Robot Grasping for Prosthetic Applications

Aggeliki Tsoli[1] and Odest Chadwicke Jenkins[2]

[1] Department of Computer Science, Brown University, Providence, RI, 02912-1910, USA
aggeliki@cs.brown.edu
[2] Department of Computer Science, Brown University, Providence, RI, 02912-1910, USA
cjenkins@cs.brown.edu

Summary. Neurally controlled prosthetic devices capable of environmental manipulation have much potential towards restoring the physical functionality of disabled people. However, the number of user input variables provided by current neural decoding systems is much less than the number of control degrees-of-freedom (DOFs) of a prosthetic hand and/or arm. To address this **sparse control** problem, we propose the use of low-dimensional subspaces embedded within the pose space of a robotic limb. These subspaces are extracted using dimension reduction techniques to compress captured human hand motion into a (often two-dimensional) subspace that can be spanned by the output of neural decoding systems. To evaluate our approach, we explore a set of current state-of-the-art dimension reduction techniques and show results for effective control of a 13 DOF robot hand performing basic grasping tasks taking place in both static and dynamic environments.

1 INTRODUCTION

During the last few decades, there has been a lot of attention on the use of prosthetic limbs for rehabilitation purposes. The goal of these devices has been to enhance a disabled person's capability for independent living and vocational productivity by

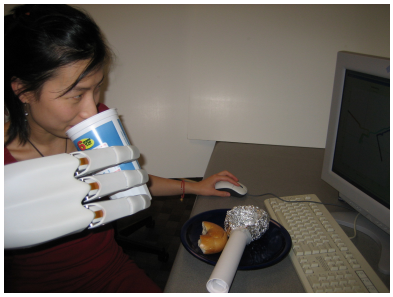


Fig. 1. Snapshot of our sparse control system driving a DLR/HIT robot hand to assist a human eat a meal. The user selects points on the 2D control space displayed on the screen using the mouse.

restoring his (or her) physical functionality. For instance, a disabled person can use a prosthetic hand to perform daily tasks, such as preparing and consuming a meal without the aid of other humans, only by using the hand to grasp and manipulate objects in his environment. The benefits from prosthetic devices become more obvious in the case of tetraplegics where the interaction of the human with his environment depends solely on the prosthetic devices ultimately controlled by the human's brain.

However, controlling a prosthetic limb, especially one with manipulating capabilities, through a humans brain signal has an inherent difficulty forming a problem that we call **sparse control problem**. The notion of the sparse control problem is that the amount of information a human can reasonably specify within a sufficiently small update interval is often far less than a robot's degrees-of-freedom (DOFs). Feasible sensing technologies, such as electroencephalogram (EEG) [4], electromyography (EMG) [20, 2, 1], and cortical neural implants [5, 16], provide a very limited channel for user input due to the sparsity and noise of the sensed signals. Specifically for **neural decoding**, efforts to decode user neural activity into control signals have demonstrated success limited to 2-3 variables with bandwidth around 15 bits/sec. On the other hand, robot hands with complex structure that will allow for performance similar to the human one can have up to 25 DOF. We can easily see that controlling the hand by mapping directly from the space of the decoded brain signal to the space of the robot hand configurations can lose a lot of useful information depending on the dimensionality of the latter space.

Considering the abovementioned limitation, previous attempts to control robot systems from low-dimensional user input have focused on low-DOF systems. Example control applications are 2D cursor control [15], planar mobile robots [4], and discrete control of 4 DOF robot arms [8, 3]. Bitzer and van der Smagt [1] have performed high-DOF robot hand control by reducing the DOFs to a discrete set of poses that can be indexed through kernel-based classification. Additionally, Michelman and Allen [13] approached this problem through combining user input with robot autonomy. The control approach in [9], where our work is based on, attempts a better mapping between the low-dimensional user input space and the high-dimensional space of robot hand configurations giving the opportunity of controlling robots with a larger number of DOFs. This mapping is performed through the use of dimension reduction techniques. Representing the functionality of the robot hand by a set of motion capture data, dimension reduction techniques try to map these data in a space with lower dimensionality preserving as much variance of the data as possible. However, viable robot hand control coming from the approach in [9] could not be shown mainly due to the noisy datasets used for training.

