Assistive Grasping with an Augmented Reality UI

Jonathan Weisz, Alexander G. Barszap*, Sanjay S. Joshi*, Carmen Elvezio, Benjamin Shababo and Peter K. Allen
Department of Computer Science, Columbia University
*Department of Mechanical and Aerospace Engineering, University of California, Davis

***Abstract*—Assisting impaired individuals with robotic devices is an emerging and potentially transformative technology. This paper describes the design of an assistive robotic grasping system that allows impaired individuals to interact with the system in a novel Human-in-the-Loop (HitL) manner, including the use of a new surface electromyography (sEMG) device we have built. The system uses 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. The system uses grasp quality measurements that generate more robust grasps by considering the local geometry of the object as well as how uncertainty will affect the proposed grasp. This interface is validated by testing with real users, both healthy and impaired. This work forms the foundation for a flexible, fully featured HitL system that allows users to grasp known and unknown objects in cluttered spaces using novel, practical Human Robot Interaction paradigms that have the potential to bring HiTL assistive devices out of the research environment and into the lives of those that need them.**

I. Introduction

A. The Promise of Assistive Robotics

With recent advances in robotics and computer vision, it is possible to imagine a robotic system to assist people with severely limiting disabilities in activities of daily living, improving their quality of life. Common daily activities frequently require the user to grasp an object stably in a context aware way. Complex hands and manipulators increase the flexibility and grasping capabilities of a robotic assistant, but at the cost of requiring more complex control of many simultaneous degrees of freedom (DOF).

This work presents an assistive grasping system for people with upper limb mobility impairments using a human-in-the-loop paradigm that allows a disabled user to grasp objects from a table using a novel, non-invasive surface electromyography (sEMG) based input device even in somewhat cluttered scenes. The novel device measures only a single differential sEMG signal at one muscle site on the user. The system puts the user in control of a multi-phase grasping pipeline that includes object recognition, integrated pre-planned and on-line grasp planning with feedback to help the user plan robust grasps in near real-time

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. Some current methods include sEMG, electroencephalography(EEG), eye-tracking, and sip-puff devices. In general, these devices are restricted to low bandwidth, noisy signals. Therefore, using these devices to control high DOF assistive grasping device poses many challenges. Our solution is to combine intelligent online grasp planning with limited HitL assistance.

B. The Challenge of Robotic Grasping

Irrespective of the problems posed by limited input devices, robotic grasping is challenging for a number of reasons. Complex robotic hands have many DOFs, so the space of possible grasps is difficult to explore. Standard approaches to planning in high dimensional state spaces are likely to fail with multi-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. Second, evaluating grasps involves several properties that are difficult to model, such as friction and closed chain kinematics. Many state-of-the-art analysis tools 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 designing generic grasp planning algorithms 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 the the desired use of the object. Thus 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.

This paper describes an sEMG driven assistive grasping platform integrating human in the loop planning in planning through an augmented reality interface. We present the iterative development process we have used to arrive at our final system, comparing different user interface paradigms and grasp planners presented in our previous papers addressing this problem Weisz *et al.* [2012, 2013, 2014]. We present a final user validation of our sEMG paradigm in a set of new experiments not published previously that addresses grasping in clutter with both known and unknown objects. The key contributions of this work include:

- Design and comparison of three different user interfaces for assistive grasping.
- Integration with a novel sEMG input device which relies on only a single muscle site.
- A new UI that improves the disabled users ability to understand the scene and produce correct grasps in complex, cluttered environments.
- Grasp reachability analysis and feedback to the user.
- Online assessment of the desired approach direction.

- Evaluation of this system on an impaired user in a remote location.
- A demonstration that this new UI is descriptive enough for the user to operate in an environment that they have never seen
- The ability to grasp known and unknown objects amidst clutter
- Demonstration of the system on two different sets of hardware
- Experimental results from both healthy and impaired users.

II. Related Work

A. Human Computer Interfaces for Assistive Robotics

There is a long history of engineering assistive systems using electrophysiological signals to drive robotic devices, such as Schmidl [1965]. A commercial version was introduced in Sherman and Lippay [1965]. In the time since, there have been myriad approaches and refinements of proposed of interfaces for 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 systems. One way is to categorize a system by its 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, electrooculagraphy (EOG), and EEG. Within this category, modalities can be further divided by 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 control 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 real-time. Below we describe the different control paradigms used in related systems using human-robot interface (HRI) devices suitable for impaired individuals.

1) 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 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 obstacles. A mapping from the subject's joints to the robotic arm can be found that ensures that the robot stays approximately the same distance from its 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 many seriously impaired individuals have lost exactly the capability used as the control input to this type of interface.

2) Joint Level Control: If direct mimicry is impractical, the user can be given 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 DOF manipulator with a gripper, the user must move 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.

3) 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 end effector location is the variable that the user most directly observes. These works 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 achieve cartesian end effector control 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 degrees of freedom simultaneously, the user's control is directly in the task space. This allows the user to decouple the different controlled degrees of freedom.

4) Discrete State Level Control: In discrete mode control, the user is able to switch between a set of predetermined configurations. The aforementioned 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. 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 schemes 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 steep learning curve.

5) Task Level Control: The key challenge of using non-invasive human robot 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 to, would be more effective. Indeed, it has been shown that users find HRI 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 Stäubli 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 four-phase pipeline where EEG signals are used to control transitions between phases. 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.

The drawback of this approach is that while the system presents the user with a set of high level choices, the user is not able to effect the process by which the choices are generated. In complex situations, the software agent may not present the user with appropriate choices.

6) Task Oriented Shared Control: An emerging alternative to the purely task oriented approach is to blend end effector control and task oriented control. In this approach, the user's input demonstrates some approximation of the desired solution or constraint which an automated planner can make use of. Because solutions are only expected to be approximate, the lower accuracy HRI devices can still be used to complete complex tasks with high reliability.

This paradigm is very useful for tasks in which the user's intent or task constraints can be difficult to explicitly encode, but can be demonstrated, such as grasping. In Ciocarlie et al. [2009], our lab introduced the the Eigengrasp Planner, which allowed the user to grasp objects reliably by demonstrating only an approximate approach direction. In this work, we have expanded upon the Eigengrasp planner to show that task oriented shared control is a practical approach for allowing the flexibility of lower level control schemes with the ease of use of higher level task level control.

B. The Eigengrasp Grasp Planner

The Eigengrasp Planner allows a user to interact with an online grasp planner in a virtual environment to plan grasps online in real-time. The user is given control of a virtual representation of the hand which they use to indicate approximately where they would like to grasp the object. A grasp planner runs in the background and presents the user with a set of options for completing the grasp. This strategy requires a responsive planner that can handle the complex problem of grasp planning in near real-time. To make this computationally tractable, Ciocarlie et al. introduced *Eigengrasps*, a dimensionality reduction technique in which control of the hand is mapped to principle components identified in human grasping studies. With this dimensionality reduction, stochastic sampling techniques can be used to generate reasonably good grasps in real-time using relatively simple grasp quality metrics.

