2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

Single Muscle Site sEMG Interface for Assistive Grasping

Jonathan Weisz, Alexander G. Barszap, Sanjay S. Joshi, and Peter K. Allen

Abstract— **We present a joint demonstration between the Robotics, Autonomous Systems, and Controls Laboratory (RASCAL) at UC Davis and the Columbia University Robotics Group, wherein a human-in-the-loop robotic grasping platform in the Columbia lab (New York, NY) is controlled to select and grasp an object by a C3-C4 spinal cord injury (SCI) subject in the UC Davis lab (Davis, CA) using a new single-signal, multi-degree-of-freedom surface electromyography (sEMG) human-robot interface. The grasping system breaks the grasping task into a multi-stage pipeline that can be navigated with only a few inputs. It integrates pre-planned grasps with on-line grasp planning capability and an object recognition and target selection system capable of handling multi-object scenes with moderate occlusion. Previous work performed in the RASCAL lab demonstrated that by continuously modulating the power in two individual bands in the frequency spectrum of a single sEMG signal, users were able to control a cursor in 2D for cursor to target tasks. Using this paradigm, four targets were presented in order for the subject to command the multi-stage grasping pipeline. We demonstrate that using this system, operators are able to grasp objects in a remote location using a robotic grasping platform.**

I. Introduction

Grasping objects is critical to many activities of daily life that present challenges to people with upper limb mobility impairments. This work presents an assistive grasping system using a human-in-the-loop for a disabled user to grasp objects from a table using a novel, non-invasive sEMG based input device even in somewhat cluttered scenes. The novel device measures only a single differential sEMG signal at one muscle site on the user. The system puts the user in control of a multi-stage grasping pipeline that includes object recognition, integrated pre-planned and on-line grasp planning with feedback to help the user plan robust grasps in near real time. We show that an impaired user can successfully use this system to grasp objects using a real robotic grasping platform. The key contributions of this work include: *a*) Integration with a novel sEMG input device which relies on only a single muscle site. *b*) A new UI that improves the disabled user's ability to understand the scene and produce correct grasps in complex, cluttered environments. *c*) Online reachability analysis and feedback. *d*) Online assessment of the desired approach direction. *e*) Evaluation of this system on an impaired user in a remote location. *f*) A demonstration that this new UI is descriptive enough for the user to operate in an environment that they have never seen.

J. Weisz and P. Allen are with the Department of Computer Science, Columbia University, NY 10027, USA. E-mail: {jweisz, allen}@cs.columbia.edu A. Barszap and S. Joshi are with the Department of Mechanical and Aerospace Engineering Deparment, University of California, Davis, CA , USA. E-mail: {abarszap, maejoshi}@ucdavis.edu . This work has been funded by NSF Grants IIS-1208153.

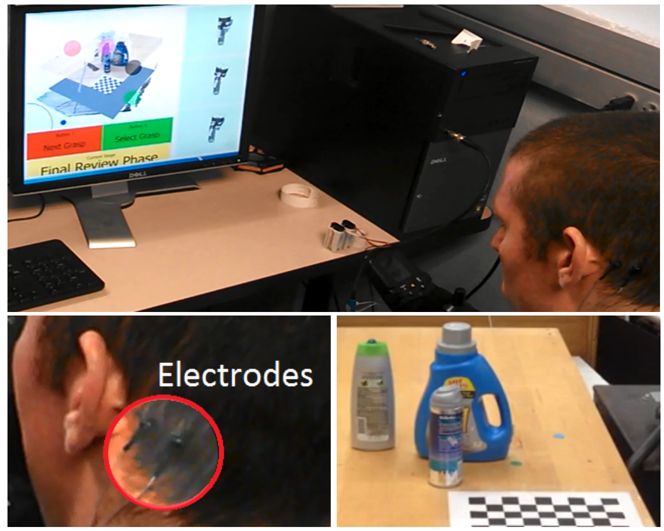

Fig. 1: An impaired subject in the UC Davis RASCAL lab (top) operating our sEMG-Assistive Grasping interface to grasp a shaving gel bottle in the Columbia Robotics Group Laboratory (bottom right). The two small black clips behind the subject's ear (bottom left) are surface EMG electrodes (used in differential mode) to detect activation of the Auricularis Posterior (AP) muscle to direct the system to pick up the object in this multi-object scene.

Fig. 1 gives an overview of the system described in this paper. In previous work [1], we presented a human-in-the-loop grasp planning system using an Emotiv Epoc EEG headset, a low throughput human input device that reads facial gestures and EEG signals. We used the Epoc to provide input that demonstrated user intent to a grasp planner that would plan grasps in near real time. We showed with a cohort of five subjects that within a few half hour long sessions, subjects were able to learn to produce stable grasps for a variety of objects.

