Initial Review Phase: After the subject selects the object, the Grasp View pane on the right is populated with a set of grasps from a database. Grasps that are reachable appear on a green background, while unreachable grasps are red. A robot hand appears that the user moves to demonstrate a desired starting pose. This demonstration hand is constrained to follow the two circular guides around the z and x axes of the object shown above. The top most grasp in the grasp view window is the currently selected grasp, which is rendered in the planning scene with the planner hand.	59
4.14 An impaired subject in the UC Davis RASCAL lab (top) operating our sEMG-Assistive Grasping interface to grasp a shaving gel bottle in the Columbia Robotics Group Laboratory (bottom right). The two small black clips behind the subjects ear (bottom left) are surface EMG electrodes (used in differential mode) to detect activation of the Auricularis Posterior (AP) muscle to direct the system to pick up the object in this multi-object scene.	62
4.15 A diagram of the kinematics of the Mico arm. The 60° offsets of the the final two arm joints is very unusual, and causes difficulties for some robotics libraries.	66
4.16 The Object Recognition and Selection state user interface	69
4.17 The UI in all of the second and third of the pipeline. The buttons on the right function as both guides for the result of hitting the color coded input options that will be presented to the use, as well as buttons that the user and experimentor use during the training stage	71
4.18 The final two stages of version 4 of the BCI grasping UI. This version of the UI is more dynamic, with the layout of the guide window on the right side of the screen changing as the user progresses through the pipeline. While this may seem like unnecessary complexity, the UI is more visibly different in each stage and less extraneous information is presented.	72

4.19 The sEMG system setup for the updated pipeline experiments. In these experiments, we placed the sEMG device behind the ear of the subject to measure contractions of the PA muscle. In order to stabilize the device and reduce noise due to motion of the wires, we stabilize the electrodes by wrapping the head of the electrodes in Silly Putty brand silicone putty.	73
4.20 The sEMG interface overlayed on the planning scene with the selected target highlighted in green.	74
4.21 The training version of the sEMG interface in Matlab, showing the desired target in red, and the selected target in green.	75
4.22 A typical grasp of the shampoo bottle from the side in the cluttered scene. Note that the hand is just able to fit between the other objects to grasp the desired target. Note that the ability to plan this grasp in such a restricted environment is an indication that this system is very successful at handling the cluttered scene.	77

List of Tables

3.1	Summary of Simulation Results: These results quantify how robust each quality measure is to pose uncertainty. For each grasp, the expected value and standard deviation of the respective quality measures are calculated across the poses sampled. These measures are normalized by dividing them by the planned quality of the unperturbed grasp to allow comparison of the effects of uncertainty across grasps. $P(fc)$ denotes the measured probability of achieving a force closed grasp, which we have calculated as $P(\epsilon_{GWS} > .001)$	23
3.2	This table quantifies the effect of sampling parameters on the estimate of quality measure. The first column describes the sampling density along each dimension of the error model. The other columns describe the average change in the subsampled $P(fc)$ as compared to the densest sampling rate. These results demonstrate that lower sampling rates may feasibly be used to approximate the $P(fc)$ at the densist sampling rate without unacceptable loss of accuracy.	30
4.1	This table shows the initial control flow of the grasping pipeline using facial gestures detected by the Emotiv Epoc	34
4.2	This table shows the initial control flow of the grasping pipeline using facial gestures detected by the Emotiv Epoc	48
4.3	Results from Experiment 2	50
4.4	sEMG Experiment 1 Results	64

4.5 Results from Experiment 3. On average, the subjects were successful in grasping 77% of the objects within 85 seconds of the first time their cursor left rest area.	78
---	----

Acknowledgments

The acknowledgments go here. The acknowledgments go here.

Dedication text

Chapter 1

Introduction

1.1 The Promise of Assistive Robotics

Although progress in the robotics field has been swift, it is unlikely that truly independent operation of robots in situations where they will interact closely with objects, obstacles, and perhaps even other people in their environment will evolve in the immediate future. However, with the help of a human operator, it is possible to achieve robust, safe operation in complex environments. This work describes a system that can accomplish this with minimal interfaces that are accessible even to individuals with impairments, which will enable the development of more capable assistive devices for these individuals.

Grasping is important for many activities of daily living (ADL) that physically impaired individuals need assistance performing, such as fetching food or communication devices. There is a large and growing population with upper limb mobility issues in the United States. There are currently 400,000 individuals with spinal cord injuries, 50% of whom suffer upper limb mobility issues, 5 million individuals that have suffered from strokes, and a generally aging worldwide population. This presents an urgent need for better assistive technologies. The individuals with the greatest need for assistive technologies are those with severe impairments. By the nature of their health challenges, they are often limited in their ability to provide input to an assistive device. To address this concern, there has been a great deal of interest in Brain Computer Interface (BCI) devices over the last few years. In general, these devices are restricted to low bandwidth, noisy signals. Therefore, using these

devices to control an assistive device poses many challenges.

1.2 The Challenge of Robotic Grasping

Irrespective of the problems of BCI interface, robotic grasping is challenging for a number of reasons. First, complex robotic hands have many degrees of freedom, which makes the space of possible grasps extremely difficult to explore. Standard approaches to planning in high dimensional state spaces are likely to fail with many fingered hands, especially as the grasp itself involves purposeful collision with the object, but most of the "near grasp" states will be overlapping the object in some way. Secondly, evaluating grasps involves several properties that are difficult to model, such as friction and closed chain kinematics. Many analysis tools that currently the state of the art are only effective if the contact points can be perfectly predicted and the grasp acquisition can be perfectly controlled so that the object is not moved. Finally, robotic hands are extremely heterogeneous in terms of their physical size, the arrangement of their sensors, and their actuators, which makes finding generic strategies difficult.

In addition to all of these issues, in natural environments, any set of grasps that is preplanned may overlap with obstacles in the environment or fail to grasp the object in a way that is well suited to the desired use of the object, so grasp planning algorithms must be fast enough to run online, and be able to reflect the intent of the use of the grasp beyond simple stability. Until these challenges are overcome, robots will remain of limited utility as helpers for improving the lives of people outside of the lab.

1.3 The Eigengrasp Planner

Previous work in our lab by Ciocarlie et al. has attempted to address the main challenges of grasp planning by producing the Eigengrasp Planner. The planner addresses these issues by following a dimensionality reduction strategy that reduces the state space of the hand so that a sampling based stochastic optimization strategy can be used with a relatively simple grasp quality metric. The quality metric that is used by the planner evaluates a projection of the fingers on to the target object, which makes the state space easier to evaluate because

the quality function is valid in configurations where the hand is not in contact with the object. The nature of the optimization approach, which gradually moves towards lower values of the quality function, should produce solutions where nearby finger contacts will also provide similar quality grasps because grasps where nearby points are poor will have narrow basins of attraction that are less likely to be found. This implies a certain amount of robustness to the small displacements of the object during grasp acquisition. The planner is easy to generalize because the only robot specific parameters are the state space reduction strategy and a set of desirable contact locations, which can be easily specified for any given robot.

The planner is fast enough to run in near real time, and can be made responsive to user intent by biasing the search of the stochastic optimization towards a pose that is demonstrated by a human operator. In this model, the desired intent of the user is indicated by demonstrating a desired direction and approximate location to grasp from, possibly by moving a virtual hand in a simulated environment.

The efficacy of Eigengrasp Planner's approach was demonstrated by having an operator hold a robotic hand's base and move it around a target object while running the planner. When the planner found a grasp that appeared reasonable, the operator would grasp the object by simply clicking a single key. The approach of the hand to the object and the application of force to the hand as it contacted the object were controlled manually by the operator. This demonstration was extremely compelling, and the problem appeared to be largely solved. However, some limitations of this approach were exposed in attempting to translate this work from the initial demonstration to a platform that could be used by a disabled person, given the limitations of the input devices available.

1.4 An Integrated Assistive Robotic Grasping Platform

To begin addressing even the most basic needs of an impaired individual with assistive robotics technologies, we need an underlying platform for robust grasping. Such a system would need to integrate path planning, object localization and identification, and grasp planning with a user interface suitable for whatever signals the user is able to supply. Al-

though fully automated approaches for each of these components have been the subject of extensive and ongoing research, integrating user input to create a shared-control environment that uses as much input as the user is able to supply is still a relatively unexplored field. There are many possible paradigms for integrating BCIs with a shared-control assistive robotics device. Traditional EMG and EEG setups are expensive and difficult to deploy. Recently, cheap and portable EEG headsets have become available, and new devices such as portable near-infrared-spectroscopy (NIRS) devices and single electrode EMG devices are also reaching the market.

These devices expand the potential for a practical, multimodal BCI system that can be taken out of the laboratory to create an assistive robotic grasping system for less structured environments. However, grasp planning remains a difficult, unsolved problem. Online planning and grasp planning under sensor uncertainty are each independently unsolved problems that need further exploration. Putting the human in the loop when planning and executing the grasp in real time fundamentally changes the nature of the problem as compared to a fully automated system. The key part of the problem becomes conveying information to the user effectively about the state of the system and then using the low bandwidth information gained from the user efficiently. This requires us to think carefully about the interfaces provided to the user and how we infer intent from the users input. Additionally, in order to present the user with reasonable grasping options, we need to extend the existing grasp stability analyses to deal with the most common problem that arises in unstructured environments, object localization errors due to sensor noise.

1.5 The Contribution Of This Thesis

This thesis extends and refines the initial work on the eigengrasp planner to a flexible integrated assistive robotics grasping platform, and demonstrates its application to a number of different input devices designed for severely impaired individuals. Two main issues are addressed. Without the benefits of complete control over the grasping process and the assumption of perfect localization of the target object, significantly more care has to be giving to selecting robust grasps. Additionally, the user cannot be assumed to have fine control

over the demonstration pose because their input device may not provide the bandwidth that such control requires. This means we need to devise a user interface that can accommodate a more limited input

To address the first issue, I present the large scale first analysis of how the seminal grasp quality measures proposed by Ferrari and Canny is expected to vary under realistic models of object location uncertainty in ?? . This analysis shows that the grasps produced by the EigenGrasp planner are not sufficiently robust under even modest location uncertainties. From this analysis, I derive an extension of the grasp quality measure that produces significantly more robust grasps.

I will present the novel platform that we have developed to apply this planner using several different brain computer interfaces. These interfaces take advantage of the fact that many severely impaired individuals who lose function in their limbs still maintain control over their facial muscles, from which we can extract a few control signals reliably. These devices are relatively low cost and noninvasive compared with most other modalities of getting signals from impaired individuals. In chapter ?? I will present the development of the integrated system, including the object localization methodology, the online grasp filtering, and further refinements to the grasp planning system to improve it's speed and quality. In chapter ??, I will describe implementation of this interface with the Emotiv Epoc, a commercial EEG/EMG device, and a combination of facial gestures and EEG control, and the results achieved by a healthy cohort of subjects. In chapter ??, I will describe an interface which uses much more minimal and practical surface EMG device that is much more easily deployable than the Epoc and other similar cap devices, because it requires only a single recording site and a few wires. We will see that an impaired individual is able to grasp objects using this minimalistic device after only a short introduction to the grasping platform. Based on these experiments, we further refined this system and present these refinements as well as a healthy cohorts capabilities with the system in chapter ??

Chapter 2

Related Work

2.1 Brain Computer Interfaces with Robotics

2.1.1 Overview

There is a long historical precedent for engineering assistive systems using electrophysiological signals to drive robotic devices [Schmidl, 1965], with a commercial version being introduced by [Sherman and Lippay, 1965] as early as 1965. In the time since, there have been myriad approaches and refinements of proposed to interfacing disabled individuals with robotic assistive systems, and this work will not review even a small fraction of them. There are two ways of categorizing these types of systems. One way is to categorize the system by the input modality; That is whether it uses physical buttons or pointing devices, some external sensor of motion such as eye or hand trackers, or some specific electrophysiological signal such as EMG, EOG, and EEG. Another consideration is where the signals are recorded from. EMG can be recorded from distal muscle sites, which may be larger, easier to record from, and produce larger signals. However, more impaired individuals tend to maintain control over muscle functions closer to the head

Another way to categorize the systems is by the type of control they engender - whether the control is at a task level, allowing the user to designate what is to be done, or at a state level, allowing the user to specify joint angles, or end effector positions.

This work presents a system at an intermediate level, in which the user has some state

level control that is task oriented. This requires an online planning system that generates robust grasps in realtime. Below I describe the different control paradigms used in related systems using BCI devices.

2.1.2 Direct Demonstration

The most intuitive, low level of control of a robotic arm involves having the robot arm directly mimic the motions of the user. It is possible to reconstruct a user's movements using surface distal limb surface EMG signals, as in [Artemiadis and Kyriakopoulos, 2011; Castellini and van der Smagt, 2009]. This type of control allows the user to express their desires explicitly, allowing the user to specify how the arm is to avoid objects. A mapping from the subjects joints to the robotic arm can be found that ensures that the robot stays approximately the same distance from it's joint limits as the subject does, which ensures that the robot's workspace will be compatible with the intent of the user in manipulating objects.

This paradigm is not suitable for assistive robotic interfaces because the impaired individuals have lost exactly the capability used as the control input to this type of interface.

2.1.3 Joint Level Control

Rather than implicitly controlling the arm, some paradigms allow the user explicit control of joints of the robot. This generally imposes a much higher cognitive load on the user, as they have to attend closely to each joint. The movement of the joints are not directly related to user's goal of manipulating some object. Control of a manipulator through such an interface is generally not possible because they have many joints. The manipulator is generally controlled by simple open and close commands. For example, in [Horki *et al.*, 2011] hand opening/closing and elbow flexion/extension are controlled by EEG signals.

For a prosthetic arm with essentially two degrees of freedom, this sort of control may be appropriate, but more degrees of freedom will strain the bit rate of these types of devices and their unreliability will make the coordination necessary to perform complex tasks in natural environments difficult or impossible. For example, to grasp an object using a six degree manipulator with a gripper, the user must in a straight line towards an object, or

the gripper will not move straight and the finger may knock over the object while moving the palm into place. This requires that the user simultaneously send coordinated signals to all six degrees in exactly the right ratios, or it may not move in anything close to a straight line.

2.1.4 End Effector Cartesian Control

The main goal of a robotic manipulator is to interact with the world with some end effector. Giving the user direct control over the end effector location can be more intuitive, because the user generally has an end effector location is the variable that the user most directly observes. These works are range in invasiveness and signal quality from using a high fidelity implanted electrode array to reconstruct the desired end effector movement, as shown by [Vogel *et al.*, 2010], to using much lower bandwidth surface electrode based systems for tracking eye movement, as in [Postelnicu *et al.*, 2011]. Several authors have proposed using various facial EMG reading systems to do achieve cartesian end effector control, such as these [[Sagawa and Kimura, 2005; Gomez-Gil *et al.*, 2011; Ranky and Adamovich, 2010; Shenoy *et al.*, 2008]].

Although this approach is similar to joint level control in requiring continuous attention to a relatively large number of simultaneously, the user's control is directly in the task space. Control that is directly in the task space allows the user to more easily decouple the different controlled degrees of freedom. This makes it more practical to use facial muscles for control for a number of reasons. First, because facial muscles are close together, it is more complicated to separate the source of the input signals picked up by an electrode at a particular muscle site if several muscles are activated at the same time. Additionally, control over these muscles is generally less fine grained, as individuals generally don't use their facial muscles for fine manipulation. Being able to decouple the controlled degrees of freedom can allow the user to make reasonable progress towards their goal while closely attending to only one degree of freedom at a time, which makes predictable, fine grained control less important.

2.1.5 Discrete Mode Control

In discrete mode control, the user is able to switch between a set of predetermined configurations. The aformentioned control methods involve control over some continuous state space. These modes require continuous user attention, high data rates, and require a great deal of understanding of the robot itself from the individual. Additionally, because manipulators are have a large number of degrees of freedom, these continuous methods do not attempt to control them at all. Rather, they switch the manipulator between two simple discrete states, open and closed. In some paradigms, the user is able to switch between a number of different discrete modes that control the configuration of the hand[*Yang et al.*, 2009; *Woczowski and Kurzyski*, 2010; *Ho et al.*, 2011; *Cipriani et al.*, 2008; *Matrone et al.*, 2011].

These control schemes represent a tradeoff between flexibility and simplicity of use. This tradeoff is especially important for complex, multijointed robotic hands. Direct control over the fingers of a manipulator is not feasible for more complex hands for a number of reasons. There are generally many degrees of freedom, and the configuration of each finger is important to whether the hand will unintentionally interact with the environment as the end effector is used. Additionally, the fingers may have overlapping workspaces, so without extremely accurate control, it is likely that they will hit each other. By allowing the user to switch between discrete states, they are able to get utility out of hands that are more complex than the simple grippers that are generally used in the simple open/close schemes. More complex hands can more accurately approximate the surface of the object that the user wants to interact with, which increases the number of possible interactions and makes them more stable.

However, these schemes limit the user's flexibility to the preset configurations. Additionally, the user has to remember how to get to the configuration that they want to use at a given time, which may require multiple steps through a branching decision tree. In continuous control schemes, the user can make small changes and observe the outcome to get the robot to do what they want, but in discrete control scheme the result of the next input may not be related to the previous one. Because there is not necessarily an easy way of associating the path that they need to take in that decision tree with the goal they want to reach, these control schemes have a high learning curve.

2.1.6 Task Level Control

The key challenge of using noninvasive brain computer interfaces is that the bit rate is low and that the input is somewhat unreliable. In addition, the user experiences limited feedback, which makes direct control difficult. Under these conditions, it would seem intuitive that users would find task level control, where the user directs the robot on what to do but has little input as to how, would be more effective. Indeed, it has been shown that users find BCI control easier using even higher level, goal oriented paradigms [Royer *et al.*, 2011], and we have begun to see work that attempts to exploit higher level abstractions to allow users to perform more complex tasks with robotic arms.

In [Bell *et al.*, 2008], EEG signals were used to select targets for pick and place operations for a small humanoid robot. [Waytowich *et al.*,] used EEG signals to control pick and place operations of a 4-DOF St aubli robot. [M. Bryan, V. Thomas, G. Nicoll, L. Chang and Rao, 2011] presented preliminary work extending this approach to a grasping pipeline on the PR2 robot. In that work, a 3D perception pipeline is used to find and identify target objects for grasping and EEG signals are used to choose between them. In [Müller-Putz *et al.*, 2005], grasping is decomposed to a 4 stage pipeline where EEG signals are used to control transitions between stages. And in [Scherer *et al.*, 2011], the authors demonstrate an interface to navigate in two dimensions and select goals in a complex virtual environment and propose a hierarchical control scheme for learning high level tasks dynamically.

2.2 Robust Grasping

In this work, we will present users with appropriate grasps for the object. The system will present the user with options, and they will select the option that best reflects their intent. Because the user does not exercise direct control over grasp acquisition, we need to be able to present the user with grasps that the manipulator can acquire reliably in the face of sensor noise. Analyzing the robustness of a grasp in the presence of object localization errors and external force perturbations is an important and difficult problem, which has thus been studied by a large number of researchers following many different strategies.

2.2.1 Wrench Space Metrics

A key criterion that has been used to define a stable grasp is that the contact points of the grasp are able to generate a wrench in an arbitrary direction, termed force closure. This property is considered fundamental in grasp analysis because it provides a simple, necessary condition for grasp stability, as shown by [Salisbury and Roth, 1983]. This property is determined by the set of wrenches that can be produced from the contact points of the grasp, termed the grasp wrench space (GWS). Force closure provides a binary analysis of grasp robustness. In order to analyze the strength of the force closure, we commonly use the epsilon quality (ϵ_{GWS}), first described in [Ferrari and Canny, 1992]. ϵ_{GWS} is defined as the radius of the largest ball around the origin that fits in the convex hull of the grasps contact wrench space. This radius is the magnitude of the minimum norm wrench that will break the object free of the grasp.

2.2.2 Wrench Space Metrics Robust to Contact Location Uncertainty

In previous work, measures have been proposed to quantify the robustness of contact location based quality metrics, with particular focus on the ϵ_{GWS} . One well developed approach considers the effect of contact model or contact location uncertainty with respect to the object. The independent contact region approach (ICR) introduced in [?] considers the effect of contact location uncertainty on the ϵ_{GWS} of a grasp. This approach grows a convex set of acceptable contact locations for each planned finger contact on the object. Any set of contact locations chosen from this set produces a grasp whose quality metric is lower bounded by a preselected quality threshold. Faverjon et. al. developed analytical approaches to finding ICRs for complex 2D and simple 3D models[Ponce and Faverjon, 1991; Ponce *et al.*, 1996; Faverjon and Ponce, 1991]. In Roa and Suarez [Roa and Suarez, 2009], this approach is extended to complex three dimensional shapes and minimal thresholds on the ϵ_{GWS} . [Pollard , 2004] followed a similar approach with a focus on larger numbers of original contacts and with extensions to task wrench space quality metrics, and also presented a method for generalizing sets of ICRs between objects. Zheng and Qian proposed a quality metric based on a similar region growing approach by measuring the radius of the smallest ball in the configuration space of the regions produced for the grasp parametrized

by contact location, contact normal, or frictional coefficients[Zheng and Qian, 2005]. All of these approaches consider the effects of contact location uncertainty independently, but do not directly address the effect of calibration error and object pose uncertainty on the ϵ_{GWS} . These approaches assume that uncertainty causes the contacts to land near the planned contact points on the object. However, simulating the entire grasping procedure often produces very different sets of contacts in the presence of uncertainty. In fact, none of the ICR approaches described are able to fully account for the effects of these resulting deviations from the planned contact locations.

2.2.3 Simplified Analytical models of Robustness

Rather than analyzing the wrench space of the applied grasp, another approach to grasp synthesis is to model the fingers as a point or hemisphere where the contacts are represented as a set of spring-like elements connect the finger to the object. With this simplification, it is possible to analytically calculate the wrench applied to the object by the fingers. By using some optimization approach, it is then possible to find the joint torques that apply zero total wrench to the object while maintaining some minimum internal force which stabilizes the object through friction. By combining this approach with assumptions about the geometry of the fingertip and the smoothness of the contact location on the object, it is possible to use an analytical approach or an optimization approach to find regions with large stability boundaries using the same joint torques. Such models have been pursued by authors under some simplified conditions in 2D [Sugar and Kumar, 2000] or simple 3D models with simple contact formulations[Yamada *et al.*, 2001]. However, it is not clear how to generalize these results to arbitrary hands and objects. Additionally, these models require many parameters of the analytical model to be chosen ad hoc.