In this paper, we extend the work in [9] and we show that this approach can lead to successful robot grasping tasks. While the mismatch in robot pose and control dimensionality is problematic, it is likely that plausible hand configurations exist in a significantly lower dimensional subspace arising from biomechanical redundancy and statistical studies in human movement [11, 18, 12]. In this work, we examine an enhanced set of linear and nonlinear dimension reduction techniques and we highlight the ones that produce low-dimensional subspaces that facilitate the sparse control of a robot hand. In addition, we define a set of quantitative metrics to assess the

quality of the resulting low-dimensional control subspaces. These metrics are based on the amount of consistency (two dissimilar hand poses are not proximal/close in the subspace) and continuity (hand poses near in sequence should remain proximal in the subspace) over time in a low-dimensional subspace. The proposed approach is applied on the control of the DLR/HIT robot hand with 13 DOFs.

In the next section, we will give an overview of the sparse control problem as well as of the dimension reduction techniques that we are going to use. In Section 3, we will present our results and Section 4 will discuss assumptions and extensions of our work.

2 METHODOLOGY

2.1 The Sparse Control Problem

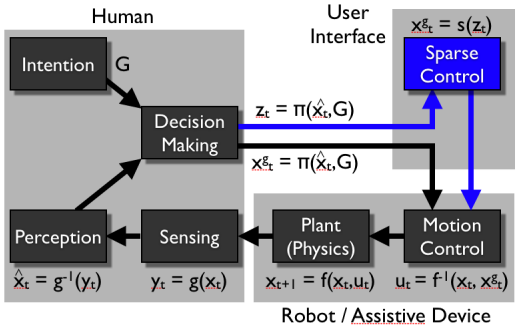


Fig. 2. A diagram of the feedback control consisting of a human decision maker perciev-ing robot state and generating control commands. Highlighted in blue, sparse control addresses the situation when the human has limited (sparse) communications with the robot, significantly less than robot's DOFs.

The essence of the sparse control problem is to estimate a control mapping $s : Z \rightarrow X$ that maps the low-dimensional user input $z \in \Re^d$ into the high-dimensional space of hand poses $x \in \Re^D$. d and D are the number of DOFs expressing human user input and robot hand pose, respectively. Shown in Figure 2, sparse control mapping sits within a feedback control loop of a human-robot team, serving as a control intermediary between a human decision maker and a robot platform. In this scenario, we assume an environment described at time t with state x_t whose physical integration is described by a (not necessarily known) function f. The robot is also assumed to have a motion control routine that produces appropriate motor commands given a desired state x_t^g (or goal robot pose) from a decision making routine. The human in the loop is responsible for making decisions to produce x_t^g, based on an internal policy π that relies on the human intensions G and the perception $\hat{x}_t$ of the state at time t coming from the partial observations y_t. The user

interface is the medium through which the human user specifies the desired pose x_t^g to the robot platform. The problem addressed in sparse control pertains to when the human cannot or is overly burdened in specifying the desired pose x_t^g for the robot. Specifically, we assume that $D >> d^*$. Also, $x_t^g \in \Re^D$ and $z_t^* \in \Re^{d^*}$ where z_t^* is the optimal amount of information a given human can communicate to a robot. Obviously, z_t^* is a moving limit that varies for each individual human user.

The estimation of the mapping s is founded upon the assumption that the space of plausible hand poses is intrinsically parameterized by a low-dimensional manifold subspace. We assume each hand pose achieved by a human is an example generated within this manifold subspace. It is given that the true manifold subspace of hand poses is likely to have dimensionality greater than two. With an appropriate dimension reduction technique, however, we can preserve as much of the intrinsic variance as possible. As improvements in decoding occur, the dimensionality of the input signal will increase but we will still leverage the same control mapping.

