In Proceedings of the *International Conference on Development and Learning* 2006

Imitation learning for reaching and grasping in virtual environments

Nigel Singh and Emanuel Todorov

Cognitive Science Department

University of California San Diego

9500 Gilman Drive

La Jolla, CA 92093-0515

Correspondence to: todorov@cogsci.ucsd.edu

ABSTRACT

In addition to trial and error learning, learning from observed teacher generated examples may greatly facilitate acquisition of motor skills. I present results for an imitation learning reaching and reaching-to-grasp controller for a Virtual Reality environment with simulated physics. The controller follows the unforced dynamics of a localized vector field around the human example data with no specified contact points needed. Trajectory, speed, and aperture profiles compare well with human data.

I. INTRODUCTION

A. Motivation

Effective and efficient motor control is one of the most complex and important skills we learn throughout our lifetime as we go from uncoordinated infants to skillful tool users and makers. In addition to trial and error learning, the acquisition of these motor skills may be greatly facilitated by learning from observed teacher generated examples.

Computational approaches to learning from examples vary. Early methods involved parsing movements into a set of if-then rules to create a finite state machine controller. These methods were highly limited by the computational power available in the early 1980s, and movement examples consisted primarily of a human pushing the robot through certain movements and then using the proprioceptive information gathered from on-board sensors to extract the if-then rules. Recently, researchers exploiting the increase in computational power have utilized less purely symbolic paradigms which include artificial neural networks, fuzzy logic, statistical learning, dynamical systems attractors, and biologically motivated architectures. [13, 14, for a review see 2] Our approach in this paper brings together several core methodologies from seemingly unrelated literatures to develop a controller which learns to reach and grasp from human examples.

B. Computer Graphics Approaches

A source of promising methods for imitation learning comes from the computer graphics world. (e.g. [9, 5]). Many of the same problems have to be solved when either making a biologically inspired motor controller or an animated avatar that has realistic biological movement and interaction with the animated environment. One such problem in motor control is the issue of high redundancy. If there are more degrees of freedom in the limb than in the space in the task space, there are infinitely many postures corresponding to each limb location and orientation. There are also infinitely many paths to the same goal whether or not there are excess degrees of freedom. How to construct a biologically plausible computational solution in such a redundant system has not been solved. The computer graphics community has long realized that making use of captured human movement avoids this problem of high redundancy by taking advantage of the most probable movements that humans make and not bother with all possible movements. This approach by essentially "stitching together" previously recorded discretized movements greatly aids in the production of novel animated motion, and the techniques available for the utilization of these example motions are currently quite sophisticated. Unfortunately, attempts at imitation learning in the motor control community have not been as successful. This lacklustre performance arises partly because of the difficulty of controlling movement in a continuous space and in modelling noisy interactions in the real world. In animation, there is less of this uncertainty since correct physics for object manipulation is not as important as what appears visually plausible. Only recently have the graphics approaches gone beyond just an interpolation of the motion capture movements to considering contact constraints with the environment.

Lee et al. [9] offers an architecture in which contacts are managed for control of a virtual avatar using a database of human motion capture data. The idea is to preprocess this motion database for flexibility in behavior and efficient searching. Flexible behavior is enabled by creating connecting transitions between frames where good matches in pose, velocity, and contact state exist. Efficient searching for a plausible sequence of frames to generate the desired movement is made possible by pruning transitions with relatively low probability, and also by the clustering of frames. A movement is found by maximizing the joint probability of a cluster sequence and frame sequence. This paper takes a similar clustering approach for reaching control but with the advantage that control is done in continuous space.

C. Dynamical Systems Approaches

Another approach similar to our model of imitation learning is a dynamical system with attractors around example trajectories [13]. The specific features of the landscape such as point attractors and limit cycles are first initialized by curve fitting the example trajectories and then refined through reinforcement learning. A PD controller is used for tracking of these trajectories. This imitation-reinforcement learning scheme achieves very impressive results such as robotic demonstrations of drumming and swinging a tennis racket. The drawback of such approaches is that a different dynamical system is used for each degree of freedom to guarantee monotonic global convergence. Independence of each degree of freedom in the human body is not an accurate assumption. In contrast, the model in this proposal does not make this independence assumption and should automatically discover couplings of degrees of freedom.

In this paper we combine the strengths of the computer graphics and dynamical systems approaches. Both the teacher examples and computer controller actions are in a virtual reality environment with simulated physics. This enables exact environment state information and solves the correspondence problem by having both the human and the computer controller use the same end effectors for manipulation. Also, the vector field is constructed around nearest neighbour example trajectories found in principal components space rather than the original Euclidean space.

