

Assistive grasping with an augmented reality user interface

Jonathan Weisz¹, Peter K Allen¹, Alexander G Barszap², and Sanjay S Joshi²

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 human-in-the-loop manner, including the use of a novel cranio-facial electromyography input device. The system uses an augmented reality interface that allows users to plan grasps online that match their task-oriented intents. The system uses grasp quality measurements that generate more robust grasps by considering the local geometry of the object and the effect of uncertainty during grasp acquisition. This interface is validated by testing with real users, both healthy and impaired. This work forms the foundation for a flexible, fully featured human-in-the-loop 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 human-in-the-loop assistive devices out of the research environment and into the lives of those that need them.

Keywords

Human-robot interaction, grasp planning, shared control

1. Introduction

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, noninvasive surface electromyography (sEMG) based input device, even in somewhat cluttered scenes. The sEMG technique involves positioning electrodes on the surface of the skin and measuring the combined electrical activation of underlying motor units in the vicinity of the electrodes (Cram and Criswell, 2011). 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-planning, and online 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. Because of these impairments, individuals 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 human-in-the-loop assistance.

Irrespective of the problems posed by limited input devices, robotic grasping is challenging for a number of reasons. Complex robotic hands have many degrees of freedom, so the space of possible grasps is large and computationally expensive to explore. Standard approaches to planning in high-dimensional state spaces are likely to fail with multi-fingered hands, especially as the grasp itself

¹Department of Computer Science, Columbia University, New York, NY, USA

²Department of Mechanical and Aerospace Engineering, University of California, Davis, USA

Corresponding author:

Jonathan Weisz, Department of Computer Science, Columbia University, 500 West 120 Street, New York, NY 10027, USA.

Email: jweisz@cs.columbia.edu

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 analyzing 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 to 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 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). Then we present new results from our final user validation study of our sEMG paradigm, which 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 user’s ability to understand the scene and produce correct grasps in complex, cluttered environments.

2. Related work

2.1. Human–computer interfaces for assistive robotics

There is a long history of assistive robotic systems using electrophysiological signals as input, with work going back as far as Schmidl (1965) and Sherman and Lippay (1965). In the time since, there have been myriad approaches and refinements of proposed 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; i.e. 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, electrooculography (EOG), or EEG. Within this category, modalities can be further divided according to where the signals are recorded

from. For example, 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. Next we describe the different control paradigms used in related systems using human–robot interface devices suitable for impaired individuals.

2.1.1. Direct joint mimicry. The most intuitive, low level of control of a robotic arm involves having the robot arm directly mimic the motions of the user. The joint angles of the robot are directly mapped to joint positions of a user moving his or her own arm naturally. It is possible to reconstruct a user’s movements using distal limb surface EMG signals (Artemiadis and Kyriakopoulos, 2011; Castellini and van der Smagt, 2009). One advantage of this type of control allows users to express their desires explicitly, enabling them to specify how the arm is to avoid obstacles. 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.

2.1.2. 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, who has to attend closely to each joint. The movements of the joints are not directly related to the user’s goal of manipulating some object. Control of a manipulator through such an interface is generally not possible because manipulators have many joints. The manipulator is generally controlled by simple open and close commands. For example, in Horki et al. (2011), hand opening or closing and elbow flexion or extension are controlled by EEG signals.

For a prosthetic arm with few degrees of freedom, such as the two-joint arm of Horki et al. (2011), this control strategy can be effective. However, higher-DoF arms require more control channels and a high degree of coordination to perform complex tasks. For example, to grasp an object using a six-DoF arm and a simple gripper, the user must move in a straight line toward an object or the gripper might knock the object over before it is in position. This requires precise real-time coordination of all six joint velocities, which requires lower noise, higher bandwidth control inputs than are available through noninvasive interfaces.

2.1.3. 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. This control scheme has been implemented using both invasive high throughput systems (Vogel et al., 2010), and less invasive systems with lower throughput, such as surface facial EMG and EOG (Gomez-Gil et al., 2011; Postelnicu et al., 2011; Ranky and Adamovich, 2010; Sagawa and Kimura, 2005; 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.

2.1.4. Discrete state-level control. Robotic hands typically have too many degrees of freedom to control directly. One alternative is to allow a user to switch between hand postures designed for a specific task. A simple open and close command is sufficient for some basic grasping applications. More complex hands may allow several preset grasp configurations, such as a wider spread of the fingers for holding a ball or a narrower spread for pinching a pencil. Other configurations for common tasks, like button pressing, can also be useful. Many such strategies have been explored, as in Cipriani et al. (2008); Ho et al. (2011); Matrone et al. (2011); Wołczowski and Kurzyński (2010); Yang et al. (2009).

These control schemes represent a tradeoff between flexibility and simplicity of use. This tradeoff is especially important as the complexity of the hand increases. To directly control the fingers for a hand as complex as, for example, the BarrettHand, the user would need to coordinate four separate signals (one for each finger, and one for the angle between the fingers) in real time to avoid tangling the fingers or knocking over the object. Discrete-level control allows more degrees of freedom to be controlled safely with fewer inputs.

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. Because there is not necessarily an easy way of associating the path that must be taken in that decision tree with the desired goal, these control schemes have a steep learning curve.

2.1.5. Task-level control. The key challenges of using non-invasive human–robot interfaces are 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 do it, more effective. Indeed, it has been shown that users find human–robot interface 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.

Bell et al. (2008) used EEG signals to select targets for pick and place operations for a small humanoid robot. Wytowich et al. (2010) used EEG signals to control pick and place operations of a four-DoF Stäubli robot. Bryan et al. (2011) presented preliminary work, extending this approach to a grasping pipeline on the PR2 robot. In that work, a 3D perception pipeline was used to find and identify target objects for grasping and EEG signals were 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. Scherer et al. (2011) 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 might not present the user with appropriate choices.

2.1.6. Task-oriented shared control. An emerging alternative to the purely task-oriented approach is to blend end effector control and task-oriented control (for example, Javdani et al., 2015). In this approach, the user’s input demonstrates some approximation of the desired solution or constraint that an automated planner can make use of. Mülling et al. (2015) showed that this strategy can improve performance in grasping tasks even when using an invasive brain–computer interface (BCI) device with relatively high bandwidth. In our work, we show that this strategy can allow noninvasive devices with much lower bandwidth to exercise similar performance in accomplishing complex tasks with high reliability.

2.2. The Eigengrasp grasp planner

In Ciocarlie and Allen (2009), our laboratory introduced the Eigengrasp planner, which allowed the user to grasp objects reliably by demonstrating only an approximate approach direction. In this work, we have expanded on 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.

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 and uses this to indicate approximately where 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, Eigengrasps were introduced, 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.

This approach has a number of practical advantages. The nature of the optimization approach, which gradually moves toward 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 and Allen (2009), the planner was demonstrated by having the operator manually 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 paper, 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.

2.3. Roadmap of this paper

To address this challenge, we have iterated through three designs of our assistive robotics system, denoted Systems 1 to 3. Section 3 describes user experiments using *System 1* with five unimpaired subjects. In Section 4, we describe *System 2*, which integrates the novel EMG interface device and enables grasping in cluttered scenes, and test its efficacy with an impaired user in a remote location. In Section 5, we describe *System 3*, which uses a different set of hardware and also improves the speed and reliability of the user interface and demonstrates it for a cohort of unimpaired subjects.

