Biomimetic Grasp Planning for Cortical Control of a Robotic Hand

Matei T. Ciocarlie, Samuel T. Clanton, M. Chance Spalding, and Peter K. Allen

Abstract— In this paper we outline a grasp planning system designed to augment the cortical control of a prosthetic arm and hand. A key aspect of this task is the presence of on-line user input, which will ultimately be obtained by identifying and extracting the relevant signals from brain activity. Our grasping system can combine partial or noisy user input and autonomous planning to enable the robot to perform stable grasping tasks. We use principal component analysis applied to the observed kinematics of physiologic grasping to reduce the dimensionality of hand posture space and simplify the planning task for on-line use. The planner then accepts control input in this reduced-dimensionality space, and uses it as a seed for a hand posture optimization algorithm based on simulated annealing. We present two applications of this algorithm, using data collected from both primate and human subjects during grasping, to demonstrate its ability to synthesize stable grasps using partial control input in real or near-real time.

I. CORTICAL CONTROL OF REACHING AND GRASPING ROBOTS

Since the discovery of the relationship between the activity of the neurons in the motor cortex and movement of the upper limb [1], there has been an effort to use these signals to control computers and robots. Clinical brain-machine interface prosthetics based on this research could provide restoration of function to those with amylotrophic lateral sclerosis, quadriplegia, or other pathologies that cause the loss of motor function.

Consider the goal of building an anthropomorphic prosthetic arm and hand that are controlled by cortical output. A number of complex factors comprise dexterous grasping and manipulation, including positioning the arm, orienting the wrist, and shaping the fingers. Together, a high number of degrees of freedom (DOFs) of control are needed to successfully complete a grasping or manipulation task.

In [2], Taylor *et al.* enable a primate to directly control the linear velocity of the endpoint of a robot arm through 3 DOFs in real time. This control was achieved by measuring the activity of individual cortical neurons that correspond to individual preferred directions of each neuron in space. The vector sum of preferred directions of a population of neurons,

each scaled by their individual unit activity, provides the velocity of robotic end-effector movement. Inverse kinematics are used to command 3-DOF endpoint velocity with an arm actuated in 4 dimensions. As reported more recently, Velliste *et al.* [3] also demonstrate the additional proportional cortical control of a robotic pincer.

While this approach has been effective for controlling the transport function of arm and simple pinching, the differing nature of dexterous grasping makes it difficult to directly apply the same algorithm. The arm is a 4-DOF manipulator whose main function in grasping is to bring the hand to a certain point in 3-D space. As a prosthetic device, any actuated system that can achieve placement of its end effector over a similar workspace is sufficient to allow control of hand transport. In contrast, the human hand is a 24-DOF manipulator whose function is to conform to the complex shape of the object to be grasped, then close stably on it in a way that allows the desired type of manipulation. The activity of some motor neurons are correlated with finger movements, indicated by joint angles [4]. However, it is not expected that recorded neural data will provide enough information about grasp shape to completely describe a grasp, at least for the foreseeable future. Furthermore, robotic hands generally have different kinematics than physiologic hands, meaning that the controllable degrees of freedom of the robot hand will not correspond to those observed during physiologic grasping. Direct cortical control of a robotic hand will therefore require methods to a) make use of incomplete or noisy information obtained through neural intent and b) translate it to actuation of a robot with a particular non-physiologic kinematic configuration.

A. Shared Control Through Online Grasp Planning

In this paper we propose a type of automatic grasp planning system that may be appropriate for integration into a robotic reach and grasp control system driven by cortical activity. This system will be able to operate in real or near-real time to control degrees of freedom of a reach and grasp robot that are uncontrolled or noisily controlled by user input. Additionally, it is important that the system be able to adapt to subjects as they learn the control of individual DOFs. The first intended application for this system will be with primate subjects learning to control a grasping and manipulating robot, which underscores the need for the system to accommodate operator learning.

We consider that a grasp planning system will incorporate the following criteria in order to be appropriate for neural-prosthetic shared control:

- *functionality and interactivity*: to be able to find a stable grasp of a target object fast enough to allow for on-line interaction with an operator.
- *adjustable cortical/computer control*: the level of operator control versus synthetic behavior should be directly adjustable along different dimensions, allowing both learning and adaptation to subjects with differing control ability.
- *biomimetic synthesis*: in the absence of complete user control, grasps will be created such that automatically controlled parameters will resemble observed physiologic behavior as much as possible.