In the remaining of this section, we overview various techniques that approximately cover the current state-of-the-art in dimension reduction.

2.2 Dimension Reduction Overview

In addressing the sparse control problem, we explore the use of six dimension reduction methods for constructing control mappings: Principal Components Analysis (PCA) [7], geodesic Multidimensional Scaling (Isomap) [17], fast Maximum Variance Unfolding (FastMVU) [19], Hessian LLE (HLLE) [6], temporally-windowed Isomap (Windowed Isomap), and Spatio-Temporal Isomap (ST-Isomap) [10]. The first method, PCA, constructs a linear transform A of rank 2 about the mean of the data μ:

$$z = s^{-1}(x) = A(x - \mu) \tag{1}$$

PCA is equivalent to performing multidimensional scaling (MDS) [7] with pairwise Euclidean distance. MDS is a procedure that takes a full matrix M, where each element $M_{x,x'}$ specifies the distance between a datapair, and produces an embedding that attempts to preserve distances between all pairs. MDS is an optimization that minimizes the "stress" E of the pairwise distances:

$$E = \sqrt{\sum_x \sum_{x'} (\sqrt{(f(x) - f(x'))^2}) - M_{x,x'})^2} \tag{2}$$

The second method, Isomap, is MDS where shortest-path distances are contained in M as an approximation of the geodesic distances between any pair of points:

$$M_{x,x'} = \min_p \sum_i M'(p_i, p_{i+1}) \tag{3}$$

where M' is a sparse graph of local distances between nearest neighbors and p is a sequence of points through M' indicating the shortest path. Isomap performs nonlinear

dimension reduction by transforming the data such that the geodesic distances along the underlying manifold become Euclidean distances in the embedding. A canonical example of this transformation is the "Swiss roll" dataset, where input data generated by a 2D manifold is contorted into a roll in 3D. Given sufficient density of samples, Isomap is able to flatten this Swiss roll data into its original 2D structure, within an affine transformation.

FastMVU attempts to find a set of low-dimensional coordinates z of the input points x by approximating the Euclidean distances between points z with the global geodesic distances between points x. This is done by constructing a convex optimization over both z and the dimension m of the embedding space. In the case where m is not the desired dimension (e.g. 2), the solution of the convex optimization is first projected to the desired low-dimensional space. Then, the resulting solution is used as a starting point for gradient descent on a function very similar to the "stress" function of MDS.

PCA and Isomap assume that the input data are unordered (independent identically distributed) samples from the underlying manifold, ignoring the temporal dependencies between successive data points. However, human hand motion has a sequential order due to the temporal nature of acting in the real world. Windowed Isomap accounts for these temporal dependencies in a straightforward manner by windowing the input data $\tilde{x}_i = x_i \dots x_{i+w}$ over a horizon of length w.

ST-Isomap further augments the local distances between the nearest neighbors of windowed Isomap. In particular, it reduces the distance between the local neighbors that are spatio-temporal correspondences, i.e. representative of the k best matching trajectories, to each data point. Distances between a point and its spatio-temporal correspondences are reduced by some factor and propagated into global correspondences by the shortest-path computation. The resulting embedding attempts to register these correspondences into proximity. In this respect, ST-Isomap is related to time-series registration using Dynamic Time Warping [14]. The distinctions are that ST-Isomap does not require segmented or labeled time-series and uses a shortest-path algorithm for dynamic programming.

All of these methods assume the structure of the manifold subspace is convex. Hessian LLE [6] avoids this convexity limitation by preserving local shape. Local shape about every data point is defined by an estimate of the Hessian $H_{x,x'}$ tangent to every point. $H_{x,x'}$ specifies the similarity of x to its neighboring points x'; non-neighboring points have no explicit relationship to x.

