Human Robot Interface for Assistive Grasping

Jacob Varley[1], Srihari Sridhar[1], Jonathan Weisz[1], Ethan Rand[2], Kenneth Lyons[3]
Sanjay Joshi[3], Joel Stein[2], Peter Allen[1]

***Abstract*— This work describes a new and improved Human-in-the-Loop (HitL) grasping system for assistive robotics. The system is controlled via a surface electromyography (sEMG) device positioned behind the subject's ear. The device provides a low bandwidth unobtrusive interface enabling the control of a robotic manipulator. This work describes the current system setup as well as a number of improvements which have been made to the interface. The main contributions of this work are a new and improved sEMG assistive robotics platform as well as the results of a user study with the new system. The system has been designed with a focus on reducing the workload of the user, clarifying how the state of the system is presented, enabling grasping in cluttered scenes, and making the future consequences of the user's actions more apparent.**

I. INTRODUCTION

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

Grasping an object generally requires contextual knowledge of the object and the intent of the user, particularly in cluttered scenes. We have developed a user interface that allows the user to effectively express their intent. This interface is validated by testing users with a novel surface electromyography(sEMG) device that can be calibrated very easily and an enhanced GUI that improves user experience and the capability of the system. This system has the potential to bring HiTL assistive devices out of the research environment and into the lives of those that need them.

The main contribution of this work is a flexible, fully featured HitL system that allows users to effectively grasp objects in cluttered environments via shared autonomy. The system uses a novel, practical sEMG device controlled by any of a variety of muscle regions. Our system is based on previous work that explored a number of different sEMG sensing devices and user interfaces (UI) which allow HiTL grasping [26][25][24]. In testing this system on both healthy and disabled subjects, we have identified areas where improvements are needed. These improvements include a cleaner, more intuitive user interface, lightened cognitive load on the user, and more clearly identified control flow. The intent is to make this assistive system portable and easy to calibrate, set up, and use in home and clinical environments.

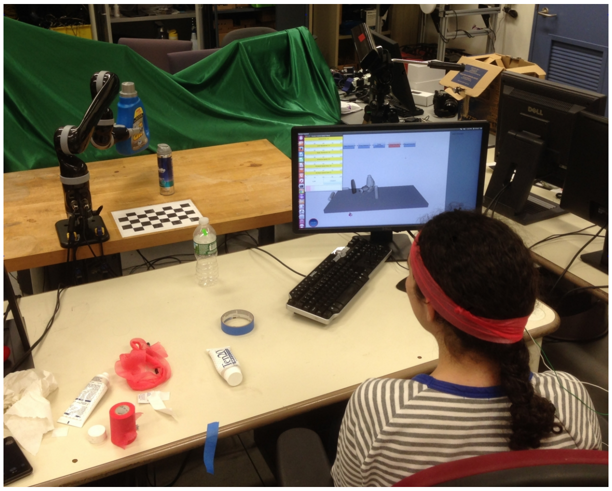


Fig. 1: Experimental setup shown after a successful grasp execution. The user interface, Mico robotic arm, Kinect depth sensor, sEMG device, and target objects are all visible.

II. RELATED WORK

There is a long history of assistive robotic systems using electrophysiological signals as input, with work going back as far as [19] and [21]. In the time since, there have been many approaches and refinements of proposed interfaces for disabled individuals with robotic assistive systems. 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, an external sensor of motion such as eye or hand trackers, or a specific electrophysiological signal such as electromyography (EMG), electrooculagraphy (EOG), and electroencephalogram (EEG). Within this category, modalities can be further divided by where the signals are recorded from. EMG can be recorded from distal muscle sites, which may be larger, easier to record from, and produce larger signals. However, more impaired individuals, such as those with spinal cord injuries, tend to maintain control over muscle functions above the level of injury.

This work was supported by NSF Grants 1208153 and 1208186 . The authors are from [1]Department of Computer Science, [2]Department of Rehabilitation Medicine, Columbia University and [3]Department of Mechanical and Aerospace Engineering, UC Davis. We thank Ida-Maria Skavhaug for helpful discussions.