In this paper we present a grasp planning algorithm that can meet these criteria; we also describe two preliminary experiments that confirm the ability of the planner to use real-life input data and provide useful output in the form of stable planned grasps. Finally, we discuss some of the factors involved with integrating this system into a complete training framework for cortically controlled prosthesis.

B. Related Work

For comprehensive reviews concerning fully autonomous robotic grasp synthesis, we refer the reader to [5], [6]. We note that our approach requires knowledge of the target object geometry, as well as its initial position relative to the hand. For application in unstructured environments, it therefore requires a complementary system for object recognition and localization such as [7]. However, we can also take advantage of the fact that our main application, operator training for cortical control, usually takes place in controlled environments where such information is directly available.

One of the most significant challenges for understanding human grasping and building effective prostheses is the high dimensionality of the human hand posture space. A discussion on different dimensionality reduction techniques applied to human hand motion capture data is presented by Tsoli and Jenkins [8]; their results show that a human operator can perform simple grasping tasks by controlling an artificial hand through a 2-dimensional input device like a computer mouse. Brown and Asada [9] present an anthropomorphic robotic hand design with a low-dimensional control system. Their architecture is inspired by the results on low-dimensional hand posture subspaces obtained by Santello *et al.* [10], which we also build on in this paper. Carrozza *et al.* [11] present the Cyberhand design, relying on passive mechanical adaptation to complement limited user control; Cipriani *et al.* [12] use a grasp planning algorithm that assumes the user can only select from a small set of pre-grasp shapes, and rely on the passive adaptation of the Cyberhand to successfully grasp a number of different objects.

Taylor *et al.* [2], Serruya *et al.* [13], and Carmena *et al.* [14] have investigated the cortical control of robot arms in reaching tasks. A shared-control approach to the cortical control of a reaching robot that augments robot trajectory generation with computer generated reflex-like command signals is described by Kim *et al.* [15]. Zecca *et al.* [16] discuss the use of electromyographic (EMG) signals for robotic hand control as an alternative to neural recordings.

However, translation of EMG information into joint positions requires the use of complex learning methods, as discussed by Afshar and Matsuoka [17] and Bitzer and van der Smagt [18]. Finally, the idea of human-robotic shared control is a common theme in robotic-assisted surgery, as in work presented by Kragic *et al.*[19] and Ang *et al.*[20].

II. LOW DIMENSIONAL HAND POSTURE SUBSPACES

While hand shape can generally be described using up to 24 variables, it is theorized that control of hand posture takes place in a lower-dimensional space of coordinated motions, or "synergies" [10]. These synergies are thought to arise due to a combination of biomechanical coupling in the hand and synchronous muscle activation. Therefore, much of the control of the hand may take place in a space with significantly less than 24 DOF. A low-dimensional projection of hand kinematics that may have physiologic significance is convenient, as it simplifies the control space of a grasp planning algorithm while producing results that normally lie within the set of biologically plausible hand postures.

In our approach we define a low-dimensional hand posture subspace by using a number of basis vectors called **eigengrasps** [21]; the implication is that these vectors can be linearly combined to obtain a wide range of hand postures for grasping tasks. Consider a hand posture $\boldsymbol{p}$ expressed as a point in the d-dimensional DOF space:

$$\boldsymbol{p} = [\theta_1 \quad \theta_2 \quad \dots \quad \theta_d] \in \mathcal{R}^d \qquad (1)$$

where θ_i is the value of i-th degree of freedom. When choosing a basis comprising b eigengrasps, a hand posture placed in the subspace defined by this basis can be expressed as a function of the amplitudes a_i along each eigengrasp direction $\boldsymbol{e}_i$:

$$\boldsymbol{p} = \boldsymbol{p}_m + \sum_{i=1}^{b} a_i \boldsymbol{e}_i \ , \ \ \boldsymbol{e}_i = [e_{i,1} \quad e_{i,2} \quad \dots \quad e_{i,d}] \qquad (2)$$

where $\boldsymbol{p}_m$ is a "mean" posture representing the origin of the subspace. Once this subspace is defined, a hand posture can be completely determined by the amplitude vector $\boldsymbol{a} = [a_1 \ \dots \ a_b] \in \mathcal{R}^b$. However, the key aspect when selecting the basis vectors is that the *resulting subspace is only useful as long as it contains enough variance in the hand posture to enable successful grasping of a large variety of objects*.