While the Emotiv is relatively low cost and convenient for devices of this type, it is bulky and prone to signal problems from the electrodes losing reliable contact with the subject's scalp. It is time consuming to find an initial position which maintains solid contacts, and it is frequently necessary to pause the experiment to re-seat the head set and retrain the system briefly. Furthermore, it uses several electrodes and a head mounting frame that could encumber the users.

Our novel input device uses only a single electrode-pair that can be placed securely in a noninvasive location behind the ear. The signal is divided into multiple power bands which are then processed into two simultaneous control channels [2], [3]. The user controls a cursor which selects among four options that control the grasp planning pipeline.

To accommodate the high visual processing demands of this paradigm, we have developed a more streamlined UI with only a few grasp options at a time. Earlier work [1] gave the user ten options. The user had to determine whether a grasp was reachable and reliable. With fewer options, it is important that we present the user with more valuable grasps first. Additionally, the subjects in [1] were able to see the experiment area to assess the reachability of different grasps using their own intuition. However, an impaired user may not be able to look around the room on their own. In this paper, we show that a user in a remote location can use the system using only the information conveyed by the interface.

To accomplish this, we have developed a more sophisticated approach to filtering grasps to make sure that the options presented are reachable and to let the user know whether the current approach direction is tenable. Using this system, an operator in the UC Davis Rascal lab (Davis, CA) was able to pick up objects in a multi-object scene remotely in the Columbia University Robotics lab (New York, NY).

II. Related Work

There is a considerable body of work on assistive robotics and controlling robots using inputs derived from electrophysiological signals. In [4], we presented a more complete review of the available literature.

Recently, [5] showed that an immobilized subject could use the BrainGate cortically implanted electrode to control a robotic manipulator. However, implanted devices enable access to many high quality channels of control, but require highly invasive surgeries that may not be suitable for many users. Alternatively, others have explored noninvasive interfaces that are restricted to a lower bandwidth. The two most commonly used signal sources are electromyography (EMG) signals from muscles activation, and electroencephalography (EEG) signals generated by brain activity. One common source of EMG signals is the forearm muscles. Some examples of this include [6], in which the authors used forearm signals to control a manipulator to pick up and place objects on a table, and several works ([7], [8], [9], [10], [11]), in which forearm muscle activation is used to preshape a hand for grasping tasks.

Many more patients retain control over the muscles in their heads and faces than forearm muscles, including those which control eye gaze direction. This makes these signals a more appropriate target for authors investigating assistive control [12], [13], [14], [15]. These works have generally focused on low level cartesian control of the end effector appropriate for simple tasks. The novelty of the interface presented in this paper is that it is generally even less invasive than those in previous works and yet allows higher level capabilities. This system records sEMG signals from a single muscle site, requiring only a single recording electrode pair.

Previous work has shown that users prefer higher level interfaces that assume some of the burden of planning how to accomplish the task [16]. In [17], the authors demonstrated using such a system for retrieving arbitrary objects from flat surfaces using a laser pointer. These systems require less input from the user, and are thus more appropriate for lower bandwidth input signals such as EEG. [18], [19] and [20] presented work using EEG signals in which the subject selects grasp targets and placement locations, and leaves the grasping and trajectory planning completely up to the automated part of the system. In [21], the authors describe using such a system to guide the whole pipeline of a complex task which allowed the user to accomplish tasks with many stages. Menu driven pipelines for accomplishing such complex tasks using low bandwidth interfaces were discussed in more detail by [22], [23].

More recent work [24], [25] has introduced a more flexible interpretation of assistive control of grasp planning, termed human-in-the-loop planning, in which the planner is guided by a user and then presents the user with options for how to complete the rest of the task. This work builds on this interpretation to extend it to the lower bandwidth signals available from many assistive interfaces such the sEMG device used in this paper.

III. The sEMG Device

In [3], [2], [26], we showed that subjects can learn to navigate a cursor on a screen by making fine-tuned contractions of the Auricularis Superior (AS) - the ear wiggling muscle located above the ear and the Extensor Pollicis Longus (EPL) - a muscle of the wrist used to stretch the thumb. In this study, we used the Auricularis Posterior (AP), which is a muscle similar to the AS but behind the ear rather than above it. The skin behind the ear has less hair, so it is easier to securely attach the electrode and get a strong sEMG signal there.

