

A User Interface for Assistive Grasping

Jonathan Weisz, Carmine Elvezio, and Peter K. Allen

Abstract—There has been considerable interest in producing grasping platforms using non-invasive, low bandwidth brain computer interfaces (BCIs). Most of this work focuses on low level control of simple hands. Using complex hands improves the versatility of a grasping platform at the cost of increasing its complexity. In order to control more complex hands with these low bandwidth signals, we need to use higher level abstractions. Here, we present a user interface which allows the user to combine the speed and convenience of offline preplanned grasps with the versatility of an online planner. This system incorporates a database of pre-planned grasps with the ability to refine these grasps using an online planner designed for arbitrarily complex hands. Only four commands are necessary to control the entire grasping pipeline, allowing us to use a low cost, noninvasive commercial BCI device to produce robust grasps that reflect user intent. We demonstrate the efficacy of this system with results from five subjects and present results using this system to grasp unknown objects.

I. BACKGROUND AND RELATED WORK

In this work, our goal is to build a shared-control system for assistive robotic grasping for complex hands that is designed to work with a brain computer interface (BCI). Grasping objects is an important component of many activities of daily living that are problematic for individuals with upper limb mobility impairments. By creating a robust system for shared control of a robotic assistant, we can enable impaired individuals to improve their quality of life.

Control of a robot using BCI signals is a difficult problem spanning many modalities and domains. Here we provide a brief overview of some of the work done in controlling manipulators and grippers using electrophysiological signals and BCI devices. For a more complete review see [1].

In Vogel et al. [2], the authors demonstrated online trajectory control of a robotic manipulator using the BrainGate cortically implanted electrode in an immobilized subject. While this was an impressive achievement, it required an invasive device capable of recording a large number of high quality signals.

Other work has established control over manipulators using less invasive, more commonly available interfaces. One such interface is electromyography (EMG) signals from forearm muscles. Shenoy et al. [3] used forearm EMG to perform basic pick and place tasks. Other authors [4], [5], [6], [7], [8] have used forearm EMG signals to switch a robotic hand between discrete shapes for grasping and manipulating.

Forearm EMG signals are only available to patients who retain control over their arms, which is not the case for many patients with impaired mobility. A larger population

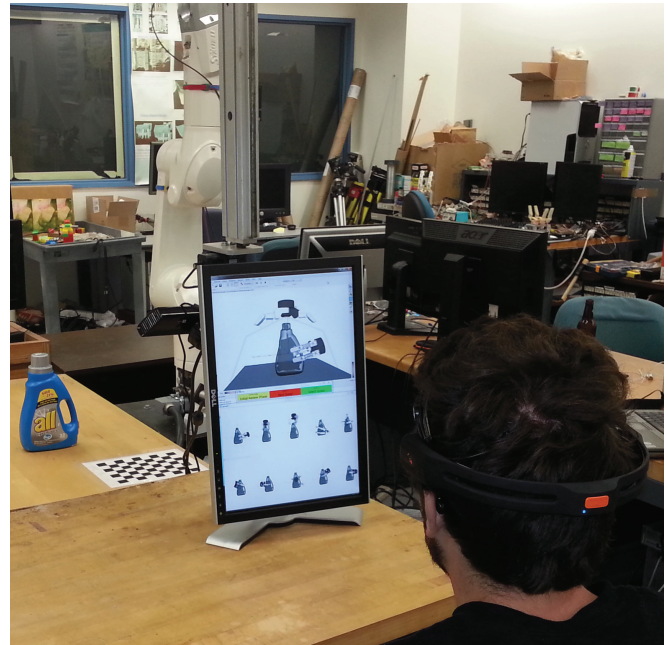


Fig. 1. A user of our BCI interface controlling the system. The subject wears an Emotiv Epoc EEG headset which is used to detect facial gestures that control the system. A point cloud is obtained by a Microsoft Kinect and used to detect the identity and position of the target object. Then the user uses the EEG headset to guide a grasp planning interface to find a good grasp for the object, and the grasp is sent to our robotic grasping platform for execution.

of patients maintain control over facial and head muscles, therefore various authors have proposed control schemes using face and head EMG signals to control robotic arms and grippers [9], [10], [11]. Eye gaze direction is also usually preserved and has been used to control 2D arm position and gripper opening and closing [12].