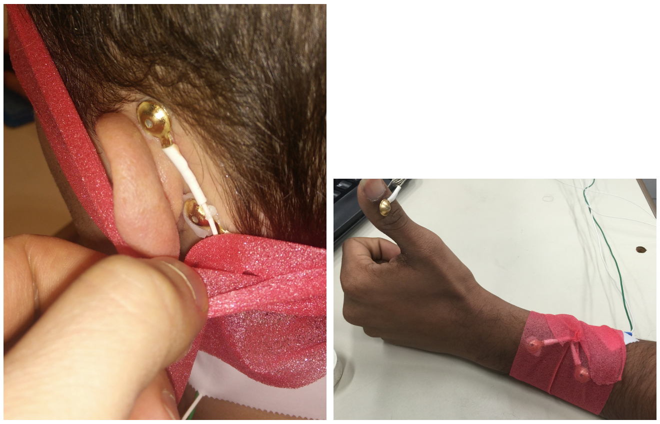


Fig. 2: For the user studies, the sEMG device was placed over the superior auricularis (SA) muscle, located behind the ear for 3 subjects, and over the extensor pollicis longus (EPL) muscles, located on the forearm for the other 3 subjects. The electrodes were attached with Ten20 conductive gel, and a small amount of prewrap (red) was used to stabilize them.

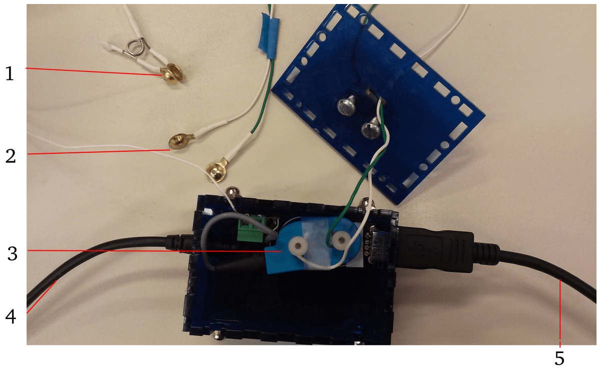
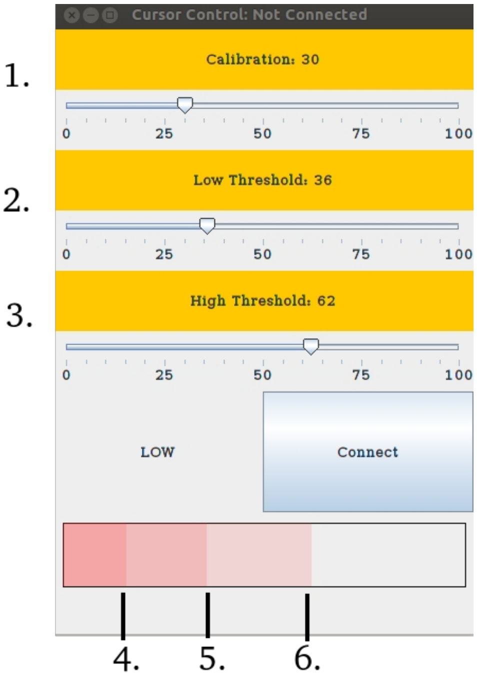


Fig. 3: The device consists of a ground wire (1), 2 electrodes (2), amplifier (3), audio cable to connected to a microphone jack (4), and a USB cable to provide power (5).

Another way to categorize the systems is by the type of control they engender - whether the control is through direct demonstration [1][4], joint level control [10], end effector cartesian control [22][15][17][8][16][20], discrete state level control [28][27][9][6][11], or task level control, allowing the user to designate what is to be done [3][2][23][18], or task oriented shared control, which tries to blend end effector control and task oriented control[13].

This work presents a system at an intermediate control level, in which the user has some state level control that is task oriented, and the input modality is a sEMG signal. This level of control has practical applications for users with motor impairments enabling them to select, grasp, and then execute a task (e.g. lifting) with a desired object.



Cursor Control: Not Connected

Calibration: 30

Low Threshold: 36

High Threshold: 62

Fig. 4: The sEMG interface contains a calibration slider (1) to set the gain applied to the sEMG signal. The low threshold slider (2) sets the threshold for the power bar that the user must exceed to change the currently selected button in the UI. The high threshold slider (3) sets the threshold for the power bar that the user must exceed in order to select a button in the UI. The power bar displays the current power output by the device (4) compared to the current low (5) and high (6) thresholds.