3 RESULTS

3.1 Experimental Setup

The application of sparse control for interactive control of the DLR/HIT hand is illustrated in Figure 3. Instead of using actual recordings from the brain of a human, decoding them and controlling the robot hand, we assumed that we already have a decoded version of his brain signal represented as a 2D signal and displayed on a 2D screen. Our assumption is enhanced by the fact that efforts in neural decoding

[8, 16, 15] have shown that disabled individuals physically incapable of moving 2D mouse can drive a 2D cursor from neural activity. This 2D signal is the result of the dimension reduction techniques on high-dimensional human hand motion capture data (x). In order to control the robot hand, the user clicks on points on a 2D screen (z) that correspond to instances of the decoded brain signal and their corresponding robot hand configuration is sent as a motor command to the robot hand. After the user observes the result of his action, he selects another point on the screen in order to achieve a predefined task.

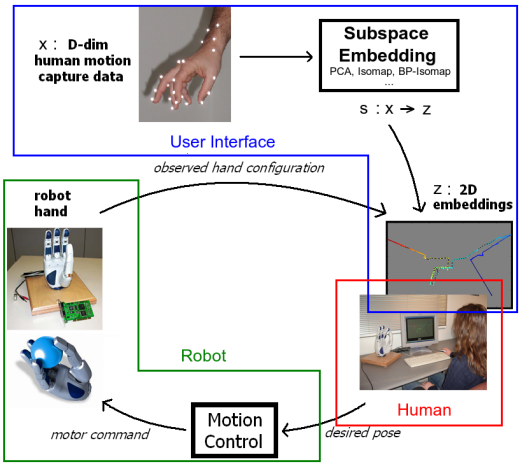
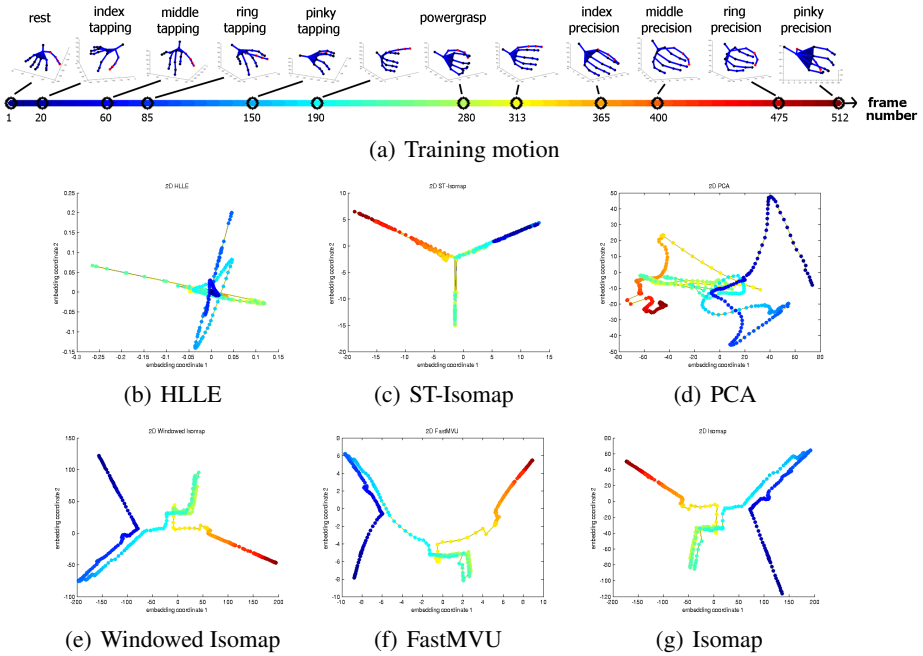


Fig. 3. Diagram for hand control by the user using human hand motion capture data for training

Our sparse control and subspace embedding systems were implemented in Matlab. Mex executables formed the bridge between our Matlab implementation and the C++ interface provided by DLR for the control of the robot hand. The robot hand used in the experiments was the DLR/HIT anthropomorphic robot hand, constructed with 4 fingers, 13 DOFs and a form factor of roughly 1.5 times the size of a human hand.