As even the most severely paralyzed individuals typically have access to muscles that are innervated at the brainstem (such as the AS and the AP), our device has the potential to support this user group in their daily lives by allowing them to manipulate the external environment through very low isomorphic contractions of a spared muscle.

We collected sEMG signals from the AP muscle with two surface Ag-AgCl cup electrodes connected to a model Y03 preamplifier (www.motion-labs.com) with input impedance higher than $10^8\Omega$, 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 Fig. 1). A third electrode was placed on the elbow as a reference. The cup electrodes were

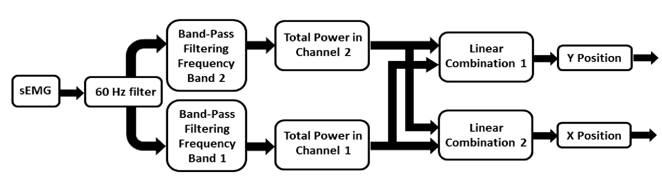

Fig. 2: The single sEMG signal is first processed through a 60 Hz noise filter. It is then run through two different band pass Buttersworth filters to extract two separate signals. The bands are then linearly combined to compute the x and y cursor positions according to Equation 1.

of the type EL254S from Biopac Systems Inc. held in place with Ten20 conductive paste.

The processing to extract control signals from the sEMG is outlined in Fig. 2. The signal is filtered using a series of notch filters as in [27] to filter out 60 Hz noise and its harmonics. The total powers of two different frequency bands of the single sEMG signal were computed using two band pass filters for 80-100 Hz (Band 1) and 130-150 Hz (Band 2). These bands were selected ad-hoc, based on previous experience. The output of the two filters produced comparable powers. The filter outputs were combined linearly as described in Equation 1. Without this transformation the cursor could not reach points along the x or y axis as there can never be zero power in either of the frequency bands. The gains for each band are set for each subject after a short calibration procedure, as described in [3] to establish the subject's comfort level maintaining a large enough voluntary muscle contraction to move the cursor to any part of the screen.

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

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

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

There are currently many different systems that provide assistive technologies to severely paralyzed persons. These operate through the use of head or muscle function still available, such as the tongue[28], eye gaze[29], facial expressions[1], voice operation[30], or breath control[31]. Although these systems are useful for many, they rely on functions that are needed for other vital functions or that may be impaired due to the person's condition. For many of these systems the user is prevented from being able to naturally interact with others. For example, tongue-operated systems prevent speech. Facial expression controlled systems prevent the user from making natural facial expressions such as laughing. In the case of severe paralysis many of these muscles are not available. Tongue and breath controlled systems can not be used when the user is on a respirator. The use of sEMG sensors for prosthetics and assistive systems control provides a noninvasive way for paralyzed persons to regain some control of their environment. Traditionally, multiple muscle sites are used to provide multiple degree-of-freedom control[32], [33], [34]. However, in many cases, such as when a paralyzed person must lay in a specific position, multiple muscle sites may not be available. Eye tracking devices may be difficult to set up for these constrained environments. For these reasons, a single muscle site in a location that does not compromise vital activities was chosen for our system. A system like this is useable even by severly impaired individuals like the one in [2], who used a previous version of this system as a TV remote in his own home.

IV. The sEMG Interface

To send signals to the grasping system, the user controls a cursor to hit one of four targets presented, which are overlayed on the grasp planning scene, as seen in Fig. 3. The user begins in a rest area and moves the cursor to one of the targets, each representing a different input option. When the target is hit, the cursor changes colors to reflect the user's selection. The user returns the cursor to the rest area, at which point the input option selected is activated. After a selection, the other targets are disabled for four seconds. If an unintended target is selected, the user can avoid returning to rest for these four seconds, canceling the selection.

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

In general, input 1 serves as a 'no' and in some stages is used to indicate that the current grasp is not suitable and proceed to the next grasp. Input 2 indicates a 'yes' and is used to allow the user to proceed to the next stage. Inputs 3 and 4 control the planner by moving a rendered image of the robot hand that indicates the user's desired approach direction around the x and z axes of the target object, respectively. We call this hand the *demonstration hand*. This is explained in more detail in section V. The rationale behind this control scheme is based on prior experience from [2], where we compared a 2 dimensional target selection paradigm with a simpler 1 dimensional paradigm in which the user was asked to hold a contraction as the system cycled through a set of options. While conceptually simpler, there is a tradeoff between how long the user has to signal their acceptance of a given option and how long the selection takes. With a 4 option list, the user's average time to reach the fourth target was around 8 seconds. In comparison, after practice using a 3 target 2 dimensional paradigm similar to ours, the user was able to select any target with an average time below 5 seconds. However, accuracy suffered significantly, dropping from 100% to around 75% for the worst case target. In this work, we have designed the pipeline that the user controls to be robust to occasional errors in target selection and implement the faster, more responsive paradigm, rather than the slower but more reliable one.