This methodology shows promise, but hasn't been demonstrated outside of these highly simplified situations.

2.2.4 Sampling Based Heuristics

Some work has addressed the effect of object pose uncertainty in grasping tasks for measures of grasp stability other than the ϵ_{GWS} . In Balasubramanian et al. the authors derive

a novel grasp measure for robust grasping from observations of human guided grasping [Balasubramanian *et al.*, 2010]. Bekiroglu et al. proposed a method to learn haptic features of grasp stability to assess grasps during their execution[Bekiroglu *et al.*, 2011]. The general problem of motion and manipulation planning robust to pose error was addressed by by Lozano-Perez et. al [Lozano-Prez *et al.*, 1984] using preimage-backchaining. In [Berenson *et al.*, 2009b], this method was extended to hand target pose regions for grasping objects on a table top. In [Brost and Christiansen, 1997], Brost and Christiansen investigated the issue of pose error robustness for a 2D gripper using a sampling based approach on a fixed trajectory plan in physical experiments. More recently, [Hsiao *et al.*, 2011] investigated Bayesian approaches to grasp planning under object pose and identity uncertainty for a simple gripper using a sampling approach to select robust grasps as measured by a grasp quality metric in simulation .

This approach is the most generic, simplest to understand, and the easiest to implement. We have followed a similar approach in analyzing the grasps produced by our grasp planner.

Chapter 3

Robust Grasping

3.1 Grasp Planning with Contact Points

Analytical models analyzing the probability of grasp success have not been found, and so far the most generic methods involve sampling. The configuration space of the “grasping” problem has several formulations, but the two most common are potential contact points or hand configurations. In analyzing potential contact points, one can either analyze points on the hand or on the object. There are numerous considerations that one can take into account that make intuitive heuristic sense, including geometric concerns such as local flatness.

Historically, it has been common to analyze the forces that can be applied at each contact point. There are no perfect models for this analysis. Models are generally classified by the physical systems that they roughly approximate. A point contact model assumes that the contact point can only apply forces in the direction opposite the normal of the surface on which it contacts. This is a conservative model. In order to generate a stable grasp from point contacts, it is necessary that any small (i.e. differential) motion cause the object to overlap the proposed contact points. This is called *form closure*. This is the equivalent of the positive span of the contact normals covering the entire space. This is an example of *form closure*. For point contacts, these constraints are *duals*, and one strictly implies the other. For the rest of the contact models which take friction into account, this is not the case - one can have *force close* without *form closure*.

This model is simple and conservative, but unfortunately it is too conservative to be use-

ful for arbitrary objects. If rotations are considered, cylindrical objects cannot be through *form closure*, because there is no contact normal that can oppose rotations along the length of the cylinder. More generic models include frictional forces in addition to pure normal forces. Point contact with friction models allow friction to be employed at each of the contact points in the plane orthogonal to the normal. In this case, the forces that can be produced at the contact point through friction are related to the normal force through the friction coefficient of the surface. This creates a cone of potential forces that can be created as the normal forces are scaled up.

Further, if the contacting surface is assumed to be ‘soft’ so that one surface conforms to another forming a contact patch, rather than a point, the contact can also apply torque around the normal direction. If the surface is significantly softer, the patch can be modelled as a set of points on the perimeter of the patch. These points will be able to apply torques around all three dimensions with respect to the original contact point, and forces along any direction in the positive halfspace of the surface normal.

In order to analyze how the contact forces effect the object, the forces and torques need to be summed up in the same reference frame. Most often, they are transformed to the centroid of the surface mesh or to the center of mass, if known. Transforming the cone of forces that can be produced by the point with friction model, because it is cannot be represented by a linear model, and so the linear transform to the new reference frame leads to a complex representation. In order to avoid this, the cone is represented by a simple linearization, making the base of the shape a regular polygon rather than a circle. Then the potential forces can be represented by the edges from the contact point to the points on the perimeter, normalized by the number of edges.

3.2 Grasp Wrench Spaces

To characterize the contribution of the contact wrenches to constructing stable grasps, we need a way to analyze the whole set of wrenches wholistically. Irrespective of the contact model used, so long as it is linearized, the contacts can be summed up in the same reference frame, and all potential forces that could possibly be generated by these contact points can

be represented. This set of forces is called the “Grasp Wrench Space.” In plain terms, the Grasp Wrench Space represents the set of wrenches that can be applied by this grasp. A key criterion that is often used to define a successful grasp is the ability to generate a wrench in an arbitrary direction, which is the generic definition of *force closure*. This property is considered fundamental in grasp analysis because it provides a simple, necessary condition for grasp stability[Salisbury and Roth, 1983].

3.3 The ϵ_{GWS} Quality Metric

The ϵ_{GWS} metric of grasping extends force closure from a binary property which is or is not satisfied to a measure of how stable a grasp is with respect to grasp forces. The ϵ_{GWS} measure is one of the most widely cited benchmark metrics in the grasp planning field[Ferrari and Canny, 1992]. Formally, for a set of contact wrenches $C \subset R^6$, one can define the neighborhood ball $B(\epsilon)$ and wrench space metric ϵ_{GWS} as follows:

$$B(\epsilon) = \{x \in R^6 \mid \|x\|_2 < \epsilon\}$$

$$\epsilon_{GWS}(C) = \max_{\epsilon} [B(\epsilon) \subseteq \text{convexhull}(C)]$$

Any grasp that does meet the minimum requirement of an $\epsilon_{GWS} > 0$ cannot apply force in some direction, which means that some infinitely small force in that direction will cause the object to begin to move. This quality metric is popular largely because it analytically addresses an intuitively necessary condition for grasp stability, the ability to resist force perturbations. One shortcoming of this measure is that it is sensitive to the choice of center of mass of the object. A variant of this measure is to consider the volume of $\text{convexhull}(C)$, denoted V_{GWS} in this work. This variant functions as an approximation of the average case ϵ_{GWS} over many estimations of the center of mass of the object.

3.4 Optimization Based Grasp Planning

Given a quality measure that is computationally tractable, it is feasible to create a stochastic optimization based grasp planner. Sampling potential contact points directly on the object leaves the problem of finding a hand configuration that matches that contact configuration

unsolved. For the generic case of arbitrary hands, this problem is difficult and unsolved. Similarly, sampling candidate contact points on the hand is equally intractable, because most hand configurations produce no contacts at all. To make the problem tractable, the state space needs to be relatively low dimensional and the quality metric must generate valid, informative values on most of that state space.

3.4.1 Potential Contact Quality Function

This lab has produced a planner that uses such a measure with simulated annealing based optimization. To make the quality metric accessible to this method and more computationally tractable, the potential contact points on the hand are projected to the object and the resulting contact wrenches are weighted by their distance and the agreement of their normal directions. This “potential contact” quality function creates a smoother optimization function.

3.4.2 Implications for pose error robustness

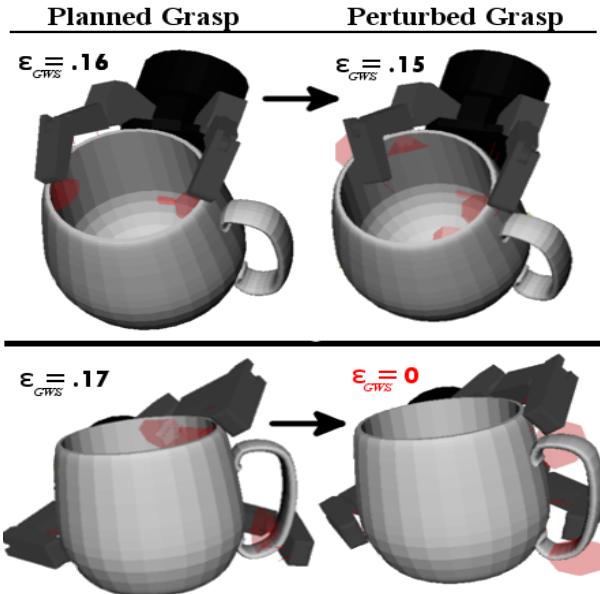


Figure 3.1: An example of a grasp where the ϵ_{GWS} is consistent with respect to the common localization errors, and one where it is not.

A stochastic optimization based approach is capable of escaping local minima in complex, nonconvex optimization functions like the projected distance and alignment between nonconvex shapes. However, the net effect of the optimization procedure is still to approximately follow a gradient. We should thus expect that the most optimal grasps that are found should usually be found in areas where the quality function is fairly smooth and the basin of attraction is wide. Given the quality function, especially the component which takes into account the relative alignment of the potential contact location and the hand, it follows that the local geometry should be expected to be relatively smooth, and that the quality function, when evaluated, should be robust to small errors in contact location.

However, in real robotic applications, if the object is known then the largest sources of contact location uncertainty are errors in perception of the location of the object itself with respect to the robot, miscalibration of the robot, and intrinsic uncertainty in control of the end effector's location. This implies that the errors in contact location are not randomly distributed, but are structured, in that they are offset in the same direction because the entire object is offset. In fact, the actual contact location is a product of the entire grasping procedure, in which a different part of the finger may make contact than is planned, and this contact location not be near the planned location. The normals may face a different enough orientation that the smoothness of the projected contact point's quality function may not be sufficient to guarantee that the actually achieved contact points produce a stable grasp. Some planned grasps may have nearly the same quality measure if the object's pose is slightly perturbed, while others may not. Fig. 3.1 shows an example of each.

3.5 Analyzing The Effects of Pose Error In Simulation

3.5.1 The Columbia Grasp Database

In order to test how robust the grasps that are produced by this planner are to localization errors, we need a large database of grasps. The Columbia Grasp Database (CGDB) had already been produced by [?] to seed a machine learning based approach to grasp planning. The CGDB was constructed by using the Eigengrasp planner on the Princeton Shape Benchmark, a set of 1814 models with class labels. Starting from this set of grasps,

we developed a simulation model to estimate the effects of localization error across many objects.

3.5.2 Grasping Pipeline

The CGDB parametrizes grasps as pre-grasp and final-grasp pairs. The pre-grasp is a collision free pose and joint angle set found by the simulated annealing planner. The final-grasp is the set of corresponding parameters found by applying a grasping heuristic to the pre-grasp that simulates closing the hand around the object until contact is made or joint limits are reached. The grasps in the CGDB are intended to be used by preshaping the hand to the pre-grasp under the assumption this is a collision free pose, but in the presence of uncertainty this may not be the case for all grasps. In this case, it is necessary to generate an initial pose which is certain to be collision free.

The general problem of optimal hand trajectory planning for grasping is complex. In this work we consider a basic three-phase grasping heuristic.

- 1a.** Move hand and fingers to an initial pose and posture far from the object along a pre-specified approach direction.
- 1b.** Move to the planned pre-grasp hand pose from the initial pose along the approach direction, stopping if contact is made.
- 2a.** Move fingers to their pre-grasp postures by linearly interpolating from initial posture, stopping each finger if contact is made.
- 2b.** (optional) For grasps with any contact points that are not on a distal most link of a finger, approach object to contact if none has occurred so far.
- 3.** Close fingers until each joint is constrained by contact or joint limits.

We use the normal to the center of the palm as the approach directions for all of the grasps considered. To determine the finger positions for the initial posture, we open the fingers so that the hand in the initial pose can wrap around itself in the pre-grasp position. We illustrate this strategy in Fig. 3.2. We open the hand at a constant rate for all joints until

the tips of each finger are outside of the projection of the second most distal link along the approach direction of the hand in the planned pre-grasp pose.

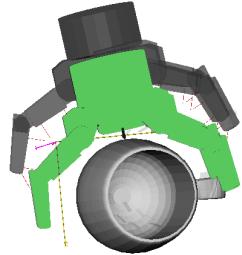


Figure 3.2: *Illustration of the initial pose and posture.* The hand is withdrawn along its approach direction and all of the fingers opened at the same constant velocity until each of the fingertips are outside of the knuckles of the hand in its pre-grasp configuration. The hand is then opened an additional ten percent of the remaining joint range. The pre-grasp configuration is shown in green.

3.5.3 Model of Pose Uncertainty

Because it is computationally intractable to densely sample the full six dimensional space of poses near the planned pregrasp pose, we consider a three dimensional error model representing an object on a support surface. We assume that each object is restricted to a set of stable poses on the surface. This error model is parametrized by $[x, y, \theta]$ as illustrated by Fig. 3.3. We use the centroid of the planned contact locations as the origin of the object because parameterizing the rotation around the center of mass of the object will have a disproportionate effect on grasps planned further from the center of mass. A reasonable range to explore using this parametrization should allow each parameter to move the relative position of the planned grasp's contact point by at least 10 mm. If we make the rough approximation that the projection of the contact points perpendicular to the axis of rotation lie on a circle with a 5 cm radius, a reasonable range for these parameters is $\theta \in [-20^\circ, 20^\circ]$ and $x, y \in [-10\text{mm}, 10\text{mm}]$. These bounds are motivated by our anecdotal experience in aligning a known object to a point cloud using common methodologies. Exploring this parameter space uniformly in increments of $1[\text{mm}, {}^\circ]$ in simulation requires

18,081 simulations per grasp. Using the GraspIt! grasping simulator [Miller and Allen, 2004] on a Xeon 2.67GHZ processor we can perform this many kinematic simulations of the grasping pipeline in under two hours for the worst case. The time required for each simulation is object and grasp dependent. With cluster computing this approach can be used to analyze large numbers of grasps from a database off-line.

3.5.4 Simulation Results

These simulations resulted in a large dataset of grasps, perturbations from planned positions, and associated quality functions. From this data, we sought to answer two questions. First, how well do the quality measures of the planned grasp predict expected grasp qualities after perturbation. Second, how do grasp qualities effect the probability of attaining a *force-closed* grasp in the presence of pose uncertainty.

3.5.4.1 Expected Grasp Quality

The grasps we analyzed were produced by the optimization procedure described in section . As we argued previously, it is reasonable to expect that the grasps produced by that planner should be locally near optimal with respect the hand position. The resulting planned grasps

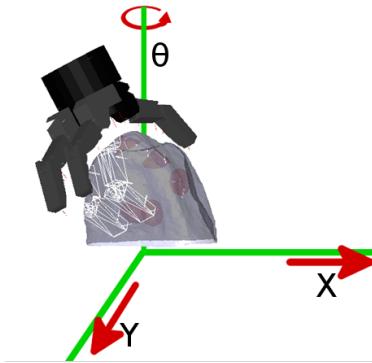


Figure 3.3: Illustration of the tabletop error model used in this paper. The object is can be translated $[-10, 10]$ mm in both X and Y and rotated $[-20, 20]^{\circ}$ around θ . The red cones mark the planned contact points. The white wire frame indicates the planned end configuration of the fingers.

are thus locally optimal with respect to the pose of the hand. To quantify how much the imposed pose perturbation effects a particular grasp, we can look at the expected value and standard deviation of its ϵ_{GWS} and V_{GWS} . However, to understand these effects on average across many grasps and objects, we normalize the expected value and standard deviation of these qualities by the quality values of the planned grasp with no error. We call the simulated quality measures of the planned grasp with no error the 'planned' quality values. Because the planned quality values are locally optimal, the expected value of the quality normalized to the planned value should have a maximum value of 1 for very robust grasps and a minimum value of 0 for extremely fragile grasps. We then consider the average of these quantities across all 480 grasps.

3.5.4.2 Probability of Achieving Force Closure

Another important measure that contributes to grasp success is how likely the perturbed grasp is to fail to be *force-closed*(*fc*), which is defined as having an $\epsilon_{GWS} = 0$. These grasps are expected to be fundamentally unstable, and so a grasp with a higher average quality but a larger variance, leading to more non-*fc* positions may succeed less often than a grasp with a lower average quality that has lower variance. To measure this effect directly, we counted the number of *fc* grasps of the perturbed samples and divided that by the total number of samples, giving us a measure we have called $P(fc)$, as it represents the probability of achieving a force closed grasp given that our sampled perturbations are uniformly likely.

Calculations involving convex hulls, such as the ϵ_{GWS} are known to be sensitive to numerical inaccuracies, and the cluster we used to generate this dataset was heterogeneous in both hardware and software environments. This initially lead to grasps that appeared *fc* with a small ϵ_{GWS} values to appear non-*fc* when evaluated on a different machine in the cluster. To achieve consistent results, we had to chose an arbitrary small positive δ that is larger than the cross-machine variability that we observed, and redefine *fc* to require that $\epsilon_{GWS} > \delta$. In this work, we arbitrarily chose $\delta > 0.001$, as this was several orders of magnitude above the largest discrepancy we ever observed between two machines analyzing the same grasp, and several orders of magnitude lower than the quality of the average unperturbed grasp that we evaluated.

Measure across sampled perturbations	Average across 'Tool' grasp dataset
normalized expected ϵ_{GWS}	0.5091
normalized std(ϵ_{GWS})	0.2915
normalized expected V_{GWS}	0.679
normalized std(V_{GWS})	0.3588
$P(fc)$	0.7957

Table 3.1: Summary of Simulation Results: These results quantify how robust each quality measure is to pose uncertainty. For each grasp, the expected value and standard deviation of the respective quality measures are calculated across the poses sampled. These measures are normalized by dividing them by the planned quality of the unperturbed grasp to allow comparison of the effects of uncertainty across grasps. $P(fc)$ denotes the measured probability of achieving a force closed grasp, which we have calculated as $P(\epsilon_{GWS} > .001)$

We calculated these statistics across all of our grasps. These results are summarized in Table 3.1. We see that the expected ϵ_{GWS} and V_{GWS} are 50% and 67% of their planned values, respectively. This shows that pose uncertainty has a large effect on the expected value of these measures. Additionally, we see that there relatively high variance of these measures across all of the sampled perturbations for a single grasp. The average standard deviation of the measures across the perturbed samples of a single grasp is 29% of the ϵ_{GWS} and 35% of the V_{GWS} .

The observation that the within grasp variability is high is what led us to consider whether the number of non- fc perturbed samples varied significantly across grasps with similar ϵ_{GWS} qualities. We found that the average $P(fc)$ is around 80%. This implies that even the best five grasps for the object should be expected to fail 20% of the time, if we only filter grasps by their ϵ_{GWS} quality.

3.5.5 Chosing optimal grasps: ϵ_{GWS} vs $P(fc)$

In the case of chosing a single grasp out of a set of many, we are faced with the problem of ranking grasps. The fundamental question we seek to answer with these research can be

rephased simply as whether performing that ranking relying solely on the ϵ_{GWS} is sufficient to chose the grasp most likely to achieve stable *force – closure* in the presence of location uncertainty. .

In Fig.3.4, we have plotted the planned ϵ_{GWS} against the estimate of the probability of achieving force closure, for each grasp. $\overline{\epsilon}^*_{GWS}$ indicates the average ϵ_{GWS} for the best grasp available for each object, which is 0.095. For most objects, the best grasp by ϵ_{GWS} lies in the range where $\epsilon_{GWS} \in (0, 0.2)$, bordered by the red line on the right of the figure. This figure shows that there is no correlation between ϵ_{GWS} and $P(fc)$ in that range.

This implies that the ϵ_{GWS} cannot be used to chose grasps that are expected to be robust. On average the best grasp by $P(fc)$ for each object has a $P(fc)$ which is .21 greater than the $P(fc)$ of the highest ϵ_{GWS} grasp. Under the error model described, choosing the highest $P(fc)$ grasp will result in a force closed grasp 21% percent more often than choosing the best grasp by ϵ_{GWS} . This analysis yields approximately the same results for the V_{GWS} in comparison to the $P(fc)$.

3.6 Physical Experiments

These simulation results show that using $P(fc)$ as a quality metric for ranking grasps from a database may select grasps which are more reliable in the face of object pose estimation error. To test whether these simulation results can be verified against physical experiments, we attempted to grasp ten objects for which we had high quality mesh models readily available courtesy of the the DARPA ARM-S project and the Willow Garage object database [23]. We added these models to our database, and calculated the $P(fc)$ for the top five grasps by ϵ_{GWS} value.

To test these grasps on a real robotic grasping system we used a Barrett 280 model hand attached to a Staubli TX60L 6 degree of freedom arm to implement the grasping pipeline described in section II-A. To emulate the friction coefficient used in our simulations, we wrapped the contacting surfaces of the hand in rubberized shelf liner. One object attempted was a blue detergent bottle made of very smooth plastic. We added a rubberized tape to the contact locations on this object to increase its friction coefficient. For each object, we

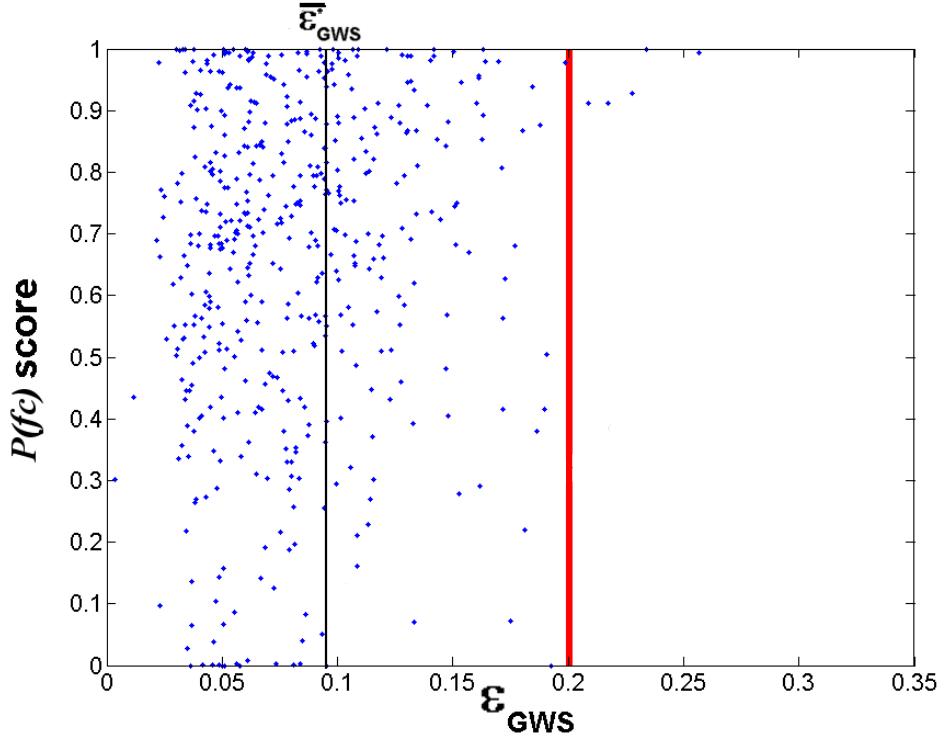


Figure 3.4: ϵ_{GWS} plotted against $P(fc)$, which is equivalent to the probability of attaining a force closed grasp under our error model from a planned starting grasp configuration. 480 grasps over 96 objects are presented in this figure. This figure demonstrates that the ϵ_{GWS} is generally a poor predictor of pose error robustness as measured by the $P(fc)$. The mean ϵ_{GWS} , denoted by the black line, is 0.095. There appears to be a cutoff, denoted by a red vertical line on the graph, around $\epsilon_{GWS} > 0.2$, after which all available grasps have a high $P(fc)$, but note how few grasps there are in the database with quality values that high. Only five objects have a grasp with a quality that high, which implies that for most objects, ϵ_{GWS} cannot be used to predict $P(fc)$.

selected the best grasp ranked by ϵ_{GWS} and the best grasp ranked by $P(fc)$. The objects were calibrated to the robot using a printed template of the bottom outline of the objects. We align the edge of each template to a known coordinate system in our robots workspace. We selected a perturbed position by sampling our error model and then applying the grasp pipeline as though the objects origin was at the perturbed location.