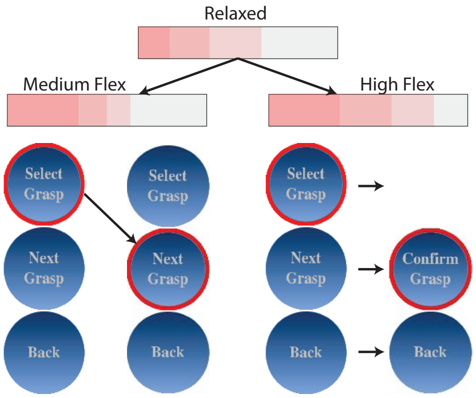
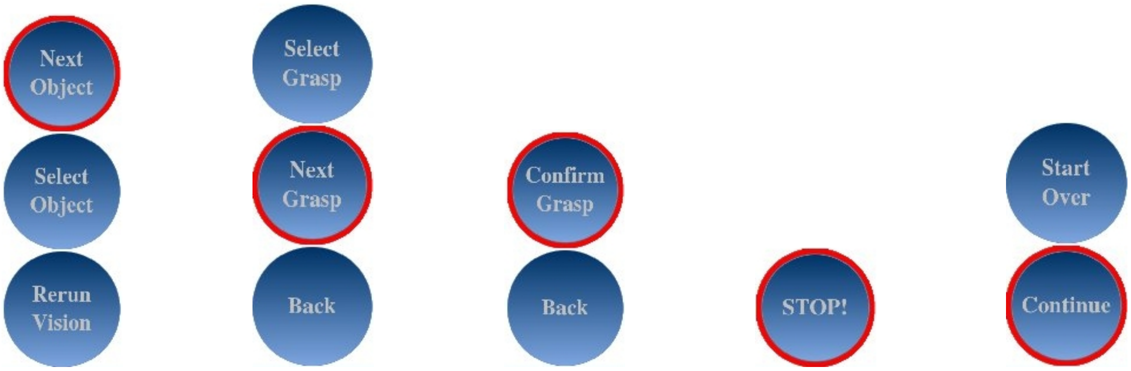


Fig. 5: While the subject is in a relaxed state, no changes occur with the UI. The subject is able to change the currently highlighted button with a medium flex, and able to select the currently highlighted button with a high or strong flex.

III. AN HRI GRASPING PLATFORM

Our previous work in developing assistive systems has outlined areas where improvements are needed, and below we describe each of these areas and the enhancements made to address them. Fig. 1 shows the experimental setup. We are using a Kinova lightweight MICO arm which has a two-fingered gripper, and a Microsoft Kinect RGB-D camera to provide point clouds of the scene. We use a pre-computed database of stable grasps so that we can pre-plan a set of

Fig. 6: Execution Pipeline: The pipeline is placed at the top of each stage to clarify what stage the user is currently in, and what stage they will be sent to if they navigate forwards or backwards. In the Paused Execution state, the user is also given the ability to restart the entire system.



(a) Object Selection (b) Grasp Selection (c) Grasp Confirmation (d) Grasp Execution (e) Paused/Finished Execution

Fig. 7: Stages to the Grasping Pipeline. The stages have different options allowing the user to navigate the interface. While the user is navigating the system, the current option is circled in red. A medium flex will change which button is selected, and a hard flex will select the current button.

grasps for each object [7]. This provides a method for the user to skip a slower online planning phase, since we will already rely more on the user and less on the automated grasp quality analysis. Using a grasp database also allows us to manually design appropriate grasps for particular affordances that are difficult for an automated planner to recognize. Our UI has been developed as a plugin for the GraspIt! [12] simulator.

A. *sEMG Device Sensing and Control*

We have devised a control method which allows a user to control a cursor in 2 modes using a single sEMG sensor. The sensor is shown attached to a subject in Fig. 2, and its individual components are labeled in Fig. 3. The device is powered through a usb port, and the sEMG signal is amplified and passed through a standard audio jack. The device consists of Gold cup electrodes connected to a Motion Labs Systems Y03 amplifier (x300 gain, 100 dB CMRR, -3 dB bandwidth from 15 Hz to 2 kHz). The sEMG interface shown in Fig. 4 maps a smoothed Root Mean Squared(RMS) value of a single EMG signal to one of several levels, where each level maps to a specific cursor action in the main UI as shown in Fig. 5. A light muscle contraction (medium-level signal) causes the cursor to move between a set of options displayed on the screen. A stronger muscle contraction (high-level signal) chooses one of the options, creating a simple yet effective interface that lets the user navigate and select actions from a menu. Further, the device has been refined to make the user feel in control of the system at all times by seeking to minimize the number of accidental responses recorded by way of spurious spikes in signal strength. This is done by the fact that the user cannot select a button unless they flex strongly. Medium flexes are much more likely to happen accidentally than strong flexes, but they never cause anything to occur, they only change the currently highlighted button.