The human hand motion sequences that were used for training were captured by a Vicon optical motion capture system. The high-dimensional pose space is created by considering each frame of the mocap data as a point in this space. More specifically, each high-dimensional point is defined as the 3D endpoints of the fingers in the hand's local coordinate system, resulting in a 12-dimensional vector. Because the DLR hand has only 3 fingers and a thumb, data for the fifth human finger (pinky) is omitted. Joint angles used for motion control of the hand were computed using an inverse kinematics procedure that minimized the distance between each finger's endpoint position with respect to the knuckle of the finger.

The datasets we used to evaluate our approach were the Basic Motions (BM) and the Special Motions (SM) datasets. The BM dataset (Figure 4a) consists of finger



(a) Training motion

(b) HLLE (c) ST-Isomap (d) PCA

(e) Windowed Isomap (f) FastMVU (g) Isomap

Fig. 4. ***Basic Motions (BM) Dataset*****: (a) training motion sequence and embeddings from (b) HLLE, (c) ST-Isomap, (d) FastMVU, (e) PCA, (f) Windowed Isomap, (g) Isomap**

tapping motions, powergrasps and precision grasps using all the fingers. The SM dataset (Figure 5a) consists of a subset of the BM dataset and specialized motions like pointing with the index finger, finger abduction, fingers forming the OK sign.

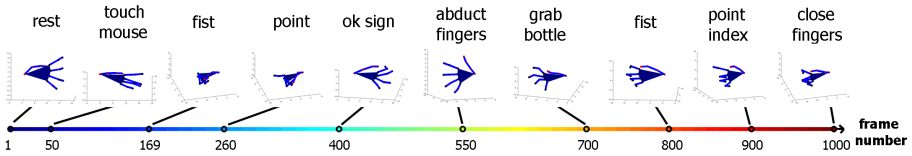
3.2 Dimension Reduction Analysis

The first step towards evaluating our approach is to assess the quality of the embeddings derived using each of the six methods mentioned before. Figures 4c-h, 5c-h show the embeddings of the several dimension reduction techniques applied on the BM and SM dataset respectively.

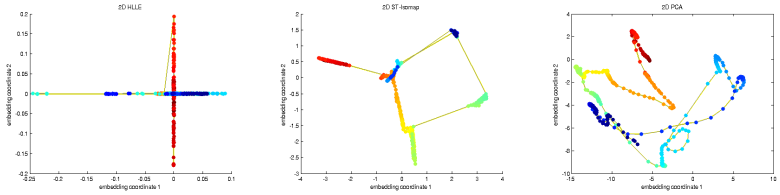
The quality of the embeddings for the BM, SM datasets is shown in Tables 1 and 2. More specifically, the quality criteria are based on the amount of consistency and continuity over time:

- ***Consistency*** suggests that two poses that are dissimilar in the high-dimensional space should not be proximal in the low-dimensional space and our error metric for consistency evaluation is the following:

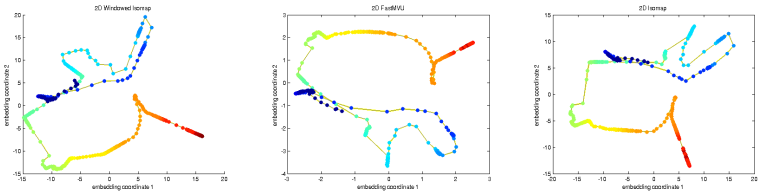
$$E = \sum_{i} \sum_{j:y_j \notin N(y_i)} g(i,j), \text{ where } g(i,j) = \begin{cases} 1 & \text{, if } j \in N(x_i) \\ 0 & \text{, else} \end{cases}$$



(a) Training motion



(b) HLLE (c) ST-Isomap (d) PCA



(e) Windowed Isomap (f) FastMVU (g) Isomap

Fig. 5. ***Special Motions (SM) Dataset*****: (a) training motion sequence and embeddings from (b) HLLE, (c) ST-Isomap, (d) FastMVU, (e) PCA, (f) Windowed Isomap, (g) Isomap**

where $N(y_i)$ is the set of the k nearest neighbors of point i in the high-dimensional space and $N(x_i)$ is the set of the k nearest neighbors of point in the low-dimensional space.