D. Grasping Models

Grasping is one of the most complex motor task we learn (see [20] for a review), evidenced by the high expectation for a child to break a fragile object. Early theories of grasping divided the task into two independent visuomotor channels, one for controlling the transport of the hand, and the other for controlling the actual grip [7]. This view was thought to be convenient because it corresponds nicely to distinct anatomical structures at the levels of joints, muscles, and corticospinal connections. [16], however, showed that grip size depends on both intrinsic and extrinsic properties of the object, as well as the transport component. Thus, no general independence can be claimed for the two proposed visuomotor channels. Also, movements of the distal joints in the fingers are made by the activation of both the distal intrinsic muscles of the hand and the proximal extrinsic muscles in the lower arm. In Jeannerod's classic view, the definition of the transport component is difficult and the usual work around defines this component at the wrist. The wrist, however, is not transported all the way to the object, but to a couple inches away at a location that depends on the wrist joint angle.

Smeets et al. [16] also demonstrated that grasping can be thought of as nothing more than pointing with the thumb and finger toward selected positions on the surface of the object using a minimum jerk trajectory [4]. This model successfully shows the curved finger path trajectories which approach the object's contact points perpendicularly to the surface plane. It also predicts that the maximum grip aperture occurs in the 2nd half of the movement and occurs later for larger objects. Smeets extended his model to match human data for grasping under perturbations using an abort-replan scheme [16]. Unfortunately, this minimum jerk model does not address the issue of how to pick object contact points, nor does it attempt to model interactions with all the surfaces of the finger and palm. The imitation learning framework in this proposal attempts to solve these problems indirectly.

II. Experimental Set-Up

A. Hardware/Software Components

We have developed a virtual reality environment (VRE) using Liquid Crystal (LC) shutter glasses and a reflective mirror set-up. The monitor outputs an effective refresh rate to each eye of 60Hz, which is synchronized to the glasses. The VRE is constructed using an OpenGL 3D graphics library with physical interactions modelled using the Open Dynamics Engine (ODE). ODE is an open source, high performance library for simulating rigid body dynamics. It is a fully featured, stable, mature and platform independent with an easy to use C/C++ API. It has advanced joint types and integrated collision detection with friction for a large number of object shape primitives. Almost arbitrarily complex virtual objects can be created through composites combining these shape primitives or a molded mesh of triangles. ODE's Berkeley Software Distribution (BSD) license allows developers to use the source code free of charge in commercial and research products.

B. Methods for User Interaction and Immersion

For interaction, Polhemus magnetic sensors, capable of data collection rates up to 240 Hz, are attached to finger tips of the user at the thumb, index, and middle finger, and at the back of the hand by use of a glove. This attachment is made possible by embedding the sensors in Orthoplast, moldable rubber-based sheets used for splints. The Orthoplast is shaped for a custom fit on each fingernail. Medical tape is used to secure these Orthoplast molds onto the fingers. The weight of the sensor cords would most likely change subjects motor control strategy, so the cords are held secure at the wrist by a wrist brace. See Fig. 1.

The user interacts with the world using three spheres which coincide with the location of the thumb, index, and middle fingertips. Thus, the three virtual finger pads correspond to these three human effectors. Currently our VRE only supports experiments that utilize the human finger tips for grasping and reaching experiments but can be extended to use full hand interactions.

Several tricks are used to aid the feeling of immersion. Head tracking allows the perspective viewpoint of the VE to change based on the user's head movement. This is accomplished by attaching a Polhemus RX2 sensor to the LC glasses, passing the sensor measurements through a low pass filter to eliminate high frequency noise introduced by the monitor, and calculating the vector offset from the sensor to each eye. Low pass filtering head movements does not introduce noticeable lag because typical head motions are slow. For the finger sensors, the noise is less obvious since the fingers are further away from the monitor and, unlike the head tracker, this noise does not translate to a change in viewing perspective which would disorient the user's visual system. However, the lag between the finger sensor movement and the virtual finger is noticeable, especially at large velocities, resulting in the need for some kind of position prediction algorithm which would ideally take advantage of the human visual system for perceptual realism. Our solution combines some mild low pass filtering of positions along with some prediction of where the sensor is moving. For prediction, the current velocity of the sensor is passed into an exponential weighting function where small velocities are not exaggerated but larger velocities are exponentially exaggerated. No prediction or filtering of the virtual effector is done if it is currently in contact with a virtual object because these movements typically have small velocities and any modification of the effector's motion will likely be detected due to increased user attention to that area of the workspace. Therefore, small visual lags for low velocity movements and movements when touching a virtual object are maintained, while visual lags for high velocity movements are significantly decreased. Under normal experimental conditions the user's eye saccades to the position of the virtual fingers when doing a rapid movement. Human visual perception is hindered while performing such quick eye movements [15]. During quick body movements the direction of motion is unlikely to change significantly, thus aiding in the perceptual realism of the prediction technique. Even during the worst case scenario of fast circular movements lags are almost imperceivable. To further aid an immersive feeling, the virtual fingers are placed at a vector offset from the sensor location to align the virtual finger-pad surface with the user's physical finger-pad surface.