B. *Phases of the Planner*

The UI contains 6 states shown Fig. 6 which the user navigates through in order to plan and execute a grasp on a target object. The buttons available in each state as well as the state transitions are shown in Fig. 7 and consist of the following:

Object Recognition: This stage initiates the object recognition system which detects objects in the scene using the point cloud captured with a Microsoft Kinect. The point cloud is first oriented to align with the detected checkerboard on the table. Then the point cloud is sent through a pass through filter to remove all points that are not above the table. This filtered point cloud is then sent through a recognition algorithm. The recognition algorithm entails a RANSAC based approach to matching the point cloud data with mesh models of objects stored in the database. More details on this algorithm can be found in[14]. While the recognition service is running in the background, the user is shown a pop up icon which instructing them to wait a moment while the object recognition process completes. The icon fades away once the request is complete, and the detected objects are inserted into the UI scene. The system can identify multiple objects, and the next stage allows the user to identify the target object amidst clutter.

Object Selection: This stage presents the scene populated with objects recognized from the object recognition stage. User intent here is dictated by three main buttons dedicated to selecting the current object, switching to the next object and rerunning the vision system to account for any changes in the scene, respectively. A combination of red and green markers are used to differentiate between the object currently selected and the other objects that could potentially be chosen next. All objects except for the one currently selected are marked red. An example of what the UI looks like to the user in this stage is shown in Fig. 9 (a) Where the currently selected object is shown in green, and the other detected objects are highlighted in red.

Grasp Selection: This stage presents the user with a set of grasps that are pulled from the database for the target object. The user can choose to either select the grasp currently highlighted or cycle through the available grasps until a suitable one is found. Additionally, the user can also navigate back to the previous state in case the chosen object has to be changed. In the background, the grasps are sent to MoveIt![5] and are analyzed for reachability. If the grasp is unreachable, then it is marked as red. If it is reachable, it is marked as green.

Grasp Confirmation: This stage acts as the last confirmation step before proceeding to the actual execution phase. The confirm button sends out the planned grasp and its associated trajectories to the arm for execution while the other button is used to get back to the Grasp Selection stage for further modifications to the grasp.

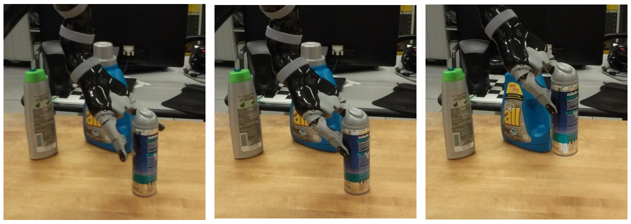
Grasp Execution: When this stage is entered, the currently selected grasp which has already been checked for reachability is sent to the arm to be executed using the process shown in Fig. 8. While the chosen grasp is being executed a single button is available for the user to select in order to pause the current execution procedure. This state is intentionally made very simple with a single button, so that the user is able to quickly stop the arm if required.

Finished/Paused Execution: After execution is either finished or stopped by the user, the system enters into the paused execution stage. Here, the user is presented with options to restart the system and go back to the Object Selection State or to continue with the currently paused grasp execution.

C. *Interface Improvements*

We have improved the usability of the interface in several different ways using basic human factors principles.

Removed Point Cloud from Planning Scene: Prior versions of the user interface had the point cloud of the scene overlaid on the planning scene. We found that while this was useful for developers to debug the system, it confused users as most of them do not often view point clouds, and found it easier to view the scene directly in front of them. In that line of thought, we have optimized this interface to work for someone who is able to view the scene directly in front of them rather than having to provide all information to the user through the interface.



(a) Approach (b) Grasp (c) Lift

Fig. 8: Grasp Execution consists of 3 stages. First the gripper is moved to a position that is backed off from the final grasp position. The gripper then moves to the object, and the fingers close. Finally, the object is lifted off the table.

Oriented Planning Scene To User's Perspective: Prior versions of the user interface showed the scene from the perspective of the Kinect which was capturing the scene. We rotated the planning scene within the user interface to show the scene from the same perspective the user is viewing it from.