Santello *et al.* have showed that two such basis vectors can account for more than 80% of the human hand grasp posture variance over a set of 57 common objects [10]. In previous work [22], we have introduced an efficient algorithm which operates in this 2-dimensional subspace to synthesize grasps of simulated objects with a human hand model. In general, postures where the hand conformed *exactly* to the surface of the target could not be found in eigengrasp space. However, by searching this subspace we can find effective pre-grasp postures that produce stable grasps when the fingers are simply closed to object contact.

Planning in a low-dimensional subspace is also appropriate in the context of cortically controlled prosthetics, where a

limited amount of control information from the operator may be present. As an example, consider the case where an external operator can only provide one-dimensional input to the grasp planner. When operating in the full-dimensional hand posture space, specifying a desired value for a single DOF has very little practical effect in determining a particular grasp. However, a single eigengrasp direction can encapsulate a significant amount of the variance required for establishing a grasp. *We posit that a low-dimensional automated grasp planner can effectively provide an interface for a user with limited control capability to perform effective grasping tasks with an upper limb robot.* With this goal in mind, we have enhanced the low-dimensional grasp planning algorithm presented in [22] to accept real-time user input along an arbitrary number of dimensions.

III. GRASP PLANNING WITH FLEXIBLE CONSTRAINTS

A. Optimization Algorithm

We approach grasp synthesis as the task of finding the combination of intrinsic and extrinsic DOFs that produces a stable grasp, according to a given grasp quality metric. We are therefore solving an optimization problem, seeking to maximize the value of grasp quality Q expressed as a function of hand posture and wrist position:

$$Q = f(\boldsymbol{p}, \boldsymbol{w}) \ , \quad \boldsymbol{p} \in \mathcal{R}^d \ , \quad \boldsymbol{w} \in \mathcal{R}^6 \qquad (3)$$

where d is the number of intrinsic hand DOFs, $\boldsymbol{p}$ represents the joint angles of the hand and $\boldsymbol{w}$ contains the position and orientation of the hand as a whole. However, when the search domain is a b- dimensional eigengrasp space, hand posture can be specified in a more compact form using the amplitudes vector $\boldsymbol{a}$. We thus take advantage of the fact that a small number of eigengrasp variables can often describe relevant variance in a much larger number of joint angles ($b \ll d$). The grasp quality function becomes:

$$Q = f(\boldsymbol{a}, \boldsymbol{w}) \ , \quad \boldsymbol{a} \in \mathcal{R}^b \ , \quad \boldsymbol{w} \in \mathcal{R}^6 \qquad (4)$$

Optimization of the quality measure is performed using simulated annealing. In our work, we use the efficient variant of the simulated annealing algorithm presented in [23]; please see [22] for an in-depth discussion of the implementation of this grasp planning method, the formulation of the quality function as well as performance evaluations.

B. Target Values and Confidence Levels

We now modify this grasp planning algorithm to meet the criterion of *adjustable cortical/computer control* that we require for our shared control system. The simulated annealing approach lends itself well to using external inputs, due to its anisotropic nature: each variable is treated independently, allowing us to control the degree to which the external "target" values are relied upon.

Consider a variable $\alpha \in [\alpha_{min}, \alpha_{max}]$ that is part of the domain of the grasp quality function Q (either an eigengrasp amplitude or a wrist position DOF). We assume that, at annealing step k, the value of this variable is α_k. The

algorithm requires that a "neighbor" value α_{k+1} be generated randomly for annealing step $k + 1$. In general, the simulated annealing algorithm finds an optimal solution if neighbors are chosen using the following guidelines: a) among early iterations of the algorithm, it allows large changes of the search variables, samples the entire domain of the optimized function; b) as the algorithm progresses, it predominantly samples increasingly smaller neighborhoods of the current solution, performing fine-grained optimization.