The quality metric that is used by the planner evaluates a projection of the desired contact points on the hand to the target object. This projection provides a smooth energy gradient in regions where the hand is not in contact with the object. When good candidates are found, the planner simulates completing the grasp by approaching the object along a pre-specified direction orthogonal to the "palm" of the robot hand and then closing the fingers.

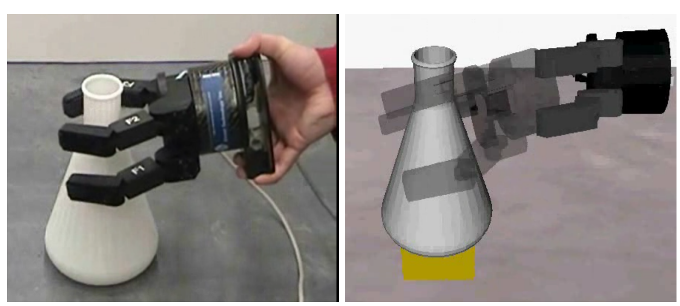

Fig. 1: An operator demonstrating the Eigengrasp Planner by manually guiding the robotic hand to guide the planner in the virtual environment.Ciocarlie and Allen [2009]

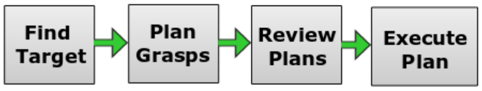

Fig. 2: The four phases of a basic grasp planning task. Breaking the task into phases allows customization of the user interface for each phase independently to make optimal use of low input bandwidth.

This approach has a number of practical advantages. The nature of the optimization approach, which gradually moves towards lower values of the quality function, produces solutions where nearby finger contacts will also provide similar quality grasps. Grasps where the qualities of nearby configurations are much poorer will have narrow basins of attraction that are less likely to be found. This implies a certain amount of robustness to small displacements and occlusion of the object from nearby clutter 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.

In Ciocarlie et al. [2009], the planner was demonstrated by having the operator manually directly move the end effector in real-time (see Figure 1). This is analogous to an extremely high bandwidth, low noise interface with perfect knowledge of the environment. In this work, we have fleshed out this demonstration to more realistic, complex situations. This required development of a full robotic grasping platform that can handle cluttered scenes, realistic input devices appropriate for disabled people, and an augmented reality user interface. The development of such a system is the central challenge addressed in this paper.

C. Roadmap of this paper

To address this challenge, we have iterated through four designs of our assistive robotics system, denoted System 1-4. Section III presents an initial prototype, named *System 1*, tested on a single user and first reported in Weisz *et al.* [2012], which we very briefly summarize. Section IV describes user experiments using *System 2*, which improves on the prototype system with additional UI elements that allow more flexible and effective user control. In Section V, we describe *System 3* which integrates the novel EMG interface device and allows grasping in cluttered scenes, and test it's efficacy with an impaired user. In Section VI, we describe in *System 4* which allow improves the speed and reliability of the user interface and demonstrate it on a cohort of healthy subjects.

System 1: An HRI Grasping Platform Prototype

To begin addressing even the most basic needs of an impaired individual with assistive robotics technologies, we need an underlying platform for robust grasping and a user input paradigm. The grasping task can be decomposed into a four subtasks: Target object identification and localization, generation of grasp plans, picking an optimal plan, and executing the plan on the robot. Each subtask can be fulfilled by different modules which benefit from different user interaction strategies. By decomposing the tasks into explicit phases of a pipelined process, as in Figure 2, we can optimize user's interaction for each phase to make the best use of input modalities with limited bandwidth while guiding the grasping platform. Although fully automated approaches for each of these subtasks 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 HRIs with a shared-control assistive robotic device. Traditional EMG and EEG setups are expensive and difficult to deploy. In this work, we wanted to explore the boundaries of what can be achieved with devices that are more practical for a real world assistive device, both in terms of convenience and cost. We experimented with two low cost devices for detecting EMG, the Emotiv Epoc (Emotiv Systems Inc., San Francisco, CA, USA) and a custom device described in Section II-A.

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

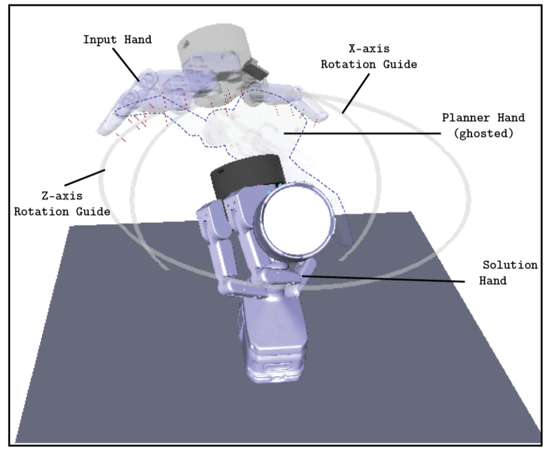

Fig. 3: An annotated screenshot of the prototype grasp planning user interface in GraspIt!. During online planning, the user is presented with an augmented reality view of the target object and three renderings of the hand interacting with the scene. The *Planner Hand*, which is the most transparent 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 best grasp quality.

bandwidth information gained from the user efficiently. This requires careful design of the interfaces provided to the user and of the control scheme for inferring intent from the user's input. Additionally, in order to present the user with reasonable grasping options, we need to extend the existing grasp stability analysis to deal with the most common problem that arises in unstructured environments, object localization errors due to sensor noise (See Weisz and Allen [2012] for an analysis of the effect of sensor noise on the Eigengrasp Planner). To explore these user interface issues, we developed an initial prototype platform, first described in Weisz *et al.* [2012].

A. Prototype Design Components

The manipulator arm for the initial prototype was composed of an industrial Stäubli TX60L robotic arm and a BarrettHand gripper. The object localization system was based on point clouds captured by a Microsoft Kinect depth 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. The Emotiv Epoc was chosen because of the convenience of its wireless form factor and relatively low cost.

EMG Input Processing: The Emotiv Epoc comes with three built-in signal processing modalities designed to detect emotional affect, facial movement, and EEG evoked responses. Combining these classifiers, we were able to derive a training paradigm for detection of four facial gestures robustly. For details see Weisz *et al.* [2013].

B. User Interface

We augmented the Eigengrasp Planner GUI in the GraspIt! simulator (Miller and Allen [2004]) 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 Figure 3.

The four facial gestures captured from the Epoc are treated as discrete on/off signals. We found some facial gestures, such as eyebrow raising, to be easier to maintain than others such as winking. These were assigned to control signals whose duration controlled some continuous value, such as position of the end effector along the guides. Two of the gestures are mapped to ”Yes” and ”No” inputs at decision points, while the remaining two control the rotation of the *Input Hand* along the guides shown in Figure 3. A video demonstrating the operation of this prototype can be found here: https://youtu.be/_b5ecKbIHBQ.