Some work has focused on a higher level, goal oriented paradigm which reduces the burden of controlling the robot on the user [13]. In [14], electroencephalography (EEG) signals were used to select targets for a small humanoid grasping platform. Similarly, Waytowich et al. [15] used EEG signals to grasp and place objects using 4-DOF Stäubli robot. Bryan et al. [16] presented preliminary work extending this approach to a grasping pipeline on the PR2 robot. Ciocarlie et al. [17] introduced a human-in-the-loop interface for grasping with the PR2.

In order to be useful, a BCI interface to grasping and manipulation needs to be placed in a greater context for controlling the high level goals of the robot. Such high level paradigms are beginning to emerge [18], [19].

However, in spite of this broad interest in developing

J. Weisz and P. Allen are with the Department of Computer Science, Columbia University, NY 10027, USA. E-mail: {jweisz, allen}@cs.columbia.edu

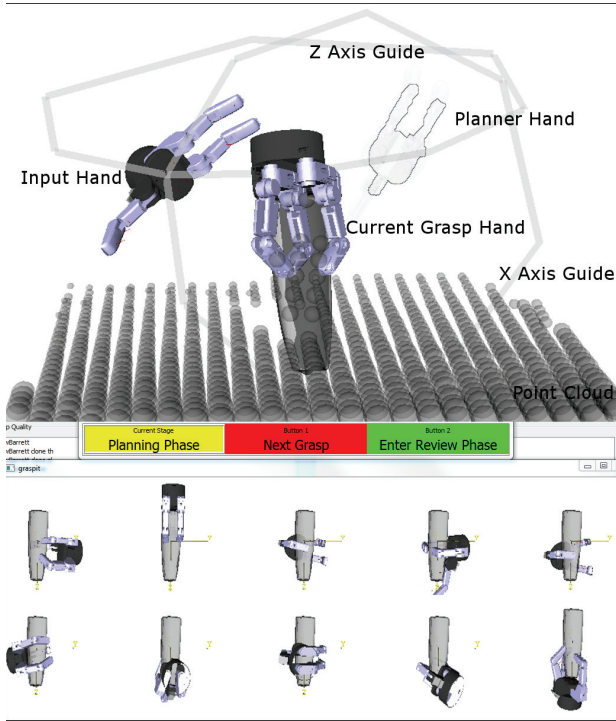


Fig. 2. The user interface for the semi-autonomous grasp planning in the *Online Planner* phase. The user interface is comprised of three windows: The main window containing three labeled robot hands and the target object with the aligned point cloud, the pipeline guide window containing hints for the user to guide their interaction with each phase of the planner, and the grasp view window containing rendering of the ten best grasps found by the planner thus far. The object shown in this figure is a novel object that the planner does not have a model of (see Fig. 5(b)). The point cloud allows the user to visualize the fit of the model and act accordingly.

BCI interfaces for grasping and manipulation, most work has focused on simple grippers. In previous work [1], we demonstrated the first *end-to-end* shared autonomy grasping system for complex hands. This prototype system allowed a user to perform the basic interactions necessary for grasping an object. In this paper we extend our prior work in a number of ways:

- Introducing a more flexible interface that reduces the effort required by the user
- Presenting user studies of our system to measure the efficacy and ease of use of the interface
- Integrating the planner with a database of preplanned grasps from a fully automated grasp planner
- Adding the ability to add semantically relevant grasps based on object functionality even though they may not be geometrically stable
- Allowing the use of both the pre-planned grasp database and the online grasp planner depending on which best reflects the users intent.

II. GRASPING PIPELINE

In assistive robotics, only low dimensional, noisy signals are available from most patient populations with significant impairments. Therefore, assistive robotics systems typically

use only simple grippers and exercise simple control algorithms over them. In order to control a high dimensional robotic manipulator using a low dimensional, noisy signal you need a robust, high level interface. Using more complex and higher DOF robotic hands increases the versatility of the system but at the cost of higher complexity for control.