- ***Continuity*** over time suggests that poses that are near in time in the training motion sequence should remain proximal in the low-dimensional space. The error metric for continuity evaluation is

$$E = \sum_i \sum_{j=i+1} g(i,j), \text{ where } g(i,j) = \begin{cases} 1 & \text{, if } j \notin N(x_j) \\ 0 & \text{, else} \end{cases}$$

As we can see from the tables, the methods that performed best were Isomap and FastMVU. Windowed Isomap seems to be a very good approximation to Isomap. The performance of the rest of the methods, although not easily revealed from the metrics, indicates a lack of consistent local continuous and spatio-temporal structure. Given more repetitions of the hand mocap training trials, we would expect better embeddings for ST-Isomap and Hessian LLE.

Method	*Consistency Error*	*Continuity Error*
HLLE	0.5168	0.2505
ST-Isomap	0.7556	0.6712
PCA	0.2839	0.1037
Windowed Isomap	0.3149	**0.0763**
FastMVU	0.2388	0.0900
Isomap	**0.2195**	0.0861

Table 1. Consistency and Continuity Errors for the hand poses embeddings of the Basic Motions (BM) datasets

Method	*Consistency Error*	*Continuity Error*
HLLE	0.6990	0.4695
ST-Isomap	0.5897	0.3944
PCA	0.2804	0.1281
Windowed Isomap	0.2719	**0.0761**
FastMVU	**0.2473**	0.1171
Isomap	0.2634	0.1251

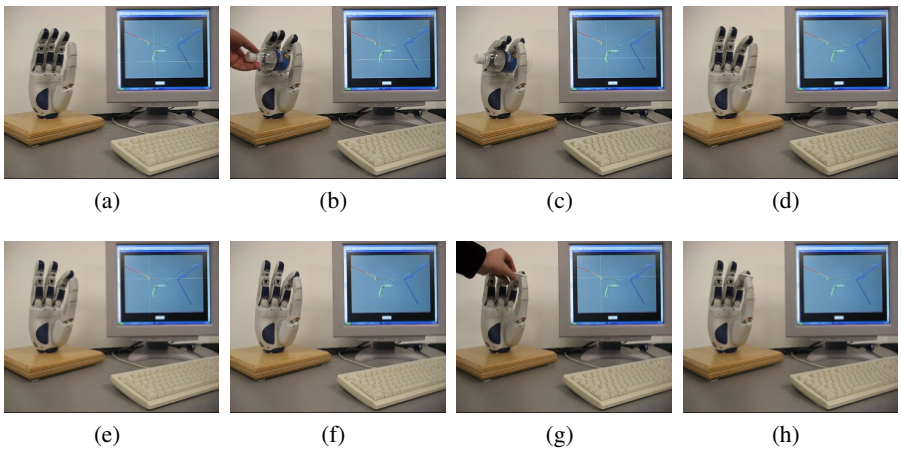
Table 2. Consistency and Continuity Errors for the hand poses embeddings of the Special Motions (SM) datasets

3.3 Performance trials

In order to show that our approach is viable for robot hand control we performed several grasping tasks in both static and dynamic environments. The embedding we used was the Isomap embedding of the BM dataset and the results are shown in Figures 6, 7, 8. In Figure 6 we performed two interactive tasks using our sparse control system: power grasp of a water bottle and precision grasp of an eraser between the thumb and index fingers. Figures 6(a-d) show the different phases of the power grasp before, during and after performing the actual grasp. The phases of the precision grasp are illustrated in figures 3(e-h). In Figure 7 we performed grasping of a box in a dynamic environment. While the box was oscillating near the place of the hand, finger tapping motions were used to position the box inside the palm. Power grasps and precision grasps were used for its actual grasping. Finally, in Figure 8, highlighting the ultimate goal of our work for prosthetic applications, we attempted to simulate the grasping actions taking place in a dining scenario: grasping a spoon in order to eat, pulling the lid of a bottle, pouring soda into a cup, drinking from the cup. As the Figures 6, 7, 8 and the accompanying video show, control of the robot hand is performed in a very reasonable manner. The objects are successfully grasped and postures of the robot hand resemble the postures of a human hand during grasping.