The neighbor generation process is seeded by sampling a **uniform** distribution $U[-1, 1]$ to obtain a random variable u called the *generating variable*. This variable is used as input to the *neighbor generating function*

$$y_k = y(u, T_k) \in [-1, 1] \qquad (5)$$

The neighbor generating function is designed such that its probability distribution satisfies criteria a) and b) presented above; in our implementation, we use the generating function introduced in [23]. After the value of y_k has been determined, the new value of α_{k+1} is generated as:

$$\alpha_{k+1} = \alpha_k + y_k (\alpha_{max} - \alpha_{min}) \qquad (6)$$

We now assume that, for the variable α, there exists a target value α_t specified by an external operator, along with a confidence level $\sigma \in [0, 1]$, with $\sigma = 0$ meaning lowest confidence and $\sigma = 1$ meaning highest confidence. The target value α_t is first normalized relative to the current value α_k as well as the total range of the variable to obtain the value of the target jump:

$$target\_jmp = \frac{\alpha_t - \alpha_k}{\alpha_{max} - \alpha_{min}} \qquad (7)$$

As can be seen from (6), a value of the neighbor function of $target\_jmp$ would create a jump such that $\alpha_{k+1} = \alpha_t$. The value of the generating variable u that is required for this jump is therefore:

$$u_t = y^{-1}(target\_jmp) \qquad (8)$$

We are now ready to compute a new value for α_{k+1}. Again, we start with the generating variable u, but instead of a uniform distribution, we use a **normal distribution of mean u_t and variance** $1 - \sigma$ (such a distribution can be obtained from the random number generator, *e.g.* using the Box-Muller transform). The generating variable u, which is now distributed mainly around the value of u_t, is used to compute the neighbor function $y(u)$ which, in turn, is used to compute α_{k+1} as discussed above.

By composing the neighbor generating function with a normal distribution centered at the desired jump value, we bias the annealing algorithm to spend more effort in the vicinity of the target. However, the stochastic nature and the main characteristics of the algorithm (large jumps early, small jumps late) are preserved. Furthermore, the algorithm can identify and refine solutions with $\alpha \neq \alpha_t$. By changing the value of the confidence level σ, the user can further influence the behavior of the algorithm and request that most of the effort is spent a smaller or larger vicinity of the target.

In particular, we note that if, at any point in the search, the confidence level is set to 1, the algorithm is guaranteed to jump to the target value in a single step. Additionally, the one-dimensional discussion presented here applies to each of the input variables; the user can establish independent target values and confidence levels in each of the dimensions of the quality function input.

IV. APPLICATIONS

We now present the results of two experiments designed to test the planning system in terms of *biomimetic synthesis*, *functionality and interactivity*, and *adjustable cortical/computer control*. As this system is being designed for use in primate cortical-control setup, the first experiment uses recorded data from a monkey to plan grasps in the observed eigengrasp space of a monkey hand. In the second experiment, we test the ability of the planner to compose synthetic grasps in near-real time using partial kinematic control provided by a human operator. For each experiment, the planning system proceeds through the following steps:

• grasp information is recorded from monkey or human operator. Recorded data includes all finger joint angles as well as wrist position and orientation.

• the recorded operator hand posture is projected into a low-dimensional eigengrasp subspace, resulting in a set of eigengrasp amplitudes.

• a partial description of the grasp, containing eigengrasp amplitudes and wrist position and orientation, is provided to the planner at different levels of confidence. We note that, instead of the complete set of 24 DOFs that can exactly identify the input grasp, the planner is only provided with a very low-dimensional approximation of the hand posture, simulating the level of information that is expected to be available through neural recordings.

• the planner searches for a form-closure grasp of the object given the input data.

Once a solution is found, we directly compare the planned grasp against the complete description of the recorded or "live" grasp. We measure the ability of the system to adapt to noisy and incomplete input, as well as the average time it requires to compose stable grasps.