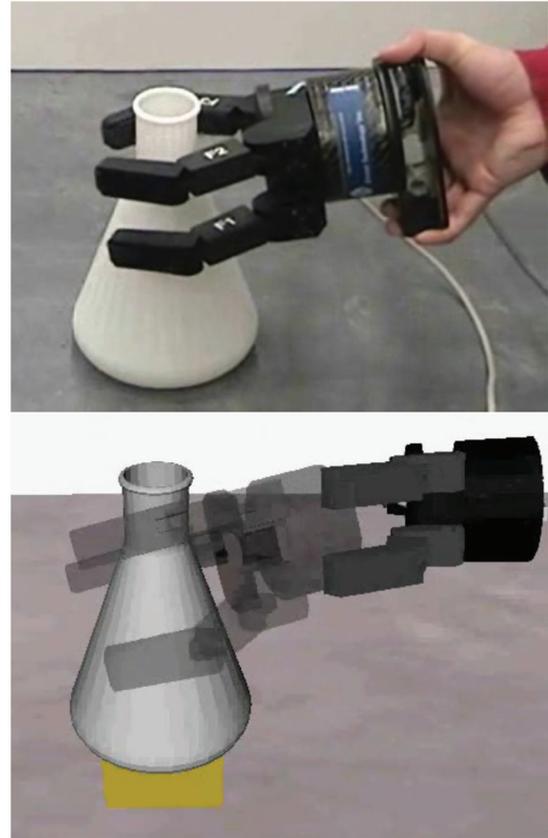


Fig. 1. An operator demonstrating the Eigengrasp planner by manually guiding the robotic hand to guide the planner in the virtual environment (Ciocarlie and Allen, 2009).

3. System 1: A human–robot interface grasping platform

In Weisz et al. (2013), we described a BCI-enabled grasping platform, through which we outlined a general strategy for an online assistive grasping system, based on an earlier system, which we described in Weisz et al. (2012). The grasping task can be decomposed into 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.

Putting the human in the loop when planning and executing the grasp in real time fundamentally changes the nature of the problem, as compared with a fully automated system. The key challenge becomes that of conveying information

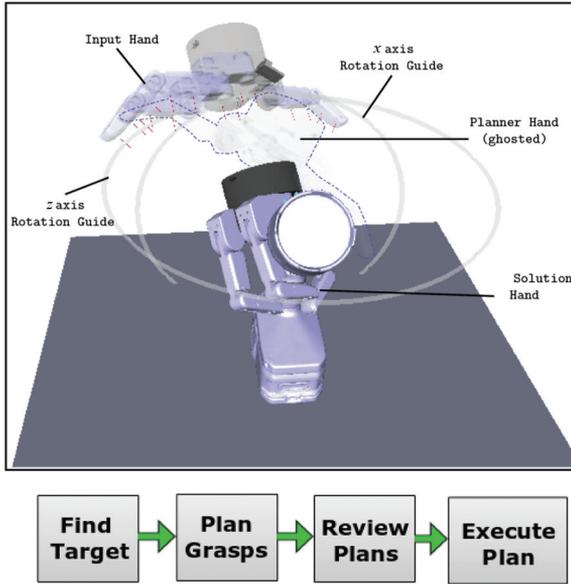


Fig. 2. Top: Annotated screenshot of the prototype grasp planning user interface in GrasPIt!. During online planning, the user is presented with an augmented reality view of the target object and three renderings of the hand interacting with the scene. The *planner hand*, which is the most transparent 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 the available control directions. The *solution hand*, which is fully opaque, demonstrates the best grasp currently available. This is the grasp that is closest to the approach direction that the *input hand* is demonstrating and that also has the best grasp quality. Bottom: 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.

to the user effectively about the state of the system and then using the low 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, 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.

3.1. 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

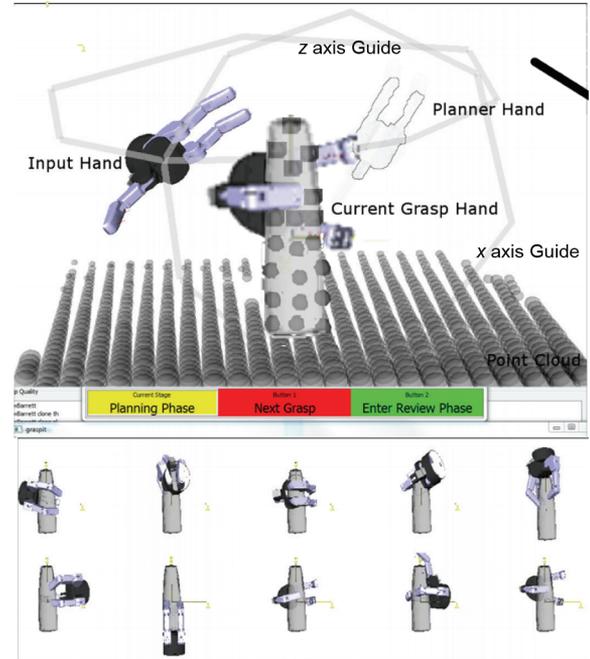


Fig. 3. System 1 interface—online planner phase. The user interface contains three windows: The main window containing three labeled robot hands and the target object with the aligned point cloud, the pipeline guide window containing hints for the user to guide interaction with each phase of the planner, and the grasp view window containing rendering of the ten best grasps found by the planner so far.

camera. There are many possible paradigms for integrating human–robot interfaces 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, in terms of convenience and cost. We experimented with low-cost devices for detecting EMG, the Emotiv Epoc (Emotiv Systems Inc., San Francisco, CA, USA). 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 (Weisz et al., 2013).

3.2. 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 virtual fixtures that allow the user to guide the planner fully inside the simulator. The augmented grasp planning scene is illustrated in Figures 2 and 3. To find the most easily detected facial gestures for each subject, we asked the subjects to lift both eyebrows, wink with each eye, clench each side of the jaw, and smile. This is a subset of the facial



Fig. 4. Results of the object recognition system with known and unknown objects. (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 system's object database, a shampoo bottle, in white. (b) The two objects that are aligned on the right of (a). 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.

gestures that are detectable by the Epoc's facial gesture classifiers that are easy to explain and demonstrate to subjects. For each subject, we selected the four gestures that produced the least cross talk in the facial gesture detector on a cursory examination.

Some facial gestures, such as eyebrow raising, are easier to maintain than others, such as winking. These were assigned to control signals whose duration controlled some continuous values, such as the position of the end effector rotating along the guides, as shown in Figure 2. Two of the gestures that are not as easy to maintain were mapped to signals analogous to “yes” and “no” at decision points.

3.3. Software platform

3.3.1. Planning and kinematics. Planning for the motion of the arm is done in OpenRAVE using a bidirectional random tree planner (Berenson et al., 2009a), and small linear motions near the object are planned using the TX60L's built-in inverse kinematics planner.

3.3.2. 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 offline and put into a feature database for testing model hypotheses. Features are sampled from the sensor data matches in the database. 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. The implementation used in this system takes between 15 s and 30 s to process the scene.

Handling novel objects. 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 solution and wait for another. An example of this alignment can be seen in Figure 4(a). 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 was modified to include a down-sampled point cloud from the depth camera. The user is responsible for rerunning the vision system until a reasonable alignment of the sensor data and detected model is displayed. 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.

3.4. 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 allows users to grasp objects more quickly in cases where they judge that one of the available preplanned grasps will work. In cases where there is no reachable grasp from the direction users want to use the object, for example, because of workspace constraints or obstacles, they can still activate the online Eigengrasp planner to find new grasps that are more appropriate to the current situation than the generic grasps in the database. This can provide an optimized experience for common cases while allowing the user flexibility and control.

Using a grasp database also allows us to manually design good grasps for particular affordances that are difficult for an automated planner to recognize. Figure 5 demonstrates such a grasp, which is realizable by the BarrettHand only

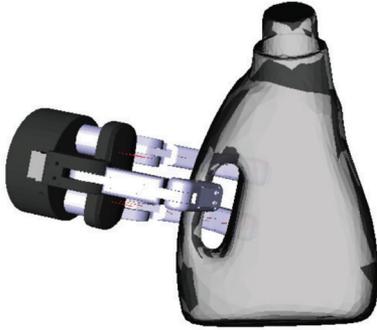


Fig. 5. This handle grasp for the detergent bottle is not a force closure grasp, but when chosen by the subjects in our experiments it succeeded 100% of the time. Adding a grasp database allows such semantically relevant grasps to be used in our system.