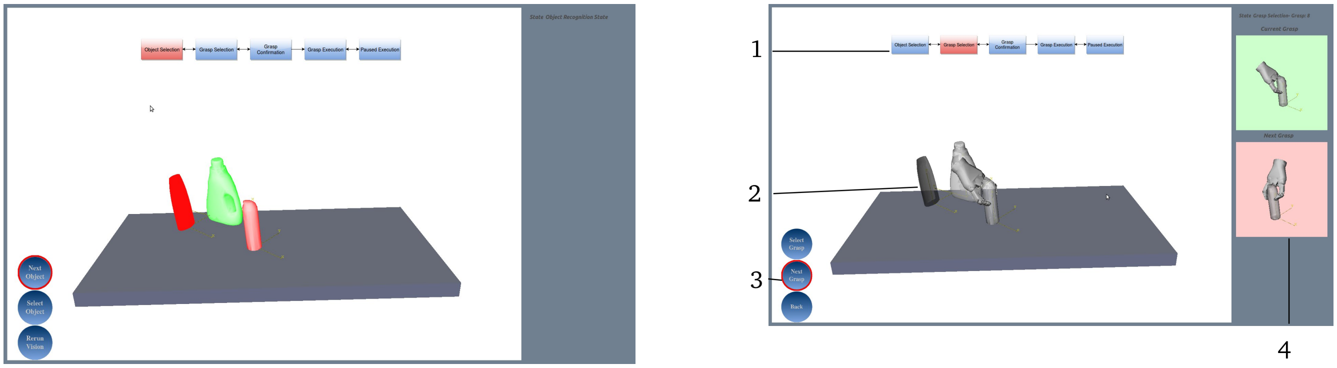
Pipeline Stage Diagram: In order to help clarify the grasping process, the pipeline stage diagram shown in Fig. 6 is illustrated above the planning scene. The pipeline stage diagram shows what stage the user is currently in, what will come next and what they have already done. This makes the consequences of different button selections significantly more clear. For example, it is now clear to a user that they are in the confirmation stage, and that the next stage should they click to move forward is the execution stage which will actually cause the arm to move.

Running Recognition Notification: A large notification is displayed when object recognition is running. It is the only stage of the pipeline where the user is not able to provide input. As this process takes several moments, it is helpful to make it clear to the user that they just need to wait until object recognition is finished.

Enhanced and Accessible Code Base: Several modifications were made to the system to improve its reliability and also to simplify the development process. The most prominent of these changes is the decision to isolate the interface and the sEMG interaction component from the core GraspIt! Application code and repackaging it into a lightweight plugin. This improved the scalability of the system from the developer's perspective as it allowed changes to be made to the underlying system at a rapid pace. The GraspIt! plugin code is available on Github along with detailed setup instructions. Several stereotypical views of the UI are shown in Fig. 9 demonstrating all of the above changes.

IV. USER STUDY

In order to explore the performance of the new system, we ran a study where 6 healthy human subjects were trained and then timed using the new system. The subjects had never



(a) Object Selection Stage: Here, the currently selected object is highlighted in green, while the other detected objects are highlighted in red.

(b) Grasp Selection Stage: The pipeline figure (3) at the top shows where in the process the user is currently at. The middle (2) contains the planning scene. The current available options (3) are displayed in the bottom left. The images on the right (4) show information relevant to the particular stage of the pipeline the user is at.

Fig. 9: Stereotypical views of the user interface as displayed to the user.

used the system prior to their participation in this study. The scene contained 3 separate target objects, an All Brand Detergent Bottle, Gillette Shaving Cream Can, and Garnier Shampoo Bottle. Each subject ran 3 trials, one for each of the target objects. The training process consisted of two steps: (1) a system tutorial where the user navigates through the system using a computer mouse while the user interface is explained. (2) Calibration, the sEMG device is attached to the user and the user learns to consistently produce either a medium or a high signal at will. During this step, the gains, and thresholds are tuned, and the electrodes may be repositioned to improve the signal. This normally takes 2-3 minutes for the forearm, and 5-15 minutes for behind the ear. We then have the subject use the system 3 times, once for each target object. A video detailing the system as used in the user study is available at: `http://www.cs.columbia.edu/~jvarley/SEMGGrasping.ogg`