A. User Interface

A user of our interface can be seen in Fig. 1. The subject is wearing an Emotiv Epoc EEG headset which is used to control the system through facial gestures. A kinect mounted behind the computer monitor obtains a point cloud that is used to identify and localize the target object. The user interacts with a simulation environment to plan grasps for the object, which are then carried out by our robotic grasping platform.

The interface for grasping is shown in Fig. 2. This interface is composed of the simulation window, the pipeline guide window, and the grasp view window. In the main window, the user is able to interface with a simulated world where they visualize and control the grasp planner. Below it, the pipeline guide window shows the user what stage of the grasping process they are in and what the commands do in the current stage. On the bottom, the grasp view window contains smaller visualizations of the ten best grasps the planner is aware of. We have found that ten is the maximum number that we can effectively fit on screen and provides a reasonable number of choices.

The main window contains three renderings of the robot hand and a rendering of the target object. Each of the robot hand renderings are distinguished by their level of opacity. The *current grasp hand* is completely opaque, the *input hand* has an opacity of 80%, and the final *grasp planner indicator* hand has an opacity of only 50%. The purpose of the *input hand* is to allow the user to visualize and control the desired approach direction that is used as part of the grasp planning process. The *current grasp hand* allows the user to visualize a particular grasp for the object and shows the user their current choice. The *grasp planning indicator* exists only while the planner is running and intermittently shows the user what grasps the grasp planner is currently considering. This shows the user how their input is effecting the planner in real time. The main window also contains the *x* and *z* axis guides which show the user how their input will affect the approach direction of the *input hand* as well as a decimated point cloud that allows the user to understand how well the aligned object used by the planner represents the data in the real world. In the example in Fig. 2, a known object is aligned to a novel object, and this user interface allows the user to grasp the object even in the absence of an exact model. These objects can be seen in Fig. 5(b)

One of our observations in our previous work was that although the grasp planner sometimes produced grasps which were unreliable, the user was able to distinguish reliable from unreliable grasps very accurately and achieve a 100% success rate in grasping objects in our preliminary tests. Using this insight, we have integrated a database of grasps with our

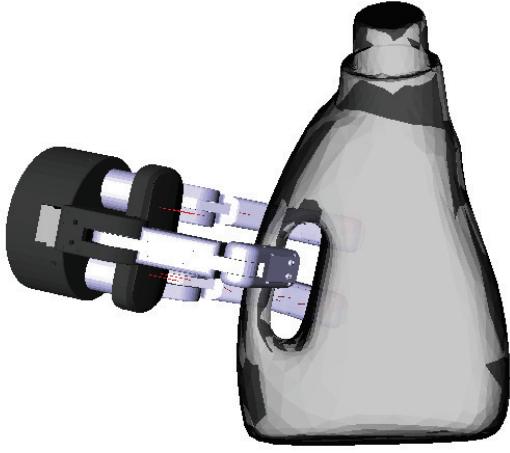


Fig. 3. This handle grasp for the all bottle is not a force closure grasp, but when chosen by the subjects in our experiments it succeeded 100% of the time. Adding a grasp database allows such semantically relevant grasps to be used in our system.

previous work in a way that allows the user to quickly select among them and further refine them to stable grasps as necessary. Some of the grasps in this database are from an automated planner, while some grasps have been hand tuned to have a semantic meaning, such as grasping a container by its handle such as that in Fig. 3. These grasps may not be stable in the force closure sense but may reflect higher level knowledge about affordances for the object. The grasps in the database are ranked such that grasps with a semantic meaning come first while grasps with a high stability as measured by their ability to resist force perturbations follow. This approach melds the approaches taken in our previous work in [20] and [21].