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. In experiments, this grasp was successful 100% of the time. Capturing this behavior in a simulator would require the modeling of dynamic object deformations. Currently, accurate simulations of such properties are too slow for sampling-based planners, so human annotation of such grasps is necessary.

To generate the grasp database, we ran the Eigengrasp planner offline six times for 20 min 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, the highest quality grasps were selected from among all of the available grasps until a full set of ten was available.

3.5. Grasping pipeline

Figure 6 includes a more detailed description of the pipeline, incorporating 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: (1) Control the vision system’s object detection and localization, (2) review the set of available grasps retrieved from the preplanned database, and (3) choose whether 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. In this pipeline, there are two possible paths to the final phase of grasp execution, trading complexity for potential speed improvements.

To help the user manage this complexity, we added a “guide window” that displays the transition table for the current phase, as shown in Figure 3. Throughout the pipeline, two of the facial gestures are associated with transitions between phases of the pipeline, while the other two provide direction to the online planner. In the “guide 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. Showing the current phase in the guide window acts as an important cue to help the user manage the transitions between phases of the pipeline, confirming that the chosen actions having their intended effect, and helping users remember the goal of each phase of the task. This is especially important when the user intended to make a different decision than the one that was detected, either by making the wrong gesture or because the classifier was mistaken.

Below the “guide window,” we visualize the top ten grasps available to the planner. This allows the user to judge options as they are retrieved from the database or repopulated by the online grasp planner.

3.6. Experiments

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 all three systems can be found in the multimedia extension or at <https://youtu.be/vuiW02i3y44>.

3.6.1. Task. Each subject was asked to grasp and lift three objects using an Emotiv Eloc as input. The experimental setup is illustrated in Figure 7. The subject is seated in front of a computer monitor about 2 m from the workspace of the robot, and is able to observe the robot and the target object. The depth camera is aimed at the center of the workspace from 1.5 m away. 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, the alignment provided by the vision system may be off center, and the subject has to decide which direction is the best aligned. Because the direction was not specified, the subjects were simply asked to grasp the object five times. The object was placed in view of the subject on a marker on the table that indicated a region in which the robot arm has high manipulability. The exact position of the object was not tightly controlled.

3.6.2. Training. The subject was asked to perform each facial gesture ten times each to train the Emotiv Eloc classifiers and choose reasonable parameters for the classifiers. 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.

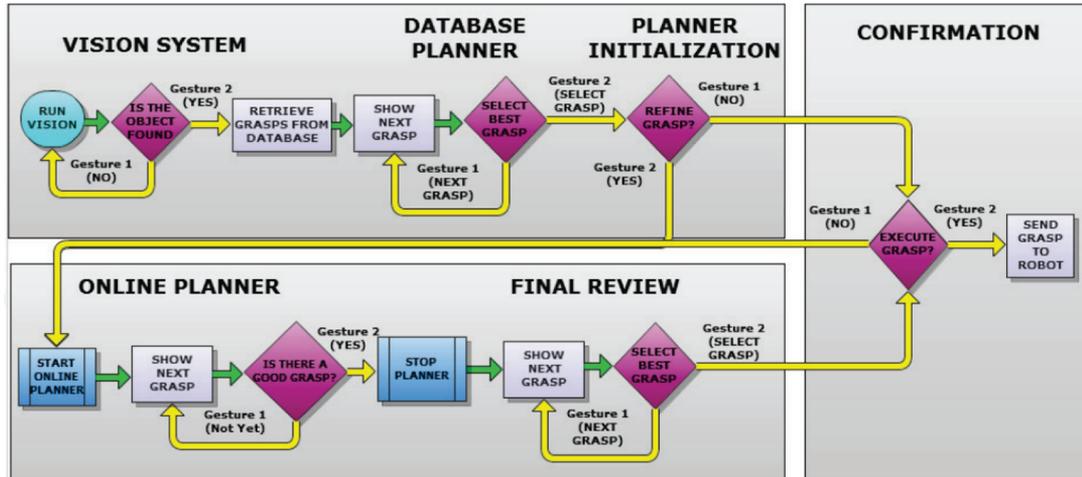


Fig. 6. Phases of the modified grasping pipeline, incorporating more control over the vision system and the grasp database. If the user chooses the n th grasp from the database, $n + 4$ user inputs are required. If none of the grasps is suitable, the online planner can be invoked with a few simple inputs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

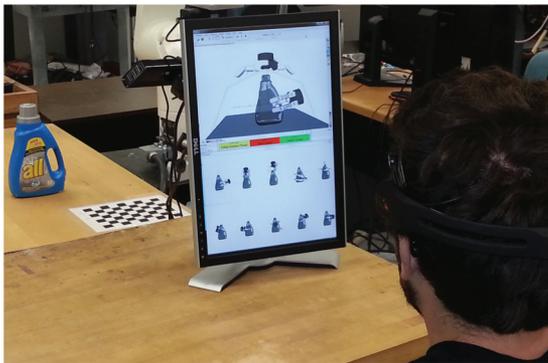


Fig. 7. A user grasping a laundry detergent bottle using System 1.

3.6.3. Results. The results of the experiments are reported in Table 1. For each subject, we report the mean time to completion and the 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, which is the time taken to plan the grasp. Overall, the average planning time was 104 s on the known objects and 86 s on the unknown object. The mean success rate was 80%, with most failures on the first attempt at attempting a new grasp. This shows that users are able to learn to use the system effectively in only a few attempts.

At the end of the experiment, subjects were asked whether they had experienced pain or discomfort during the experiment. All five subjects reported facial muscle fatigue. The subjects were asked if there was any part of the experience they particularly disliked. Three of the subjects mentioned that the setup time, which had taken over an hour for all three tasks, was too long. Three of the subjects also mentioned that they had trouble reading the guide window while concentrating on the screen.

Table 1. Results using System 1: Subjects using this system successfully grasped the target object in 80% of trials, with an average grasping time of 104 s.

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

In these experiments, grasps from the side demonstrated significantly more robustness and shorter 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, and which resulted in longer planning times while the subjects searched for a better grasp. In general, grasping roughly cylindrical objects, such as the top of the detergent bottle from above is somewhat problematic for the BarrettHand, owing 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 from among the grasps pulled directly from the database. The difference in planning times reflects the benefit of integrating the offline planning phase.

3.7. Discussion

These experiments revealed a number of problems with the device used in this experiment. 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 device to recognize their intended actions, especially with false negatives making it difficult to continue to the next phase of the pipeline at will. This led to overemphasis of the facial gestures, which caused muscle fatigue. The subjects reported significant muscle fatigue and some level of frustration with the lengthy setup. Although these issues point at major problems with the practicality of the input device, the subjects were able to complete the task well enough to validate the general strategy and user interface.

These difficulties are especially a large problem in testing this system on a disabled subject. Since they are dependent on carers, an indeterminately long setup time poses a major problem in performing studies with that population.

In addition to these issues, we also wanted a system that would handle more complex scenes with multiple objects and clutter in the target area. The clutter in the scene blocks grasps from many directions. To find grasps that are not blocked by clutter or outside the reachable region of the arm, we needed to perform online reachability assessments and provide feedback to users to allow them to understand when their chosen approach direction is blocked or unreachable. Although we did not measure cognitive load explicitly during this experiment, the subjects' comments relating difficulty in both focusing on the task and reading the guide window indicate that demands on the subjects' visual attention were high. To address this, the visual interface must be streamlined and simplified.