A. Monkey Grasping Experiment

Monkey hand kinematic information was collected from a rhesus monkey fitted with a customized glove mounted with 23 3mm reflective markers on his right hand and lower arm. Hand motion was recorded by a Vicon motion analysis system while an industrial robot presented objects of different shapes and orientations within reach of the subject (Fig. 1). At the beginning of each trial, the monkey was trained to reach and grasp the presented object, squeezing top and bottom mounted pressure sensors. If force greater than a threshold registered on the sensors, the trial was saved and the monkey was given a water reward. Marker data was then processed in order to derive a kinematic model of the monkey hand and measure the angles of each joint of the hand during entire trials. Principal Component Analysis (PCA)
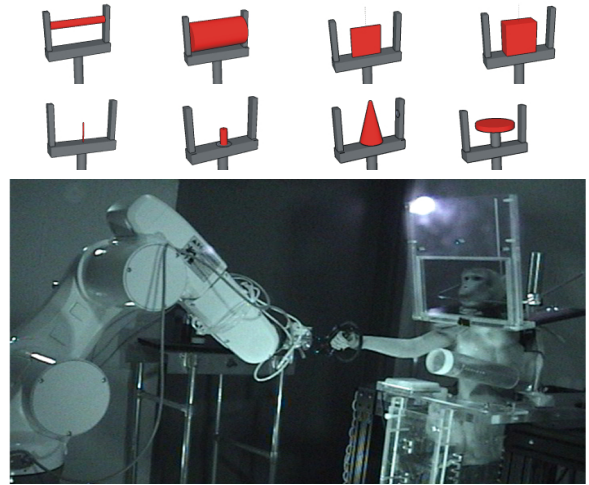

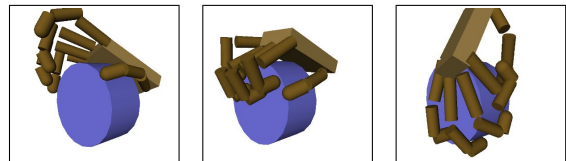Fig. 1. Experimental setup and object set used for recording primate grasps.

Fig. 2. **Left**: recorded monkey grasp used as reference pose. **Middle**: grasp planned using reference pose as input with high confidence level ($\sigma = 0.95$). **Right**: grasp planned without using reference pose.
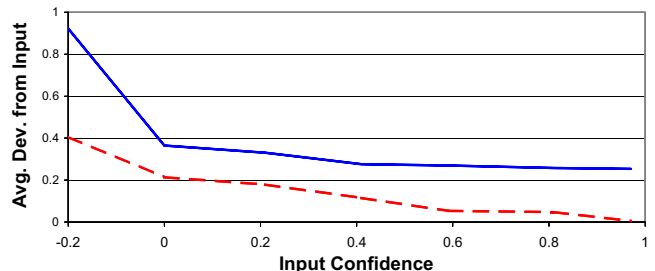

Fig. 3. Difference between planned grasps and input grasps, shown as normalized distance between the variables that define the grasp: solid line shows the difference in the amplitude of the first eigengrasp, while dashed line shows the difference in wrist orientation. A value of -0.2 was used as a starting point for the Input Confidence axes to represent the case where planning was carried out without any kind of input.

was applied to the joint angle data to find the eigengrasps that characterize monkey hand motion during grasping. PCA results showed that a 3-dimensional subspace contains 85% of the variance in hand posture, suggesting the use of 3 eigengrasp amplitudes for grasp planning experiments.

Recorded grasps were then provided as input to the automated grasp planner, as discussed above. The level of confidence in the input degrees of freedom was then varied, simulating incomplete or low-fidelity external control, and resulting grasps were compared to recorded ones. To illustrate this process, an example showing one particular grasp is shown in Fig. 2. This example is representative for the general behavior of the grasp planner: when the input pose is used with a high confidence level, the generated grasp is in the vicinity of the target posture. However, in the absence of on-line input, the planner can generate grasps at random locations around the object.

Fig. 4. Grasp planning using human data. **Left**: grasp executed by human operator. **Middle**: Low-dimensional approximation of human grasp, also affected by measurement noise. This posture, which only approximates a correct grasp, is used as input for the grasp planner. **Right**: form-closure grasp found by the planner based on the input data.