B. Pipeline Details

The grasping pipeline, illustrated in Fig. 4, is divided up into seven stages: object identification and alignment, database planning, planner initialization, online planning, grasp review, confirmation, and grasp execution, which are described below. This pipeline is controlled using only four facial gestures. The use of these gestures in each stage of the pipeline is explained in Table I. In general, gesture 1 serves as a “no” and in stages is used to indicate that the current grasp is not suitable and proceed to the next grasp. Gesture 2 indicates a “yes” and is used to allow the user to proceed to the next stage. False positive readings of these two gestures have strong consequences, and are best associated with a concise and strong facial gesture such as closing one eye or clenching the jaw. Gestures 3 and 4 always control the approach direction of the *input hand* relative to the object. These gestures can be maintained to generate *continuous* motion of the hand over two degrees of freedom and therefore are best associated with gestures that can be contracted for several seconds without too much twitching or fatigue.

1) *Object Recognition*: For a complete description of the vision system used in this paper, see [22]. Briefly, this system

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

TABLE I

A DESCRIPTION OF THE USER INTERFACE AS THE USER PROGRESSES THROUGH PHASES OF THE PIPELINE.

uses RANSAC on features derived from oriented pairs of points to find potential correspondences in a hash table from points in the scene to points in a known set of objects. Fig. 5(a) shows a correctly chosen model aligned with the range scan taken with a Microsoft Kinect. This method is robust and fast enough to demonstrate the efficacy of our BCI-grasping pipeline. Even when no exact model is in the database, a reasonable model can be well aligned to the object, as seen on the right side of Fig. 5(a).

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

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

4) *Online Planning*: If the user needs to refine their chosen grasp, they can elect to start the online planner. The planner generates a starting pre-grasp pose by moving the input hand of the main window to mirror the desired grasp after the hand has been opened and withdrawn several centimeters along a pre-set approach direction. The planner then runs, replacing the grasps in the grasp view window as new solutions are found that more closely adhere to the desired approach direction demonstrated by the input hand. The user is able to control the desired approach direction by moving the hand along the circular guides demonstrated by Fig. 2 using Gesture 3 to rotate around the z axis of the object and Gesture 4 to rotate around the x -axis of the object. As in the *Initial Grasp Review* phase, Gesture 1 allows the user to browse through the current list of grasps, while Gesture 2 signals acceptance of a particular grasp.

For a detailed discussion of the Online Eigengrasp Grasp Planner used in this work, see [21]. Briefly, the planner uses