4 Conclusion

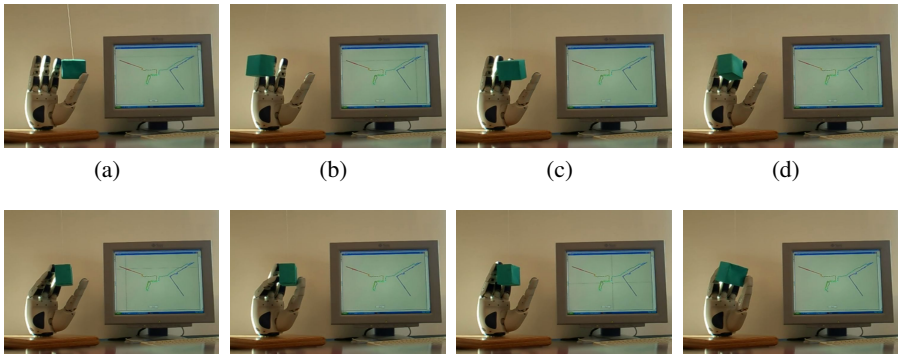
In this paper, we have attempted to address the problem of sparse control of a high-DOF robot hand with an emphasis towards leveraging current neural decoding capa-



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 6. Performances of interactive sparse control for power grasping on a water bottle (top, a-d) and precision grasping of a small eraser (bottom, e-h).

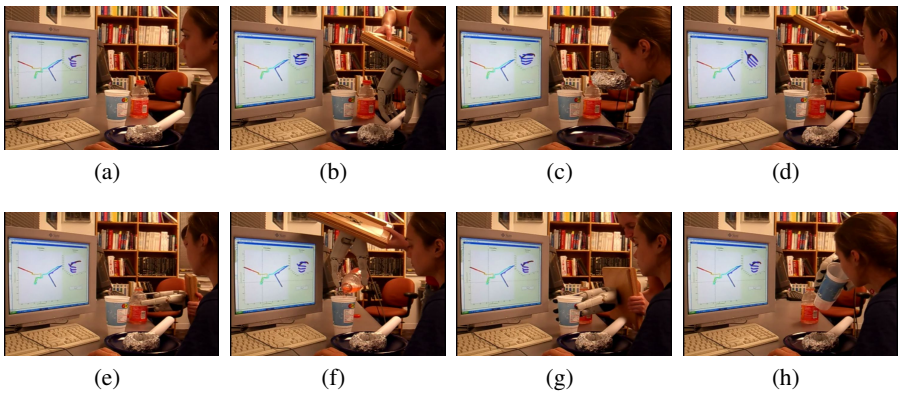


(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7. Performance of interactive sparse control for grasping an oscillating box. Finger tapping motions are used to position the box inside the palm; power grasps and precision grasps are used for its actual grasping.

bilities. We have taken a data-driven approach to this problem by using dimension reduction to embed motion performed by humans into 2D spaces. One of the next steps of our work is to test our system in the case where a robot arm is attached to the robot hand as well as use our system in combination with currently existing systems that perform 2D cursor control through decoding neural activity. We believe that these experiments will give us useful insight on the advantages and limitations of our method and will be the best way to test how robust to neural signal noise our system is.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 8. Performance of interactive sparse control for a real life scenario: the robot hand assists a human eat her meal. (a) The 2D control space and the current pose of the hand are displayed on the screen, (b) the robot hand grasps the spoon and (c) it feeds the human. (d) The robot hand pulls the lid out of the bottle, (e) grasps the bottle, (f) pours soda into the cup, (g) grasps the cup and (h) approaches the cup to the human.