C. Software Platform

1) 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 TX60L's built-in inverse kinematics planner.

2) Recognition System: We use the *Model Ransac* method described in Papazov and Burschka [2011] to identify and localize the target object in the scene. This method generates features from pairs of oriented points on the surface of the object. Prospective models are processed off-line and put into 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 variant 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 good robustness and is extensible to multi-object scenes.

D. Evaluation and Shortcomings

Using this grasping platform, the experimenters were able to develop an interface for the Eigengrasp Planner and access it through a noisy, EMG based interface. However, we found that users with less experience and patience were not able to interpret the information in the grasp planning scene. In order to run user experiments, we needed a more informative UI and some faster planning options.

System 2: Adding Visual Feedback

After initial experiences with *System 1*, we noted several shortcomings in the design of the user interface. With additional visual cues, we were able to add several features and allow non-expert users to successfully navigate the grasping pipeline. We added elements to the UI that are illustrative of the current state of the planner and the available options at each phase of the planner (see Figure 4). Additionally, this extended UI gives feedback about the position of localized object with respect to the sensed environment, allowing the user to interact directly with the localization system. With these modifications, we also enable the user to grasp ”novel” objects that are not in the object database and use our pre-planned grasp database.

A. Handling Novel Objects

In order to handle objects that are not in the recognition system, we rely on the stochastic nature of the planning and recognition system and the discernment of the user. When automated systems fail, the user can reject the proposed solutions and wait for another. The parameters of the object recognition system can be tuned to recognize objects with similar parts by increasing the allowed error in the hypothesis testing stage of RANSAC. An example of this alignment can be seen in Figure 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 down-sampled point cloud from the depth camera. The user is responsible for rerunning the vision system until they see a reasonable alignment of the sensor data and detected model. This interaction also comes into play in the grasp planning phase, in which we rely on the user to reject grasps that may seem appropriate for the detected model but do not fit the actual unknown model well.

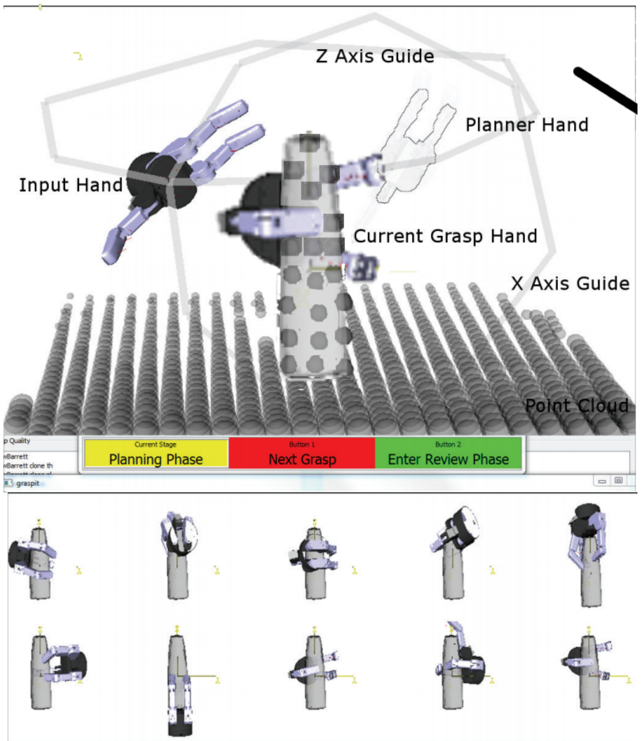

Fig. 4: *System 2 Interface - Online Planner Phase*. The user interface contains three windows: The main window containing three labeled robot hands and the target object with the aligned point cloud, the pipeline guide window containing hints for the user to guide their interaction with each phase of the planner, and the grasp view window containing rendering of the ten best grasps found by the planner thus far.

(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 shampoo bottle can be an appropriate proxy for the juice bottle in the planner.

Fig. 5: Results of the object recognition system with known and unknown objects.

B. Incorporating a Grasp Database

One useful aspect of mapping the object in the scene to a set of objects from a database is that we can also pre-plan 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 object from the database acting as a proxy for the target 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. 7 demonstrates such a grasp. In experiments, this grasp was successful 100% of the time, but only because the soft plastic surface of the object deforms during grasp acquisition to allow the finger to pass through the hole in the handle region of the bottle. Capturing this behavior in a simulator would require modeling dynamic object deformations. Currently, accurate simulations of such properties are too slow for sampling based planners, and so human annotation of such grasps is necessary. Where such grasps are available and match up with the object geometry, they are very useful.

To seed the grasp database, we ran the Eigengrasp Planner off-line six times for twenty minutes each 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.

C. Grasping Pipeline

These improvements result in a more flexible, but complex grasping pipeline. In this pipeline, there are three additional phases before the *planning-review-execute* pipeline outlined previously in which the augmented visualizations are used to provide the user with extra flexibility in initializing the planner. These phases allow the user to first control the vision systems object detection and localization, then review the set of available grasps retrieved from the pre-planned database, and finally to chose whether to initialize the planner to activate the online refinement or simply execute the retrieved grasp. The phases of the pipeline and transitions between them are outlined as a state machine in Figure 6 and Table I. Notice that these modifications create two possible paths to the final

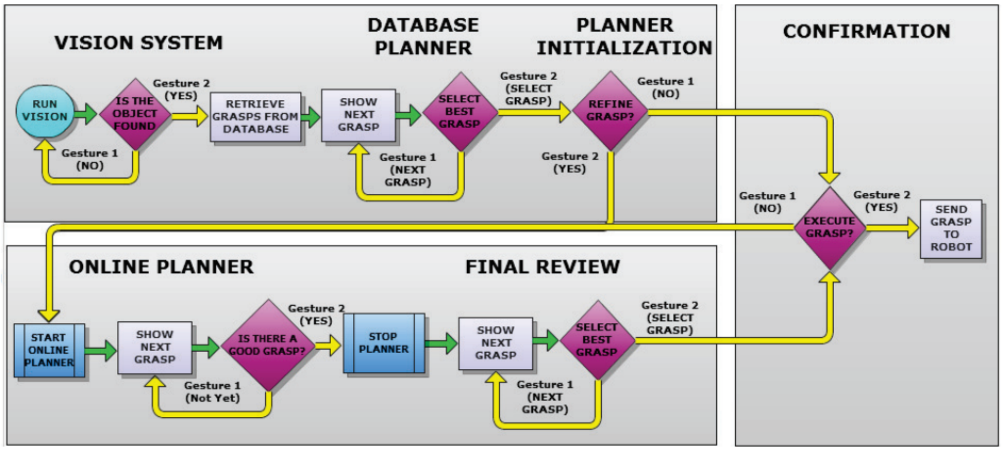

Fig. 6: The phases of the modified grasping pipeline incorporating more control over the vision system and the grasp database. If the user chooses the *nth* grasp from the database, *n+4* user inputs are required. If none of the grasps are suitable, the online planner can be invoked with a few simple inputs to refine one of the grasps further.

phase of grasp execution, trading off complexity for potential speed improvements.