V. The Grasping Pipeline

In our scenario, the user is presented with a live pointcloud of a table with several objects to be grasped. The user is asked to select an object and attempt to lift it off the table using the sEMG interface. Central to this method is the use of our grasping simulator, GraspIt! [35]. GraspIt! allows us to simulate robot configuration using data from a vision system that acquires pointclouds in real-time. This simulation environment allows us to evaluate grasps using various quality measures, which we use to synthesize good grasps. Then we can present the user with a visualization of potential grasps aligned to the live vision data. Combined

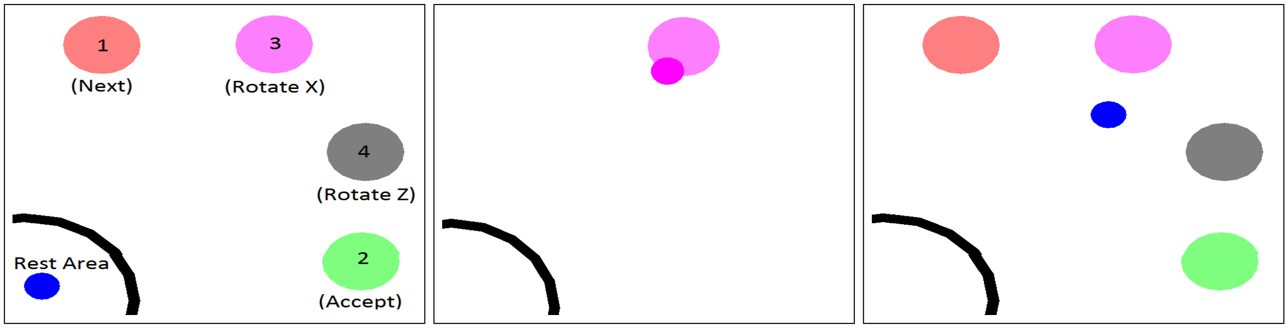

Fig. 3: *The sEMG Interface*: (a) Hitting one particular target changes the color of the cursor to reflect the selection and makes the other targets unavailable. (b) If the user does not return to the rest area after a few seconds, the selection times out and is deselected and all targets become available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

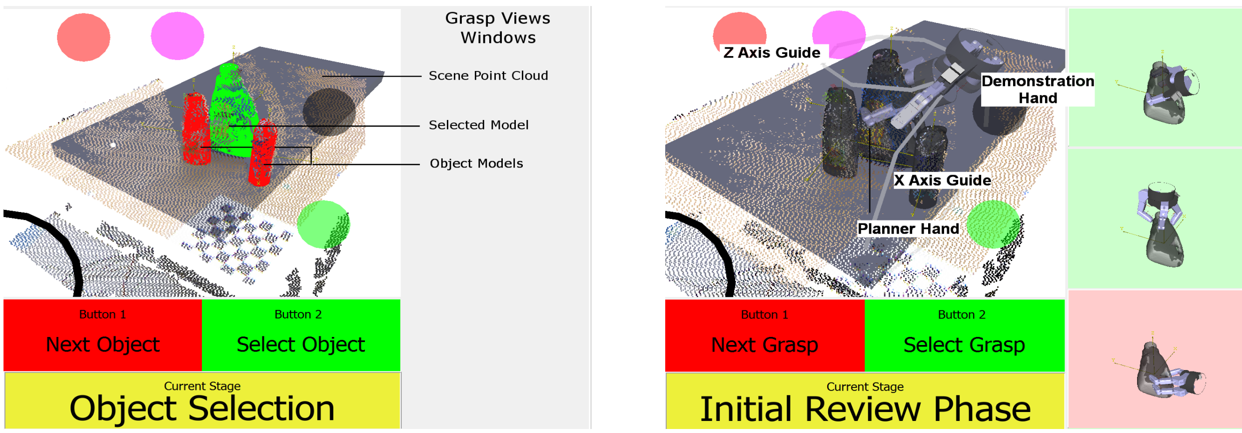

Fig. 4: *The Grasp Planning Interface:* (a) Object Selection: The subject is able to see the planning scene in the main UI window. The window on the bottom tells the user the current phase and what the green and red inputs will do in this phase. In this phase, the subject sees the the pointcloud and hits the red target until the object they wish to grasp is highlighted in green. Then they hit the green target to proceed to the next phase. (b) Initial Review Phase: After the subject selects the object, the Grasp View pane on the right is populated with a set of grasps from a database. Grasps that are reachable appear on a green background, while unreachable grasps are red. A robot hand appears that the user moves to demonstrate a desired starting pose. This *demonstration hand* is constrained to follow the two circular guides around the z and x axes of the object shown above. The top most grasp in the grasp view window is the currently selected grasp, which is rendered in the planning scene with the *planner hand*.