A. *Lessons Learned*

For the user study, a trial was called successful if the first time the grasp execution stage was reached was intentional, and a failure if the grasp execution stage was ever reached accidentally. The subjects using the device on the forearm were 100% (9/9) successful in only reaching the execution stage intentionally. The subjects using the device behind their ear were successful 78% (7/9) of the time. For the forearm subjects, we were able to successfully lift the object 80% of the time. For the behind-the-ear subjects, we ran into a communication issue between the Mico arm and the computer causing the arm to arbitrarily timeout while following trajectories roughly a third of the time. This made evaluation of the end to end use of the system difficult to analyze, although when the timeouts did not occur, it was able to pick up the object. Fortunately, we were still able to run users through the system and evaluate their performance in navigating through the UI and reaching the Grasp Execution Stage of the pipeline.

The timing results are shown in Table I and Table II. One large take away from our user study is that while it is possible to navigate the interface using the behind the ear control (90.8 seconds on average), it is more difficult than when using the sEMG device on the forearm (67.6 seconds on average). In addition, every stage in the pipeline was visited by the behind the ear subjects on average more often than by the forearm subjects. During development using the forearm is helpful because it is easier to setup, but this gap in performance should be kept in mind as it does take longer to use the device behind the ear.

Overall, the users understood how they were supposed to navigate the system and were aware of what the system was doing at any given point in time. From our user studies we were able to both verify that our improvements to the new system had the intended effect, and several new areas of improvement became apparent.

One lesson learned from the experiments was that the unreachable grasps which were displayed in red confused some subjects. It took a bit of explaining to inform the subjects what the red meant, with several users initially wanted to try unreachable grasps during training as the grasp best aligned with the users intention. We originally believed visualizing the grasps as unreachable would help the user understand that the system had tried to generate valid grasps from a given direction and that they were not feasible, as opposed to not showing anything, and leaving the user wondering why they had no options from a given angle. Instead it seems showing the unreachable grasps to the user causes more confusion than help, and also creates more choices to navigate through.

B. *Subject Survey*

After completion of the user study, the subjects filled out a short questionnaire about their experience with the

TABLE I: Behind Ear: Average times for each of the stages of the pipeline are recorded for each of the subjects. In addition to the number of times each state was entered by each user over all 3 of their trials.

	Average Time Per Stage - Behind Ear Subjects								
Subject	Object Recognition	#	Object Selection	#	Grasp Selection	#	Confirmation	#	Average Total (s)
1	25.3	3	13.7	8	13.1	9	2.4	5	105.1
2	25.1	3	8.4	6	16.7	6	1.8	3	77.1
3	28.4	3	7.4	8	16.7	7	3.0	3	90.1
Total (s)	236.8		219.2		334.8		26.4		
Average (s)	26.3	3	10.0	7.3	15.2	7.3	2.4	3.7	90.8

TABLE II: Forearm: Average times for each of the stages of the pipeline are recorded for each of the subjects. In addition to the number of times each state was entered by each user over all 3 of their trials.

	Average Time Per Stage - Forearm Subjects								
Subject	Object Recognition	#	Object Selection	#	Grasp Selection	#	Confirmation	#	Average Total (s)
1	23.4	3	23.7	3	20.1	3	2.7	3	69.9
2	23.5	3	10.5	5	9.3	5	2.4	3	58.9
3	23.3	3	23.6	3	25.0	3	2.0	3	73.9
Total (s)	70.2		194.4		181.8		21.3		
Average (s)	23.4	3	17.7	3.6	16.5	3.6	2.4	3	67.6

system. From the survey, it was apparent that leaving in the unreachable grasps was not found to be beneficial by the users. We also learned that the control paradigm was highly intuitive for our subjects. This makes sense as there are only two main parameters that need to be adjusted in order to use the device. These parameters are the low to medium threshold, and the medium to high threshold, and adjusting either of these has easy to understand effects.

V. CONCLUSION AND FUTURE WORK

This work has presented a new and improved interface for assistive robotics. The interface provides a higher level of autonomy than prior assistive grasping systems we have developed. It is also easier for users to understand how to effectively interact with the system due to improvements in how both the state of the system and consequences of a user's actions are displayed. This system is being migrated to the Columbia University Medical Center in order to support extensive user studies on subjects with spinal cord injuries with the goal being to provide assistance with activities of daily living.