In order to quantify the robustness of the system as well as its sensitivity to the recorded input, we compared the values of generated grasp variables to ones recorded from the monkey for a large set of more than 500 planned grasps, using a large variety of input postures over the complete set of test objects. The results, presented in Fig. 3, show the mean difference in selected variables between generated and recorded grasps, normalized to the maximum range of each variable. To account for the stochastic nature of the algorithm, the results were averaged over multiple form-closure grasps for each combination of target object and level of confidence. The results show that planned hand orientation (red dashed line) was very sensitive to the confidence level specified by the user; the distance between the orientation of the planned form-closure grasps and the orientation specified as input decreased towards 0 as the confidence level approached 1. On the other hand, even with high confidence levels, the system did not find form-closure grasps where the hand conformed *exactly* to the eigengrasp amplitudes specified in the input (blue solid line). However, the system was effective in finding form-closure grasps within a given neighborhood of the specified eigengrasp input. This result can be partly explained by the fact that the geometry of the monkey hand model is not exact, so slightly different grasps will be needed to achieve form-closure.

In our experiments, the average time required to find a form-closure grasp using operator input was 3.3 seconds, approaching the speed required for real-time operation. All the experiments were performed on a commodity computer with a dual-core Intel Pentium 1.8GHz processor. Generally, computation time increased with tighter bounds placed on the given inputs. While these results appear counter-intuitive, we note that the presence of on-line input can be restrictive, requiring the planner to find a particular solution in line with operator intent, rather than opening up entire ranges of variables for sampling. In the final real-time system, situations where the planner cannot find a form-closure grasp in a given amount of time will be aborted; an important aspect of training is letting the execution of the grasp fail if the extrinsically controlled variables are too far from a form-closure solution for the automated planner to be effective.

B. Human Operator Input

In the case of grasp planning experiments using data from a human operator, we used a 2-dimensional eigengrasp

subspace presented by Santello *et al.* [10]. Unlike the monkey experiments, in this case the human operator interacted directly with our system: as the operator grasped a target object, "live" data was provided to the automated planner in real time using a Cyberglove and a Flock of Birds magnetic tracker. Due to the low-dimensional posture representation, as well as measurement noise, this data only provided an approximation of the actual grasp. A typical example is presented in Fig. 4, showing the difference between the actual grasp and the low-dimensional recording.

This system can be used interactively by presenting the planned grasps to the operator as soon as they are computed. This allows the operator to assess the connection between the example grasp and the planned result, effectively learning to grasp using very few dimensions of hand control. A more extensive set of examples of this interaction, showing the operator's hand, as well as the output of the system is presented in Fig. 5.

The responsiveness of the system was measured by attempting a large number of grasps of a target object, with different hand postures as well as approach directions. For a given target grasp, if the planner failed to provide a form-closure solution within 10 seconds, the attempt was deemed a failure and the operator selected a new approach direction. In our test, the planner succeeded in finding a solution for the operator-provided target posture in 86% of the cases (55 out of 64). The average time required to find a new form-closure grasp for a given target pose was 3.9 seconds.