To help the user manage this complexity, we've added several elements to the user interface, shown illustrated in Figure 4. Two of the facial gestures are associated with transitions between phases of the pipeline. Below the main illustration of the grasp planning scene, we have added a new *Guide Window* to tell the user what each of those gestures will do in the current phase. In this window the left side of the window shows the current phase in yellow, the center of the window shows the result of the Gesture 1 transition in red, and the right side of the window shows the result of the Gesture 2 transition in green.

The bottom window of the interface shows an illustration of the top ten grasps in the image to the user. This allows the user to judge their options as they are retrieved from the database or repopulated by the online grasp planner.

Each phase of the grasping pipeline is outlined in detail below:

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, described in detail in Section II-C2, uses a depth camera as input and is . In our first set of experiments, we used a single unoccluded object, and assumed that the recognition system works perfectly, so no user interaction was required. In these experiments, the user is responsible for ensuring that the detected model aligns to the object well where the user intends to grasp the object. As seen in Figures 5a the alignment found does not need to reflect the semantic or geometric reality perfectly, as the user is presented with many grasping options and can exercise their own judgment. The user can 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: 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 the *nth* grasp, only *n+4* inputs are required.

Grasp Planning Phase: 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 planner. While the planner is running, the "input hand" is made 30% transparent. A second, fully opaque "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 with 80% transparency showing the last tested state. The user interface presented while the subject is running the system is presented in Fig. 4. While the planner is running, Gesture 1 iterates through the grasp list and shows that grasp in the grasp planning scene using the "Current Grasp Hand". Gesture 2 stops the planner and enters the Review Phase. Whichever grasp is currently being shown is the one that is selected when the review phase is entered. Whichever grasp is currently being shown is the one that is selected when

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 I: The phase transitions table of a state machine representing the *System 2* pipeline. Each transition is evoked by a facial gesture. In this pipeline, gestures 3 and 4 only control the input to the online planner, and never evoke a state transition.

the review phase is entered.

Review Phase: The planner is stopped and the user has an opportunity to review the list of best grasps and choose one for execution. As in the previous phase, 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 phase 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 their intent and restart the planning process.

Confirmation: In this phase, Gesture 1 will restart the planner in the online planning phase, while Gesture 2 executes the grasp. The most input error is mistaken detection of two consecutive signals of the same type when only one is intended. Switching the gesture used to progress forwards seems unintuitive, but it is necessary to stop the user from inadvertently executing grasps through inattention or misclassification.

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 system's estimate of the object's position in the world. If the planner cannot find a solution to execute the grasp, the user is notified by flashing the screen black to indicate an error, similar to the *visual bell* used to indicate errors in terminal emulators. If this occurs, the user can restart the planning phase to find a new grasp. If the planner can find a solution, the grasp is executed with the actual arm and hand.

D. Experiments

In order to test the efficacy of our system, we recruited five healthy subjects to participate in an experiment to use the system to lift three objects from a table. All testing was approved by the Institutional Review Board of Columbia University under Protocol AAAJ6951. The results of these experiments were published in Weisz *et al.* [2013], and a video illustrating the experiments can be found at https://youtu.be/3YnbxVsJKs0.

1) Task: Each subject was asked to grasp and lift three objects using an Emotiv Epoc as input. Two of the objects, a flashlight and a detergent bottle, were in the database and available to the vision system. One of the objects, 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 asked to grasp the object five times, irrespective of direction.

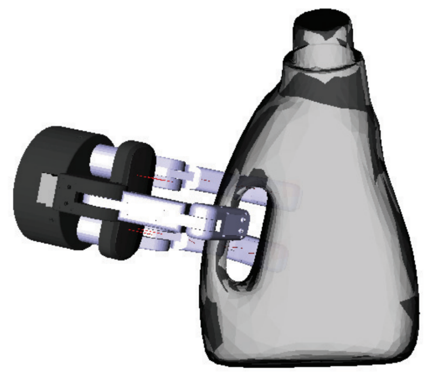

Fig. 7: This handle grasp for the detergent bottle is not a force closure grasp, but when chosen by the subjects in our experiments it succeeded 100% of the time. Adding a grasp database allows such semantically relevant grasps to be used in our system.

2) Training: The subject trained the Emotiv Cognitiv suite 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 to 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. To train the subject to perform the task, the subject was asked to perform the task twice in the virtual environment without executing the final grasp on the arm.

3) Results: The results of the experiments are reported in Tab. II. 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 contained only 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

Grasp	Subject	Successes	Mean Time(s)
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 II: Results from Experiment 2

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.

E. Discussion

During the experiment, we found that it was necessary to re-wet 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 initially, 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 phase of the pipeline at will. This lead to overemphasis of the facial gestures, which caused muscle fatigue. 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. In more cluttered scenes, this issue would be problematic. Finally, users reported that the pipeline guide window was difficult to read while focused on the task.

In the next section, we describe a different interface device that is designed specifically to measure facial EMG signals, along with some of the changes we made to the user interface to address the concerns subjects expressed during this experiment.

System 3: Novel sEMG Device with Impaired User Study

A. Surface EMG Recording

Although the Epoc made a reasonable early testbed, it has a number of practical issues. The device has a fairly large profile around the whole head, which is inconvenient for individuals with severe impairments that require head support. Additionally, we it was challenging to maintain consistent signal quality from the device, requiring repeated re-seatings and maintenance to make sure all of the electrodes were still making good contact. To address some of these issues, we adopted a novel input device under development at UC Davis which is designed to be used by severely impaired individuals. This device has an extremely noninvasive profile, requiring only a single sEMG recording site behind the ear.

As the muscles behind the ear are are innervated by nerves that 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 affect more distal muscles first, and more proximal muscles near the face and hands retain their function in more impaired individuals. The Superior Auricular (SA) 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.

A series of works including Joshi *et al.* [2011]; Perez-Maldonado *et al.* [2010]; Skavhaug *et al.* [2012] have shown that the input device 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 Figure 8. 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 Buttersworth 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

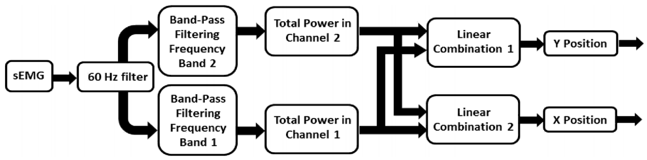

Fig. 8: The single sEMG signal is first processed through a 60 Hz noise filter. It is then run through two different band pass Buttersworth filters to extract two separate signals. The bands are then linearly combined to compute the x and y cursor positions.

filter outputs were combined linearly as described in Equation 1.

$$x_{pos} = 1.75\frac{ChannelPower_1}{gain_1} - 0.75\frac{ChannelPower_2}{gain_2} \tag{1a}$$

$$y_{pos} = 1.75\frac{ChannelPower_2}{gain_2} - 0.75\frac{ChannelPower_1}{gain_1} \tag{1b}$$

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 PA 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 Figure 12. 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.