The minimal time this system can make a plan and execute a grasp is around 45 s, owing to the time taken by the vision system and the slow speed at which we run the grasping trajectory. The performance of subjects was within $\approx 300\%$ of this minimum. Optimizations to the vision pipeline and increasing the speed of arm motion as subjects become more comfortable with the robot will significantly decrease the time it takes to grasp an object.

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 that subjects expressed during this experiment.

4. System 2: Novel sEMG device with impaired user study

4.1. Introduction

In this section, we will introduce a significantly different paradigm for interacting with our online planning system, which is controlled through a cursor-based selection scheme rather than facial gestures. We will also describe revisions to the user interface to reduce visual clutter and provide more useful online feedback about the reachability of the demonstrated goals. We will then describe a short pilot study with an impaired user.

4.2. Surface EMG recording

To address some of the issues experienced using the Epop, we adopted a novel input device under development at the University of California 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.

The muscles behind the ear are innervated by nerves that come directly from the brain stem, without ever entering the spine. Even individuals with the most severe spinal cord paralysis can still access these muscles. Friedman et al. (1999) noted that this makes the auricular muscles a promising target for control of assistive devices. Although some individuals are able to move their ears independently, we have found that even individuals who 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 produces signals that can be detected by electrodes mounted on the surface of the skin.

A series of works (Joshi et al., 2011; Perez-Maldonado et al., 2010; Skavhaug et al., 2012, 2016) 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 of the muscles near the recording site in order to control the power voluntarily 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 wheelchair.

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 Butterworth filters to extract two separate signals. The bands are then linearly combined to compute the x and y cursor positions. This

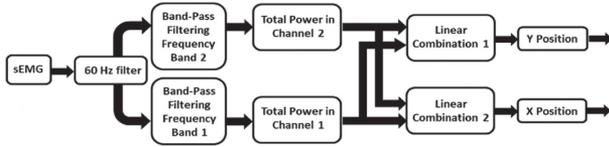


Fig. 8. The single sEMG signal is first processed through a 60 Hz noise filter. It is then run through two different band-pass Butterworth filters to extract two separate signals. The bands are then linearly combined to compute the cursor positions.

linear combination is necessary to generate independent control channels, since there are no perfect band-pass filters, and the subject might not be able to completely isolate 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 during maximum voluntary contraction. The filter outputs were combined linearly, as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1.75\text{gain}_1 & -0.75\text{gain}_2 \\ -0.75\text{gain}_1 & 1.75\text{gain}_2 \end{bmatrix} \begin{bmatrix} \text{Channel}_1 \\ \text{Channel}_2 \end{bmatrix} \quad (1)$$

Without this transformation, the cursor could not reach points along the x or y axis, as there can never be zero power in either of the frequency bands. The gains for each band are set for each subject after a short calibration procedure, as described by Perez-Maldonado et al. (2010), to establish the subject’s comfort level in maintaining a large enough voluntary muscle contraction to move the cursor to any part of the screen.

The sEMG signals are collected from the posterior auricular muscle with two surface Ag–AgCl cup electrodes connected to a model Y03 preamplifier (www.motion-labs.com) with input impedance higher than $10^8 \Omega$, 15–2000 Hz signal bandwidth, and a gain of 300. The electrodes were placed behind the subject’s left ear along the axis of the muscle, with approximately 1.5 cm inter-electrode distance (see Figure 9). A third electrode was placed on the elbow as a reference. The cup electrodes were type EL254S from Biopac Systems Inc. and were held in place with Ten20 conductive paste.

To adapt this system to our use, we added some additional smoothing steps similar to those of Vernon and Joshi (2011). The cursor position is further filtered through a low-pass filter with a cutoff frequency of 0.5 Hz. This produces a new position at 4 Hz. To smooth the visualization of the cursor motion, we linearly interpolate seven 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 0.25 s delay between the calculated position and

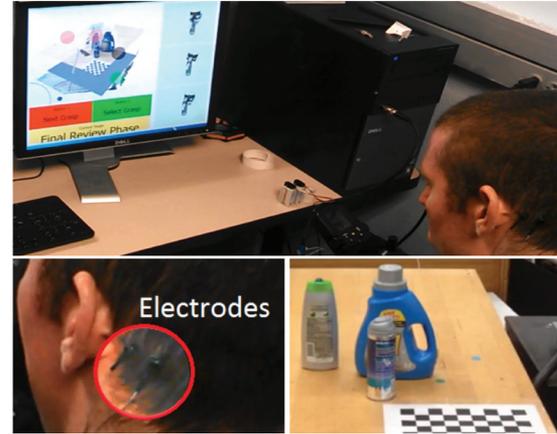


Fig. 9. An impaired subject in the UC Davis RASCAL lab (top) operating our sEMG assistive grasping interface to grasp a shaving gel bottle in the Columbia Robotics Group Laboratory (bottom right). The two small black clips behind the subject’s ear (bottom left) are surface EMG electrodes (used in differential mode) to detect activation of the posterior auricular muscle to direct the system to pick up the object in this multi-object scene.

the visualization. This delay is not noticeable when the subject is making smooth, controlled motions and has been used successfully in previous experiments.

4.3. sEMG GUI

To send signals to the grasping system, the user controls a cursor to hit one of four targets, as illustrated in Figure 10. Each target represents a different input option. During grasp planning, the targets are overlaid on the augmented reality display. The user begins in a rest area and moves the cursor to one of the targets. 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 4 s. If an unintended target is selected, the user can force the selection to *timeout* by avoiding rest for these 4 s, canceling the selection.

We map these inputs following a strategy similar to that used for each facial gesture in System 1. For the red and green targets, denoted Inputs 1 and 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 hand’s approach direction.

4.4. Handling cluttered scenes

In addition to an improved input device, we extended 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

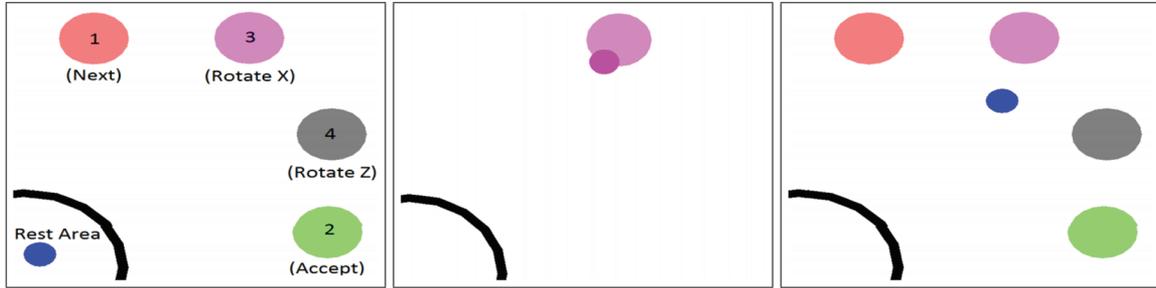


Fig. 10. The sEMG interface. (a) The user interface is composed of four targets overlaid on the grasping scene. Target 2 usually signals acceptance of the current option. Target 1 toggles the next option. Targets 3 and 4 provide input to the planner. (b) Hitting one particular target changes the color 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.

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 that are too close together for the fingers to surround the object without colliding with other objects. We tested grasp planning scenes where there was at least 5 cm of empty space between each object.

Handling cluttered scenes introduces 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, creating 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 might not have a reachable path to grasp. In System 1, 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.

To address the first two issues, we implemented an online reachability test that a user can clearly interpret. When good grasp candidates are found, System 2 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(b)). 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. Valid grasps are ranked by their distance from the demonstration hand and alignment with its approach direction. This makes the planner more responsive in cluttered scenes. The list is re-sorted as the demonstration hand is moved.