with a real time, on-line grasp planner, this allows the user to interface with the real scene interactively by guiding the quasi-autonomous system through a grasping pipeline.

The grasping pipeline is divided into a number of stages: Object recognition and initialization, target selection, initial grasp review, planner initialization, on-line grasp refinement, final grasp review, and grasp choice confirmation. This pipeline is controlled via the four inputs described above. Fig. 4 shows the window presented to the user during this process. In each phase, the current phase can be seen in the user input prompt window on the bottom. The result of hitting the red and green targets is described in the corresponding fields of the window.

A. Pipeline

1) Initialization: The grasp planning scene is initially populated by an object recognition system based on [36]. First, a pointcloud generated by a Microsoft Kinect is added to the scene. Then, a set of objects from a 3D model database are recognized and aligned to the scene using a variant of RANSAC with oriented pairs of points as features. We assume a friction coefficient of 1.0. The inertial parameters of the model are approximated by assuming a uniform density of the surface of the model as in [35]. The object models are added to the planning scene in the simulator to complete the shape detected by the pointcloud for collision detection and grasp quality analysis. The user can see the objects in the simulator and select the target object for the grasp planner. The other objects are treated as obstacles to be avoided. The user sends input 1 to activate the object recognition system. If the recognized objects align well with the point cloud sent, they can accept the results with input 1. If not, they can rerun the recognition system with input 2.

2) Object Selection: The first object is highlighted as the target object. To select an object as a target, the user sends input 2. To switch to the next object in the recognized object list the user sends input 1.

3) Initial Review: The user is presented with a list of pre-planned grasps from a precomputed database. The database is built by an offline planner, described in subsection V-B, that can compute the best stable grasps for an object based on a grasp quality measure. By precomputing a set of stable grasps offline from many directions, we can often present the user with a good grasp choice immediately. The user sends input 1 to increment through the grasp list. When the user finds a reasonable looking grasp, they send input 2 to select the grasp.

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

5) Grasp Refinement: In order to refine their chosen grasp, the user is able to position the *demonstration hand* which shows the planner the approximate approach direction the user wants to grasp the object from. The grasp is refined by starting the online Eigengrasp planner described in subsection V-B using the selected grasp's approach direction. The planner then generates a set of stable grasps, and the user selects their desired grasp from among them. The user is able to rotate the *demonstration hand* around the x axis of the object using input 3 and around the z axis using input 4. The planner re-ranks the grasps to prioritize those closer to the demonstrated approach direction. New grasps are generated using the demonstrated hand direction as a seed and added to the available grasp list. If the planner is not able to generate reachable grasps from the demonstrated pose, the hand is tinted red to indicate to the user that there is a problem. Sending input 2 stops the planner and proceeds to the Final Grasp Review stage.

6) Final Grasp Review: The user sends input 1 to select the next grasp on the grasp list. They send input 2 to select that grasp.

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

B. Grasp Planner

Grasps are planned using our previously developed Eigengrasp Grasp Planner[37]. This planner is used both to populate our grasp database[38] and to refine grasps on-line at the user's direction. The planner uses simulated annealing on a lower dimensional representation of the joint angles of the hand. The objective function used attempts to align a set of predetermined contact points on the hand to the object. In this case, a desired contact location is placed on the center of each fingertip and inner finger link. There are four additional contacts on the palm, one in the center of each quadrant. For each desired contact point, the planner projects along the direction normal to the hand to find any contacts with the target object. If the distance to the palm is greater than .05 m or the angle between the normals is more than $45°$, the projected contact is discarded. Otherwise, the magnitude of the maximum wrenches produced by this contact point are scaled proportionately to their distance and angle to the contact object, and the *epsilon quality*[39] of the wrench space produced by these projected contact points is calculated. Grasps that have a better quality than the current best solution are added to the list of potential grasp candidates.

For these grasp candidates, a more computational expensive second quality measure is calculated in which the hand moves along a predetermined approach direction, closes the fingers, and then calculates the *epsilon quality* of the actual simulated contacts, rather than the projection of them.