We tested up to ten randomly selected perturbed positions for each grasp. The same perturbations were applied to both of the tested grasps for each object. We moved the arm and hand at very slow speeds to simulate quasi-static conditions. In the first two stages of the pipeline, we moved the arm at 1% of its maximum speed, and the fingers of the Barrett hand at 10% of their maximum speed. In the final closing step, initial contact had been achieved or joint limits have been reached for each finger. We then raised the finger speed to 50% to drive the underactuated fingers to their final positions. After the fingers stopped moving, we lifted the object 3 cm perpendicular to the tables surface. If any part of the object was still touching the table when the arm came to a stop, we graded the trial as a failure, otherwise we graded the trial as a success.

We find that the grasps ranked best by the $P(fc)$ are successful more often. The results of this experiment along with the grasps and objects analyzed are in Fig. 3.5 and Fig. 3.6. Overall, for these ten objects the $P(fc)$ ranking selected successful grasps on 85 out of 93 trials (91%), whereas the grasps selected by the ϵ_{GWS} ranking succeeded on 63 of 93 trials (67%). For some objects, nearly all of the grasp attempts were successful irrespective of the $P(fc)$ score. We separated the objects in our experiment into two groups. One group is composed of larger, more complex objects shown in Fig. 3.6 and the other group is composed of light, roughly cylindrical objects shown in Fig. 3.5. We see that all of the objects in the roughly cylindrical category were grasped very robustly irrespective of $P(fc)$ score. For these objects, the Eigengrasp planner finds grasps which more or less encompass the object. In contrast, for the five objects which could not be enveloped by the hand, we see that the $P(fc)$ ranked grasp is successful in 35 out of 43 trials (81%), as compared to 16 out of 43 trials (37%) using the ϵ_{GWS} grasp. These results show that the $P(fc)$ ranking chooses more successful grasps; especially on the larger, more complex objects where the success rate is doubled.

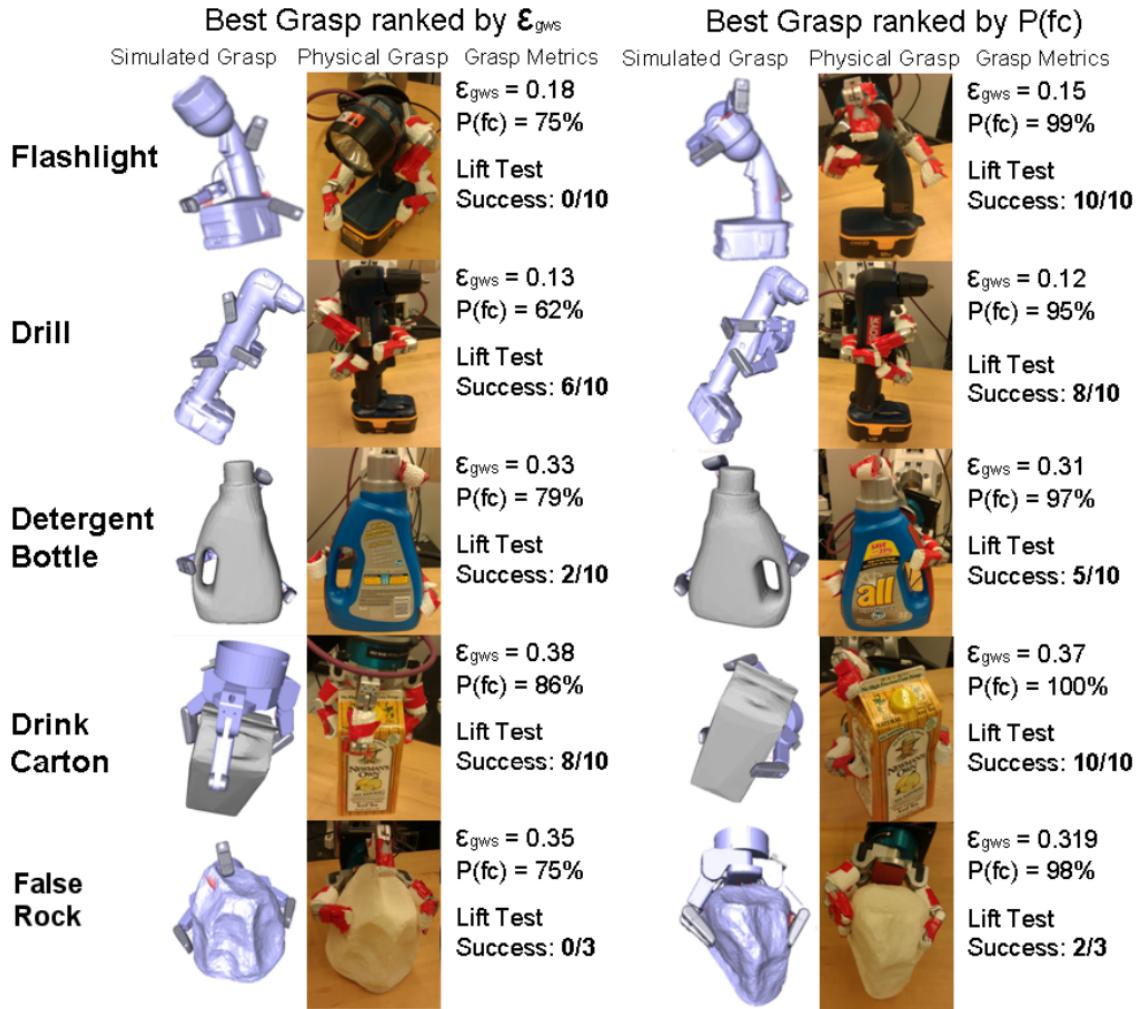


Figure 3.5: *Results A*: The results of physical experiments on five large, complex objects. In the left column are the best grasps for each object ranked by ϵ_{GWS} . An example of the simulated and physically realized grasp is illustrated for each grasp attempted. Additionally, the figure gives the $P(fc)$ and ϵ_{GWS} for each grasp attempted, as well as the fraction of attempts which succeeded in lifting the object by 3 cm in physical experiments. The higher $P(fc)$ grasp is successful on many more trials than the higher ϵ_{GWS} grasp for these objects.



Figure 3.6: *Results B*: The results of physical experiments on five smaller, simpler objects. In the left column are the best grasps for each object ranked by ϵ_{GWS} . In the right column are the best grasps by $P(fc)$. An example of the simulated and physically realized grasp is illustrated for each grasp attempted. Additionally, the figure gives the $P(fc)$ and ϵ_{GWS} for each grasp attempted, as well as the fraction of attempts which succeeded in lifting the object by 3 cm in physical experiments.

3.7 Discussion

In this work, we have shown that the planned ϵ_{GWS} of a planned grasp is not predictive of the probability of achieving a force closed grasp in the presence of uncertainty, neither in simulation nor physical experiments. To analyze this in simulation we generated the measure $P(fc)$ as an approximation of this probability. We then showed that the $P(fc)$ can itself be used to rank grasps from a preplanned grasp database and that this reranking predicts grasping success in physical experiments better than the ϵ_{GWS} .

The analysis presented here is less effective at discriminating fragile grasps on smaller objects than it is on larger objects. For small objects, the object is fully encompassed by the hand, and the highest ϵ_{GWS} grasp is always an enveloping powergrasp, and are always spherical, as described in [?]. As seen in 3.6, in such grasps the palm behind is behind the object, one finger pressing downwards, with each finger roughly evenly spaced apart. For a hand capable of achieving this configuration, it is always a high ϵ_{GWS} grasp, because the normals of the contact points are evenly spaced around the center of the grasp, which maximizes their ability to directly oppose perturbation forces. It is also a configuration which is relatively invariant to perturbations that do not move the object outside of the working envelope of the hand.

Under more constrained conditions, such grasps are not always attainable. The object may not be in the center of the workspace, there may be objects in the way, or the desired purpose to which the object will be put requires a different grasp. These cases are better represented by the results on the large objects, which couldn't be fully encompassed by the hand. Under these conditions, the higher ϵ_{GWS} grasp often fails, and we see better results ranking by $P(fc)$

3.7.1 Online Extensions

Although the full sampling rate is too computationally expensive, in Tab. 3.2 we examine the possibility of approximating $P(fc)$ at lower sampling rates. By using a sampling step size of (10mm, 10mm, 10 °), we see an average difference of 6% with respect to the denser sampling. In none of our samples did this change the order of our rankings. At this sampling

Sampling Step Size (X, Y, θ) (mm, mm, $^{\circ}$)	Mean Difference	Standard Deviation	Max Difference
2-2-4	0.0128	0.0122	0.0667
2-2-4(random)	0.0097	0.0086	0.0493
10-10-10	0.0653	0.0551	0.3205
10-10-20	0.0840	0.0713	0.4782

Table 3.2: This table quantifies the effect of sampling parameters on the estimate of quality measure. The first column describes the sampling density along each dimension of the error model. The other columns describe the average change in the subsampled $P(fc)$ as compared to the densest sampling rate. These results demonstrate that lower sampling rates may feasibly be used to approximate the $P(fc)$ at the densist sampling rate without unacceptable loss of accuracy.

rate, we can generate an estimate in under a second, which make it suitable as a filter for an online grasp planning system. At lower sampling rates, the rankings do become reordered, which may lead us to select less robust grasps. In our shared control online planner, we use the lower resolution $P(fc)$ to weight the grasp quality measure that controls the order in which grasps are tested for reachability, and the order in which they are presented to the user. Without this filter, many grasps that are presented to the user are fragile to object pose estimation errors.

Chapter 4

A Grasp Planning UI for Facial Gestures

4.1 A BCI input device prototyping

In order to develop a full prototype system for a grasping system usable by a disabled individual, we developed an initial test bed with a number of test objects using an industrial Stuabli robotic arm, a BarrettHand gripper, and an object localization system based fiducials and a camera. We sought an input device that might be representative of what could be achieved by a low cost BCI device in order to evaluate a user interface under realistic conditions of error and throughput. A number of low cost wireless EEG devices began entering the consumer market in 2009. Two EEG headsets we evaluated as potential EEG devices were the Neurosky Mindwave and the Emotiv Epo.

4.1.1 Mindwave

The Mindwave is an approximately \$100 headset which has a single dry electrode which is reported to measure a one dimensional axis of voluntary control that the developers map to “focus” and “relaxation.” The main advandages of this device are it’s low cost and ease of application. Unfortunately, we were not able to develop any to extract useful signals from the Mindwave device and abandoned attempts after several hours of training using

the vendor provided software failed to produce results among three researchers in our lab.

4.1.2 The Emotiv Epoc

The Emotiv Epoc is an approximately \$400 headset with 14 wet electrodes. Of the 16 electrodes, 14 are positioned at the International 10-20 locations corresponding to AF3, AF4, F3, F4, F7, F8, FC5, FC6, P7, P8, T7, T8, O1, O2. The remaining 2 channels are references and are positioned at P3 and P4. Each electrode has a sampling rate of roughly 128 hz and a resolution of 16 bits (14 bits effective). While substantially more invasive than the Mindwave, this device produces more signals with higher fidelity. The api provided does not allow access to the raw signals produced by the Emotiv, but there are a number of classifiers that are made available, separated into three suites representing modalities. These classifiers are essentially black boxes, but we can describe their stated purpose.

4.1.2.1 Classifier Suites

Affectiv Suite The first modality is similar to that of the Mindwave, but measuring five different channels related to emotional state - engagement, frustration, meditation, instantaneous excitement, and long term excitement. These channels vary rather slowly and are not well suited to use as an input method. It is also not clear that they can be made subject to voluntary control.

Expressiv Suite The second modality interprets the input signal as facial movements, reporting events such as “wink”, “jaw clench”, and “smile.” This interface doesn’t allow any training of the classifier, but does allow a threshold to be set for the action to be reported. Multiple actions may be reported at the same time, and there is some amount of crosstalk between each modality.

Cognitiv Suite The second modality claims to measure some mental action command. Unlike the other two modalities, this modality has a training interface that the subject can use to train the classifier on up to four independent commands. The training interface for the classifier for this modality allows the user to effect the position of a block in space using verbs such as ‘lift’, ‘push’, and ‘pull’, and is thus presumably

analogous to the imagined motion paradigm presented in other literature. As the number of trained commands increases, the difficulty of training additional commands also increases.

4.1.3 UI Design

The Cognitiv Suite appeared ideal, as it allows up to four user elicited output options with analog output levels. Of the three researchers developing the UI, none of us were able to achieve reasonable classification rates using mental imagery even with a single option. We abandoned the EEG modality in favor of using facial gestures. The Expressiv Suite could provide reliable detection of jaw clenching, but other facial gestures could not be reliably detected without interfering with casual breathing, blinking, and speaking. However, we could train the Cognitiv Suite classifier on up to three facial gestures relatively quickly, with the fourth gesture classified by the Expressiv Suite. While these outputs are only discrete values, we could train some gestures to be easier to maintain, so that they can be used as continuous control signals.

In this work, GraspIt! is used both to plan grasps and to present the user with feedback. We augmented the Eigengrasp Planner GUI with a visualization of the grasp planning scene that includes a number of guides and fiducials that allow the user to guide the planner fully inside the simulator. The augmented grasp planning scene is illustrated in ??.

4.1.3.1 Grasping Pipeline

With these constraints in mind, we broke the process of grasping on object down into a multiple stages that could be controlled using four inputs - two continuous control signals and two discrete signals. These stages are outlined in Fig. 4.1.

The grasping pipeline is divided up into four stages: object identification and alignment, grasp planning, grasp review, and grasp execution, which are described below. This pipeline is controlled using only four facial gestures. The use of these gestures in each stage of the pipeline is explained in Table 4.1. In general, gesture 1 serves as a click and moves the user through the pipeline. The exception to this is at points of decision for the user. In these cases, gesture 2 serves as the YES option and gesture 1 becomes NO and returns

Phase Gesture	Run Planner	Review Grasps	Execution
1	start/stop planner	cycle through grasps	restart
2	n/a	select grasp	confirm grasp
3	rotate around x axis	n/a	n/a
4	rotate around z axis	n/a	n/a

Table 4.1: This table shows the initial control flow of the grasping pipeline using facial gestures detected by the Emotiv EPOC

the user to an earlier point in the pipeline. Because false positive readings of gesture 1 and 2 have strong consequences, we found that both are best associated with a concise and strong gesture such as closing one eye or clenching the jaw. Gestures 3 and 4 control the approach direction of the hand relative to the object during the grasp planning stage of the pipeline. These gestures can be maintained to generate continuous motion of the hand over two degrees of freedom and therefore are best associated with gestures that can be contracted for several seconds without too much twitching or fatigue.

Object Localization: The Eigengrasp planner requires a complete description of the geometry and location of the target object. The object localization system we have incorporated is uses a depth camera as input and is described in greater detail in 4.1.4.3. In the initial work, we used a single unoccluded object, and assumed that the recognition system works perfectly, so no user interaction is required.

Grasp Planning Phase: Grasp Planning Phase: Once the object is localized, we center the hand around the object and initialize the Eigengrasp planner. In this phase the user can move the hand around the object using gestures 3 and 4, which allow “continuous” input. Each gesture moves an “input hand” around a corresponding axis guide, which allows the user to visualize how the hand will move. Gesture 3 rotates around the X axis of the object, while gesture 4 rotates around the object’s Z axis. When the planner is run, the demonstrated pose is used as a part of the objective function for the simulated annealing planner. Gesture 1 starts the planning thread. The demonstrated pose can still

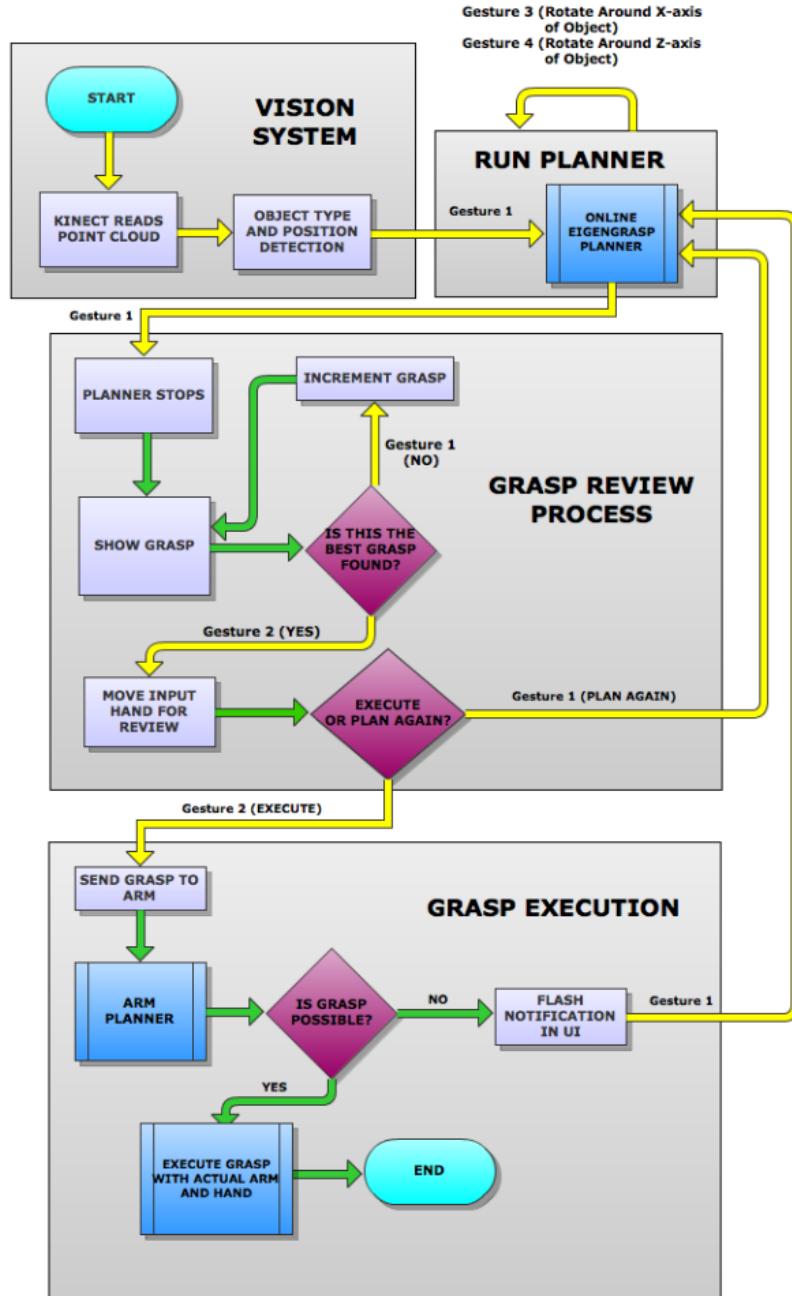


Figure 4.1: This diagram shows the initial Emotiv Epoc based grasping pipeline. An overview of the grasping pipeline. In the first phase, the system uses the kinect to identify and localize the object in the scene. In the second phase, the user guides the planner to an appropriate approach direction. In the third phase, the user stops the planner and reviews the available grasps. In the fourth phase, the user sends the grasp to the robot for execution.

be adjusted while the planner is running, however it can take some time to adjust to the demonstrated approach direction, so it is beneficial to start from a reasonable pose. While the planner is running, the “input hand” is made 30% transparent. A second, fully opaque “best solution hand” is rendered showing the highest quality grasp from nearby the current approach direction. To give the user feedback about the progress of the planner, a third “planner hand” is rendered showing the last tested state. The user interface presented while the subject is running the system is presented in 4.2 and 4.3. While the planner is running, the top ten grasps are stored. When the user sees the “best solution hand” in a position they believe will produce a stable grasp, they can stop the planner with a second instance of Gesture 1.

Review Phase: Once the planner is stopped the user has an opportunity to review the list of best grasps and choose one for execution. As in the previous stage, the solution hand is used to display the grasps to the user. At any point, the user can select a grasp which removes the “solution hand” from the world and closes the “input hand” into the chosen grasp. Now the user can evaluate the grasp more closely and examine the quality metrics for the grasp. If the user is satisfied, they can confirm selection of the grasp and send it to the next stage of the pipeline. If the user does not want to execute any of the found grasps, they can select the grasp that is closest to what they have in mind and restart the planning process.