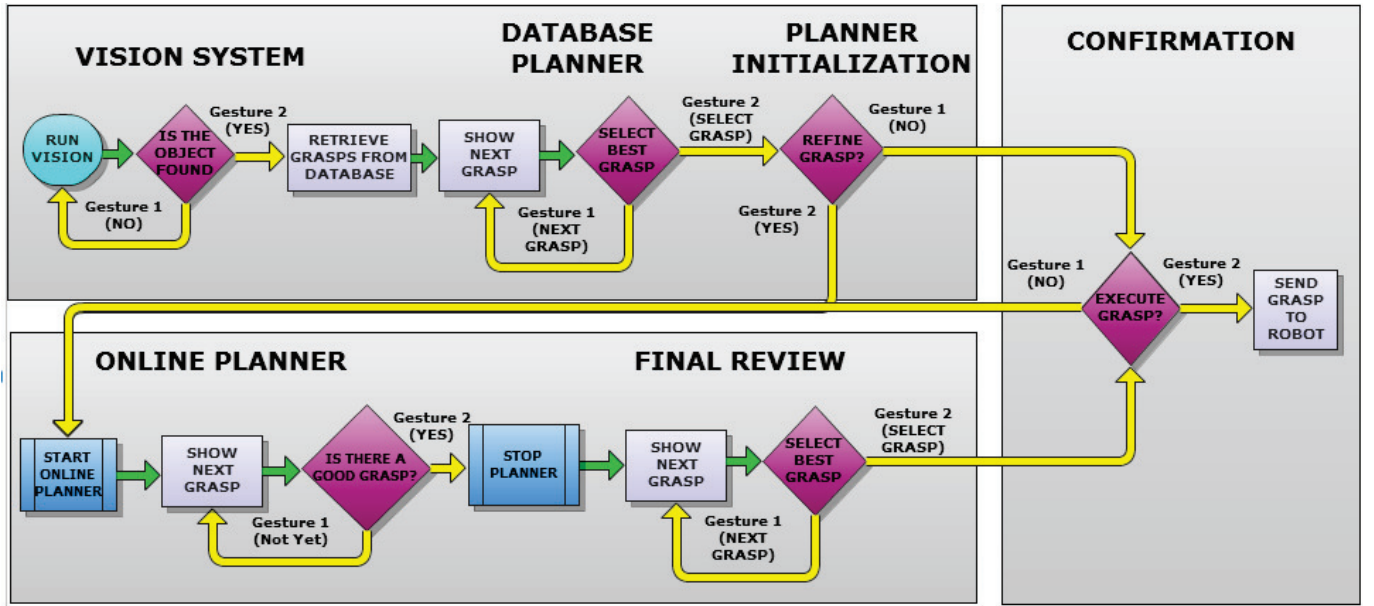


Fig. 4. The phases of our grasping pipeline. The purple diamonds reflect decision points in the pipeline that require input from the user. If the user chooses grasp n from the database, $n+4$ user inputs are required. If none of the grasps are suitable, the online planner can be invoked with a few simple inputs to refine one of the grasps further.

a two stage process to find a set of grasps for a particular object. In the first stage, simulated annealing to optimize a cost function based on the distance from pre-planned contact points on the fingers to the target object and the normal between the projection of those points to the object. The optimization process is constrained to remain roughly in the neighborhood of an example pose, where for each dimension of the example pose a *confidence* variable controls how far the planner is allowed to deviate from the example pose. The example pose is set by the *input hand* in the main window. As the user moves the *input hand*, the planner produces solutions that track its motion.

In order to make this approach computationally tractable, a lower dimensional linear subspace for the joint postures of the hand is explored by the first stage of the optimization process. This dimensionality reduction is motivated by computational motor control experiments that show that 80% of the motion of the human hand in the pre-shaping phase of grasping can be explained by only two dimensions [23], [24]. In the second phase of planning, promising grasps from the simulated annealing phase are refined by approaching the object and closing the fingers along a prespecified trajectory until each finger makes contact, potentially leaving the postural subspace explored during the simulated annealing phase to conform to the object.

5) *Review Grasps*: With the planner stopped and the list of grasps stable, the user is able to continue browsing through the list using Gesture 1, or continue to the final confirmation phase using Gesture 2.

6) *Confirm Grasp*: Having viewed the possible grasps and selected the best possible option, the user may use Gesture 1 to restart the planner if they reject all of the grasps or use Gesture 2 to execute the selected grasp.

7) *Execute Grasp*: In this phase, the planner attempts to find a path to the planned grasp. If the planner is unable to find a path due to kinematic constraints of the arm or collisions with the environment, the main window interface flashes black briefly to signal the user that the planner has failed. The user is then able to return to the *Review Phase* using Gesture 1. Once in the *Review Phase* the user will be able to select a different grasp or restart the planning process from the *Online Planning* phase.

III. EXPERIMENTS

A. Task

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

B. Training

The Emotiv Epoc EEG headset uses 14 electrodes sampled at 128 hz with 16 bits of resolution. The signals are communicated over Bluetooth to the Cognitiv and Expressiv classifier suites from Emotiv. In this work, we use four gestures. Gesture 1 is the jaw clench classified by the Expressiv classifier. We have trained the Cognitiv suite on three signals, right eye winking as gesture 2, left side jaw clenching as gesture 3, and eyebrow raising as gesture 4.



(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 juice bottle on the left is used as a *novel* object in this experiment. In Fig. 2 and Fig. 5(a), we show the model of the shampoo bottle aligned with the point cloud data from the juice bottle. Although the objects are relatively different in size and shape, the alignment found by the vision system is sufficient to run the grasp planner on the bottle.

Fig. 5.