To adapt this system to our use, we have adding 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.

B. sEMG GUI

To send signals to the grasping system, the user controls a cursor to hit one of four targets, as illustrated in Figure 9. During grasp planning, they are overlaid on the augmented reality display. 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 user's 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 force the selection to *timeout* by avoiding rest for these four seconds, canceling the selection.

We map these inputs to following a similar strategy to that used for each facial gesture in *System 2*. 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 exert near continuous control over the approach direction.

C. 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 that many of the grasps for the objects may collide with other 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 5cm of empty space between each objects.

Handling cluttered scenes brings up a number of challenges. First, it slows down the online planning phase. There are many fewer possible grasps and the obstacles divide the state space into discontinuous regions, which creates more local minima in the value of the quality function, which slows down convergence. Additionally, adding more geometry to the planning scene slows down collision detection, which is a bottleneck for grasp planning. Second, many of the grasps produced by the planner may not have a reachable path to grasp. Previously we made the optimistic assumption that most grasps were reachable, but in clutter this is no longer a valid assumption. 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 the first two issues, we implemented a fast, online reachability test which the user can clearly interpret. When good grasp candidates are found, *System 3* checks that an entire valid trajectory can be generated using the CBiRRT planner described in Berenson *et al.* [2009b]. Unreachable grasps are placed at the end of the list of grasps and colored red in the grasp preview window (see Figure 11). This allows the user to see that progress is being made even when no new reachable grasps are being generated.

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

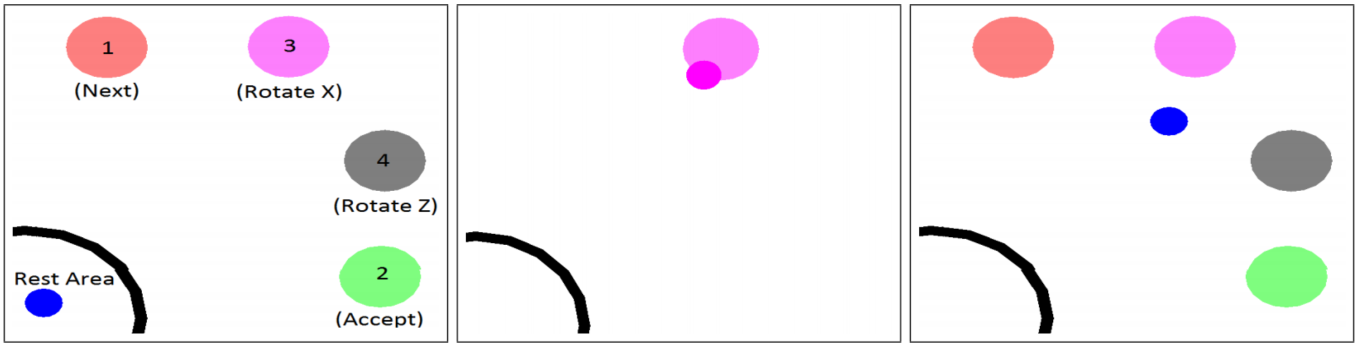

Fig. 9: The sEMG Interface: (a) The user interface is composed of 4 targets overlaid on the grasping scene. Target 1 usually signals acceptance of the current option. Target 2 toggles the next option. Targets 3 and 4 provide input to the planner.(b) Hitting one particular target changes the color of the cursor to reflect the selection and makes the other targets unavailable. (c) 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.

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 have the intuition that the region they are trying to grasp from is not within the robots workspace.

D. GUI Modifications For Clutter and sEMG Interface

A number of changes have been made to the user interface to accommodate both the added visual complexity of overlaying the sEMG control interface on the planning scene and the added difficulty of interpreting the multi-object scene. We have redesigned the UI with a cleaner look and feel that implements a number of new features.

The *System 3* interface layout is outlined in Figure 10, 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 judgment 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 scene. 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 so that all detected objects are visible and that the user's intuition is unimpeded by deformations due to the aspect ratio. Fourth, we removed 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 rendered as a translucent layer on top of the grasp planning scene, allowing the subject to see both at the same time. The scene is chosen so that the relevant objects are centered in the scene. While the user is not actively using the cursor it 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 between an accidental selection of these two options. The middle two options modify the demonstrated approach direction, and accidental selections have minimal impact.

E. System 3 Pipeline

Initialization: This phase has been modified from *System 2* in that the view is aligned to the true view of the camera, rather than being centered on the target object. This improves the user's overall situational awareness. 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: The first detected 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 non-target objects are all highlighted in red. The non-target objects are replaced with lower resolution models when a target is selected, which makes the planning phases faster.

Initial Review: As in *System 2*, the user is presented with a list of pre-planned grasps from a precomputed database. This phase has been modified from *System 2* to present the user with a clearer visualization and reachability information. As

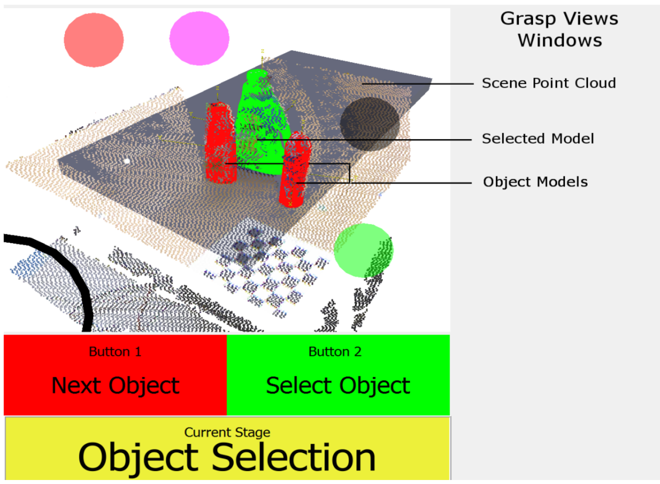

Fig. 10: *System 3 Interface - Object Selection Phase.* The subject is able to see the planning scene in the main UI window. The window on the bottom tells the user the current phase and what the green and red inputs will do in this phase. In this phase, the subject sees the the point cloud and hits 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.