Execution Phase: Once the robotic control software receives the selected grasp which is planned relative to the objects coordinate system, the arm planner decides if that grasp is achievable given the kinematic constraints of the arm and the vision systems estimate of the objects position in the world. If the planner cannot find a solution to execute the grasp, it will send a message back to GraspIt! which then notifies the user. If this occurs, the user can restart the planning phase to find a new grasp. If the planner can find a solution, then the grasp is executed with the actual arm and hand.

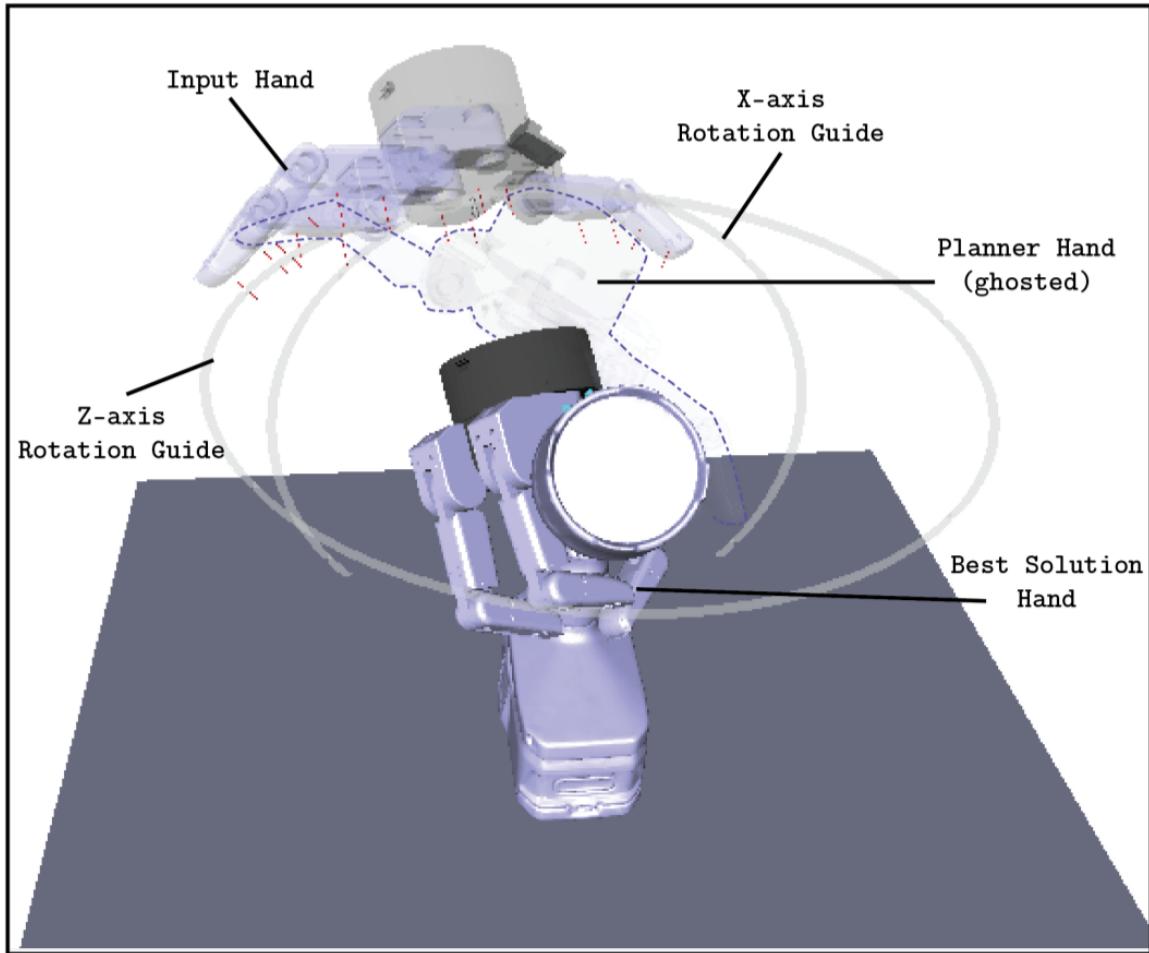


Figure 4.2: An illustration of the grasp planning user interface in GraspIt!. Here we demonstrate the three hands of the system. The Planner Hand, which is most transparent hand, demonstrates the current state of the planner. The Input Hand, which is of intermediate transparency, is the hand through which the user directs the planning system. Here you can see the rotational guides which allow the user to visualize their available control directions. The Solution Hand, which is fully opaque, demonstrates the best grasp currently available. This is the grasp which is closest to the approach direction that the Input Hand is demonstrating and which also has the minimal grasp energy.

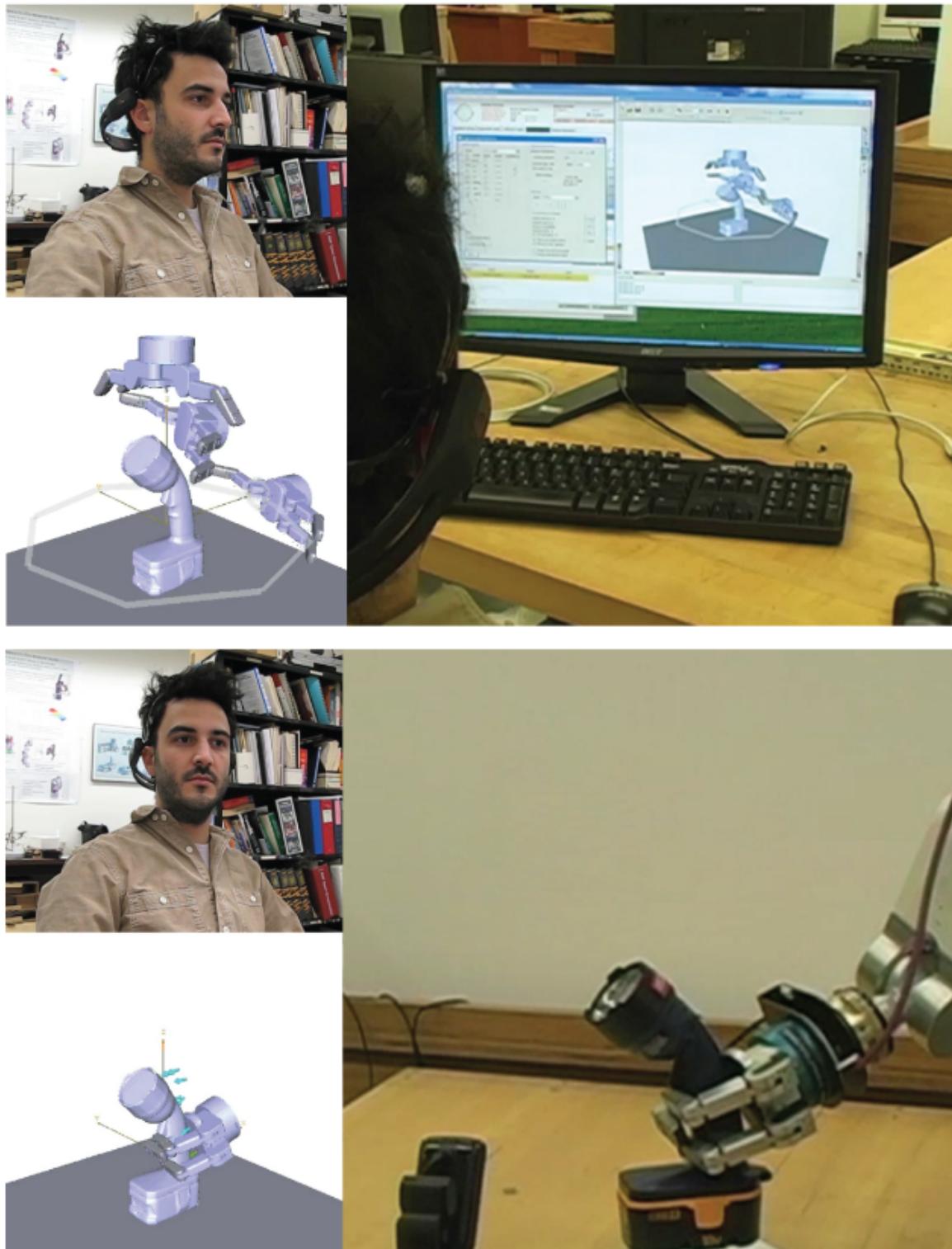


Figure 4.3: Two images captured as a user grasps an object with this system. Top: Clockwise from the top-left are an image of the user wearing the Emotiv headset and operating the system, the computer monitor with user interface, and a screen capture of the simulation. Bottom: User watching the robotic arm implementing the selecting grasp as shown in the simulator window on the lower left.

4.1.4 Grasping Platform

4.1.4.1 Hardware

In this initial work, our grasping platform is comprised of a 280 model BarrettHand mounted on a Stäubli TX60L 6-DOF robotic arm. The Stäubli arm is an industrial arm which we have found to be accurate to within 3 mm in our setup. This is more accurate than a lower cost, portable arm is likely to be, so the only uncertainty in this experiment comes from the vision system. A Microsoft Kinect sensor is used to generate point clouds of the object.

4.1.4.2 Planning and Kinematics

Planning for the motion of the arm is done in OpenRave using a bidirectional random tree planner in [Berenson *et al.*, 2009a], and small linear motions near the object are planned using the built-in inverse kinematics planner.

4.1.4.3 Recognition System

The raw point cloud data is gathered by a Microsoft Kinect. In this work, we assume the target is a member of a known set of ten objects. We use the method described in [Papazov and Burschka, 2011] to identify and localize the target object in the scene. Briefly, this method generates features from pairs of oriented points on the surface of the object. Prospective models are processed offline and put in to a hash table. Features are sampled from the sensor data and tested for collision in the hash table. If a sufficient number of collisions occurs with points on the same model, a varient of RANSAC is used to test the hypothesis that a set of points in the sensor data corresponds to a particular model at a particular location. This method has demonstrated a high robustness and is extensible to multi-object scenes.

4.1.5 Validation

4.1.5.1 Training

In order to validate this system, we had one of the researchers train the Emotiv Cognitiv suite for on three facial gestures for 10 successive training periods each. Each training period

lasts eight seconds. During some sessions, the subject either was not able to maintain the facial gestures due involuntary motions such as coughing, and these training sessions were discarded and repeated. During training, the subject is given visual feedback via the motion of a cube and a power meter both of which represent the strength of current classification. After the Cognitiv Suite gestures are trained, we set the threshold for detection of jaw clenching in the Expressiv Suite to a level that does not trigger for the other facial gestures.

4.1.5.2 Grasping Experiments

The subject used the system to grasp and pick up five common objects: a flashlight, a flask, a bottle of laundry detergent, a bottle of shampoo, and a canister of shaving cream. For each setup, the Stäubli arm starts such that it is completely vertical with a the BarrettHand at the top. A simulation world consisting of the hand, the object, and a surface is presented to the subject who can move the input hand and start the planner at any time. For each object, the subject executed grasps from two different approach directions. The subject was able to grasp all of the objects successfully from either of the two directions.

The resulting grasps are shown in 4.4. In each case, we demonstrate the grasp planned in simulation and the final grasp achieved by the physical robot. We found that using this grasp planning environment, we were easily and reliably able to grasp objects. In each case, the physical grasp was successful on the first attempt. Planning the grasp took on the order of 10-30 seconds. The review process took an additional 10-15 seconds. Finally, the arm motion trajectory planning took 30 seconds.

4.1.6 Conclusions

These results showed significant promise in extending the Online Eigengrasp Planner to assistive devices. However, this initial work showed only that an expert user with experience in robotics and GrapIt! was able to use the system. Additionally, the grasping scenario was rather simplistic, involving only a single object in free space. This interface is also only suitable for grasping known objects. From this initial work, we wanted to extend to more realistic grasping scenarios incorporating multiple, possibly unknown objects and validate the UI on a more general population.

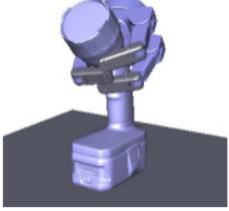
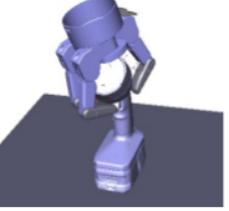
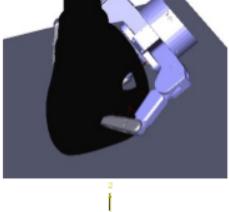
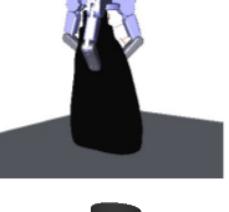
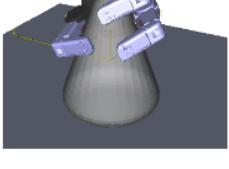
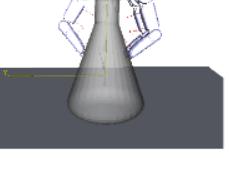
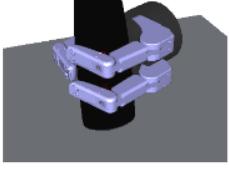
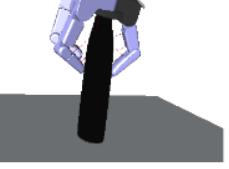
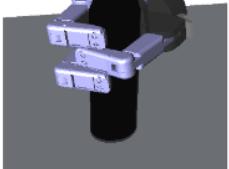
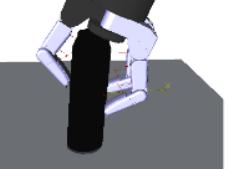
Object	Results			
	Approach 1		Approach 2	
	Simulation Grasp	Physical Grasp	Simulation Grasp	Physical Grasp
Flashlight				
Detergent Bottle				
Flask				
Shampoo Bottle				
Shaving Cream				

Figure 4.4: The results of ten different attempts to grasp five objects using two different user-selected approach directions per object. In each case, the user was able to generate an appropriate grasp using the simulator which was then transferred to the robot.

4.2 Handling Novel Objects

The parameters of the object recognition system described in [Papazov and Burschka, 2011] can be tuned to recognize objects with similar parts. To allow the user to grasp objects which are not in the database, we rely on the user to act as a filter to reject grasps that the planning system cannot know are inappropriate because it does not have the exact object model. An example of this alignment can be seen in Fig. 4.5a. In order to allow the user to discern how well the detected object aligns to the true geometry of the novel object in the scene, the UI has been modified to include a downsampled point cloud of from the depth camera. Additionally, since the user will need to exercise more judgement as to which grasps are reasonable, we added a display that shows all of the grasps available. This augmented UI can be seen in Fig. 4.6

4.2.1 Incorporating a grasp database

One useful aspect of mapping the object in the scene to a limited set of objects from a database is that we can also preplan a set of grasps for each object. This provides a method for the user to skip the slower online planning phase, since we will already rely more on the user and less on the automated grasp quality analysis. Additionally, it is more important to start the grasp planner from a particular conformation, since this allows the user to restrict the planner to the region of the proxy object that best conforms to the true object in the scene.

Using a grasp database also allows us to manually design good grasps for particular affordances. Such affordances are not generally grasped proficiently by generic grasp planners. Fig. 4.7 demonstrates such a grasp. In experiments, this grasp was successful 100% of the time, but only because the object deforms to allow the finger to pass through the whole in the handle region of the bottle. A simulation that can incorporate this behavior in any kind of real-time online planning system is beyond the state of the art. Where such grasps are available and match up with the object geometry, they are very useful.

To seed the grasp database, we ran the Online Eigengrasp planner six times for twenty minutes a piece with the approach direction of the palm aligned to the major axes in the

positive and negative directions, using the best grasp from each direction in the database. If there were fewer than ten grasps in the database, including manually inserted grasps, then the highest quality grasps were selected from among all of the available grasps until a full ten are available.

4.2.1.1 Pipeline Version 2

These modifications require additional user interaction. Whereas the our initial experiments assumed the vision system would succeed, when the true object may not be present the output of the vision system is much more variable. Because the vision system is stochastic, running the system again may present the user with different options. The second version of the pipeline allows the user to keep rerunning the vision system until a reasonable match is acquired. As seen in 4.6 where the object is detected upside-down, the alignment found does not need to reflect the semantic or geometric reality perfectly, as the user is presented with many options and can exercise their own judgement. Incorporating the database also allows us to short circuit the online planning phase when possible, which requires it's own phase. Finally, to guard against false positives in the grasp confirmation stage sending inappropriate grasps, we added a separate review and confirmation stage which returns to the beginning of the online planner stage. This new six stage pipeline is illustrated in Fig 4.8, with the user input controls in each stage mapped in Tab. 4.2. You can see that this table is less sparse than the first version as the pipeline is more complex. Because of this added complexity, we have added a new pipeline guide window, seen between the grasp views and the grasp planning scene in 4.6. In this window the left side of the window shows the current phase in yellow, the center of the window shows the result of Gesture 1 in red, and the right side of the window shows the result of Gesture 2 in green.

The second version of the pipeline differs from the version one in several of the states: *Object Localization*: This stage now accepts user input, allowing the user to send Gesture 1 to run the object recognition system again and attempt to find a better fit. All data from the previous iteration is discarded. Gesture 2 accepts the current fit and moves on to the next phase.

Database Planning: This phase is inserted before the planner initialization phase. Once

the object is identified, the planner loads a set of pre-planned grasps from a database. These grasps are presented to the user in the grasp view window. Gesture 1 allows the user to browse through the list of grasps and visualize them in the larger main window. The user is able to chose the grasp that best reflects their intent, and then signal acceptance of this grasp with Gesture 2.

Planner Initialization: Having selected a grasp from the initial set of pre-planned grasps, the user can chose to either execute this grasp using Gesture 1, shortcutting the grasp planning phase and proceeding to the grasp Confirm Grasp phase, or they can run an automated grasp planner using the selected grasp as a guide using Gesture 2. By choosing one of the grasps from the database and skipping straight to the confirmation phase, the user can significantly reduce the amount of effort required to grasp an object. To choose grasp n, only $n+4$ inputs are required.

Online Planning: This phase remains the same as in version 1, except that Gesture 1 iterates through the grasp list and shows that grasp in the grasp planning scene using the “Current Grasp Hand”. Whichever grasp is currently being shown is the one that is selected when the review phase is entered.

Grasp Review: This phase is the same as version 1.

Confirmation: This phase is new in version 2. In this phase, Gesture 1 will restart the planner in the online planning phase, while Gesture 2 executes the grasp. This additional phase adds robustness to false positives in the grasp review stage.

Execution: This phase is the same as in version 1.

4.2.2 Experiments

4.2.2.1 Task

In order to test the efficacy of our system, we asked five subjects to grasp and lift three objects using an Emotiv Epoc, a low cost, noninvasive commercially available EEG headset, as input. Two of the objects, a flashlight and a detergent bottle, were in the database and used for the vision system and one object, a small juice bottle, was novel. Each subject was asked to perform two grasps, one from the top of the object and one from the side of the object. Each grasp was repeated three times. For the novel object, subjects were simply



(a) Point clouds with RGB texture from the vision system. On the left is a flashlight along with its aligned point cloud in white. On the right is the point cloud of a juice bottle along with the best model from the vision systems object database, a shampoo bottle, in white.



(b) The two objects which are aligned on the right of the figure above. The shampoo bottle is in the object database. The juice bottle is not. The two are roughly the same width, and this the shampoo bottle can be an appropriate proxy for the juice bottle in the planner.

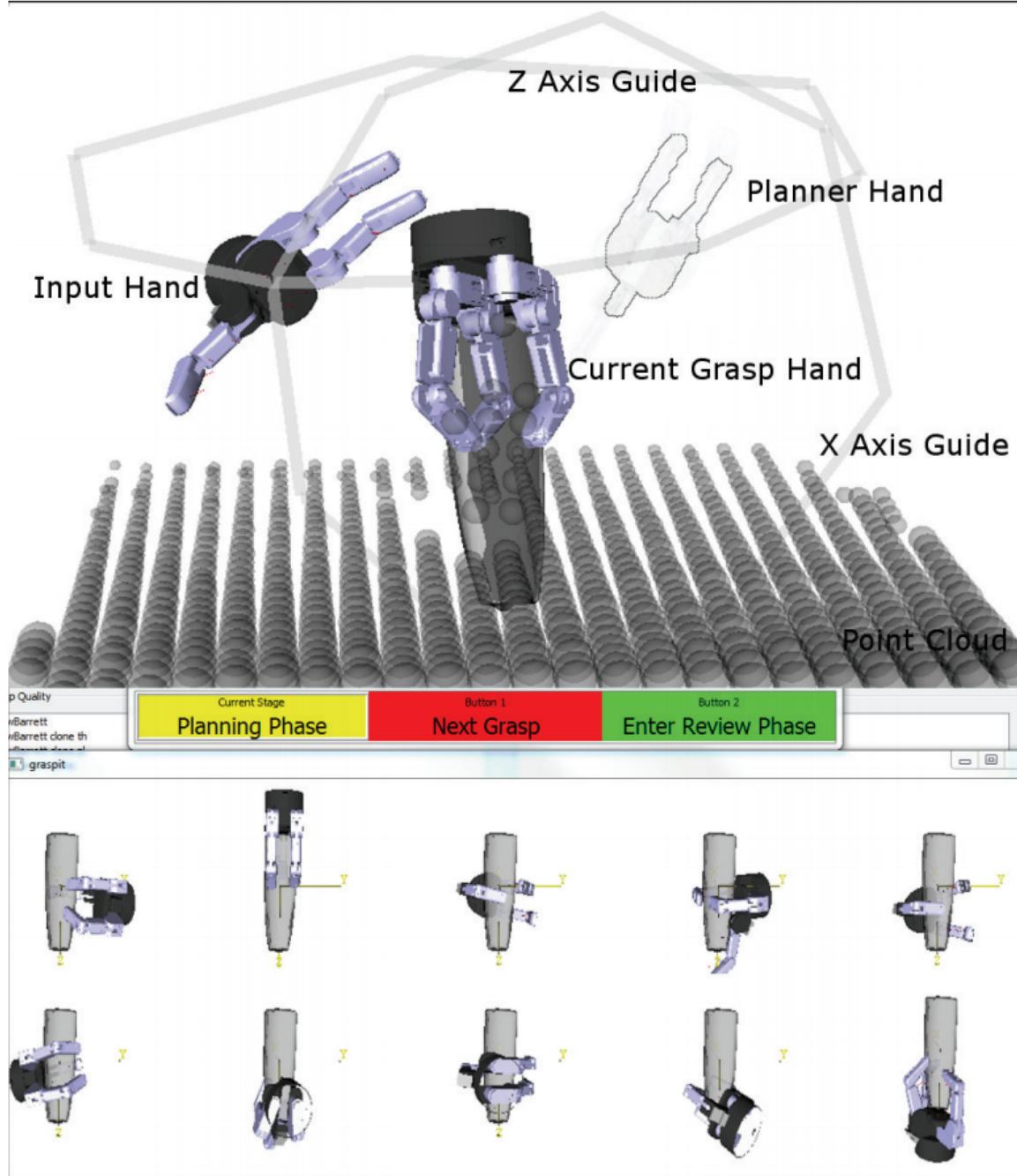


Figure 4.6: The user interface for the semi-autonomous grasp planning in the Online Planner phase. The user interface is comprised of three windows: The main window containing three labeled robot hands and the target object with the aligned point cloud, the pipeline guide window containing hints for the user to guide their interaction with each phase of the planner, and the grasp view window containing rendering of the ten best grasps found by the planner thus far. The object shown in this figure is a novel object that the planner does not have a model of (see Fig. 4.5a). The point cloud allows the user to visualize the fit of the model and act accordingly.