In this work, we introduce a final step to the grasp ranking to test for reachability among the obstacles in the current workspace. We use the CBiRRT motion planner[40] to attempt to find an arm trajectory to reach the grasping pose. Previous iterations of this planner used a simpler, built in inverse kinematics solver to reject unreachable states. In contrast, this version checks that an entire valid trajectory can be generated. Unreachable grasps are placed at the end of the list of grasps and colored red in the *grasp preview window* (see Fig. 4b). We maintain the list of unreachable grasps so that we can reject nearby grasps without running more computationally expensive analyses. The valid grasps are ranked by their distance to the demonstration hand and alignment to its approach direction. This makes the planner more responsive in cluttered scenes. The list is re-sorted as the demonstration hand is moved.

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

VI. Experiment

A. Setup

1) Hardware: Our grasping platform consists of a Staubli TX60L robot arm with a BarrettHand. Pointclouds are gathered by the object recognition system using a Microsoft Kinect. Communication between the user interface at UC Davis and the grasping platform at Columbia University is through standard TCP/IP sockets using common message passing libraries. Delays in communication over the remote link are not considerable, and are not the focus of this paper.

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

During the task, the subject reported which target they were trying to reach and we tracked the number of mistaken target activations, which would lead the user to loop back

Fig. 5: Above is the subject confirming a grasp for the handle of the detergent bottle at the RASCAL lab. Below is the realized grasp of the successfully lifted detergent bottle at Columbia University.

through part of the pipeline. After the grasp is selected, the target object is lifted off of the table automatically so that the user can see whether the grasp is stable. If no part of the target object remains on the table, we consider the trial a success.

3) Training: In this preliminary experiment, we had one C3-C4 spinal cord injury patient with limited upper limb mobility who has had extensive prior experience with this sEMG device. All testing was approved by the Institutional Review Board of the University of California, Davis under Protocol 251192-10. To familiarize the subject with the interface, we demonstrated the pipeline two times with the subject just watching and asking questions along the way. We then went through the pipeline with the subject two more times while verbally instructing them on which target to hit while the experimenter controlled the cursor with a computer mouse. This allowed the subject to familiarize themself with the pipeline and navigate their way through it without having to also focus on the task of hitting targets with the sEMG interface. Once they appeared to be conversant with the system, we turned over control to the subject's sEMG interface.

B. Results

The results of the experiment are shown in Table I. The subject was able to grasp the objects successfully on every

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

TABLE I: Experiment Results

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

These results demonstrate a reasonable level of success for an impaired subject operating a novel system, although they indicate that future revisions of the system should try to reduce accidental target selections that lead to timeouts by increasing the target size or modifying the target layout. Fig. 5 shows one of the objects being grasped in the Columbia Robotics laboratory after being planned at the RASCAL lab in UC Davis. Although performance of this particular impaired subject was somewhat slow, experience with other impaired subjects leads us to believe that with additional practice and improvements in system calibration, performance may be significantly improved. A video explaining the system and demonstrating the user's performance can be found here: `http://robotics.cs.columbia.edu/jweisz/sEMGGraspingIROS2014`

VII. Discussion and Future Work

These results demonstrate that the grasping system we have described can be used by an impaired subject even without the subject having had previous experience with the robot in question, and without ever being in the same room with it. This shows that the subject doesn't need any input other than what the user interface presents to successfully grasp an object, demonstrating that our grasp analysis is producing reasonable results. In our prior work[1], the subjects were able to see the real robot and scene, and we found that the subject's own intuition played an important role in helping them filter the presented grasps to find successful grasps. Here, we have replaced much of that intuition with a richer user interface and smarter planning. This interface was successful in spite of the additional visual complexity of overlaying the sEMG user interface on the planning scene. The novel device is easier to place on the user and proved more reliable. Additionally, it is suitable for subjects with a higher level of impairment. We see this work as an initial step towards a full featured interface to a mobile robotic assistive system for highly impaired individuals.

Even at this early stage of development, user feedback has been important. Prior to the successful trials described above, the subject attempted the experiment with much less success. We learned that it was important to train the subject by first explaining to him the entire pipeline of events, so that he would understand the big picture, rather than guiding them step by step. Small software changes in the interface also made large improvements in usability. In early experiments, selecting a target did not deactivate the other targets. This setup proved to be very difficult for the subject to correctly choose the target they were attempting to hit. In many cases the correct target would be hit but an incorrect target would be hit on the cursor's way back to the rest area, resulting in the wrong command being triggered. Simply deactivating non-selected targets while the cursor returned to the rest area greatly improved performance.