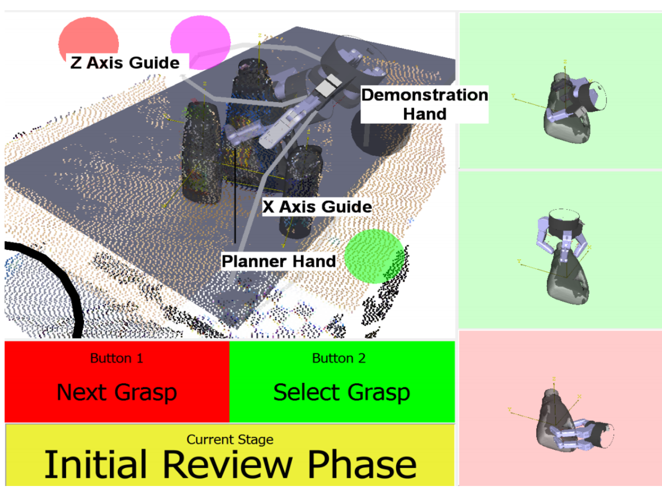

Fig. 11: *System 3 Interface - Initial Review Phase.* After the subject selects the object, the Grasp View pane on the right is populated with a set of grasps from a database. Grasps that are reachable appear on a green background, while unreachable grasps are red. A robot hand appears that the user 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 topmost grasp in the grasp view window is the currently selected grasp, which is rendered in the planning scene with the planner hand.

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. Moving the demonstration hand will also cause the grasp list to re-sort bringing the new approach direction to the top of the list. Reachable grasps are presented on a green background, while unreachable grasps are presented on a red background. 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 *System 2*. The user is presented with the choice to either accept the grasp from the third phase with input 1, proceeding straight to the Grasp Choice Confirmation phase or they can send input 2 to refine their chosen grasp further.

Grasp Refinement: This phase has been modified from *System 2* to provide feedback about the feasibility of their demonstrated approach direction and grasps. New grasps 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. Sending input 2 stops the planner and proceeds to the Final Grasp Review pahse.

Final Grasp Review: This phase is similar to *System 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.

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

F. Validation

To validate this system, we recruited a male 25-30 year old impaired subject with limited upper limb mobility due to a C3-C4 spinal injury. All testing of *System 3* was approved by the Institutional Review Board of the University of California, Davis under Protocol 251192-10. 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 subject's PA muscle to avoid the need to shave the subjects hair. 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 the Columbia University Robotics Group lab. The setup is shown in Figure 12. The top of the figure shows the subject with the device attached and using the grasp planning system. The bottom left image is a closeup 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. The results of this experiment were published in Weisz *et al.* [2014], and a video illustrating the experiment can be found at https://youtu.be/tRPXmb9yUbA.

1) Task: Due to limitations on the impaired subject's time, we were only able to complete three trials using the system. In these three trials, the subject was asked to pick up an object from a cluttered, multi-object scene. In the first two attempts, he was asked to use the online planner to refine one of the pre-planned grasps. In the first attempt, he grasped the laundry detergent bottle. In the second attempt, he grasped the shaving

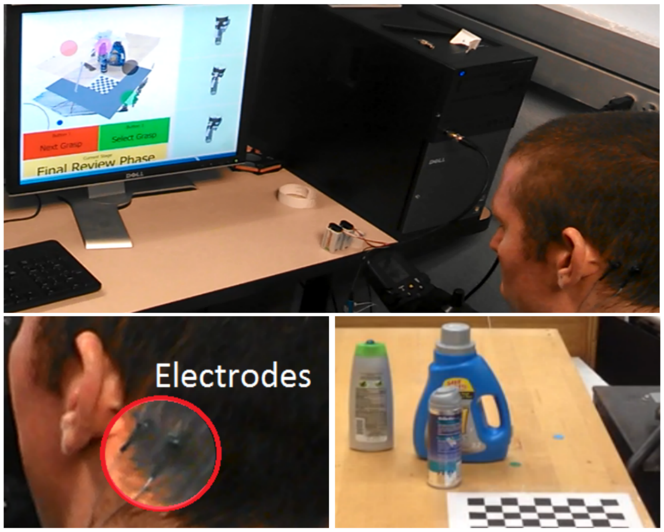

Fig. 12: An impaired subject in the UC Davis RASCAL lab (top) operating our sEMG-Assistive Grasping interface to grasp a shaving gel bottle in the Columbia Robotics Group Laboratory (bottom right). The two small black clips behind the subjects ear (bottom left) are surface EMG electrodes (used in differential mode) to detect activation of the Posterior Auricular (PA) muscle to direct the system to pick up the object in this multi-object scene.

gel bottle. In the third attempt, he 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 he was 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 he was trying to reach and we tracked the number of mistaken target activations, which would lead the user to loop back through that 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.

2) Training: 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 him on which target to hit while the experimenter controlled the cursor with a computer mouse. This allowed the subject to familiarize himself 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 he appeared to be conversant with the system, we turned over control to the subjects sEMG interface.

3) Results: The results of the experiment are shown in Tab. III. The subject was able to grasp the objects successfully on all three attempts. 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 an average of 25 timeouts, and 1 mistakenly selected targets per

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 III: sEMG Experiment 1 Results

attempt. 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 the previous step where necessary to correct mistakes. Several mistakes in a row are necessary to actually realize mistaken actions on the robot.

4) Discussion: These results, while promising from the perspective that a single experiment participant 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 trained to use the PA muscle, which is smaller and has more variable performance. Time restraints did not allow for extensive training of the subject, and so when switching from training to task, the additional cognitive load appears to have degraded performance. A second problem was that the online reachability tester is fairly slow using the CBiRRT 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 he 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.

System 4: A practical assistive grasping platform

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. Our initial prototype used an industrial arm, which is extremely accurate and has a large workspace, but is too heavy and expensive to be part of an assistive robotic setup. Additionally, this large, high precision arm does not reflect the performance characteristics of an arm which is affordable and practical for a robotic wheel chair. The Kinova Mico arm is more 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.

A. Adaptations For the Mico Manipulator

The Kinova Mico arm is a six DOF arm with a two finger gripper. The fingers each have two joints coupled to a passive under-actuation mechanism that enables both enveloping grasps of convex cross sections of objects and fingertip grasps. These fingers are made of a hard plastic which has relatively little friction, which implies that the fingers of the hand must be well aligned to the surface of the object to achieve a stable grasp.

The transmission of the under-actuation mechanism of the hand is designed such that the fingertips remain at roughly same angle relative to the palm through most of the range of the finger's motion, similar to the motion of a parallel jaw gripper. For hands of this type, we can trivially estimate the contact point of the grasps without performing the kinematic simulation of closing the hand in GraspIt!, which is the most computationally expensive aspect of grasp analysis. In this work, we applied 10x multiplier to the quality measure of grasps whose estimated contacts aligned to within 3° of the normal to the nearest surface. This was sufficient to generate only well aligned, reasonable grasp candidates.

B. Improved Online Reachability Checking

Given our previous insight that the online reachability testing is a bottleneck for the online grasp refinement, we wanted to explore a different options for online reachability checking. This motivated us replace the OpenRave trajectory planner with with the MoveIt! planning environment (see Sucan and Chitta [2013]), which interfaces with a large number of planners in the OMPL planning library Şucan *et al.* [2012].