The results of the reachability test are also used to train a nearest-neighbors classifier. When the user moves the demonstration hand, we find the five grasps for which the normal of the palm of the hand is closest to the normal of

the demonstrated pose. If at least 50% of these grasps are unreachable, we designate the current demonstration pose as being in an unreachable region, which is indicated to the user by highlighting the demonstration hand in the planner interface in red. These measures are crucial for a naive user who is not familiar with the kinematics of the robot arm and might not have the intuition that the region from which grasping is attempted is not within the robot’s workspace.

4.5. GUI modifications for sEMG interface

A number of changes were 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 redesigned the UI with a cleaner look and feel that implements a number of new features.

The System 2 interface layout is outlined in Figure 11(a), 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 he or she might not be physically present to observe it first hand. Second, we display only three grasp options instead of ten 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 might 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 that is more suitable for non-expert users. The sEMG interface is rendered as a translucent layer on top of the grasp planning scene, allowing the subject to see

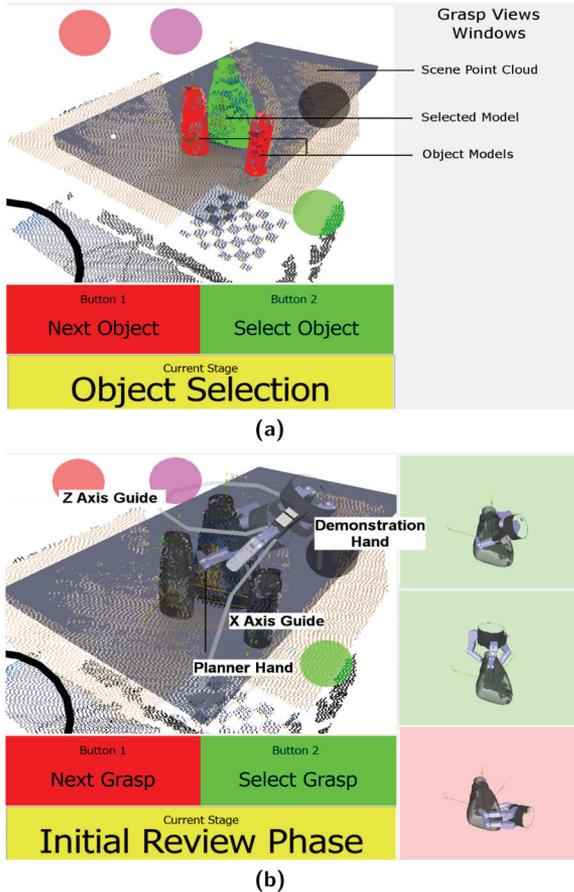


Fig. 11. The System 2 interface. (a) System 2 interface—object selection phase. The subject is able to see the planning scene in the main UI window. The window on the bottom tells the user the current phase and what the green and red inputs will do in this phase. In this phase, the subject sees the point cloud and hits the red target until the desired object is highlighted in green and then hits the green target to proceed to the next phase. (b) System 2 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 given a red background. The topmost grasp in the grasp view window is the currently selected grasp, which is rendered in the planning scene with the planner hand.

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.

4.6. System 2 pipeline

We made a number of modifications to several of the stages of the pipeline shown in Figure 6.

4.6.1. Initialization. The subject is presented with a view of the scene from the perspective of the depth camera. The user sends Input 1 to activate the object recognition system. If the recognized objects align well with the point cloud sent, the results can be accepted with Input 1. If not, the recognition system can be rerun with Input 2.

4.6.2. 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.

4.6.3. Initial review. The user is presented with a list of preplanned grasps from a precomputed database. This phase has been modified from System 1 to present a clearer visualization and reachability information. As the user iterates through the grasp list, the grasps in the middle and bottom rows shift up and the next grasp in the list moves to the bottom position. When the user moves the demonstration hand, the grasp list is re-sorted, bringing grasps from 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 grasp, Input 2 is sent to select the grasp.

4.6.4. Planner initialization. The user is presented with the choice of accepting the grasp from the previous phase with Input 1, proceeding straight to the grasp choice confirmation phase, or sending Input 2 to refine the chosen grasp further.

4.6.5. Grasp refinement. The online planner runs, presenting the user with updated options as new grasps are found. The 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. Moving the demonstration hand causes the planner to generate grasps from the demonstrated approach direction. Sending Input 2 stops the planner and proceeds to the final grasp review phase.

4.6.6. Final grasp review. The list of available grasps is static and the user is able to review all of the available grasps before making a final selection from the list. The user interface is the same as that presented in the *initial review phase* (Figure 11(b)), except for the current stage indicator.

The user is presented with three grasps at a time, which can be iterated through to select one that represents the user’s intent. The user sends Input 1 to switch to the next grasp on the grasp list or Input 2 to select the current grasp.

4.6.7. 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.

4.7. Validation

To validate this system, we recruited a 30-year-old man with limited upper limb mobility owing to a C3–C4 spinal injury. All testing of System 2 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 in using this interface. For this work, we measured the activity in the subject’s posterior auricular muscle to avoid the need to shave his 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 lab. The setup is shown in Figure 9.

As with the previous task, the objects were placed with their centers roughly aligned to markers on the table in the middle of the workspace of the arm, in a region with high manipulability. The objects were separated by at least 5 cm, which allows the BarrettHand sufficient space to reach between them without having to singulate.

4.7.1. Task. Owing 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 preplanned grasps. In the first attempt, he grasped the laundry detergent bottle. In the second attempt, he grasped the shaving gel bottle. In the third attempt, he was asked to grasp the detergent bottle using one of the preplanned 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 objects.

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 from 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.

Table 2. System 2: Experimental validation results.

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

4.7.2. 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 twice more 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 his way through it without having also to focus on the task of hitting targets with the sEMG interface. Once he appeared to be conversant with the system, we turned control over to the subject’s sEMG interface.

4.7.3. Results. The results of the experiment are shown in Table 2. The subject was able to grasp the objects successfully on all three attempts. On average, it took the subject 694 s to grasp each object, including about 60 s for the vision system to detect the objects in the scene. There were an average of 25 timeouts, and one mistakenly selected target per attempt. Timeouts are an expected part of this interface, which allows the user to re-select the 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.

At the end of the experiment, the subject was asked whether he had experienced any pain or discomfort, but did not report any. He was also asked whether there was any part of the experience he particularly disliked. He reported trouble controlling the cursor accurately enough, and also stated that he had gotten confused several times about what phase of the pipeline he was currently in whenever there was a phase transition he did not intend.

4.7.4. Discussion. This study was a short, single-user proof of concept with only a few results. Although these results are quite promising from the perspective that a single participant was able to understand and utilize the system fairly quickly, the experiment also revealed some shortcomings of the 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 posterior auricular muscle, which is smaller and has more variable performance. Time constraints 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. Additionally, the subject's difficulty in keeping track of his current phase in the pipeline indicates that the visual attention requirements of the user interface are still too high to allow the subject to attend to both the task and the phase indicator window at the same time. A second problem was that the online reachability tester is fairly slow in 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 that were not reflective of his 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, this tactic was not sufficient near border regions where grasps were possible but unlikely because of occlusions. In light of the severity of these issues, we proceeded directly to another iteration of the system rather than following up with further user studies.

5. System 3: A practical assistive grasping platform

These results showed enough efficacy of this system that we developed an enhanced system, System 3, which uses a smaller, lighter robotic arm, the Kinova Mico. Systems 1 and 2 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 that is affordable and practical for a portable assistive device that could be mounted on either a table or a wheelchair. 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.

5.1. 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 that 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 the same angle relative to the palm through most of the range of the finger's motion, like the motion of a parallel 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 a $10\times$ 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.

5.2. 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 different options for online reachability checking. This motivated us to replace the CBiRRT trajectory planner with the MoveIt! planning environment (see Sucas and Chitta, 2013), which interfaces with a large number of planners in the OMPL planning library (Sucas et al., 2012).