Our design choices were motivated by previous experience with a different impaired subject using a similar device in [2]. The subject in this paper's performance was significantly slower than the subject reported in that previous work. In future work, we will need to explore why that was the case for this subject and whether it is reasonable to have an optional switch that allows the user to change to something similar to the 1-D paradigm described in section IV, or even if this shift should occur automatically based on how many options are relevant for the current stage of the pipeline or subject performance. Clearly, the tradeoffs in different design choices are important and will be a subject of future research. Our future work will continue to explore these issues and advances in user training and UI design. In order to make this interface more useful, we will need to extend it beyond the acquisition of target objects for grasping to more complex tasks. We will also explore more complex signal processing strategies to improve subject performance, extending this device to a three dimensional cursor, and improving the accuracy of target selection by recalibrating the gains of the system on-line.

REFERENCES

[1] J. Weisz, C. Elvezio, and P. K. Allen, "A user interface for assistive grasping," in *Proc. IROS*. IEEE, Nov. 2013.

[2] S. Vernon and S. S. Joshi, "Brain-muscle-computer interface: mobile-phone prototype development and testing." *TITB*, vol. 15, no. 4, pp. 531–538, 2011.

[3] C. Perez-Maldonado, A. Wexler, and S. Joshi, "Two dimensional cursor-to-target control from single muscle site semg signals," *Trans. of Neural Systems and Rehabilitation Engineering*, vol. 18, April 2010.

[4] J. Weisz, B. Shababo, and P. K. Allen, "Grasping with your face," in *Proc. ISER*. Springer, 2012.

[5] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, 2012.

[6] P. Shenoy, K. J. Miller, B. Crawford, and R. N. Rao, "Online electromyographic control of a robotic prosthesis." *TBME*, Mar. 2008.

[7] D. Yang, J. Zhao, Y. Gu, L. Jiang, and H. Liu, "EMG pattern recognition and grasping force estimation: Improvement to the myocontrol of multi-DOF prosthetic hands," in *IROS*. IEEE, Oct. 2009.

[8] A. Woczowski and M. Kurzyski, "Human-machine interface in bioprosthesis control using EMG signal classification," *Expert Systems*, vol. 27, no. 1, pp. 53–70, Feb. 2010.

[9] N. S. K. Ho, K. Y. Tong, X. L. Hu, K. L. Fung, X. J. Wei, W. Rong, and E. A. Susanto, "An EMG-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation," in *Int. Conf. on Rehabilitation Robotics*. IEEE, 2011.

[10] C. Cipriani, F. Zaccone, S. Micera, and M. Carrozza, "On the Shared Control of an EMG-Controlled Prosthetic Hand: Analysis of User Prosthesis Interaction," *IEEE Transactions on Robotics*, Feb. 2008.

[11] G. Matrone, C. Cipriani, M. C. Carrozza, and G. Magenes, "Two-channel real-time EMG control of a dexterous hand prosthesis," in *Proc. Int. Conf. on Neural Engineering*, Apr. 2011, pp. 554–557.

[12] K. Sagawa and O. Kimura, "Control of robot manipulator using EMG generated from face," in *ICMIT*, Dec. 2005.

[13] J. Gomez-Gil, I. San-Jose-Gonzalez, L. F. Nicolas-Alonso, and S. Alonso-Garcia, "Steering a Tractor by Means of an EMG-Based Human-Machine Interface," *Sensors*, vol. 11, no. 7, 2011.

[14] G. N. Ranky and S. Adamovich, "Analysis of a commercial EEG device for the control of a robot arm," in *Proc. NEBEC*, Mar. 2010.

[15] C.-c. Postelnicu, D. Talaba, and M.-i. Toma, "Controlling a Robotic Arm by Brainwaves and Eye," *IFID*, 2011.

[16] A. S. Royer, M. L. Rose, and B. He, "Goal selection versus process control while learning to use a brain-computer interface." *Journal of Neural Engineering*, vol. 8, no. 3, p. 036012, June 2011.

[17] A. Jain and C. C. Kemp, "El-e: an assistive mobile manipulator that autonomously fetches objects from flat surfaces," *Auton. Robots*, vol. 28, no. 1, pp. 45–64, Jan. 2010.

[18] C. J. Bell, P. Shenoy, R. Chalodhorn, and R. P. N. Rao, "Control of a humanoid robot by a noninvasive brain-computer interface in humans." *Journal of Neural Engineering*, vol. 5, no. 2, pp. 214–20, Jun 2008.