The OMPL planners have different strategies with different performance properties. In order to investigate which one is appropriate to grasping in the cluttered scenes with the Mico Arm, we captured 10 scenes similar to that in Figure 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.

The online reachability check is the final stage of filtering before grasps are presented to the user. The sampling nature of the planner implies that there will be a great deal of temporal correlation between grasp requests. In order to take advantage of this correlation, 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.

Because the trajectory planners are stochastic, their performance is highly task specific and sensitive to parameters such as minimum segment length and allowed planning time. 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. Two of the Probablistic Road Map (PRM) planners (Kavraki *et al.* [1996]) planners performed the best using the caching scheme, succeeding in 43% of the grasps, and the vanilla PRM implementation 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.

The single-query bidirectional variant of the PRM planner (SBL) produced plans that seemed smoother in the region near the object. Planning grasps in a cluttered scene is state space in which there is a very narrow valid region near the goal state, and so one might expect a bidirectional planner to find a more optimal path out of that region because it will spend more resources directly on that part of the problem.

However, we found empirically, when the caching scheme failed to find a reasonable neighbor, the SBL planner's success rate dropped to 30%, whereas the PRM planner's success rate remained the same. This led to slight lag in performance as the cache was populated. 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 phase of the pipeline. [1]

C. Further UI Improvements

In our previous systems, 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 system, the purpose of these inputs can also change in each phase. In addition, our previous test user 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 system, the UI window is adaptive, providing more visual cues to the user as to what their goal is in each particular phase of the pipeline. The grasp previews are integrated with the pipeline guide display, 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 Figure 13 and Figure 14. While this may seem like unnecessary complexity, the UI is more visibly different in each phase and less extraneous information is presented. This seems to help subjects keep track of what phase they are in and what its purpose is.

[1] Although 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.

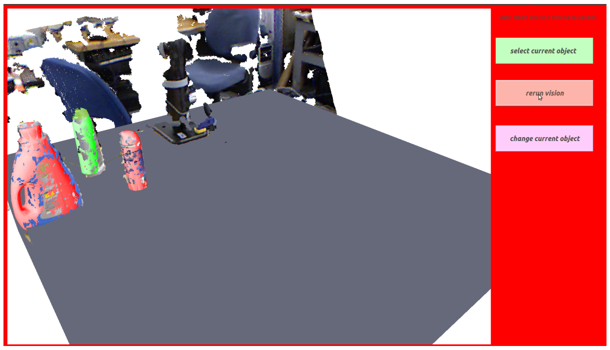

Fig. 13: *System 4 - Object Recognition and Selection State.* The graspable objects in the seen are highlighted in red and green. Sending input 1 selects the green object as the target, input 2 cycles to the next object, and input 3 triggers the object recognition system to refresh. The background UI area is rendered in red while the recognition is still processing.

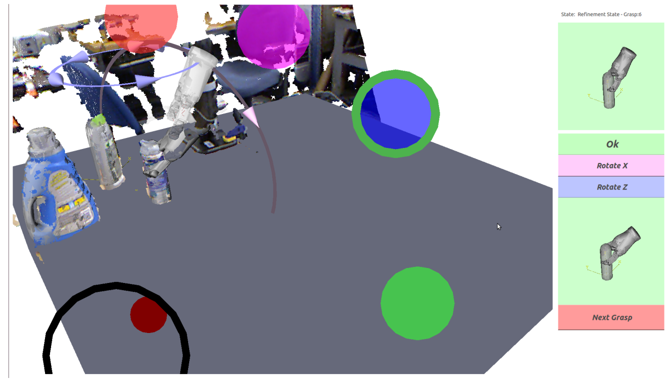

Fig. 14: *System 4 - Grasp Refinement State.* The buttons on the right function as both guides for the result of hitting the color coded input options that will be presented to the use, as well as buttons that the user and experimenter use during the training stage. The sEMG is interface overlaid on the planning scene with the selected target highlighted in green.

System 4 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 *System 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 *System 3*, the user is presented with a list of preplanned grasps from a precomputed database. The UI presented is shown in Figure 14, in which 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 phase. Input 3 will iterate through the available grasps, whereas input 4 will return to the Object Recognition and Selection Phase.

Grasp Refinement: This phase is similar to *System 3*, but with one fewer grasp displayed more prominently. 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 phase, 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: as in the previous phase, this phase has been adapted to have only two grasps, the top showing the current selection and the bottom showing the next selection. Input 1 proceeds to the Grasp Choice Confirmation phase, 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 *System 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 phase and input 2 to send the grasp for execution on the robot.

D. Validation

To validate these design decisions, we tested our pipeline on 5 healthy subjects, 2 male and 3 female, ages 22-30. All testing was approved by the Institutional Review Board of the Columbia University under Protocol AAAJ6951. 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 representing the 'selected' input, which is outlined in green as shown in Figure 14. 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. This change allowed us to focus on testing improvements the user interface and grasp planning pipeline without needing the more extensive training necessary to train a subject to achieve full 2D control over the cursor.

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$^{\text{TM}}$ silicone putty, as shown in Figure 15. 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 response to eyebrow raises and little response to jaw motion.

2) Training: sEMG Device: Each subject was trained on the sEMG user interface without the grasp planning system. In the training system, 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*

Fig. 15: The sEMG system electrodes. In these experiments, we placed the electrodes behind the ear of the subject to measure contractions of the PA muscle. We stabilize the electrodes by wrapping the head of the electrodes in Silly Putty silicone putty.

until it overlaps with the *desired target*, 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.

3) Task: We placed the objects on the table in proximity to one another as shown in Figure 14. We asked the subject to grasp each object three times, the first time from any direction they 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, it is easy to find such object positions.

4) Results: The results of the experiment for 5 subjects are tabulated in Tab. IV. On average, the subjects were successful in grasping 82% of the objects within 92 seconds of the first time their cursor left the 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

Grasp	Subject	Success	Time	Grasp	Success	Time
Detergent Bottle Top	1	Yes	75	Shampoo Bottle Open Choice	Yes	93
	2	Yes	53		Yes	121
	3	No	45		Yes	63
	4	No	122		Yes	95
	5	Yes	135		Yes	117
	Mean	60%	86		100%	98
Detergent Bottle Side	1	No	66	Shaving Gel Top	No	83
	2	Yes	40		No	123
	3	Yes	52		Yes	112
	4	Yes	80		No	139
	5	Yes	85		Yes	97
	Mean	80%	64		60%	111
Detergent Bottle Open Choice	1	Yes	50	Shaving Gel Side	Yes	65
	2	Yes	57		Yes	52
	3	Yes	53		Yes	57
	4	Yes	135		Yes	88
	5	Yes	128		Yes	92
	Mean	100%	85		100%	71
Shampoo Bottle Top	1	Yes	151	Shaving Gel Open Choice	No	73
	2	Yes	72		Yes	59
	3	Yes	60		Yes	76
	4	No	126		Yes	81
	5	No	104		Yes	85
	Mean	60%	102		80%	75
Shampoo Bottle Side	1	Yes	134	Average Per-for-mance	66%	87
	2	Yes	95		88%	75
	3	Yes	132		88%	72
	4	Yes	164		77%	114
	5	Yes	143		88%	109
	Mean	100%	133		82%	92

TABLE IV: Results from Experiment 3. On average, the subjects were successful in grasping 82% of the objects within 92 seconds of the first time their cursor left rest area.