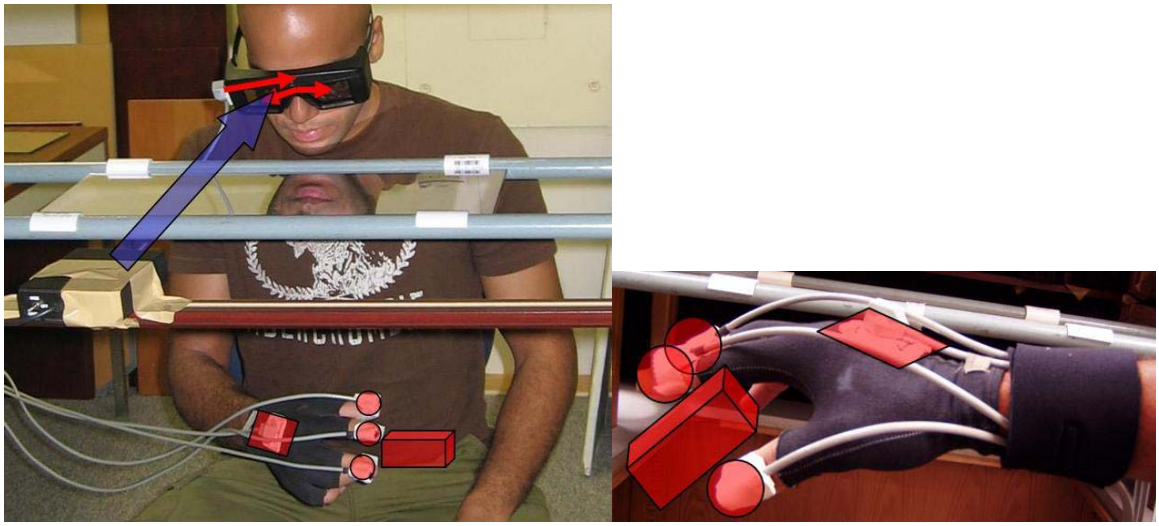

Fig. 1: (Left) Reflective mirror Virtual Reality set-up showing Polhemus sensors, eye-offsets for head-tracking, communication link of the liquid crystal glasses with the computer, and the virtual effectors

superimposed on the subject's hand. Interaction is disabled for the palm object and is used primarily to help the subject visually see his hand orientation. (Right) Enlarged image of the hand effectors grasping a box. During experiments subjects can only see the virtual effectors and not their own hand.

Along with a rigorous calibration process the above techniques result in a powerful and general experimental apparatus to record a subject's movements while having complete experimenter control of all the physics of interaction. Such an environment with exact state information allows the experimenter and control designer to focus purely on the human strategy without worrying about real-world uncertainties. The correspondence problem is also solved here because any artificial agent that learns from imitating the human strategy is not required to do any mapping from the effectors of the teacher to its own effectors because the virtual fingers used in both cases are the same.

C. Description of Tasks

Subjects were Cognitive Science graduate students naïve to interacting with virtual environments. Data from five subjects were recorded and used separately to feed examples to the controller. Performance of the controller did not change significantly from subject to subject, so the results reported are from using the examples generated from a single subject.

One of the simplest tasks for imitation learning is reaching towards a target. Subjects were instructed to use only their extended index finger to reach towards and slightly touch and stop at the location of a spherical target with as high accuracy as possible. Once the target is touched it disappears for 1 second and then reappears at a uniform random location within a specified work area. During this 1 second pause subjects are instructed to remain as still as possible and then reach toward the new target location. See Fig. 2.

In the second task, the subjects are instructed to reach to grasp a spherical target with their thumb and index finger. When both fingers touch the target, it disappears for 1 second and then reappears at a uniform random location within a specified work area. During this 1 second pause subjects are also instructed to remain as still as possible and then reach to grasp the new target.