REFERENCES

[1] P. K. Artemiadis and K. J. Kyriakopoulos. A switching regime model for the EMG-based control of a robot arm. *IEEE Trans. on Systems, Man, and Cybernetics*, 41(1):53–63, Feb. 2011.

[2] C. J. Bell, P. Shenoy, R. Chalodhorn, and R. Rao. Control of a humanoid robot by a noninvasive brain-computer interface in humans. *Journal of Neural Engineering*, 5(2):214–20, Jun 2008.

[3] M. Bryan, V. Thomas, G. Nicoll, L. Chang, J. Smith, and R. Rao. What You Think is What You Get: Brain-Controlled Interfacing for the PR2. Technical report, Iros 2011: The PR2 Workshop, San Francisco, 2011.

[4] C. Castellini and P. van der Smagt. Surface EMG in advanced hand prosthetics. *Biological cybernetics*, 100(1):35–47, Jan. 2009.

[5] S. Chitta, I. Sucan, and S. Cousins. Moveit![ros topics]. *Robotics & Automation Magazine, IEEE*, 19(1):18–19, 2012.

[6] C. Cipriani, F. Zaccone, S. Micera, and M. C. Carrozza. On the Shared Control of an EMG-Controlled Prosthetic Hand: Analysis of User Prosthesis Interaction. *IEEE Transactions on Robotics*, 24(1):170–184, Feb. 2008.

[7] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen. The columbia grasp database. In *Int. Conf. on Robotics and Automation*, pages 1710–1716. IEEE, 2009.

[8] 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*, 11(7), 2011.

[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*, pages 1–5. IEEE, June 2011.

[10] P. Horki, T. Solis-Escalante, C. Neuper, and G. Müller-Putz. Combined motor imagery and SSVEP based BCI control of a 2 DoF artificial upper limb. *Medical & Biological Engineering & Computing*, 49(5):567–77, May 2011.

[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*, pages 554–557, Apr. 2011.

[12] A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*, 11(4):110–122, 2004.

[13] K. Mülling, A. Venkatraman, J. Valois, J. Downey, J. Weiss, S. Javdani, M. Hebert, A. B. Schwartz, J. L. Collinger, and J. A. Bagnell. Autonomy infused teleoperation with application to BCI manipulation. *CoRR*, abs/1503.05451, 2015.

[14] C. Papazov and D. Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. In *Computer Vision–ACCV 2010*, pages 135–148. Springer, 2010.

[15] C.-c. Postelnicu, D. Talaba, and M.-i. Toma. Controlling a Robotic Arm by Brainwaves and Eye. *Int. Fed. For Information Processing*, pages 157–164, 2011.

[16] G. N. Ranky and S. Adamovich. Analysis of a commercial EEG device for the control of a robot arm. In *Proc. Northeast Bioengineering Conference (NEBEC)*, pages 1–2, New York, NY, Mar. 2010.

[17] K. Sagawa and O. Kimura. Control of robot manipulator using EMG generated from face. In *Proc. Int. Conf. on Manufacturing and Industrial Technologies*, volume 6042, Dec. 2005.

[18] 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*, volume 6691. Springer, June 2011.

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

[20] P. Shenoy, K. J. Miller, B. Crawford, and R. N. Rao. Online electromyographic control of a robotic prosthesis. *IEEE Transactions on Biomedical Engineering*, 55(3):1128–35, Mar. 2008.

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

[22] J. Vogel, S. Haddadin, J. D. Simeral, S. D. Stavisky, D. Bacher, L. R. Hochberg, J. P. Donoghue, and P. Van Der Smagt. Continuous Control of the DLR Light-weight Robot III by a human with tetraplegia using

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

[23] N. Waytowich, A. Henderson, D. Krusienski, and D. Cox. Robot application of a brain computer interface to staubli TX40 robots - early stages. *World Automation Congress (WAC)*, pages 1–6, 2010.

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

[25] J. Weisz, C. Elvezio, and P. K. Allen. A user interface for assistive grasping. *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013.

[26] J. Weisz, B. Shababo, L. Dong, and P. K. Allen. Grasping with your face. *Springer Tracts in Advanced Robotics*, pages 435–448, 2013.

[27] A. Wołczowski and M. Kurzyński. Human-machine interface in bioprosthesis control using EMG signal classification. *Expert Systems*, 27(1):53–70, Feb. 2010.

[28] 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 *Int. Conf. on Intelligent Robots and Systems*, pages 516–521. IEEE, Oct. 2009.