[19] N. Waytowich, A. Henderson, D. Krusienski, and D. Cox, "Robot application of a brain computer interface to staubli TX40 robots - early stages," *World Automation Congress (WAC)*, pp. 1–6.

[20] J. S. M. Bryan, V. Thomas, G. Nicoll, L. Chang and R. Rao, "What You Think is What You Get: Brain-Controlled Interfacing for the PR2," Iros 2011: The PR2 Workshop, San Francisco, Tech. Rep., 2011.

[21] S. M. Grigorescu, T. Lüth, C. Fragkopoulos, M. Cyriacks, and A. Gräser, "A bci-controlled robotic assistant for quadriplegic people in domestic and professional life," *Robotica*, vol. 30, May 2011.

[22] R. Scherer, E. C. V. Friedrich, B. Allison, M. Pröll, M. Chung, W. Cheung, R. P. N. Rao, C. Neuper, and M. Pr, "Non-invasive brain-computer interfaces: enhanced gaming and robotic control," in *Advances in Computational Intelligence*, June 2011, vol. 6691.

[23] P. Gergondet, A. Kheddar, C. Hintermuller, C. Guger, and M. Slater, "Multitask humanoid control with a brain-computer interface: user experiment with hrp-2," in *Proc. ISER*. Springer, 2012.

[24] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *HRI*, 2012.

[25] T. Chen, M. Ciocarlie, S. Cousins, P. M. Grice, K. Hawkins, K. Hsiao, C. Kemp, C.-H. King, D. Lazewatsky, A. E. Leeper, H. Nguyen, A. Paepcke, C. Pantofaru, W. Smart, and L. Takayama, "Robots for humanity: A case study in assistive mobile manipulation," *IEEE Robotics & Automation Magazine*, vol. 20, 2013.

[26] I. Skavhaug, R. Bobell, B. Vernon, and S. Joshi, "Pilot study for a brain-muscle-computer interface using the extensor pollicis longus with preselected frequency bands," in *EMB*. IEEE, August 2012.

[27] R. G. T. Mello, L. F. Oliveira, and J. Nadal, "Digital butterworth filter for subtracting noise from low magnitude surface," *Computer Methods and Programs in Biomedicine*, no. 1, pp. 28–35, Nov. 2007.

[28] X. Huo and M. Ghovanloo, "Evaluation of a wireless wearable tonguecomputer interface by individuals with high-level spinal cord injuries," *J. Neural Eng.*, vol. 7, pp. 1–12, 2010.

[29] R. Barea, L. Boquete, M. Mazo, and E. López, "System for assisted mobility using eye movements based on electrooculography," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 10, no. 4, Dec. 2002.

[30] J. A. Bilmes, J. Malkin, X. Li, S. Harada, K. Kilanski, K. Kirchhoff, R. Wright, A. Subramanya, J. Landay, P. Dowden, and H. Chizeck, "The vocal joystick," in *ICASSP*. IEEE, May 2006.

[31] N. Pellegrini, A. Pelletier, D. Orlikowski, C. Lolierou, M. Ruquet, J.-C. Raphal, and F. Lofaso, "Hand versus mouth for callbell activation by {DMD} and becker patients," *Neuromuscular Disorders*, 2007.

[32] J. Paciga, P. Richard, and R. Scott, "Error rate in five-state myoelectric control systems," *Med. and Biol. Eng. and Comput.*, 1980.

[33] D. S. Dorcas and R. N. Scott, "A three-state myoelectric control," *Med. Biol. Eng. Comput.*, vol. 4, pp. 367–370, 1966.

[34] M. Zeeca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Critical Rev. Biomed. Eng.*, vol. 30, pp. 459–485, 2002.

[35] A. T. Miller and P. K. Allen, "Graspit!: A versatile simulator for robotic grasping," *IEEE Robotics and Automation Magazine*, 2004.

[36] C. Papazov and D. Burschka, "An efficient ransac for 3d object recognition in noisy and occluded scenes," in *ACCV*, 2010.

[37] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *IJRR*, vol. 28, no. 7, pp. 851–867, 2009.

[38] C. Goldfeder, M. Ciocarlie, H. Dang, and P. Allen, "The columbia grasp database," in *Proc. ICRA*. IEEE, 2009, pp. 1710–1716.

[39] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. of the Int. Conf. on Robotics and Automation*, August 1992, pp. 2290–2295.

[40] D. Berenson, S. Srinivasa, and J. Kuffner, "Task Space Regions: A framework for pose-constrained manipulation planning," *IJRR*, Mar. 2011.