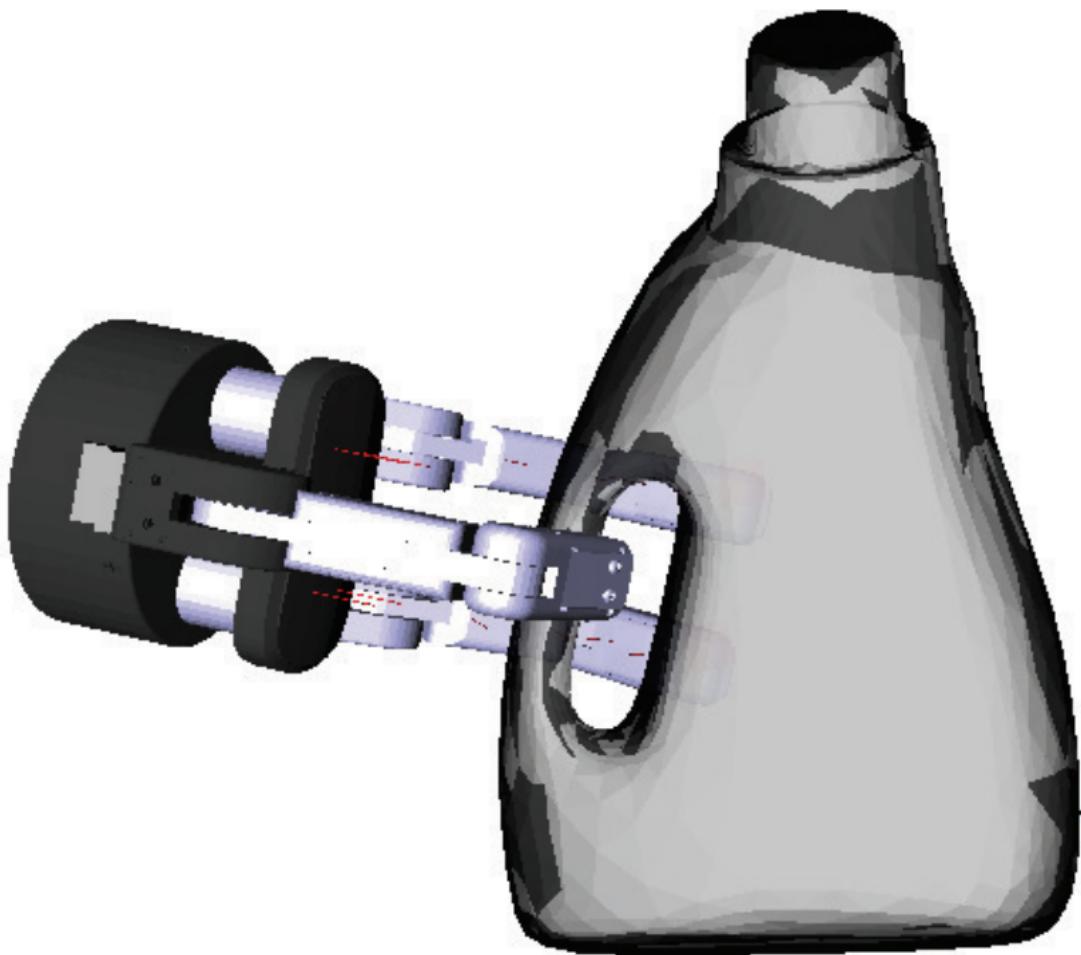


Figure 4.7: This handle grasp for the all bottle is not a force closure grasp, but when chosen by the subjects in our experiments it succeeded 100% of the time. Adding a grasp database allows such semantically relevant grasps to be used in our system.

Phase	Gesture 1	Gesture 2
Object Recognition	Rerun Recognition	Database Planning
Database Planning	Next Grasp	Planner Initialization
Planner Initialization	Confirm Grasp	Online Planning
Online Planning	Next Grasp	Review Grasps
Review Grasps	Next Grasp	Confirm Grasp
Confirm Grasp	Restart Planner	Execute Grasp
Execute Grasp	Review Grasps	N/A

Table 4.2: This table shows the initial control flow of the grasping pipeline using facial gestures detected by the Emotiv EPOC

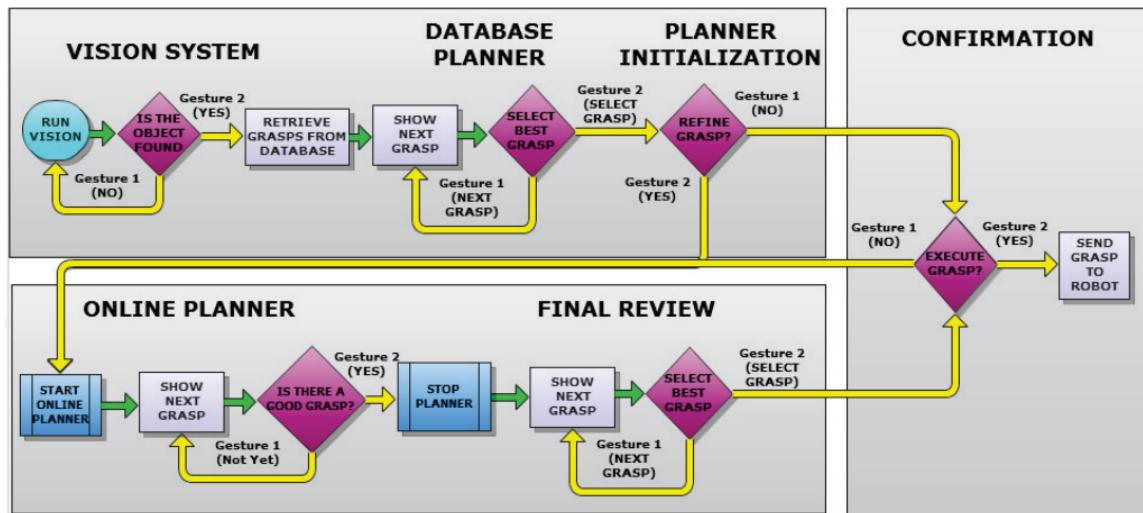


Figure 4.8: The phases modified grasping pipeline incorporating more control over the vision system and the grasp database. The purple diamonds reflect decision points in the pipeline that require input from the user. If the user chooses grasp n from the database, n+4 user inputs are required. If none of the grasps are suitable, the online planner can be invoked with a few simple inputs to refine one of the grasps further.

asked to grasp the object five times, irrespective of direction.

4.2.2.2 Training

In this work, we use four gestures. Gesture 1 is the jaw clench classified by the Expressiv classifier. We have trained the Cognitiv suite on three signals, right eye winking as gesture 2, left side jaw clenching as gesture 3, and eyebrow raising as gesture 4. Subjects were trained to use the Emotiv Epoch in four fifteen minute sessions. In the first two sessions, the classifier built into the Cognitiv Suite of the Emotiv Epoch software was trained on the three facial expressions used in the experiment. In the second two sessions, the subject was asked to perform the task in the virtual environment without executing the final grasp on the actual arm.

4.2.2.3 Results

The results of the experiments are reported in 4.3. For each subject, we report the mean time to completion and fraction of successful attempts for each grasp. Time to completion is measured from the end of the object identification phase to the beginning of the execution phase, as this represents the time taken to plan the grasp. Overall, the average planning time was 104 seconds on the known objects and 86 seconds on the unknown object. The average success rate was 80%, demonstrating that this system is efficacious in allowing the user to plan and execute a reasonable grasp for these objects. It is notable that grasps from the side demonstrated significantly more robustness and lower planning times than grasps from above. The grasp database only contained one grasp from above for each of these objects, and this grasp was a fingertip grasp which may be sensitive to pose estimation error, which resulted in longer planning times while the subjects searched for a better grasp. In general, grasping roughly cylindrical objects such as the top of the detergent bottle from above is somewhat problematic for the BarrettHand due to its configuration and the low friction of its fingertips. In contrast, subjects were able to find a reasonable grasp from the side of the object among the grasps pulled directly from the database. The difference in planning times reflects the benefit of integrating the off-line planning phase.

Grasp	Subject	Successes	Mean Time
Flashlight Side	1	3/3	125
	2	3/3	53
	3	2/3	103
	4	3/3	95
	5	3/3	82
Flashlight Top	1	3/3	132
	2	2/3	75
	3	2/3	96
	4	3/3	93
	5	2/3	125
Detergent Bottle Side	1	3/3	75
	2	3/3	57
	3	3/3	106
	4	2/3	82
	5	3/3	75
Detergent Bottle Top	1	1/3	151
	2	2/3	114
	3	2/3	142
	4	2/3	161
	5	3/3	145
Novel Bottle	1	3/5	132
	2	4/5	63
	3	4/5	95
	4	4/5	91
	5	4/5	50

Table 4.3: Results from Experiment 2

4.2.3 Discussion

During the experiment, we found that it was necessary to rewet the electrodes of the Epoc many times during the experiment. Additionally, for three of our subjects it took more than an hour to find the right thresholds and position for the headset. After the experiment, subjects were asked to describe their discomfort during the experiment and their level of control. Subjects reported little discomfort, but were frustrated with the difficulty of getting the Epoc to recognize their intended actions, especially with false negatives making it difficult to continue to the next stage of the pipeline at will. In spite of this frustration, subjects were able to complete the task.

However, these difficulties pose a major problem in testing this system on a disabled subject. Since they are dependent on caretakers, an indeterminately long setup time poses a major problem in performing studies with that population. Another major source of issues is the lack of online reachability testing in the grasp planner. In these experiments, we placed the object carefully to avoid having the planner find unreachable grasps. When we extend this system to handle more cluttered scenes, this will become a much more pressing issue. Finally, users reported that the pipeline guide window was too small.

4.3 Integrating a novel sEMG Device

4.3.1 Surface EMG Recording

Although the Epoc made a reasonable early testbed, there are a number of ways in which it is not optimal. For one thing, the device has a fairly large profile around the whole head, which is inconvenient for individuals with severe impairments that require head support. Additionally, the device was quite finicky to train and tended to need repeated reseatings and maintainance. To address some of these issues, we formed a collaboration with a group at UC Davis which had developed a novel input device designed to be used by severely impaired individuals. This device extremely noninvasive and low profile, requiring only a single surface electromyography (sEMG) recording site behind the ear. The muscles around the ear are show in Fig. 4.9. These muscles are innervated by nerves which come directly from the brain stem, without ever entering the spine. Thus, even individuals with the most

severe spinal cord paralysis can still access these muscles. Additionally, neurodegenerative disorders tend to effect more distal muscles first, and the function of more proximal muscles near the face and hands retain their function in more impaired individuals. The Superior Auricular(SA) muscle is a large muscle near the temple which is often activated by chewing motions or jaw clenches. The Posterior Auricular(PA) is a smaller muscle behind the ear. Although some individuals are able to independently move their ears, we have found that even individuals that cannot move their ears can learn to activate the muscles in that region without achieving overt motion when they are given visual feedback. This activation is produces signals that can be detected by electrodes mounted on the surface of the skin.

In a series of works [Joshi *et al.*, 2011],[Perez-Maldonado *et al.*, 2010],[Skavhaug *et al.*, 2012], our collaborators have shown that they can record two simultaneous channels from a single recording site. This is achieved by training the subject to modulate the activation frequency of the muscles near the recording site so that they can voluntarily control the ratio of power in two separate frequency bands. These two independent degrees of control are used to drive a cursor which selects options by hitting targets on a screen. In those works, the authors produced different user interfaces such as a UI for allowing a disabled individual to control a television.

The general methodology is outlined in 4.10. The single sEMG signal is first processed through a 60 Hz noise filter to remove noise from the AC power supply. It is then run through two different band pass Butterworth filters to extract two separate signals. The bands are then linearly combined to compute the x and y cursor positions. This linear combination is necessary to generate independent control channels since there are no perfect band pass filters, and the subject may not be able to completely independently activate the frequency bands.

The total powers of two different frequency bands of the single sEMG signal were computed using two band pass filters for 80-100 Hz (Band 1) and 130-150 Hz (Band 2). These bands were selected ad-hoc, based on previous experience. The output of the two filters produced comparable powers. The filter outputs were combined linearly as described in Equation 4.1. Without this transformation the cursor could not reach points along the x or y axis as there can never be zero power in either of the frequency bands. The gains

for each band are set for each subject after a short calibration procedure, as described in [Perez-Maldonado *et al.*, 2010] to establish the subjects comfort level maintaining a large enough voluntary muscle contraction to move the cursor to any part of the screen.

The sEMG signals are collected from the AP muscle with two surface Ag-AgCl cup electrodes connected to a model Y03 preamplifier (www.motion-labs.com) with input impedance higher than 108, 15-2000 Hz signal bandwidth and a gain of 300. The electrodes were placed behind the subjects left ear along the axis of the muscle with approximately 1.5 cm inter-electrode distance (see Fig. 1). A third electrode was placed on the elbow as a reference. The cup electrodes were the type EL254S from Biopac Systems Inc. held in place with Ten20 conductive paste.

$$x_{pos} = 1.75 \frac{ChannelPower_1}{gain_1} - 0.75 \frac{ChannelPower_2}{gain_2} \quad (4.1a)$$

$$y_{pos} = 1.75 \frac{ChannelPower_2}{gain_2} - 0.75 \frac{ChannelPower_1}{gain_1} \quad (4.1b)$$

To adapt this system to our use, we have added some additional smoothing steps. The cursor position is further filtered through a low-pass filter with a cutoff frequency of .5 Hz. This produces a new position at 4 Hz. To smooth the visualization of the cursor motion, we linearly interpolate 7 intermediate positions between each successive update, increasing the refresh rate of the visualization from 4 Hz to 32 Hz. This makes the system feel significantly more interactive, at the cost of a .25 second delay between the calculated position and the visualization.

4.3.2 sEMG GUI

To send signals to the grasping system, the user controls a cursor to hit one of four targets presented, as seen in Fig. 4.11. The user begins in a rest area and moves the cursor to one of the targets, each representing a different input option. When the target is hit, the cursor changes colors to reflect the users selection. The user returns the cursor to the rest area, at which point the input option selected is activated. After a selection, the other targets are disabled for four seconds. If an unintended target is selected, the user can avoid returning

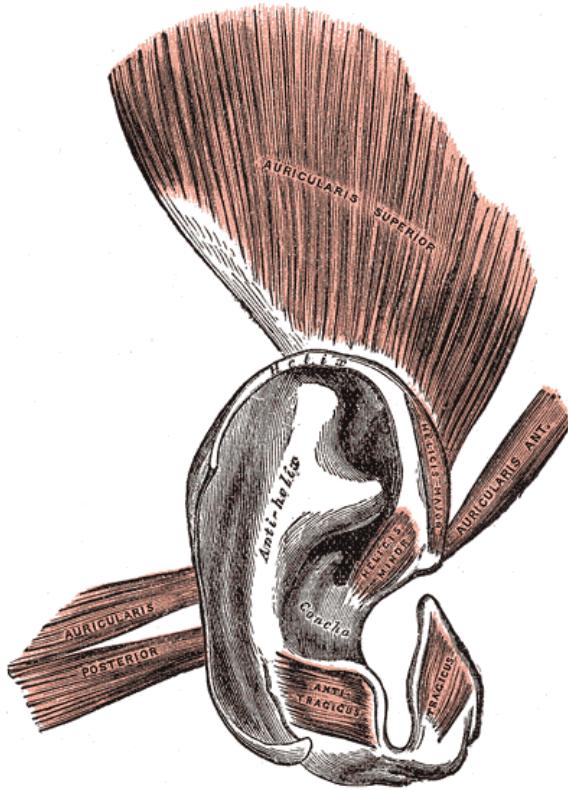


Figure 4.9: A diagram of the muscles around the ear. The sEMG device has been used on both the Posterior and Superior Auricular muscles. In order to record from the Superior Auricular (SA), it is necessary to remove the hair in the area above the ear. The Posterior Auricular(PA) is generally easier to access, however it is a smaller muscle that is more difficult to learn voluntary control over.

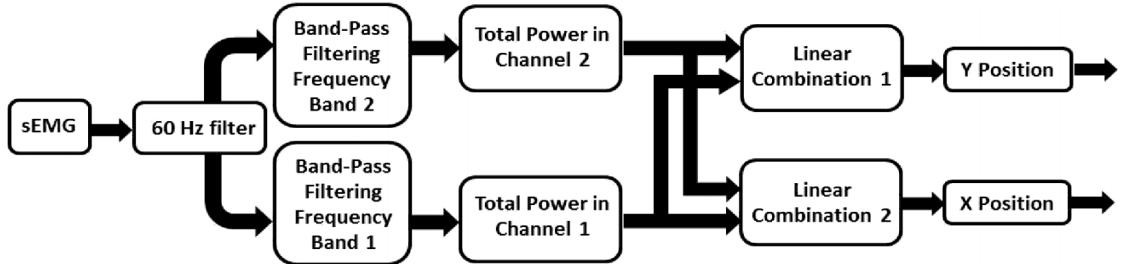


Figure 4.10: The single sEMG signal is first processed through a 60 Hz noise filter. It is then run through two different band pass Butterworth filters to extract two separate signals. The bands are then linearly combined to compute the x and y cursor positions.

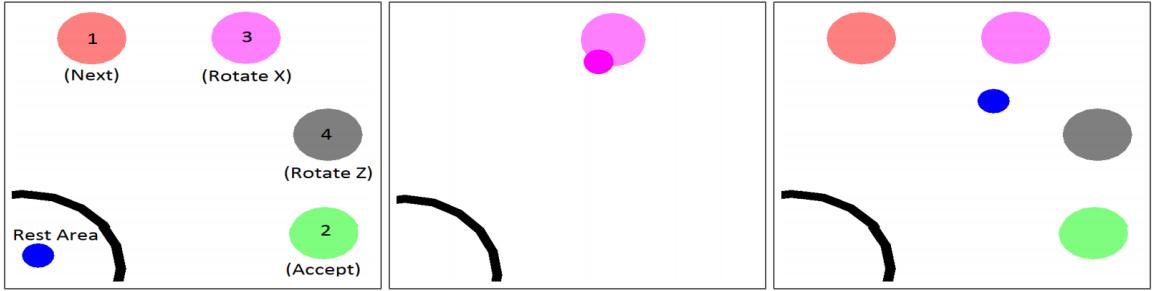


Figure 4.11: The sEMG Interface: (a) Hitting one particular target changes the color of the cursor to reflect the selection and makes the other targets unavailable. (b) If the user does not return to the rest area after a few seconds, the selection times out and is deselected and all targets become available again for selection. (c) The user interface is composed of 4 targets overlayed on the grasping scene. Target 1 usually signals acceptance of the current option. Target 2 toggles the next option. Targets 3 and 4 provide input to the planner.

to rest for these four seconds, cancelling the selection.

We map these inputs to the inputs introduced in our previous sections. For the red and green inputs, denoted input 1 and input 2, the input is activated a single time when the user returns to rest. For inputs 3 and 4, the magenta and black targets respectively, the activation is sent continuously until the user exits the rest area again. This allows the user to do near continuous control over the approach direction.

4.4 Handling Cluttered Scenes

In addition to an improved input device, we also wanted to extend our grasping system to handle more realistic scenes, including some amount of clutter. In this work, we will define “clutter” as objects being in close enough proximity some grasps for each object may collide with nearby objects, but that they are not actually in contact with one another. We did not handle the problem of singulation, which is a specialized manipulation designed to separate objects which are too close together for the fingers to surround the object without colliding with other objects. As such, we tested grasp planning scenes where there was at least 2 inches of empty space between each objects.

Handling cluttered scenes brings up a number of challenges. First, it slows down the on-

line planning phase. For one thing, there are many fewer possible grasps and the state space has more discontinuities in the value of the quality function, which slows down convergence. Additionally, when there are more possible collisions, grasp planning is significantly slower. Second, many of the grasps produced by the planner, while valid in their final poses, may not have a reachable path to grasp. Third, with more objects in the scene, there is more visual clutter and it is more difficult to produce a useful visualization.

In order to address this problem, we added reachability tests to the online grasp planning system and user interface. Previous iterations of this planner used a simpler, built in inverse kinematics solver to reject unreachable states. In contrast, this version checks that an entire valid trajectory can be generated. Unreachable grasps are placed at the end of the list of grasps and colored red in the grasp preview window (see Fig. 4.13). We maintain the list of unreachable grasps so that we can reject nearby grasps without running more computationally expensive analyses. The valid grasps are ranked by their distance to the demonstration hand and alignment to its approach direction. This makes the planner more responsive in cluttered scenes.

We maintain the list of unreachable grasps so that we can reject nearby grasps without running more computationally expensive analyses. The valid grasps are ranked by their distance to the demonstration hand and alignment to its approach direction. This makes the planner more responsive in cluttered scenes. The list is re-sorted as the demonstration hand is moved.