Subjects were trained to use the Emotiv Epoc in four fifteen minute sessions. In the first two sessions, the classifier built into the Cognitiv Suite of the Emotiv Epoc software was trained on the three facial expressions used in the experiment. In the second two sessions, the subject was asked to perform the task in the virtual environment without executing the final grasp on the actual arm.

C. Grasping Platform

Our grasping platform is composed of a 280 model BarrettHand, a Stäubli TX60L 6-DOF robotic arm, and a Kinect sensor. We use the OpenWAM driver for realtime control of our BarrettHand and the OpenRave CBIrrt planner [25] for arm motion planning.

D. Results

The results of the experiments are reported in Table II. For each subject, we report the mean time to completion and fraction of successful attempts for each grasp. Time to completion is measured from the end of the object identification phase to the beginning of the execution phase, as this represents the time taken to plan the grasp. Overall, the average planning time was 104 seconds on the known objects and 86 seconds on the unknown object. The average success rate was 80%, demonstrating that this system is efficacious in allowing the user to plan and execute a reasonable grasp for these objects. A video of the grasping process can be found at <http://robotics.cs.columbia.edu/jweiszbciGraspingIROS2013>.

After the experiment, subjects were asked to describe their discomfort during the experiment and their level of control. Subjects reported little discomfort, but were frustrated with the difficulty of getting the Epoc to recognize their intended

actions, especially with false negatives making it difficult to continue to the next stage of the pipeline at will. In spite of this frustration, subjects were able to complete the task. This demonstrates that the subjects tolerated the system reasonably well and felt that it gave them enough control to perform the task.

These results show that we have developed an effective shared control grasp planning system for complex hands. It is notable that grasps from the side demonstrated significantly more robustness and lower planning times than grasps from above. The grasp database only contained one grasp from above for each of these objects, and this grasp was a fingertip grasp which may be sensitive to pose estimation error, which resulted in longer planning times while the subjects searched for a better grasp. In general, grasping roughly cylindrical objects such as the top of the detergent bottle from above is somewhat problematic for the BarrettHand due to its configuration and the low friction of its fingertips. In contrast, subjects were able to find a reasonable grasp from the side of the object among the grasps pulled directly from the database. The difference in planning times reflects the benefit of integrating the off-line planning phase.

IV. DISCUSSION

In this work, we demonstrated our improved interface for planning grasps for complex hands in a simulator. We integrated this interface with a grasp database that allows both semantically relevant and geometrically stable grasps to be used. This produced a working system that our user study showed enabled users to reliably achieve good grasps in a reasonable amount of time.

By integrating a preplanned and online human-in-the-loop approach, we have produced a system that is flexible and

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	2/3	132
	2	2/3	75
	3	2/3	96
	4	3/3	93
	5	2/3	125
Detergent Bottle Side	1	3/3	75
	2	3/3	57
	3	3/3	106
	4	2/3	82
	5	3/3	75
Detergent Bottle Top	1	1/3	151
	2	2/3	114
	3	2/3	142
	4	2/3	161
	5	3/3	145
Novel Bottle	1	3/5	132
	2	4/5	63
	3	4/5	95
	4	4/5	91
	5	4/5	50

TABLE II
RESULTS FROM EXPERIMENTS

usable. Although we used the Emotiv Epoc as our input to the system, the system itself is agnostic as to the input device used and the binding of each gesture, as long as four input signals can be derived from it. We have begun integrating the single electrode BCI device described in [26] as a less invasive, even lower cost alternative to the Epoc.

The role of this system is to add flexibility to a more general assistive robotics environment such as that proposed in [18]. In future work we will integrate with such a system to produce a full assistive robotics environment for use on a mobile manipulator. A full system will also integrate the ability to add grasps to the database online and tag them with semantic meanings as applicable. A flexible grasp planning system such as the one we have demonstrated is a key step towards building a flexible assistive robotic manipulator.

V. ACKNOWLEDGMENTS

This work has been funded by NSF Grants IIS-0904514 and IIS-1208153.