V. CONCLUSIONS

In this work, we identify and address some of the difficulties facing the design of a cortically guided robotic arm and hand control system. As the number of controlled degrees of freedom in a prosthetic robot increases, more sophisticated ways of processing information from the cortex will be necessary to allow for training subjects to operate such devices and ultimately enable successful completion of tasks given noisy or incomplete control information. We propose a shared-control paradigm for overcoming these difficulties, where incomplete reach and grasp control information from the cortex is supplemented with control synthesized by an automatic grasp planning system. We have outlined a series of requirements for such a system to be effective for on-line control of a reaching and grasping robot. These requirements include the ability to plan grasps at interactive speeds, adjust to differing amounts of control input on individual degrees of freedom, and plan biomimetic grasps in the absence of input. In this paper, we describe a particular grasp planning system which has the potential to fulfill the outlined requirements by working in near-real time with flexible constraints along different control degrees of freedom. By varying the degree to which the planner will adhere to constraints, we can effectively encode the degree to which the system trusts possibly noisy or incomplete information in extrinsically controlled degrees of freedom. We have tested this method in two different environments that show its ability to meet the outlined requirements.
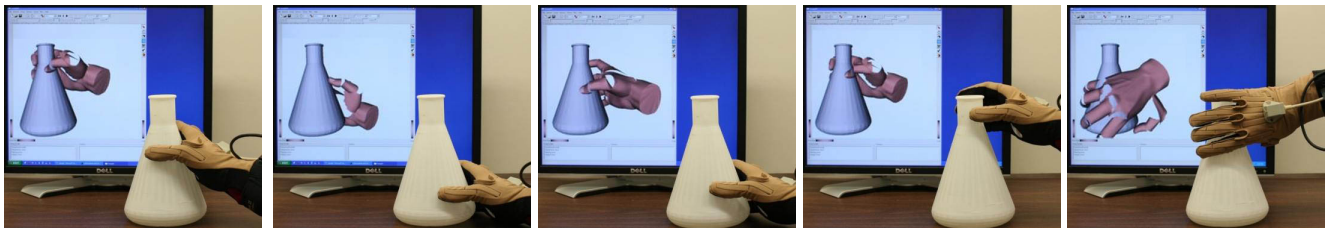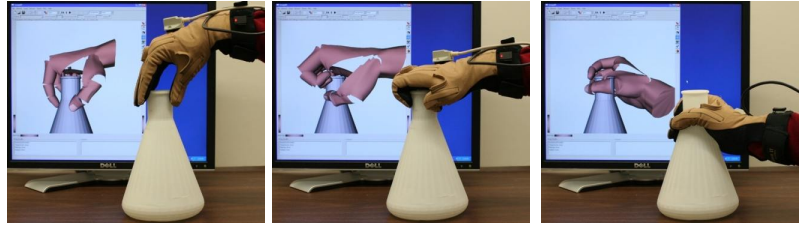
a) Grasps planned with medium confidence ($\sigma = 0.5$) in target wrist position, orientation and finger posture.
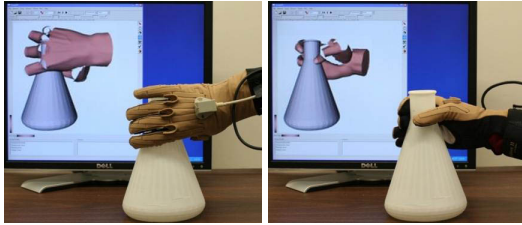
b) Grasps planned with high confidence ($\sigma = 0.95$) in target wrist orientation

c) Grasps planned with high confidence ($\sigma = 0.95$) in target finger posture, specified along two eigengrasp directions

Fig. 5. Examples of interactive grasp planning using input provided by a human operator. Operator input is recorded using a Cyberglove for finger posture and a magnetic tracker for hand location and orientation, and provided as a target pose to the grasp planner. The system reacts by outputting the first form-closure grasp that is found, usually within 2-5 seconds after the target pose is provided.

The next step in this research is to integrate the grasp planning system into actual real-time cortical control. A cortical-robotic reach control experiment is currently being modified to incorporate control of robot hand orientation and shape. To use this system as an intermediary layer between the neural information processor and the robot controller, we model the robot arm and hand in the planning system much as we have modeled the human and monkey hands in this paper. An interesting problem to be faced is determining the appropriate set of low-dimensional eigengrasps to be used if the robotic hand kinematic structure is different from its human counterpart. Current work is also aiming to improve the computational performance of the system in order to receive and integrate real-time updates of controlled variables into grasp estimates. We believe that this approach will prove a valuable component in the effort towards effective cortically controlled hand prostheses.

REFERENCES

[1] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction," *Science*, vol. 233, no. 4771, pp. 1416–1419, Sep 1986.

[2] D. M. Taylor, S. I. Tillery, and A. B. Schwartz, "Direct cortical control of 3d neuroprosthetic devices," *Science*, vol. 296, no. 5574, 2002.

[3] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–1101, Jun 2008.

[4] A. P. Georgopoulos, G. Pellizzer, A. V. Poliakov, and M. H. Schieber, "Neural coding of finger and wrist movements," *J Comput Neurosci*, vol. 6, no. 3, pp. 279–288, May-Jun 1999.

[5] K. B. Shimoga, "Robot grasp synthesis algorithms: a survey," *Intl. J. of Robotics Research*, vol. 15, pp. 230–266, 1996.