The results of the reachability test are also used to train a nearest neighbors classifier. When the user moves the demonstration hand, we find the five grasps for which the normal of the palm of the hand is closest to the normal of the demonstrated pose. If at least 50% of these grasps are unreachable, we designate the current demonstration pose as being in an unreachable region, which is indicated to the user by highlighting the demonstration hand in the planner interface in red. These measures are crucial for a naive user that is not familiar with the kinematics of the robot arm and may not understand that the region they are trying to grasp from is not within the robots workspace.

4.4.1 GUI Modifications for Clutter and sEMG Interface

A number of changes have been made to the user interface to accomodate both the added visual complexity of overlaying the sEMG control interface on the planning scene and the added difficulty of interpreting the multiobject scene. The new layout is demonstrated in Fig. ??, which illustrates the UI presented during the object selection phase. First, the point cloud displayed in the scene has been upgraded to a higher resolution, color point cloud. This change allows the user to discern the target object more effectively in the cluttered scene. It also allows the user to exercise more judgement in interpreting the scene, since they may not be physically present to observe it first hand. Second, we display only three grasp options instead of 10 to reduce the visual clutter. This also allows us to enlarge the presentation of the grasp so that the user can more easily discern how the hand may interact with the rest of the objects in the seen. Third, we moved the grasp preview window to the side of the screen and modified the way that the UI generates the view to share the aspect ratio and alignment of the depth camera that all detected objects are visible and that the user's intuition is unimpeded by deformations due to the aspect ratio. Third, we remove all of the window decorations and grasp metric displays, as the subject is not expected to be able to interpret them correctly. Overall, this provides a much cleaner, streamlined view more suitable for non-expert users.

Unfortunately, additional visual clutter is introduced by the addition of the sEMG user interface. This interface is presented with relatively high transparency, allowing the subject to see the grasp planning scene behind the UI with relative ease. The scene is chosen so that the relevant objects are relatively centered in the scene. While the user is not actively making a suggestion, the cursor remains in the lower left of the screen and the main grasp planning scene remains unoccluded.

We placed targets 1 and 2 in opposite corners of the screen because these inputs control the progress of the user through the grasping pipeline, and we wish to minimize confusion and accidental selection between these two options. The middle two options modify the demonstrated approach direction, and accidental selections have minimal impact.

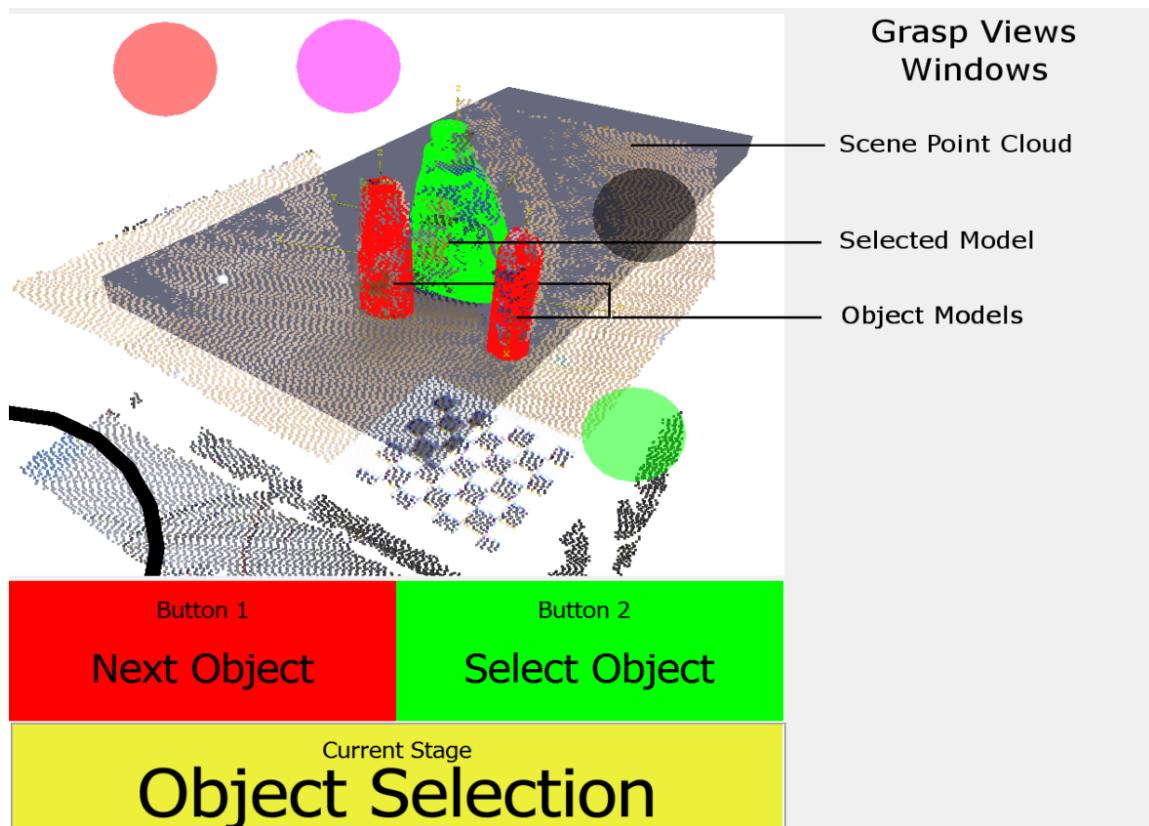


Figure 4.12: The Grasp Planning Interface - Object Selection Phase: The subject is able to see the planning scene in the main UI window. The window on the bottom tells the user the current phase and what the green and red inputs will do in this phase. In this phase, the subject sees the the pointcloud and hits the red target until the object they wish to grasp is highlighted in green. Then they hit the green target to proceed to the next phase.

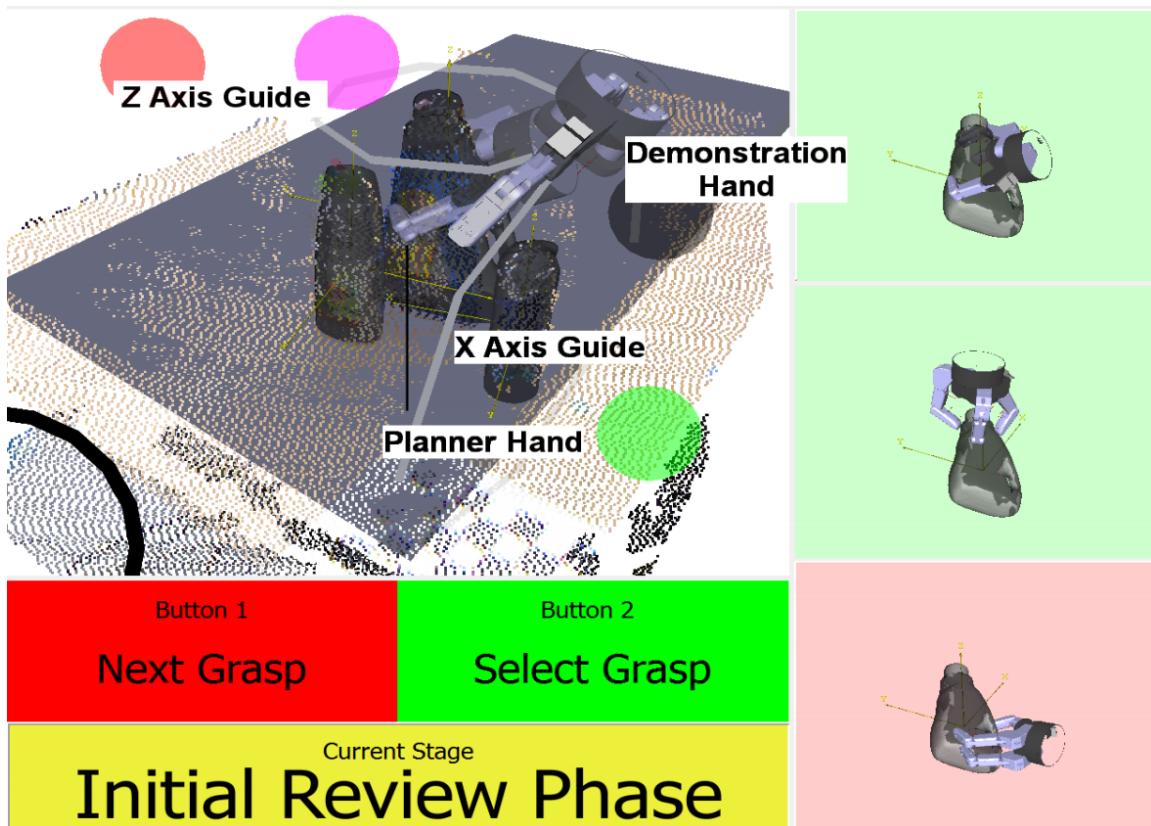


Figure 4.13: The Grasp Planning Interface - Initial Review Phase: After the subject selects the object, the Grasp View pane on the right is populated with a set of grasps from a database. Grasps that are reachable appear on a green background, while unreachable grasps are red. A robot hand appears that the user moves to demonstrate a desired starting pose. This demonstration hand is constrained to follow the two circular guides around the z and x axes of the object shown above. The top most grasp in the grasp view window is the currently selected grasp, which is rendered in the planning scene with the planner hand.

4.4.2 Pipeline Version 3

A number of changes have been made to the pipeline to accomodate the presence of other objects in the scene and the different input device. *Initialization:* This phase is the same as version 2, except that the view is aligned to the true view of the kinect, rather than being centered on the target object. The user sends input 1 to activate the object recognition system. If the recognized objects align well with the point cloud sent, they can accept the results with input 1. If not, they can rerun the recognition system with input 2.

Object Selection: This phase is new in this version of the pipeline. The first object is highlighted in green as the target object. To select an object as a target, the user sends input 2. To cycle to the next object in the recognized object list the user sends input 1. The nontarget objects are all highlighted in red. The nontarget objects are replaced with lower resolution models when a target is selected, which makes the planning phases faster.

Initial Review: As in the pipeline version 2, the user is presented with a list of preplanned grasps from a precomputed database. As they iterate through the grasp list, the grasp in the middle and bottom rows shift up and the next grasp in the list moves in to the bottom position. The grasps are evaluated for reachability in order. An unreachable grasp may take up to 15 seconds to return failure using the CBiRRT planner. The user is also able to move the planner hand to demonstrate a desired approach direction, which will cause the grasp list to re-sort to bring the closest grasps to the new approach direction to the top of the list and reset the grasp visualizations to the beginning of the cycle. The user sends input 1 to increment through the grasp list. When the user finds a reasonable looking grasp, they send input 2 to select the grasp.

Planner Initialization: This phase is the same as version 2. The user is presented with the choice to either accept the grasp from the third stage with input 1, proceeding straight to the Grasp Choice Confirmation stage or they can send input 2 to refine their chosen grasp further.

Grasp Refinement: This phase is slightly modified from the pipeline version 2. As grasps are generated, they are displayed with a white background in the grasp preview pane on the right side of the screen while they are being analyzed for reachability. If the analysis finds the grasp to be reachable, the background is highlighted in green and the list

is re-sorted to bring the closest reachable grasps to the top. Unreachable grasps are given a red background. The successful grasps are stored in a KD tree by approach direction and eigengrasp value. If the nearest reachable grasp is not within 20° of the approach direction of the demonstrated grasp or its eigengrasp values differ by more than 20%, the demonstration hand is highlighted in red to indicate that this approach direction may be a bad choice. Sending input 2 stops the planner and proceeds to the Final Grasp Review stage.

Final Grasp Review: This phase is similar to version 2, but adapted to provide more feedback with fewer grasp previews. The user sends input 1 to select the next grasp on the grasp list. They send input 2 to select that grasp. The user is able to move the demonstration hand as in the previous stages, which will still cause a re-sorting of the grasp list. This allows the user to more easily go through the grasp list, since the commands which control motion are continuous and need much less effort to send repeatedly than having to manually hit the next grasp option repeatedly.

Grasp Choice Confirmation: The user sends input 1 to go back to the Grasp Refinement stage and input 2 to send the grasp for execution on the robot.

4.4.3 Validation

To validate this system, we recruited an impaired subject with limited upper limb mobility due to a C3-C4 spinal injury. This subject had previous experience with the sEMG device, but had not been trained on this interface. For this work, we measured the activity in the subjects PA muscle, as this muscle is more accessible because the hair over it is thinner. The subject was recruited and trained at the UC Davis site, and operated the robot without ever having any interaction with it or the experiment site in the real world at in the Columbia University Robotics Group lab. The setup is shown in Fig. 4.14. The top of the figure shows the subject with the device attached and using the grasp planning system. The bottom left image is a close up of the device, which demonstrates how low profile and minimalistic this device is. The bottom right shows the target scene in the robot workspace, with three container objects.

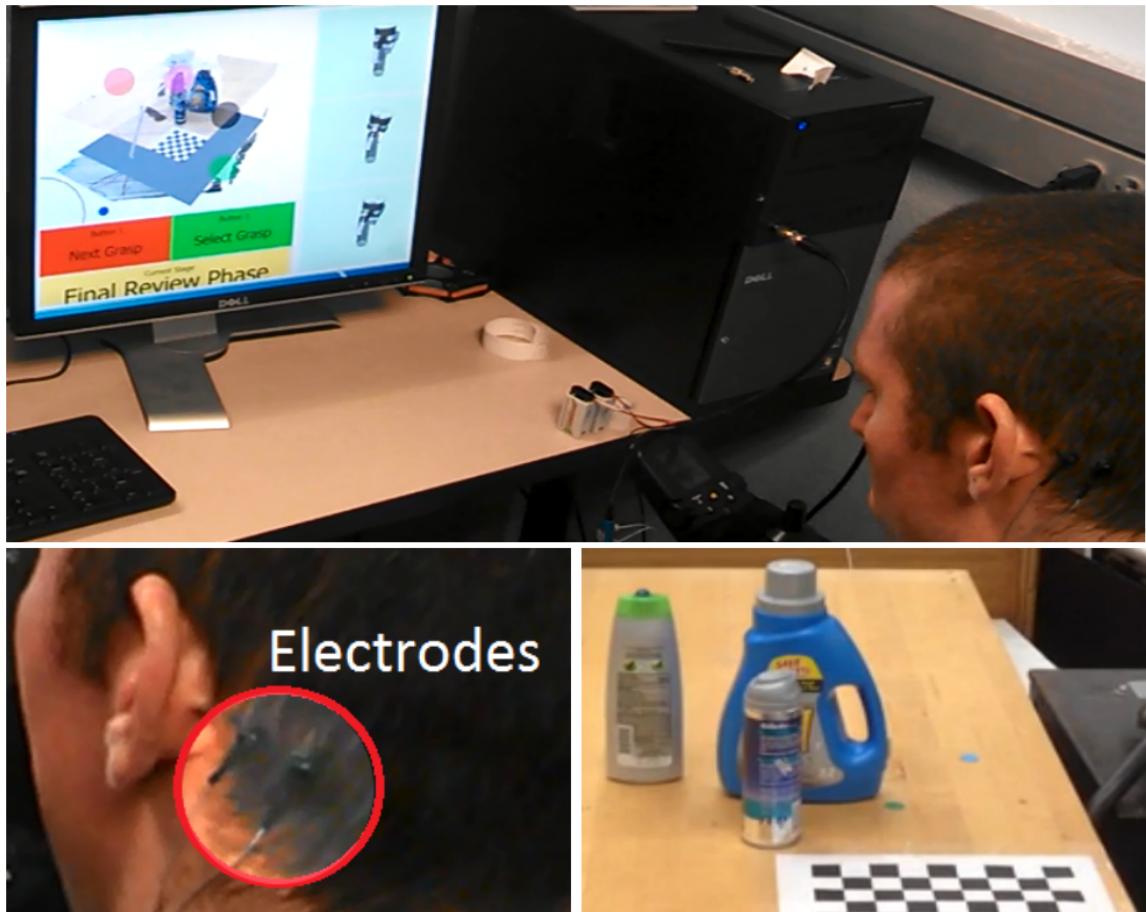


Figure 4.14: An impaired subject in the UC Davis RASCAL lab (top) operating our sEMG-Assistive Grasping interface to grasp a shaving gel bottle in the Columbia Robotics Group Laboratory (bottom right). The two small black clips behind the subjects ear (bottom left) are surface EMG electrodes (used in differential mode) to detect activation of the Auricularis Posterior (AP) muscle to direct the system to pick up the object in this multi-object scene.

4.4.3.1 Task

The subject was asked to make three attempts to pick up an object from a cluttered, multi-object scene. In the first two attempts, the user was asked to use the online planner to refine one of the pre-planned grasps. In the first attempt, the subject grasped the laundry detergent bottle. In the second attempt, the subject grasped the shaving gel bottle. In the third attempt, the subject was asked to grasp the detergent bottle using one of the pre-planned grasps directly from the grasp database. Other than the image in the planner interface, the subject was not given any information about the objects they were to grasp. However, they are all well known, household objects, so the subject can be expected to have some implicit idea of the weight and friction properties of the object.

During the task, the subject reported which target they were trying to reach and we tracked the number of mistaken target activations, which would lead the user to loop back rough part of the pipeline. After the grasp is selected, the target object is lifted off of the table automatically so that the user can see whether the grasp is stable. If no part of the target object remains on the table, we consider the trial a success.

4.4.3.2 Training

All testing was approved by the Institutional Review Board of the University of California, Davis under Protocol 251192-10. The subject had previously been trained to use their SA muscle with this device, but for this experiment we attempted to use the PA since their hair was now in the way of the SA site. To familiarize the subject with the interface, we demonstrated the pipeline two times with the subject just watching and asking questions along the way. We then went through the pipeline with the subject two more times while verbally instructing them on which target to hit while the experimenter controlled the cursor with a computer mouse. This allowed the subject to familiarize themselves with the pipeline and navigate their way through it without having to also focus on the task of hitting targets with the sEMG interface. Once they appeared to be conversant with the system, we turned over control to the subjects sEMG interface.

Grasp	Time (s)	#Inputs	#Timeouts	Mistaken Selections
Detergent 1	564	14	14	2
Detergent 2	609	9	50	0
Shaving Gel	910	12	11	1

Table 4.4: sEMG Experiment 1 Results

4.4.3.3 Results

The results of the experiment are shown in Tab. 4.4. The subject was able to grasp the objects successfully on every attempt. On average, it took the subject 694 seconds to grasp each object, including about 60 seconds for the vision system to detect the objects in the scene. There were 75 timeouts, and 3 mistakenly selected targets. Timeouts are an expected part of this interface, which allows the user to re-select their intended target if the initially selected target is incorrect. Occasional mistaken selections are also expected, and the pipeline is designed to be robust to these errors, allowing the user to go back to previous step where necessary to correct mistakes. Several mistakes in a row are necessary to actually realize mistaken actions on the robot.

4.4.3.4 Discussion

These results, while promising from the perspective that user was able to understand and utilize the system fairly quickly, demonstrated a number of shortcomings with our system. First, the user’s control over the sEMG device was not very accurate, yielding many false initial selections that had to be timed out. This may be because the user was using a new muscle, but in subsequent experiments the user was not seen to perform very well in other similar paradigms, especially with respect to finding parameters for the sEMG processing that allowed the user to return to the rest area. This user may just have too much random activity in that area for the processing pipeline we have utilized. A second problem was that the online reachability tester is fairly slow using the CBRRT planner, and thus new available grasps appeared slowly. This caused relatively few reasonable grasps to be available, and so the user had more trouble because they had to iterate through more

grasps which were not reflective of their intent while looking for a reasonable one. While indicating poor regions for grasping by shading the display hand was somewhat effective at helping the user avoid long waits in regions that were doomed to failure, it was not sufficient near border regions where grasps were possible but unlikely because of occlusions.

4.5 Further Refinements

Our initial results showed enough efficacy of this system that we developed a second prototype using a smaller, lower weight robotic arm, the Kinova Mico, which is suitable for mounting on a wheel chair. We also sought feedback from our colleagues at the Columbia Medical Center who worked with this same sEMG device in stroke patients. Their advice was that our user interface needed further streamlining. We also sought to resolve the online reachability checking issue by integrating a faster planner.

4.5.1 Adaptations for the Mico Manipulator

The Kinova Mico arm is a six DOF arm with a two finger gripper. The fingers use a passive underactuation mechanism and is capable of both enveloping and fingertip grasps. These fingers are made of a hard plastic which has relatively little friction. The transmission of the underactuation mechanism is designed such that the fingertips remain perpendicular through most of the range of the finger's motion. In order to plan grasps in GraspIt! for a such a gripper with a small friction coefficient, it is necessary to strongly bias the planner towards keeping the fingertips very well aligned to the surface of the object. The simplest way to do this is to add a strict filtering layer to the grasp quality metric which only performs the second stage of grasp analysis, which tests for force closure of the final grasp, when the virtual contacts are aligned to within 3° of the normal to the nearest surface. Because the finger closing does not change the fingertip angle much during the closing process and there is very little friction, performing the second stage is not likely to succeed unless the surface alignment is good to begin with.

Additionally, the CBiRRT planning pipeline uses an inverse kinematics library, described in [Len *et al.*, 2010] that is incompatible with the unusual 60° offsets of the Mico Arm (see Fig

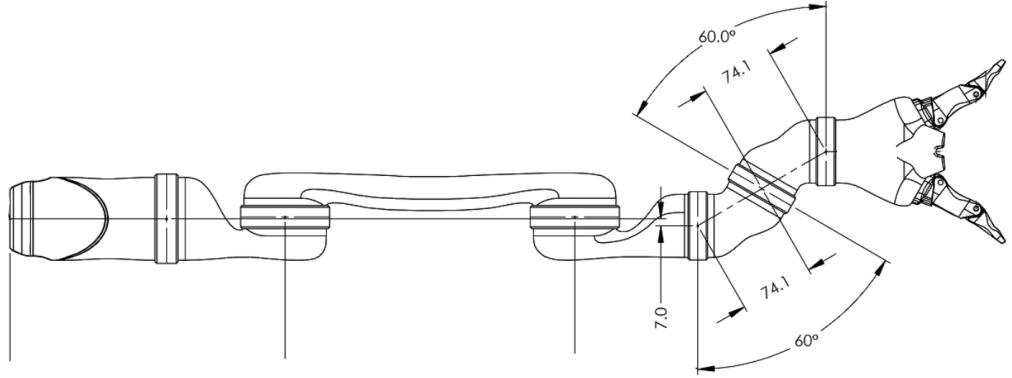


Figure 4.15: A diagram of the kinematics of the Mico arm. The 60° offsets of the the final two arm joints is very unusual, and causes difficulties for some robotics libraries.