The OMPL planners have different strategies with different performance properties. To investigate which is appropriate to grasping in the cluttered scenes with the Mico arm, we captured 10 scenes similar to that shown in Figure 12(b) 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 the 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. To take advantage of this correlation, we implemented a plan caching scheme that 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 such parameters as minimum segment length and allowed planning time. We used 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 s. This parameter controls the sampling density of the planner along the planned trajectory, and so trades between planning time and accuracy of collision detection along the trajectory.

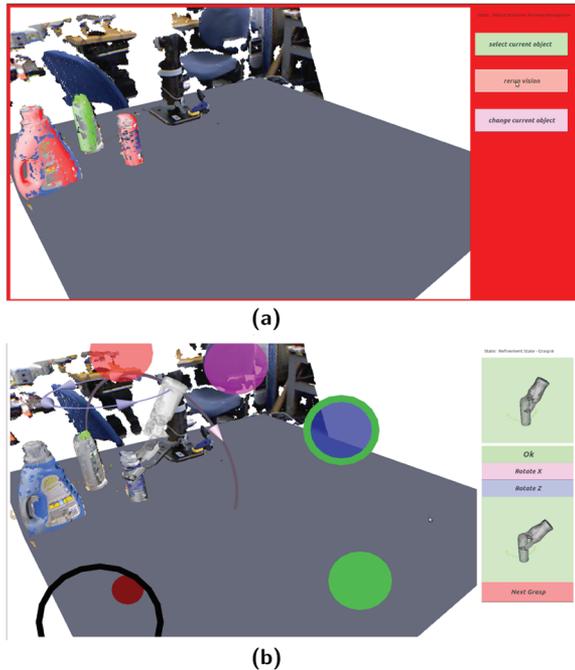


Fig. 12. Screenshots of *System 3* pipeline stages. (a) *System 3*—object recognition and selection state. The graspable objects in the scene are highlighted in red and green. Sending Input 1 selects the green object as the target, Input 2 cycles to the next object, and Input 3 triggers the object recognition system to refresh. The background UI area is rendered in red while the recognition is still processing. (b) *System 3*—grasp refinement state. The buttons on the right function as guides for the result of hitting the color-coded input options that will be presented to the user, as well as buttons that the user and experimenter use during the training stage. The sEMG is interface overlaid on the planning scene with the selected target highlighted in green.

Two of the probabilistic road map planners (Kavraki et al., 1996) performed best using the caching scheme, succeeding in 43% of the grasps, and the standard probabilistic road map implementation had the fastest planning for the caching version of the planner, with an average planning time of 5.5 s for arm motions in which the caching fails, and 0.1 s when the cache succeeds. Note that we only used single-query versions of the probabilistic road map planners, so this improved performance does not leverage the advantage that probabilistic road map planners have in multi-query planning scenarios.

The single-query bidirectional variant of the probabilistic road map planner (SBL) produced plans that seemed smoother in the region near the object. Grasping objects in a cluttered scene requires planning in a state space in which there is a very narrow valid region near the goal state, 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. This means that the planned trajectory changes direction fewer times near the

goal, which leads the user to more easily understand and supervise the motion of the robot (Dragan et al., 2013).

However, we found empirically, when the caching scheme failed to find a reasonable neighbor, that the SBL planner’s success rate dropped to 30%, whereas the probabilistic road map planner’s success rate remained the same. This led to a slight lag in performance as the cache was populated. Thus, for the online reachability verification, we used the probabilistic road map planner with a segment length of 0.05, while to produce 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. (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 ad-hoc optimization script.)

5.3. Further UI improvements

In our previous systems, the two “continuous” inputs that shifted the hand around the object were not part of the pipeline guide display that showed users which phase they were in and what their inputs would do. In every phase of the pipeline, the inputs 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 that fewer grasp options should be presented during the parts of the pipeline where they were not needed.

In this system, the UI window is adaptive, providing more visual cues to the user as to what the 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 Figures 12(a) and (b). While this may seem like unnecessary complexity, the UI is more visibly different in each phase and less extraneous information is presented. This acts as a large visual cue that does not require the subject to explicitly switch attention to check the current phase.

5.4. System 3 pipeline

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

5.4.1. Object recognition and selection. This phase combines the first two phases of *System 2*. 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 objects 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.

5.4.2. Initial review. As in System 2, the user is presented with a list of preplanned grasps from a precomputed database. The UI presented is shown in Figure 12(b), 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.

5.4.3. Grasp refinement. This phase is similar to System 2, but with one fewer grasp displayed more prominently. The first grasp is shown in the top of the window and the next grasp is shown in 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.

5.4.4. Final grasp review. The list of available grasps is static and the user is able to review all of the available grasps before making a final selection from the list. As in the previous phase, this phase has been adapted to have only two grasps, with 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.

5.4.5. Grasp choice confirmation. This phase is similar to that of System 2, 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.

5.5. Validation

To validate these design decisions, we tested our pipeline on five healthy subjects, two men and three women, aged 22–30. All testing was approved by the Institutional Review Board of 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 was constrained to move toward the target representing the “selected” input, which is outlined in green, as shown in Figure 12(b). To change 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 to the user interface and grasp planning pipeline without needing the more extensive training necessary to train a subject to achieve full 2D control of the cursor.

5.5.1. sEMG device setup. In these experiments, we placed the sEMG device behind the ear of the subject to measure contractions of the posterior auricular muscle. 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™ silicone putty, as shown in Figure 13. We find the correct placement of the device by asking the subject to clench his or her jaw gently and raise his or her eyebrows. We place the electrodes where we find a large response to eyebrow raises and little response to jaw motion.

5.5.2. 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* 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 two 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 subjects with the grasp planning system, we manually showed them three examples of grasping objects, once short circuiting the online planning and twice allowing the online refinement to run. Then we allowed the subjects to guide the pipeline themselves five times, twice without the online planner and three times with it. Then we repeated the training, allowing the subjects 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.

5.5.3. Task. We placed the objects on the table in proximity to one another, as shown in Figure 12(b), near the center of the workspace of the Mico arm. We asked the subject to grasp each object three times, the first time from any direction deemed reasonable, then 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

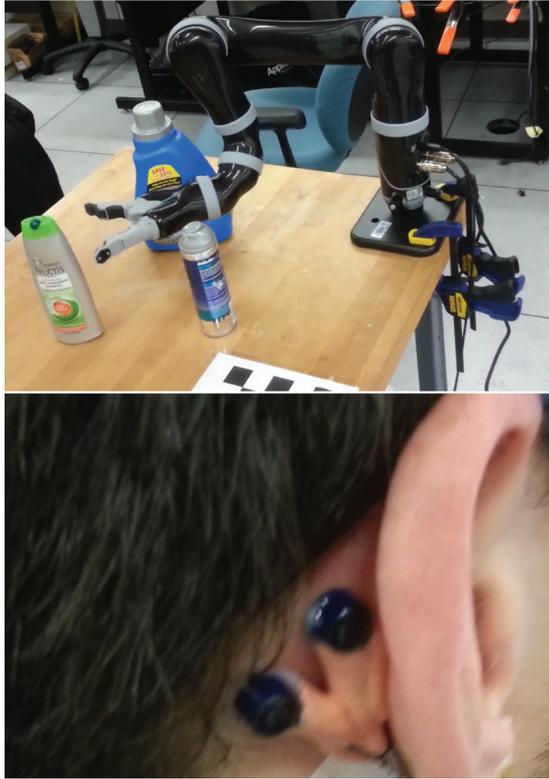


Fig. 13. Top: A typical grasp of the shampoo bottle from the side in the cluttered scene. Note that the hand is just able to fit between the other objects to grasp the desired target. Note that the ability to plan this grasp in such a restricted environment is an indication that this system is very successful at handling the cluttered scene. Bottom: The sEMG system electrodes. In these experiments, we placed the electrodes behind the ear of the subject to measure contractions of the posterior auricular muscle. We stabilize the electrodes by wrapping the head of the electrodes in silicone putty.

or top grasp required online grasp refinement. Since the workspace of the Mico arm is not very large, it is easy to find such object positions.