[6] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," *IEEE Intl. Conf. on Robotics and Automation*, pp. 348–353, 2000.

[7] D. Kragic, A. Miller, and P. Allen, "Real-time tracking meets online planning," in *IEEE Intl. Conf. on Robotics and Automation*, 2001.

[8] A. Tsoli and O. C. Jenkins, "Robot grasping for prosthetic applications," in *13th Intl. Symp. of Robotics Research*, Hiroshima, JP, November 2007.

[9] C. Brown and H. Asada, "Inter-finger coordination and postural synergies in robot hands via mechanical implementation of principal components analysis," in *IEEE-RAS Intl. Conf. on Intelligent Robots and Systems*, 2007, pp. 2877–2882.

[10] M. Santello, M. Flanders, and J. F. Soechting, "Postural hand synergies for tool use," *Journal of Neuroscience*, vol. 18, no. 23, 1998.

[11] M. C. Carrozza, G. Cappiello, S. Micera, B. B. Edin, L. Beccai, and C. Cipriani, "Design of a cybernetic hand for perception and action," *Biol. Cybern.*, vol. 95, no. 6, pp. 629–644, 2006.

[12] C. Cipriani, F. Zaccone, G. Stellin, L. Beccai, G. Cappiello, M. Carrozza, and P. Dario, "Closed-loop controller for a bio-inspired multi-fingered underactuated prosthesis," *IEEE Intl. Conf. on Robotics and Automation*, pp. 2111–2116, 2006.

[13] M. Serruya, N. Hatsopoulos, L. Paninski, M. Fellows, and J. Donoghue, "Instant neural control of a movement signal," *Nature*, no. 416, pp. 141–142, 2002.

[14] J. Carmena, M. Lebedev, J. R.E. Crist, O'Doherty, D. Santucci, D. Dimitrov, P. Patil, C. Henriquez, and M. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLOS Biol.*, no. 1, p. E42, 2003.

[15] H. K. Kim, S. J. Biggs, D. W. Schloerb, J. M. Carmena, M. A. Lebedev, M. A. Nicolelis, and M. A. Srinivasan, "Continuous shared control for stabilizing reaching and grasping with brain-machine interfaces," *IEEE Trans Biomed Eng*, vol. 53, no. 6, Jun 2006.

[16] M. Zecca, S. Micera, M. C. Carrozza, and P. Dario, "Control of multifunctional prosthetic hands by processing the electromyographic signal," *Crit. Rev. in Biomedical Eng.*, vol. 30, pp. 459–485, 2002.

[17] P. Afshar and Y. Matsuoka, "Neural-based control of a robotic hand: Evidence for distinct muscle strategies," in *IEEE Intl. Conf. on Robotics and Automation*, New Orleans, 2004, pp. 4633–4638.

[18] S. Bitzer and P. van der Smagt, "Learning emg control of a robotic hand: Towards active prostheses," in *IEEE Intl. Conf. on Robotics and Automation*, Orlando, 2006, pp. 2819–2823.

[19] D. Kragic, P. Marayong, M. Li, A. M. Okamura, and G. D. Hager, "Human-machine collaborative systems for microsurgical applications." *I. J. Robotic Res.*, vol. 24, no. 9, pp. 731–741, 2005.

[20] W. T. Ang, P. K. Khosla, and C. N. Riviere, "Active tremor compensation in microsurgery," in *Proc. 26th Annu. Intl. Conf. IEEE Eng. Med. Biol. Soc.* IEEE, 2004, pp. 2738–2741.

[21] M. Ciocarlie, C. Goldfeder, and P. Allen, "Dexterous grasping via eigengrasps: A low-dimensional approach to a high-complexity problem," in *Robotics: Science and Systems Manipulation Workshop - Sensing and Adapting to the Real World*, 2007.

[22] ——, "Dimensionality reduction for hand-independent dexterous robotic grasping," in *IEEE-RSJ Intl. Conf. on Intelligent Robots and Systems*, 2007.

[23] L. Ingber, "Very fast simulated re-annealing," *J. Mathl. Comput. Modelling*, vol. 12, no. 8, pp. 967–973, December 1989.