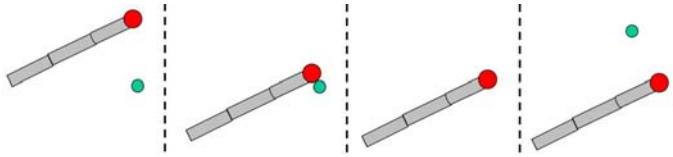

Fig. 2: Graphic depicting the sequence of motions for reaching task – (1) target appears (green); (2) subject reaches toward and touches target with virtual index finger pad (red); (3) target disappears for 1 second; (4) target reappears at a uniform random location in a specified work area. Note that this task is done in three dimensions.

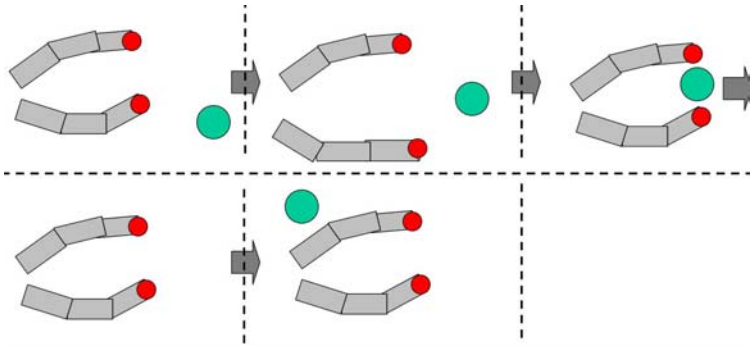

Fig. 3: Graphic depicting the sequence of motions for reach to grasp task – (1) target appears (green); (2) subject reaches toward and grasps target with virtual thumb and index finger pads (red); (3) target disappears for 1 second; (4) target reappears at a uniform random location in a specified work area. Note that this task is done in three dimensions.

III. IMITATION LEARNING ALGORITHM

A human generated example, **E**, is a sequence of frames, **F**. Each **F** consists of relevant state information, **S**, and the control, **U,** which is simply velocity in our implementation. State information is a list of the positions of the finger pads ($\mathbf{S}_{fingers}$), objects ($\mathbf{S}_{objects}$), and what contacts, **C**, currently exist. Currently only the index finger pad will be considered for state information about the hand.

In this approach to imitation learning of the human strategy a clustering approach is used where a velocity vector field is constructed based upon the closest n-dimensional cluster in the motion capture database to the current frame. The offline processing done is as follows – (A-1) for each **F** in **E** transform finger location into the reference frame of the current target; (A-2) reduce the dimensionality of these transformed frames by performing principal components analysis (PCA) on the union of all motion sequences and keep the dimensions which explains 95% of the movement variance; (A-3) save a list of initial frames for all motion sequences. The offline processing time is lengthy, but is necessary for fast online control. Transforming the frames to the reference frame of the target allows the system of the target and finger to be translated anywhere in the workspace and still have a similar control strategy. In essence, the controller is now always reaching toward the origin. In step A-2, reducing the dimensionality of the observed examples transforms the database into a more natural reference frame where the axis correspond to synergies of the hand and arm and also provides for more efficient nearest neighbour look-ups. Step A-3 is used for speed up of the online controller as will be now discussed. The online controller algorithm is depicted in Fig. 4. Note that the nearest neighbours are always found in PCA space which is a more powerful reference frame to take advantage of inherent synergies of the human body and for more efficient nearest neighbour searches. To bias the controller to exhibit bell-curve velocities typical of human reaching movements, B-1 to B-5 ensures that the vector field is constructed with movement sequences that started at a similar distance from the target location.