REFERENCES

- [1] J. Weisz, B. Shababo, and P. K. Allen, "Grasping with your face," in *Proc. of Int. Symposium on Experimental Robotics*. Springer, 2012.
- [2] 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.
- [3] P. Shenoy, K. J. Miller, B. Crawford, and R. N. Rao, "Online electromyographic control of a robotic prosthesis," *IEEE transactions on bio-medical engineering*, vol. 55, no. 3, pp. 1128–35, Mar. 2008.
- [4] 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*. IEEE, Oct. 2009, pp. 516–521.
- [5] A. Woczkowski and M. Kurzyski, "Human-machine interface in bio-prosthesis control using EMG signal classification," *Expert Systems*, vol. 27, no. 1, pp. 53–70, Feb. 2010.

- [6] 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 *2011 IEEE International Conference on Rehabilitation Robotics*. IEEE, June 2011, pp. 1–5.
- [7] C. Cipriani, F. Zaccane, S. Micera, and M. Carrozza, "On the Shared Control of an EMG-Controlled Prosthetic Hand: Analysis of User Prosthesis Interaction," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 170–184, Feb. 2008.
- [8] G. Matrone, C. Cipriani, M. C. Carrozza, and G. Magenes, "Two-channel real-time EMG control of a dexterous hand prosthesis," in *IEEE/EMBS Conference*, Apr. 2011, pp. 554–557.
- [9] K. Sagawa and O. Kimura, "Control of robot manipulator using EMG generated from face," in *ICMIT 2005: Control Systems and Robotics*, vol. 6042, no. 1, Dec. 2005, pp. 604 233–604 233–6.
- [10] 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, pp. 7110–7126, 2011.
- [11] G. N. Ranky and S. Adamovich, "Analysis of a commercial EEG device for the control of a robot arm," in *Proc. IEEE Northeast Bioengineering Conference*, New York, NY, Mar. 2010, pp. 1–2.
- [12] C.-c. Postelnicu, D. Talaba, and M.-i. Toma, "Controlling a Robotic Arm by Brainwaves and Eye," *International Federation For Information Processing*, pp. 157–164, 2011.
- [13] 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.
- [14] 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. 214–220, June 2008.
- [15] 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)*, 2010, pp. 1–6.
- [16] M. Bryan, J. Green, M. Chung, J. Smith, R. Rao, and R. a. Scherer, "Towards Hierarchical Brain-Computer Interfaces for Humanoid Robot Control," in *11th IEEE-RAS International Conference on Humanoid Robots*. IEEE-RAS, Oct. 2011.
- [17] A. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Human Robot Interaction*, 2012.
- [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*, June 2011, vol. 6691/2011, pp. 362–369.
- [19] 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. of the Int. Symposium on Experimental Robotics*. Springer, 2012.
- [20] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen, "Data-driven grasping with partial sensor data," in *Proceedings of the IEEE/RSJ Inter. Conf. on Intelligent robots and systems*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1278–1283.
- [21] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.
- [22] C. Papazov and D. Burschka, "An efficient ransac for 3d object recognition in noisy and occluded scenes," in *Computer Vision ACCV 2010*, 2011, vol. 6492, pp. 135–148.
- [23] A. Tsoli and O. C. Jenkins, "2d subspaces for user-driven robot grasping," in *RSS Workshop on Robot Manipulation: Sensing and Adapting to the Real World*, Atlanta, GA, June 2007.
- [24] M. Santello, M. Flanders, and J. F. Soechting, "Patterns of hand motion during grasping and the influence of sensory guidance," *The Journal of Neuroscience*, vol. 22, no. 4, pp. 1426–1435, 2002.
- [25] D. Berenson, S. S. Srinivasa, and J. Kuffner, "Task Space Regions: A framework for pose-constrained manipulation planning," *The International Journal of Robotics Research*, Mar. 2011.
- [26] S. Joshi, A. Wexler, C. Perez-Maldonado, and S. Vernon, "Brain-muscle-computer interface using a single surface electromyographic signal: Initial results," in *Int. IEEE/EMBS Conf. on Neural Engineering*, May 2011, pp. 342–347.