5 Acknowledgements

This work was supported in part by ONR Awards N00014-07-1-0141 and N00014-06-1-0692 and NSF Award IIS-0534858. The authors thank Michael Black, Panagiotis Artemiadis, Wenjin Zhou and Doria Jianu for useful discussions and their contribution in the experiments. The authors also thank Greg Shaknarovich and the Metamotion Inc. for motion capture data and Schunk Inc., DLR, and Patrick van der Smagt for their assistance with the DLR/HIT robot hand.

References

1. S. Bitzer and P. van der Smagt. Learning EMG control of a robotic hand: towards active prostheses. In *IEEE International Conference on Robotics and Automation*, pages 2819–2823, Orlando, May 2006.
2. B. Crawford, K. Miller, P. Shenoy, and R. Rao. Real-time classification of electromyographic signals for robotic control. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 523–528, Pittsburg, PA, July 2005.
3. E. Crawford and M. Veloso. Learning to select negotiation strategies in multi-agent meeting scheduling. In *Working Notes of the AAAI Workshop on Multiagent Learning*, 2005.
4. J. del R. Millan, F. Renkens, J. Mourino, and W. Gerstner. Brain-actuated interaction. *Artif. Intell.*, 159(1-2):241–259, 2004.
5. J. Donoghue, A. Nurmikko, G. Friehs, and M. Black. Development of neural motor prostheses for humans. *Advances in Clinical Neurophysiology (Supplements to Clinical Neurophysiology)*, 57, 2004.
6. D. Donoho and C. Grimes. Hessian eigenmaps: Locally Linear Embedding techniques for high-dimensional data. *Proc. National Academy of Sciences*, 100(10):5591–5596, 2003.

7. R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.
8. L. Hochberg, M. Serruya, G. Friehs, J. Mukand, M. Saleh, A. Caplan, A. Branner, D. Chen, R. Penn, and J. Donoghue. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, 442:164–171, June 2006.
9. O. C. Jenkins. 2D subspaces for sparse control of high-dof robots. In *Intl. Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2006)*, New York, NY, USA, Aug-Sep 2006.
10. O. C. Jenkins and M. J. Matarić. A spatio-temporal extension to Isomap nonlinear dimension reduction. In *The International Conference on Machine Learning (ICML 2004)*, pages 441–448, Banff, Alberta, Canada, Jul 2004.
11. J. Lin, Y. Wu, and T. Huang. Modeling the constraints of human hand motion. In *IEEE Workshop on Human Motion*, 2000.
12. C. R. Mason, J. E. Gomez, and T. J. Ebner. Hand synergies during reach-to-grasp. *The Journal of Neurophysiology*, 86(6):2896–2910, December 2001.
13. P. Michelman and P. Allen. Shared autonomy in a robot hand teleoperation system. In *Proceedings of the 1994 Conference on Intelligent Robotics Systems*, pages 253–259, Munich, Germany, September 1994.
14. C. S. Myers and L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. In *The Bell System Technical Journal*, pages 1389–1409, September 1981.
15. M. Serruya, A. Caplan, M. Saleh, D. Morris, and J. Donoghue. The BrainGate pilot trial: Building and testing a novel direct neural output for patients with severe motor impairment. In *Society for Neuroscience Annual meeting*, October 2004.
16. D. Taylor, S. H. Tillery, and A. Schwartz. Information conveyed through brain control: Cursor versus robot. *IEEE Transactions on Neural Systems and Rehabilition Engineering*, 11(2):195–199, June 2003.
17. J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
18. E. Todorov and Z. Ghahramani. Analysis of the synergies underlying complex hand manipulation. In *Intl. Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4637–4640, San Francisco, CA, September 2004.
19. K. Weinberger, F. Sha, Q. Zhu, and L. Saul. Graph Laplacian methods for large-scale semidefinite programming, with an application to sensor localization. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19*. MIT Press, Cambridge, MA, 2007.
20. M. Zecca, S. Micera, M. C. Carrozza, and P. Dario. Control of multifunctional prosthetic hands by processing the electromyographic signal. *Critical Reviews in Biomedical Engineering*, 30(4-6):459–485, 2002.