5.5.4. Results. The results of the experiment for five subjects are tabulated in Table 3. On average, the subjects were successful in grasping 82% of the objects within 92 s of the first time the cursor left the rest area. Subjects 2 and 3 were the best able to control the cursor, having previously been able to move their ears already, and also performed the best in these experiments. These results indicate that the underlying planning system is providing options that the less capable subjects are not exploring because they are having more difficulty with the UI.

At the end of the experiment, subjects were asked whether they had experienced pain or discomfort during the experiment. None of the subjects reported any discomfort or muscle fatigue. The subjects were asked whether there was any part of the system they particularly disliked. One subject reported a small amount of frustration with the

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

Grasp	Subject	Success	Time
Detergent bottle, top	1	Yes	75
	2	Yes	53
	3	No	45
	4	No	122
	5	Yes	135
	Mean	60%	86
Detergent bottle, side	1	No	66
	2	Yes	40
	3	Yes	52
	4	Yes	80
	5	Yes	85
	Mean	80%	64
Detergent bottle, open choice	1	Yes	50
	2	Yes	57
	3	Yes	53
	4	Yes	135
	5	Yes	128
	Mean	100%	85
Shampoo bottle, top	1	Yes	151
	2	Yes	72
	3	Yes	60
	4	No	126
	5	No	104
	Mean	60%	102
Shampoo bottle, side	1	Yes	134
	2	Yes	95
	3	Yes	132
	4	Yes	164
	5	Yes	143
	Mean	100%	133
Shampoo bottle, open choice	1	Yes	93
	2	Yes	121
	3	Yes	63
	4	Yes	95
	5	Yes	117
	Mean	100%	98
Shaving gel, top	1	No	83
	2	No	123
	3	Yes	112
	4	No	139
	5	Yes	97
	Mean	60%	111
Shaving gel, side	1	Yes	65
	2	Yes	52
	3	Yes	57
	4	Yes	88
	5	Yes	92
	Mean	100%	71
Shaving gel, open choice	1	No	73
	2	Yes	59
	3	Yes	76
	4	Yes	81
	5	Yes	85
	Mean	80%	75

(continued)

Table 3. Continued

Grasp	Subject	Success	Time
Average performance	1	66%	87
	2	88%	75
	3	88%	72
	4	77%	114
	5	88%	109
	Mean	82%	92

speed of the online planner when grasping the detergent bottle from the side in the center of the cluttered scene. Two of the subjects felt that the experiments were repetitive.

For the shampoo bottle, relatively fewer grasps can succeed, compared with 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 13. 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 13. Without the partial plan caching implemented in the online trajectory planner, the online reachability checker cannot reliably find a path to this region. With the cache enabled, plans are eventually found by connecting lower-quality grasps produced by the planner that are less occluded by clutter and therefore easier to reach. This behavior allows us to keep the allowed planning time for each individual grasp candidate low to increase the throughput of the filter. The random nature of the grasp planner combined with the solution cache allows us to compensate for the uncertainty of the feasibility checker. This allows the user the discretion to wait for a solution for particularly hard goal regions.

6. Conclusions

In this paper, we have 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 allowing a non-expert user to apply intuition to the grasping problem and provide the added value that makes the system work well in spite of sensor noise and 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.

The iterative development strategy followed in this paper has allowed us to uncover problems with each revision quickly to reach a viable platform. As the platform and its

capabilities evolved, we made the tasks more challenging. The drawback to this approach is that we cannot use it to make strong statements comparing each iteration of the system. To fully validate the improvements to the systems, we will need to conduct further experiments with larger numbers of subjects using each system to perform the same task.

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, online grasp planning, and the online reachability filter to address the most challenging issues that caused our initial user to take up to 8 min 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 2D control.

This study serves as a pilot to validate the design choices of the system on a path toward more experiments with impaired users. We did not measure explicitly 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. 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 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 of 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 wheelchair or mobile manipulator assistive platform. We have also begun integrating more complex artificial intelligence techniques, such as deep learning with online grasp planners (Varley et al., 2015), which may be able to provide better grasps to the user in spite of sensor and localization noise.

Acknowledgements

We wish to thank Dr. Joel Stein and Dr. Ethan Rand for their feedback on the UI. We also thank Ida-Maria Skavhaug and Kenneth Lyons for helpful discussions and suggestions.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Science Foundation (grant numbers IIS-1208153 and 1208186).

References

- Artemiadis PK and Kyriakopoulos KJ (2011) A switching regime model for the EMG-based control of a robot arm. *IEEE Transactions on Systems, Man, and Cybernetics* 41(1): 53–63.
- Bell CJ, Shenoy P, Chalodhorn R, et al. (2008) Control of a humanoid robot by a noninvasive brain–computer interface in humans. *Journal of Neural Engineering* 5(2): 214–220.
- Berenson D, Srinivasa S, Ferguson D, et al. (2009a) Manipulation planning on constraint manifolds. In: *IEEE international conference on robotics and automation (ICRA '09)*, Kobe, Japan, 12–17 May 2009. Piscataway, NJ: IEEE.
- Berenson D, Srinivasa SS and Kuffner JJ (2009b) Addressing pose uncertainty in manipulation planning using task space regions. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, St. Louis, MO, 10–15 October 2009. Piscataway, NJ: IEEE.
- Bryan M, Thomas V, Nicoll G, et al. (2011) What you think is what you get: Brain-controlled interfacing for the PR2. Technical report. In: *Iros 2011: The PR2 workshop*, San Francisco, CA, 25–30 September 2011. Piscataway, NJ: IEEE.
- Castellini C and van der Smagt P (2009) Surface EMG in advanced hand prosthetics. *Biological Cybernetics* 100(1): 35–47.
- Ciocarlie MT and Allen PK (2009) Hand posture subspaces for dexterous robotic grasping. *International Journal of Robotics Research* 28(7): 851–867.
- Cipriani C, Zaccone F, Micera S, et al. (2008) On the shared control of an EMG-controlled prosthetic hand: Analysis of user prosthesis interaction. *IEEE Transactions on Robotics* 24(1): 170–184.
- Cram JR and Criswell E (2011) *Cram's Introduction to Surface Electromyography*. Sudbury: Jones & Bartlett Learning.
- Dragan AD, Lee KC and Srinivasa SS (2013) Legibility and predictability of robot motion. In: *Proceedings of the 8th ACM/IEEE international conference on human–robot interaction, HRI '13*, Tokyo, Japan, 3–6 March 2013, pp. 301–308. Piscataway, NJ, USA: IEEE Press.
- Friedman RN, McMillan RG, Kincaid CJ, et al. (1999) Preliminary electrophysiological characterization of functionally vestigial muscles of the head: Potential for command signaling. *Journal of Spinal Cord Medicine* 22(3): 167–172.
- Gomez-Gil J, San-Jose-Gonzalez I, Nicolas-Alonso LF, et al. (2011) Steering a tractor by means of an EMG-based human–machine interface. *Sensors* 11(7): 7110–7126.
- Ho NSK, Tong KY, Hu XL, et al. (2011) An EMG-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation. In: *International conference on rehabilitation robotics*, Zurich, Switzerland, 29 June–1 July 2011. Piscataway, NJ: IEEE.
- Horki P, Solis-Escalante T, Neuper C, et al. (2011) Combined motor imagery and SSVEP based BCI control of a 2 DoF artificial upper limb. *Medical & Biological Engineering & Computing* 49(5): 567–577.
- Javdani S, Srinivasa S and Bagnell A (2015) Shared autonomy via hindsight optimization. In: *Proceedings of robotics: Science and systems* (ed. LE Kavraki, D Hsu, and J Buchli), Rome, Italy, 13–17 July 2015.
- Joshi SS, Wexler AS, Perez-Maldonado C, et al. (2011) Brain–muscle–computer interface using a single surface electromyographic signal: Initial results. In: *5th international IEEE/EMBS conference on neural engineering (NER)*, Cancun, Mexico, 27 April–1 May 2011, pp. 342–347. Piscataway, NJ: IEEE.
- Kavraki L, Svestka P, Latombe JC, et al. (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12(4): 566–580.
- Matrone G, Cipriani C, Carrozza MC, et al. (2011) Two-channel real-time EMG control of a dexterous hand prosthesis. In: *5th international IEEE/EMBS conference on neural engineering (NER)*, Cancun, Mexico, 27 April–1 May 2011, pp. 554–557. Piscataway, NJ: IEEE.
- Miller AT and Allen PK (2004) Graspit!: A versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine* 11: 110–122.
- Müller-Putz GR, Scherer R, Pfurtscheller G, et al. (2005) EEG-based neuroprosthesis control: A step towards clinical practice. *Neuroscience Letters* 382(1–2): 169–174.
- Mülling K, Venkatraman A, Valois J, et al. (2015) Autonomy infused teleoperation with application to BCI manipulation. *Computing Research Repository* abs/1503.05451. <http://arxiv.org/abs/1503.05451>.
- Papazov C and Burschka D (2011) An efficient RANSAC for 3D object recognition in noisy and occluded scenes. In: *Computer vision ACCV 2010* (eds. R Kimmel, R Klette, and A Sugimoto), Queenstown, New Zealand, 8–12 November 2010, vol. 6492, pp.135–148. Berlin: Springer.
- Perez-Maldonado C, Wexler AS and Joshi SS (2010) Two dimensional cursor-to-target control from single muscle site sEMG signals. *Transactions of Neural Systems and Rehabilitation Engineering* 18(2): 203–209.
- Postelnicu CC, Talaba D and Toma Mi (2011) Controlling a robotic arm by brainwaves and eye. In: *Technological Innovation for Sustainability, DoCEIS 2011* (ed LM Camarinha-Matos), Costa de Caparica, Portugal, 21–23 February 2011, vol. 349, pp.157–164. Berlin: Springer.