4.15 for an illustration) . Without this library, the planner the filtering pipeline is slightly slower and less reliable. Given our previous insight that the online reachability testing is a bottleneck for the online grasp refinement, this motivated us to explore a variety of different planners integrated with the MoveIt! planning environment [see [Sucan and Chitta, 2013]].

4.5.2 Improved Online Reachablility Checking

MoveIt! provides an interface to the OMPL planning library, which integrates a number of planners which have very different performance properties. In order to investigate which one is appropriate to grasping in the cluttered scenes with the Mico Arm, we captured a 10 scenes similar to that in 4.14 and ran the online reachability checker on the set of default grasps for each of the objects in the scene. Since many of the grasps in the online planner tend to be very similar, we perturbed the grasps by a +/- 0.005 m in each direction, testing 60 grasps for each of three objects for each scene. In order to take advantage of this proximity, we implemented a plan caching scheme which stores the start and end point of the arm trajectory in a nearest neighbors lookup tree. When planning a new trajectory for online analysis, we first attempt to plan from the end of the nearest endpoint. If that fails, we retry from the original starting position. If this second attempt succeeds, the planned path is inserted into the cache. For the actual arm motion, we retry the planning until

it succeeds from the original starting location, so long as a valid cached plan exists. This is because smoothing such plans to remove the excess waypoints introduced by the initial segment from the cached plan is still an open area of research that we did not wish to address in this work.

We did a parameter sweep of the allowed trajectory segment length from 0.01 to 0.1 in steps of 0.01 with allowed planning times up to 20 seconds. The SBL and PRM planner had around the same rate of success using the caching scheme, succeeding in 43% of the grasps, and the PRM had the fastest planning for the caching version of the planner, with an average planning time of 5.5 seconds for arm motions in which the caching fails, and 0.1 seconds when the cache succeeds. However, when not using the caching scheme, the SBL planner's success rate dropped to 30%. So, for the online reachability verification, we used the PRM planner with a segment length of 0.05, while for producing the actual grasp on the robot we used the SBL planner. These changes removed the online reachability checking as a bottleneck for the online grasp refinement stage of the pipeline.¹

4.5.3 Further UI Improvements

In our previous pipelines, the two 'continuous' inputs which shifted the hand around the object were not part of the pipeline guide display which showed the user which phase they were in and what their inputs would do. In every phase of the pipeline, they always did the same thing. However, in this version of the pipeline the purpose of these inputs can also change in each stage. In addition, our previous test user and our collaborators at the Columbia Medical Center indicated that there should be a clearer differentiation between the augmented reality region containing the grasp planning scene and the rest of the UI and fewer grasp options presented during the parts of the pipeline where they are not needed.

In this version, the UI window is adaptive, changing more to suite the needs of each stage of the pipeline. The grasp previews are integrated with the pipeline guide display,

¹Although MoveIt! includes a benchmarking suite for determining the optimal parameters for a set of problems, it cannot be used with MoveIt!'s pick and place grasping pipeline, which handles the approach and lift phases of the path planning, or with robots that have some joints with continuous joint ranges. As such, we implemented our own, less comprehensive optimization script.

and the pipeline guide areas also function as GUI buttons for the experimenter to use when familiarizing the subject with the UI. Each of the targets now has a corresponding color coded button. These new UI elements are shown in Fig. 4.16, Fig. 4.17, and Fig. Fig. 4.18. This version of the UI is more dynamic, with the layout of the guide window on the right side of the screen changing as the user progresses through the pipeline. While this may seem like unnecessary complexity, the UI is more visibly different in each stage and less extraneous information is presented. This seems to help subjects keep track of what stage they are in and what its purpose is.

4.5.4 Updated pipeline

The updated pipeline is slightly shorter and makes more varied use of inputs 3 and 4.

Object Recognition and Selection: This phase combines the first two phases of version 3. To select an object as a target, the user sends input 2. To cycle to the next object in the recognized object list the user sends input 3, which will continuously iterate through the grasps until the user leaves the rest area. To rerun the object recognition system the user sends input 1. While the recognition system is still running, the whole screen is highlighted in red and it is not possible to proceed to the next phase until the recognition finishes.

Initial Review: As in the pipeline version 3, the user is presented with a list of preplanned grasps from a precomputed database. As illustrated in Fig. ??, the currently selected grasp is shown in the window of the top of the guide area, with the color of the background again indicating the results of the online reachability checker. The next grasp is shown in the bottom of the window. Input 1 begins the online refinement stage, input 2 skips to the Final Grasp Review stage. Input 3 will iterate through the available grasps, whereas input 4 will return to the Object Recognition and Selection Stage.

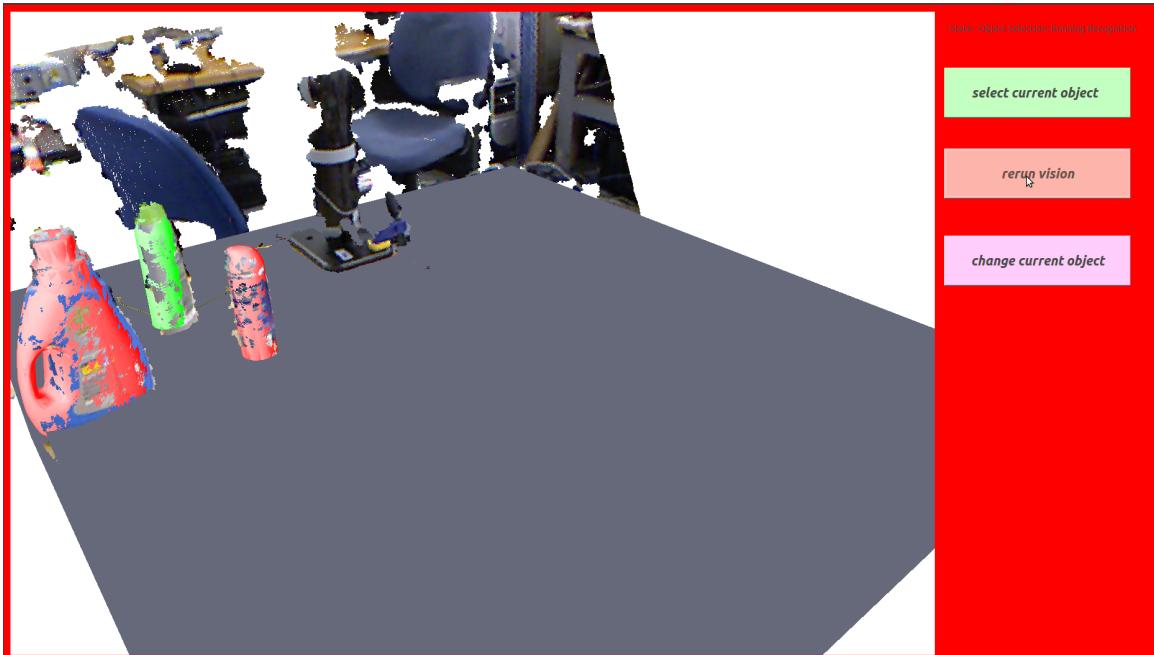
Grasp Refinement: This phase is the same as version 3, but with the first grasp shown in the top of the window and the next grasp shown on the bottom. Input 1 proceeds to the Final Grasp Review stage, input 2 aligns the hand to the next grasp and brings it up to the top window. Inputs 3 and 4 rotate the hand around the object as previously described.

Final Grasp Review: This phase is similar to version 3, but adapted as previous stages to have only two grasps, the top showing the current selection and the bottom showing the

Figure 4.16: The Object Recognition and Selection state user interface



(a) The object selection and recognition state. The user can cycle through the objects at a rate of 2 per second by sending input 3. Selecting input 1 goes to the next state.



(b) Sending input 2 triggers the object recognition state and highlights the background in red while the recognition is running.

next selection. Input 1 proceeds to the Grasp Choice Confirmation stage, input 2 aligns the hand to the next grasp and brings it up to the top window. Inputs 3 and 4 rotate the hand around the object as previously described.

Grasp Choice Confirmation: This phase is similar to version 3, but with only the selected grasp shown in the grasp preview window. The user sends input 1 to go back to the Grasp Refinement stage and input 2 to send the grasp for execution on the robot.

4.5.5 Validation

To validate these design decisions, we tested our pipeline on 5 naive subjects. To simplify testing of the UI, we did not attempt to train the subjects on the two dimensional version of the user interface. Instead, the subjects were given a similar user interface, but the cursor is constrained to move towards the target represented the 'selected' input, which is outlined in green as shown in Fig. 4.20. In order to switch which target is currently 'selected', the user leaves the rest area and returns to it without hitting a target. This cycles the 'selected' target forward by one.

4.5.5.1 sEMG Device Setup

In these experiments, we placed the sEMG device behind the ear of the subject to measure contractions of the PA muscle. In order to stabilize the device and reduce noise due to motion of the wires, we stabilize the electrodes by wrapping the head of the electrodes in Silly Putty brand silicone putty, as shown in Fig. 4.19. We find the correct placement of the device by asking the subject to clench their jaw gently and raise their eyebrows. We place the electrodes where we find a large in response to eyebrow raises and there is little response to jaw motion.

4.5.5.2 Training

sEMG Device: Each subject was trained on the sEMG user interface without the grasp planning system. In the training system, shown in Fig. 4.21, the user is given a 'desired' target highlighted in red which is randomly selected at the beginning of each trial. The user is then instructed to cycle the 'selected' target until it overlaps with the 'desired target',

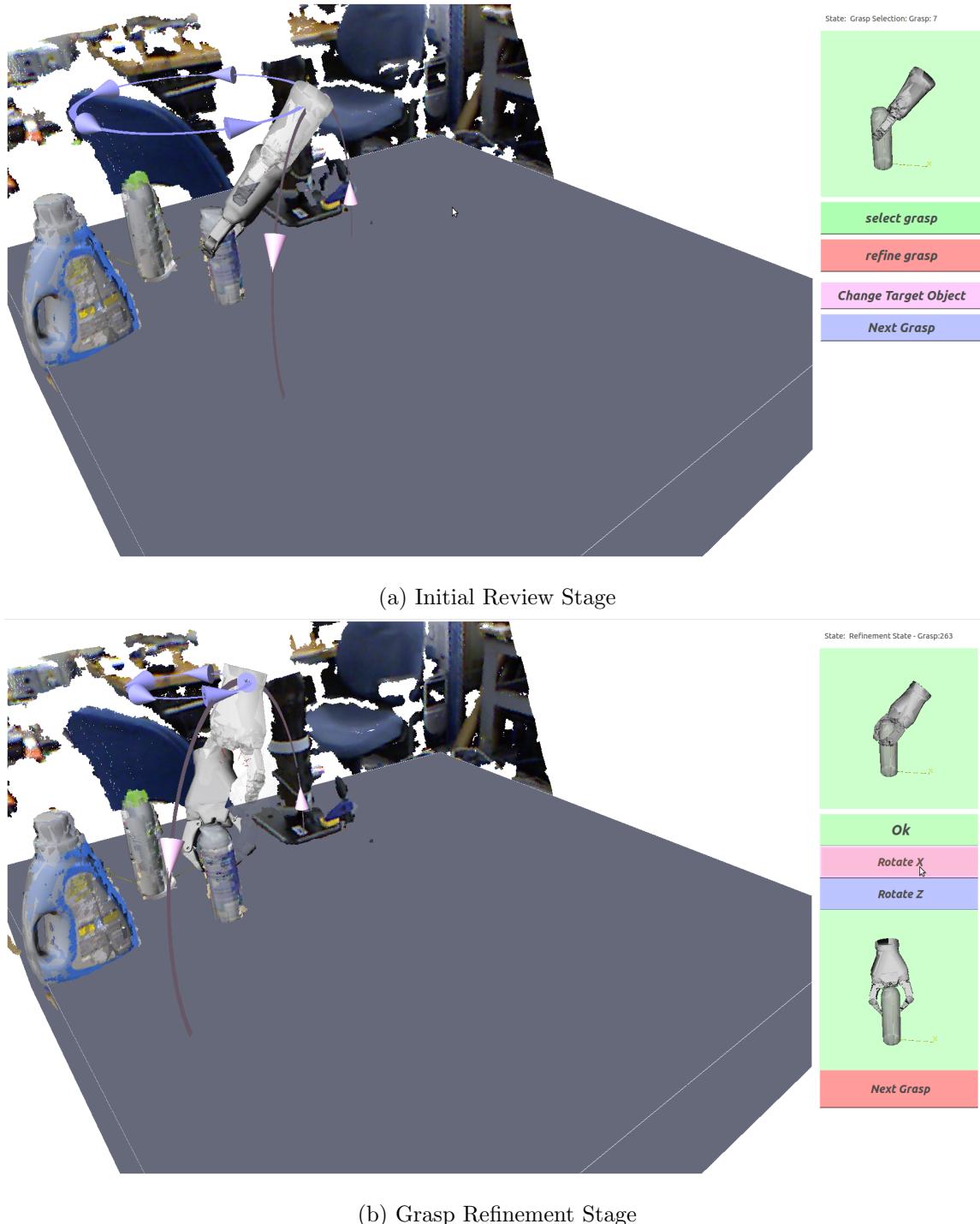


Figure 4.17: The UI in all of the second and third of the pipeline. The buttons on the right function as both guides for the result of hitting the color coded input options that will be presented to the user, as well as buttons that the user and experimentor use during the training stage

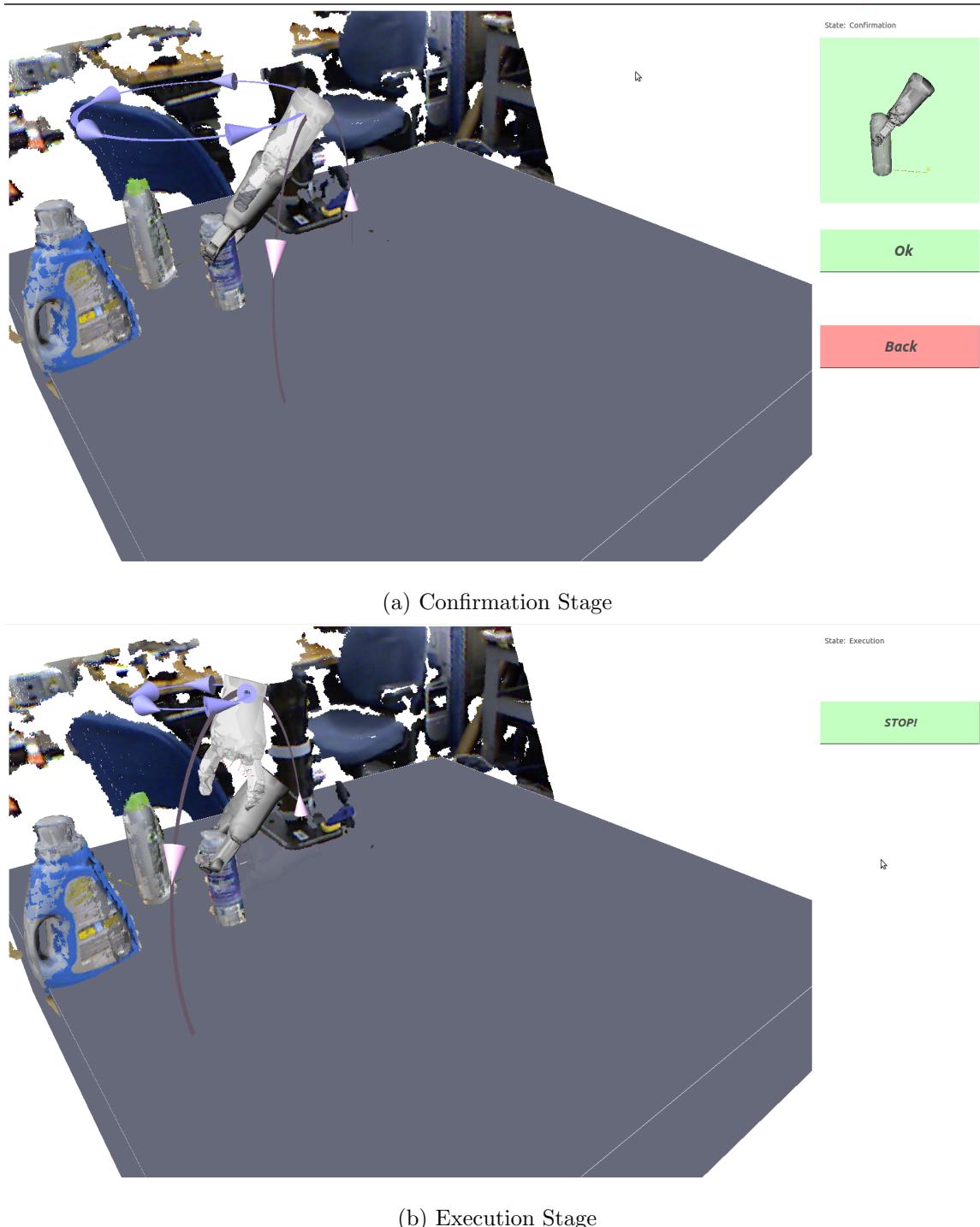


Figure 4.18: The final two stages of version 4 of the BCI grasping UI. This version of the UI is more dynamic, with the layout of the guide window on the right side of the screen changing as the user progresses through the pipeline. While this may seem like unnecessary complexity, the UI is more visibly different in each stage and less extraneous information is presented.

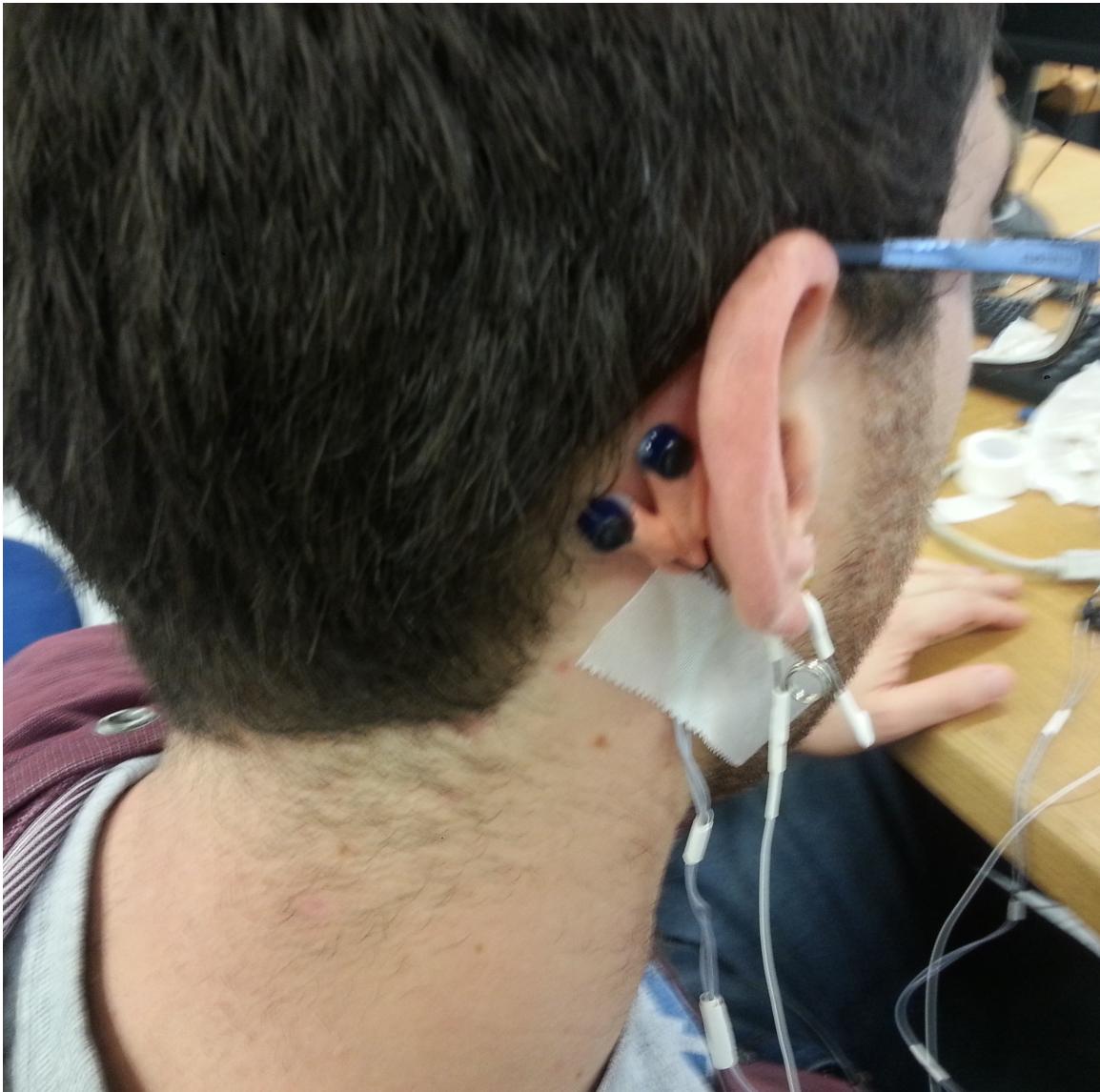


Figure 4.19: The sEMG system setup for the updated pipeline experiments. In these experiments, we placed the sEMG device behind the ear of the subject to measure contractions of the PA muscle. In order to stabilize the device and reduce noise due to motion of the wires, we stabilize the electrodes by wrapping the head of the electrodes in Silly Putty brand silicone putty.