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 Figure 16. 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 Figure 16. 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.

III. Conclusions

In this paper, we've discussed the many details involved in building a full assistive grasping system around an online grasp planner. The key challenge was to find the right balance of complexity and usability, particularly with respect to the design of the visual aspects of the interface. A clear user interface is the key to allow a non-expert user to apply their intuition to the grasping problem and provide the added value that makes the system work well in spite of sensor noise and

Fig. 16: 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.

any shortcomings in the heuristics applied by the automated parts of the system. Careful development of this platform has allowed us to produce an extremely capable system around components whose cost and complexity is not prohibitive.

Through this work with the UC-Davis sEMG device, we have pushed the boundaries of what can be accomplished with a minimally invasive, facial muscle driven input. First, we extended our basic system design to a more complex environment with multiple objects in close proximity to one another. This involved augmenting the user interface with additional phases to select the desired object, adding an online reachability tester, and producing a new UI with a dedicated interface including a cleaner UI with an integrated sEMG driven option selection overlay. After initial validation of the interface on an impaired user, we developed a series of improvements to the user interface, the online grasp planning, and online reachability filter to address the most challenging issues that caused our initial user to take up to eight minutes to make a single grasp selection. We developed a novel control paradigm for testing these changes without changing the visual interface which allowed us to validate the updated system on naive users without the extensive training necessary to train an individual to develop full 2-D control.

This study serves as a pilot to validate the design choices of the system on a path towards more experiments with impaired users. Even though this paradigm requires the user to make up to four motions for selections which had previously required one, we observed that users took on average 1/8th the time to make grasp selections in the latest version of the sEMG driven planner. We did not explicitly measure how long the users spent in each stage of the pipeline, but one of the most costly phases was observed to be the grasp refinement stage, when it was used. In order to improve performance in this stage, we would have to improve the performance of the collision detection system, which is the dominant cost of the simulated annealing driven grasp refinement. Overall, the majority of the failures to grasp an object were caused by the difficulty of grasping cylinders along the major axis with a gripper, represented by grasping the detergent bottle or shaving gel bottle from above. In these grasps, squeezing the gripper can easily cause the object to be ejected. Subjects cannot seem to learn to expect this behavior without having experienced it a number of times, as they do not have a good sense of the friction properties of the gripper. To improve this behavior, we would have to implement a more complex feedback controller during the grasping process. It is also likely that with greater experience, the subjects would have been more familiar with the kind of grasps that cause ejection.

This work demonstrates one the first EMG driven grasping systems that we know of that allows a user to grasp an object in a somewhat cluttered scene, or integrates user intent with the intermediate level of control we have proposed. The sEMG device itself is very minimalistic, and could itself be embedded in the frame of a pair of glasses, which makes this device a real candidate for evolving to a consumer level product. Future work for this paradigm will be to refine the training paradigm to make learning 2D control over the device easier, exploring new control paradigms, and extending the user interface to control the locomotion of a motorized wheel chair or mobile manipulator assistive platform. We have also begun some exploration towards more complex grasp quality measures that integrate more complex artificial intelligence techniques such as "deep learning" which may have less reliance on the accuracy of the object recognition system.

References

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.

Christian J Bell, Pradeep Shenoy, Rawichote Chalodhorn, and Rajesh Rao. Control of a humanoid robot by a noninvasive brain-computer interface in humans. *Journal of Neural Engineering*, 5(2):214–20, Jun 2008.

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.

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.

Claudio Castellini and Patrick van der Smagt. Surface EMG in advanced hand prosthetics. *Biological cybernetics*, 100(1):35–47, January 2009.

Matei T. Ciocarlie and Peter K. Allen. Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28(7):851–867, 2009.

Christian Cipriani, F Zaccone, Silvestro Micera, and Maria 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.

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.

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.

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.

Sanjay S. Joshi, Anthony S. Wexler, Claudia 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.

L.E. Kavraki, P. Svestka, J.-C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Robotics and Automation, IEEE Transactions on*, 12(4):566–580, Aug 1996.

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.

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.

Andrew T. Miller and Peter K. Allen. Graspit!: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11:110–122, 2004.

Gernot R Müller-Putz, Reinhold Scherer, Gert Pfurtscheller, and Rüdiger Rupp. EEG-based neuroprosthesis control: a step towards clinical practice. *Neuroscience Letters*, 382(1-2):169–74, 2005.

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.

Claudia Perez-Maldonado, Anthony S. Wexler, and Sanjay S. Joshi. Two dimensional cursor-to-target control from single muscle site semg signals. *Trans. of Neural Systems and Rehabilitation Engineering*, 18, April 2010.

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.

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.

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.

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.

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.

Hannes Schmidl. The inail myoelectric b/e prosthesis. *Orthop Prosth Appl J*, pages 298–303, 1965.

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.

David E. Sherman and Andrew Lippay. A russian bioelectric-controlled prosthesis. *Canadian Medical Association journal*, 92(5):243, 1965.

Ida-Maria Skavhaug, R. Bobell, Benjamin Vernon, and Sanjay 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.

Ioan A. Sucan and Sachin ”MoveIt!” Chitta, 2013. Available Online from http://moveit.ros.org.

Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. http://ompl.kavrakilab.org.

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 BrainGate2 Neural Interface System. In *International Symposium on Experimental Robotics*, 2010.

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.

Jonathan Weisz and P.K. Allen. Pose error robust grasping from contact wrench space metrics. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 557–562, May 2012.

Jonathan Weisz, Benjamin Shababo, and Peter K. Allen. Grasping with your face. In *Proc. Int. Sym. on Experimental Robotics*. Springer, 2012.

Jonathan Weisz, Carmine Elvezio, and Peter K. Allen. A user interface for assistive grasping. In *Proc. Int. Conf. on Intelligent Robots and Systems*. IEEE Press, Nov. 2013.

Jonathan Weisz, Alexander G. Barszap, Sanjay S. Joshi, and Peter K. Allen. Single muscle site semg interface for assistive grasping. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, September 14-18, 2014*, pages 2172–2178, 2014.

Andrzej Woczowski and Marek Kurzyski. Human-machine interface in bioprosthesis control using EMG signal classification. *Expert Systems*, 27(1):53–70, February 2010.

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.