- Ranky GN and Adamovich S (2010) Analysis of a commercial EEG device for the control of a robot arm. In: *Proceedings of the 2010 IEEE 36th annual northeast bioengineering conference*, New York, NY, 26–28 March 2010. Piscataway, NJ: IEEE.
- Royer AS, Rose ML and He B (2011) Goal selection versus process control while learning to use a brain-computer interface. *Journal of Neural Engineering* 8(3): 036012.
- Sagawa K and Kimura O (2005) Control of robot manipulator using EMG generated from face. *Proceedings of SPIE* 6042(2): 604233.
- Scherer R, Friedrich ECV, Allison B, et al. (2011) Non-invasive brain-computer interfaces: enhanced gaming and robotic control. In: *Advances in Computational Intelligence* (eds J Cabestany, I Rojas, and G Joya), Torremolinos, Spain, 8–10 June 2011, vol. 6691, pp.362–369. Berlin: Springer.
- Schmidl H (1965) The I.N.A.I.L. myoelectric B/E prosthesis. *Orthotics and Prosthetics* 19(4): 298–303.
- Shenoy P, Miller KJ, Crawford B, et al. (2008) Online electromyographic control of a robotic prosthesis. *IEEE Transactions on Biomedical Engineering* 55(3): 1128–35.
- Sherman DE and Lippay A (1965) A Russian bioelectric-controlled prosthesis. *Canadian Medical Association Journal* 92(5): 243.
- Skavhaug IM, Bobell R, Vernon B, et al. (2012) Pilot study for a brain-muscle-computer interface using the extensor pollicis longus with preselected frequency bands. In: *2012 34th annual international conference of the IEEE engineering in medicine and biology society*, San Diego, CA, 28 August–1 September 2012, pp.1727–1731. Piscataway, NJ: IEEE.
- Skavhaug IM, Lyons KR, Nemchuk A, et al. (2016) Learning to modulate the partial powers of a single sEMG power spectrum through a novel human-computer interface. *Human Movement Science* 47: 60–69.
- Sucan IA and Chitta S (2013) MoveIt!. <http://moveit.ros.org> (accessed 13 April 2017)
- Sucan IA, Moll M, and Kavraki LE (2012) The open motion planning library. *IEEE Robotics & Automation Magazine* 19(4): 72–82.
- Varley J, Weisz J, Weiss J, et al. (2015) Generating multi-fingered robotic grasps via deep learning. In: *2015 IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, Germany, 28 September–2 October 2015, pp.4415–4420. Piscataway, NJ: IEEE.
- Vernon S and Joshi SS (2011) Brain-muscle-computer interface: Mobile-phone prototype development and testing. *IEEE Transactions on Information Technology* 15(4): 531–538.
- Vogel J, Haddadin S, Simeral JD, et al. (2010) Continuous control of the DLR light-weight Robot III by a human with tetraplegia using the BrainGate2 Neural Interface System. In: *12th international symposium on experimental robotics* (eds O Khatib, V Kumar, and G Sukhatme), Delhi, India, 18–21 December 2010, Berlin: Springer-Verlag, vol. 79, pp.125–136.
- Waytowich N, Henderson A, Krusienski D, et al. (2010) Robot application of a brain computer interface to Stäubli TX40 robots—early stages. In: *World automation congress*, Kobe, Japan, 19–23 September 2010. Piscataway, NJ: IEEE.
- Weisz J and Allen PK (2012) Pose error robust grasping from contact wrench space metrics. In: *2012 IEEE international conference on robotics and automation*, St. Paul, MN, 14–18 May 2012, pp.557–562. Piscataway, NJ: IEEE.
- Weisz J, Barszap AG, Joshi SS, et al. (2014) Single muscle site sEMG interface for assistive grasping. In: *2014 IEEE/RSJ international conference on intelligent robots and systems*, Chicago, IL, 14–18 September 2014, pp.2172–2178. Piscataway, NJ: IEEE.
- Weisz J, Elvezio C and Allen PK (2013) A user interface for assistive grasping. *2013 IEEE/RSJ international conference on intelligent robots and systems*, Tokyo, Japan, 3–7 November 2013. Piscataway, NJ: IEEE.
- Weisz J, Shababo B and Allen PK (2012) Grasping with your face. In: *13th international symposium on experimental robotics* (JP Desai, G Dudek, O Khatib, et al. eds.), Quebec, Canada, 18–21 June 2012, vol. 88, pp.435–448. Cham: Springer.
- Wolczowski A and Kurzyński M (2010) Human-machine interface in bioprosthesis control using EMG signal classification. *Expert Systems* 27(1): 53–70.
- Yang D, Zhao J, Gu Y, et al. (2009) EMG pattern recognition and grasping force estimation: Improvement to the myocontrol of multi-DoF prosthetic hands. In: *International conference on intelligent robots and systems*, St. Louis, MO, 10–15 October 2009, pp.516–521. Piscataway, NJ: IEEE.

Appendix: Index to multimedia extension

Archives of IJRR multimedia extensions published prior to 2014 can be found at <http://www.ijrr.org>; all videos published after 2014 are available on the IJRR YouTube channel at <http://www.youtube.com/user/ijrrmultimedia>.

Table of multimedia extension

Extension	Media type	Description
1	Video	Demonstration of all three systems