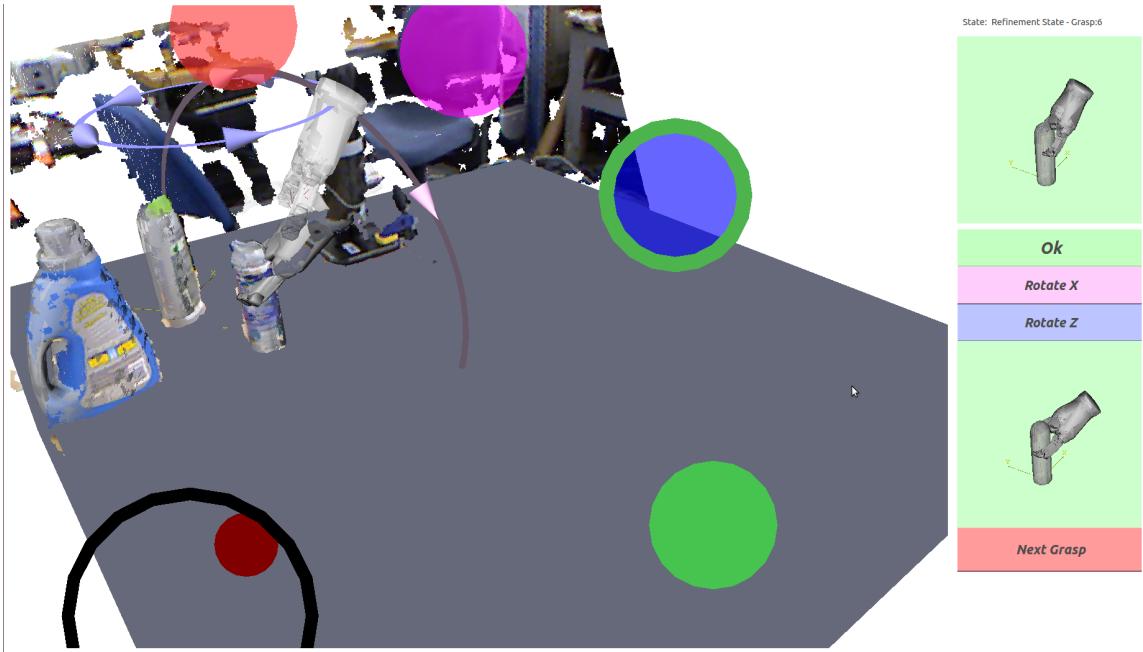


Figure 4.20: The sEMG interface overlayed on the planning scene with the selected target highlighted in green.

which is then shown in gold. The subject was asked to perform sets of 30 trial blocks until they successfully completed at least 29 of the 30 attempts. This took at most 2 blocks of trials for any subject, with subjects who already had some ability to move their ears frequently succeeding in their first block.

Grasp Planning Interface: To familiarize the subject with the grasp planning system, we manually showed the subject three examples of grasping objects, once short circuiting the online planning and twice allowing the online refinement to run. Then we allowed the subject to guide the pipeline themselves five times, twice without the online planner and three times with it. Then we repeated the training allowing the subject to guide the planner to pick up the large detergent bottle five times in whatever direction they chose using the UI through the on screen button interface.

4.5.5.3 Task

We placed the objects on the table in proximity to one another as shown in Fig. ???. We asked the subject to grasp each object three times, the first time from any direction they

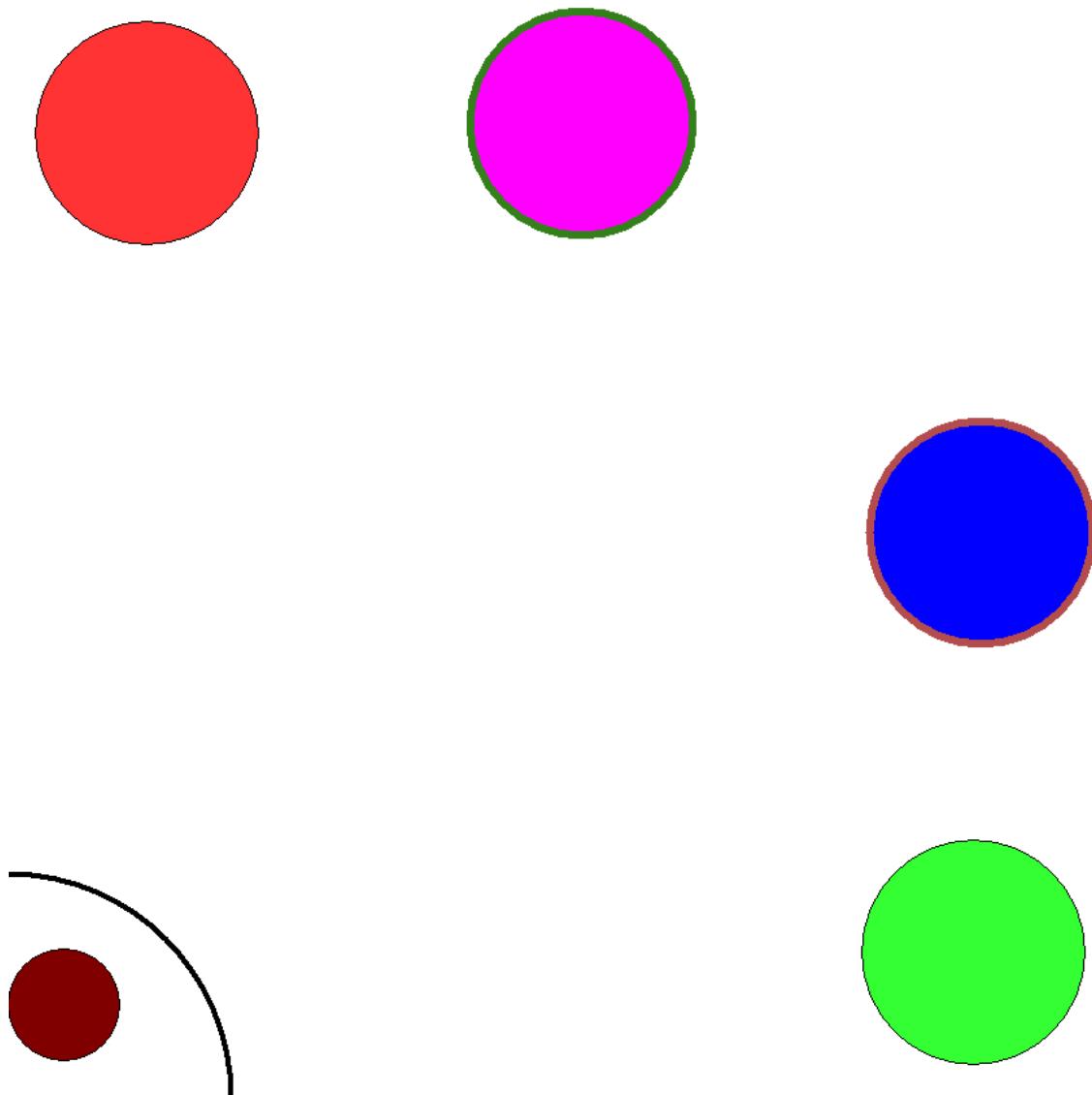


Figure 4.21: The training version of the sEMG interface in Matlab, showing the desired target in red, and the selected target in green.

deemed reasonable, once from the side, and once from above. For each object, the placement of the objects and grasps in the database were such that either the side or top grasp required the online grasp refinement. Since the workspace of the Mico arm is not very large, this was not difficult to accomplish.

4.5.5.4 Results

The results of the experiment for 5 subjects are tabulated in Tab. 4.5. On average, the subjects were successful ingrasping 77% of the objects within 85 seconds of the first time their cursor left rest area. With respect to speed, results are comparable, and indeed somewhat better than the amount of time it took subjects to grasp objects with the Emotiv EPOC, even though the subject may have to iterate over the possible options before selecting them. Subjects 2 and 3 were the best able to control the cursor, having previously been able to move their ears already, and also performed the best in these experiments. These results indicate that the underlying planning system is providing options that the less capable subjects are not exploring because they are having more difficult with the UI.

For the shampoo bottle, there are relatively fewer grasps that can succeed as compared to the rotationally symmetric shaving gel bottle and the taller, sloping detergent bottle. The only feasible grasps from the side for the shampoo bottle are directly from the side, aligned with the wide axis of the bottle, as demonstrated in 4.22. This narrow feasible region and the potential for many collisions with the other objects in the scene during the reaching motion to this region makes this a particularly difficult grasp, especially when the clearance around the grasp is as tight as it is in 4.22. Without the partial plan caching implemented in the online trajectory planner, planning grasps to this region using stochastic, sampling based planners is extremely unreliable. With the caching scheme, this grasp was successfully found 100% of the time, although the planning time is somewhat longer than the other grasp tasks.



Figure 4.22: A typical grasp of the shampoo bottle from the side in the cluttered scene. Note that the hand is just able to fit between the other objects to grasp the desired target. Note that the ability to plan this grasp in such a restricted environment is an indication that this system is very successful at handling the cluttered scene.

Grasp	Subject	Success	Time	Grasp	Subject	Success	Time
Detergent Bottle Top	1	Yes	75	Shampoo Bottle Open Choice	1	Yes	93
	2	Yes	53		2	Yes	121
	3	No	45		3	Yes	63
	4	No	122		4	Yes	95
	5	NA	NA		5	NA	NA
	Mean	50%	72		Mean	100%	93
Detergent Bottle Side	1	No	66	Shaving Gel Top	1	No	83
	2	Yes	40		2	No	123
	3	Yes	52		3	Yes	112
	4	Yes	80		4	No	139
	5	NA	NA		5	NA	NA
	Mean	75%	72		Mean	50%	93
Detergent Bottle Open Choice	1	Yes	50	Shaving Gel Side	1	Yes	65
	2	Yes	57		2	Yes	52
	3	Yes	53		3	Yes	57
	4	Yes	135		4	Yes	88
	5	NA	NA		5	NA	NA
	Mean	100%	73		Mean	100%	65
Shampoo Bottle Top	1	Yes	151	Shaving Gel Open Choice	1	No	73
	2	Yes	72		2	Yes	59
	3	Yes	60		3	Yes	76
	4	No	126		4	Yes	81
	5	NA	NA		5	NA	NA
	Mean	75%	102		Mean	75%	72
Shampoo Bottle Side	1	Yes	134	Average Perfor- mance	1	66%	87
	2	Yes	95		2	88%	73
	3	Yes	132		3	88%	72
	4	Yes	164		4	66%	109
	5	NA	NA		5	NA	NA
	Mean	100%	131		Mean	77%	85

Table 4.5: Results from Experiment 3. On average, the subjects were successful ingrasping 77% of the objects within 85 seconds of the first time their cursor left rest area.

Part I

Bibliography

Bibliography

- [Artemiadis and Kyriakopoulos, 2011] Panagiotis K Artemiadis and Kostas J Kyriakopoulos. A switching regime model for the EMG-based control of a robot arm. *IEEE Trans. on Systems, Man, and Cybernetics*, 41(1):53–63, February 2011.
- [Balasubramanian *et al.*, 2010] R. Balasubramanian, Ling Xu, P.D. Brook, J.R. Smith, and Y. Matsuoka. Human-guided grasp measures improve grasp robustness on physical robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2294–2301, May 2010.
- [Bekiroglu *et al.*, 2011] Yasemin Bekiroglu, J. Laaksonen, Jimmy Alison Jorgensen, V. Kyrki, and D. Kragic. Assessing grasp stability based on learning and haptic data. *Robotics, IEEE Transactions on*, 27(3):616–629, June 2011.
- [Bell *et al.*, 2008] Christian J Bell, Pradeep Shenoy, Rawichote Chalodhorn, and Rajesh P N Rao. Control of a humanoid robot by a noninvasive brain-computer interface in humans. *Journal of Neural Engineering*, 5(2):214–20, Jun 2008.
- [Berenson *et al.*, 2009a] Dmitry Berenson, Siddhartha Srinivasa, David Ferguson , and James Kuffner. Manipulation planning on constraint manifolds. In *IEEE International Conference on Robotics and Automation (ICRA '09)*, May 2009.
- [Berenson *et al.*, 2009b] Dmitry Berenson, Siddhartha S. Srinivasa, and James J. Kuffner. Addressing pose uncertainty in manipulation planning using task space regions. In *in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009.

- [Brost and Christiansen, 1997] RandyC. Brost and AlanD. Christiansen. Empirical verification of fine-motion planning theories. In Oussama Khatib and J.Kenneth Salisbury, editors, *Experimental Robotics IV*, volume 223 of *Lecture Notes in Control and Information Sciences*, pages 475–485. Springer Berlin Heidelberg, 1997.
- [Castellini and van der Smagt, 2009] Claudio Castellini and Patrick van der Smagt. Surface EMG in advanced hand prosthetics. *Biological cybernetics*, 100(1):35–47, January 2009.
- [Cipriani *et al.*, 2008] C. Cipriani, F. Zaccone, S. Micera, and M.C. Carrozza. On the Shared Control of an EMG-Controlled Prosthetic Hand: Analysis of User Prosthesis Interaction. *IEEE Transactions on Robotics*, 24(1):170–184, February 2008.
- [Faverjon and Ponce, 1991] B. Faverjon and J. Ponce. On computing two-finger force-closure grasps of curved 2d objects. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 424–429 vol.1, Apr 1991.
- [Ferrari and Canny, 1992] C. Ferrari and J. Canny. Planning optimal grasps. In *Proc. of the Int. Conf. on Robotics and Automation*, pages 2290–2295, August 1992.
- [Gomez-Gil *et al.*, 2011] Jaime Gomez-Gil, Israel San-Jose-Gonzalez, Luis Fernando Nicolas-Alonso, and Sergio Alonso-Garcia. Steering a Tractor by Means of an EMG-Based Human-Machine Interface. *Sensors*, 11(7), 2011.
- [Ho *et al.*, 2011] N. S. K. Ho, K. Y. Tong, X. L. Hu, K. L. Fung, X. J. Wei, W. Rong, and E. A. Susanto. An EMG-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation. In *Int. Conf. on Rehabilitation Robotics*, pages 1–5. IEEE, June 2011.
- [Horki *et al.*, 2011] Petar Horki, Teodoro Solis-Escalante, Christa Neuper, and Gernot Müller-Putz. Combined motor imagery and SSVEP based BCI control of a 2 DoF artificial upper limb. *Medical & Biological Engineering & Computing*, 49(5):567–77, May 2011.
- [Hsiao *et al.*, 2011] Kaijen Hsiao, Matei Ciocarlie, and Peter Brook. Bayesian grasp planning. 2011.

- [Joshi *et al.*, 2011] S.S. Joshi, A.S. Wexler, C. Perez-Maldonado, and S. Vernon. Brain-muscle-computer interface using a single surface electromyographic signal: Initial results. In *Int.l IEEE/EMBS Conf. on Neural Engineering (NER)*, pages 342–347, May 2011.
- [Len *et al.*, 2010] Beatriz Len, Stefan Ulbrich, Rosen Diankov, Gustavo Puche, Markus Przybylski, Antonio Morales, Tamim Asfour, Sami Moisio, Jeannette Bohg, James Kuffner, and Rüdiger Dillmann. Opengrasp: A toolkit for robot grasping simulation. In Noriaki Ando, Stephen Balakirsky, Thomas Hemker, Monica Reggiani, and Oskar von Stryk, editors, *Simulation, Modeling, and Programming for Autonomous Robots*, volume 6472 of *Lecture Notes in Computer Science*, pages 109–120. Springer Berlin Heidelberg, 2010.
- [Lozano-Prez *et al.*, 1984] Toms Lozano-Prez, Matthew T. Mason, and Russell H. Taylor. Automatic synthesis of fine-motion strategies for robots. *The International Journal of Robotics Research*, 3(1):3–24, 1984.
- [M. Bryan, V. Thomas, G. Nicoll, L. Chang and Rao, 2011] J.R. Smith M. Bryan, V. Thomas, G. Nicoll, L. Chang and R.P.N. Rao. What You Think is What You Get: Brain-Controlled Interfacing for the PR2. Technical report, Iros 2011: The PR2 Workshop, San Francisco, 2011.
- [Matrone *et al.*, 2011] G. Matrone, C. Cipriani, M. C. Carrozza, and G. Magenes. Two-channel real-time EMG control of a dexterous hand prosthesis. In *Proc. Int. Conf. on Neural Engineering*, pages 554–557, April 2011.
- [Miller and Allen, 2004] Andrew T. Miller and Peter K. Allen. Graspit!: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11:110–122, 2004.
- [Müller-Putz *et al.*, 2005] Gernot R Müller-Putz, Reinhold Scherer, Gert Pfurtscheller, and Rüdiger Rupp. EEG-based neuroprostheses control: a step towards clinical practice. *Neuroscience Letters*, 382(1-2):169–74, 2005.
- [Papazov and Burschka, 2011] Chavdar Papazov and Darius Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *Computer Vision ACCV 2010*, volume 6492, pages 135–148. 2011.

- [Perez-Maldonado *et al.*, 2010] C. Perez-Maldonado, A.S. Wexler, and S Joshi. Two dimensional cursor-to-target control from single muscle site semg signals. *Trans. of Neural Systems and Rehabilitation Engineering*, 18, April 2010.
- [Pollard , 2004] Nancy Pollard . Closure and quality equivalence for efficient synthesis of grasps from examples. *International Journal of Robotics Research*, 23(6):595–614, 2004.
- [Ponce and Faverjon, 1991] J. Ponce and B. Faverjon. On computing three-finger force-closure grasps of polygonal objects. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1018–1023 vol.2, June 1991.
- [Ponce *et al.*, 1996] Jean Ponce, Steve Sullivan, Attawith Sudsang, Jean daniel Boissonnat, and Jean-Pierre Merlet. On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *International Journal of Robotics Research*, 16:11–35, 1996.
- [Postelnicu *et al.*, 2011] Cristian-cezar Postelnicu, Doru Talaba, and Madalina-ioana Toma. Controlling a Robotic Arm by Brainwaves and Eye. *Int. Fed. For Information Processing*, pages 157–164, 2011.
- [Ranky and Adamovich, 2010] G. N. Ranky and S. Adamovich. Analysis of a commercial EEG device for the control of a robot arm. In *Proc. Northeast Bioengineering Conference (NEBEC)*, pages 1–2, New York, NY, March 2010.
- [Roa and Suarez, 2009] M.A. Roa and R. Suarez. Computation of independent contact regions for grasping 3-d objects. *Robotics, IEEE Transactions on*, 25(4):839–850, Aug 2009.
- [Royer *et al.*, 2011] Audrey S Royer, Minn L Rose, and Bin He. Goal selection versus process control while learning to use a brain-computer interface. *Journal of Neural Engineering*, 8(3):036012, June 2011.
- [Sagawa and Kimura, 2005] K. Sagawa and O. Kimura. Control of robot manipulator using EMG generated from face. In *Proc. Int. Conf. on Manufacturing and Industrial Technologies*, volume 6042, December 2005.

- [Salisbury and Roth, 1983] J.K. Salisbury and B. Roth. Kinematic and force analysis of articulated mechanical hands. *Journal of Mechanical Design*, 105, March 1983.
- [Scherer *et al.*, 2011] Reinhold Scherer, Elisabeth C. V. Friedrich, Brendan Allison, Markus Pröll, Mike Chung, Willy Cheung, Rajesh P. N. Rao, Christa Neuper, and Markus Pr. Non-invasive brain-computer interfaces: enhanced gaming and robotic control. In *Advances in Computational Intelligence*, volume 6691. June 2011.
- [Schmidl, 1965] HANNES Schmidl. The inail myoelectric b/e prosthesis. *Orthop Prosthet Appl J*, pages 298–303, 1965.
- [Shenoy *et al.*, 2008] Pradeep Shenoy, Kai J Miller, Beau Crawford, and Rajesh N Rao. Online electromyographic control of a robotic prosthesis. *IEEE Transactions on Biomedical Engineering*, 55(3):1128–35, March 2008.
- [Sherman and Lippay, 1965] E David Sherman and Andrew Lippay. A russian bioelectric-controlled prosthesis. *Canadian Medical Association journal*, 92(5):243, 1965.
- [Skavhaug *et al.*, 2012] I. Skavhaug, R. Bobell, B. Vernon, and S. Joshi. Pilot study for a brain-muscle-computer interface using the extensor pollicis longus with preselected frequency bands. In *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, pages 1727–1731, August 2012.
- [Sucan and Chitta, 2013] Ioan A. Sucan and Sachin "MoveIt!" Chitta, 2013. Available Online from <http://moveit.ros.org>.
- [Sugar and Kumar, 2000] T.G. Sugar and V. Kumar. Metrics for analysis and optimization of grasps and fixtures. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 4, pages 3561–3566 vol.4, 2000.
- [Vogel *et al.*, 2010] Jörn Vogel, Sami Haddadin, John D Simeral, Sergej D Stavisky, Dirk Bacher, Leigh R Hochberg, John P Donoghue, and Patrick Van Der Smagt. Continuous Control of the DLR Light-weight Robot III by a human with tetraplegia using the Brain-Gate2 Neural Interface System. In *International Symposium on Experimental Robotics*, 2010.

- [Waytowich *et al.*,] N. Waytowich, A. Henderson, D. Krusienski, and D. Cox. Robot application of a brain computer interface to staubli TX40 robots - early stages. *World Automation Congress (WAC)*, pages 1–6.
- [Woczowski and Kurzyski, 2010] Andrzej Woczowski and Marek Kurzyski. Human-machine interface in bioprosthesis control using EMG signal classification. *Expert Systems*, 27(1):53–70, February 2010.
- [Yamada *et al.*, 2001] T. Yamada, T. Koishikura, Y. Mizuno, N. Mimura, and Y. Funahashi. Stability analysis of 3d grasps by a multifingered hand. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 3, pages 2466–2473 vol.3, 2001.
- [Yang *et al.*, 2009] Dapeng Yang, Jingdong Zhao, Yikun Gu, Li Jiang, and Hong Liu. EMG pattern recognition and grasping force estimation: Improvement to the myocontrol of multi-DOF prosthetic hands. In *Int. Conf. on Intelligent Robots and Systems*, pages 516–521. IEEE, October 2009.
- [Zheng and Qian, 2005] Yu Zheng and Wen-Han Qian. Coping with the grasping uncertainties in force-closure analysis. *Int. J. Rob. Res.*, 24(4):311–327, April 2005.