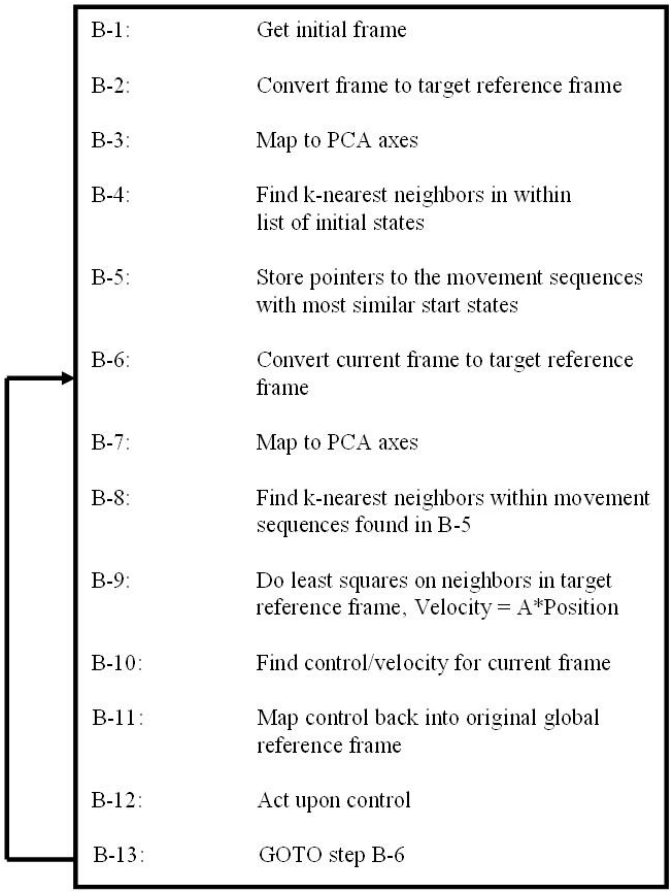

Fig. 4: Online control algorithm that follows a vector field constructed from the cluster of nearest frames.

IV. RESULTS

A. Imitation Reaching

Fig. 5 shows reaching examples from one subject in the reference frame of the target and average speed profiles for these reaching trajectories. The speed profiles are normalized by time between 0 to 1, where 0 is the start of the movement and 1 is the end of the movement when the subject touches the target. These normalized speed profiles are then averaged into 5% interval bins. The typical bell-curve shaped profiles are observed with an interesting difference. In traditional reaching experiments to real targets ending velocities are near zero because of impacting with a real object. In our VRE even though subjects were instructed to have as close to zero velocity as possible when touching the target, this velocity is not zero. This is due to lack of tactile feedback and not actually encountering a physical object to stop the reaching movement and the object disappearing as soon as it is touched. However, subjects' velocities eventually become zero at the location corresponding to the center of the spherical target. Fig. 6 shows a snapshot of step B-5 in Fig. 4. The first 100 nearest neighbors by distance of starting location of motion sequence to the target is found, and of these starting locations the 50 nearest neighbors in PCA space are then found. For the entire motion trajectory only these nearest neighbor motion sequences are used. For construction of the local velocity vector field the 500 nearest neighbor frames in PCA space are found and a first order least squares fit to this data is calculated. The control/velocity for the current frame position is determined from this local least squares solution. Each control decision takes an average of .02 seconds on an Intel P3 1.4 GHZ with 512MB RAM, resulting in essentially real-time control.

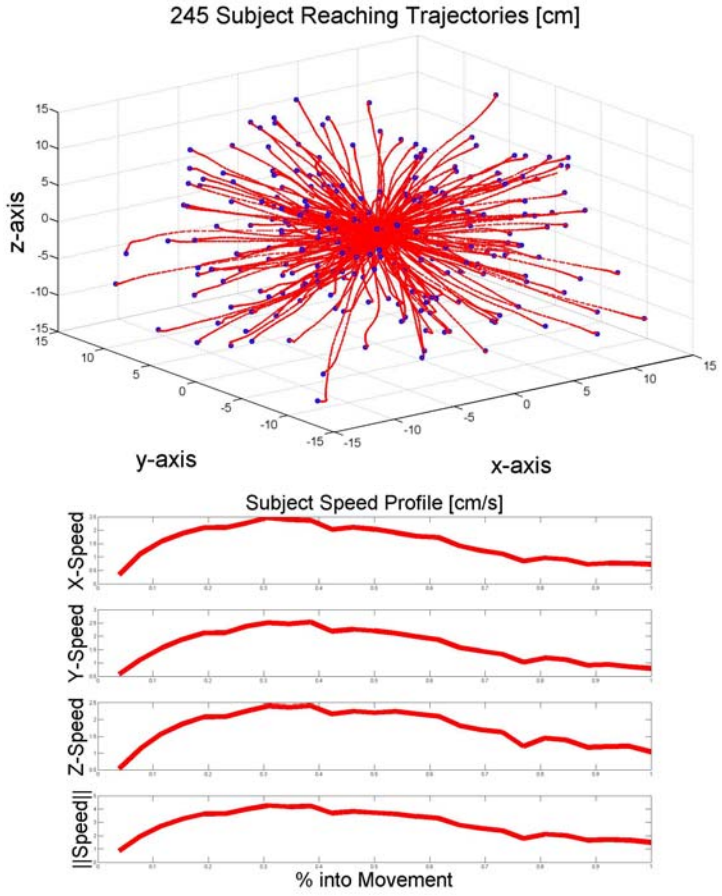

Fig. 5: (Top) 245 reaching trajectories from a single subject in the reference frame of the target. Blue points are the initial positions and red lines are the trajectories to the target (origin). (Bottom) Average bell-curved speed profiles for reaching movements time-normalized in 5% bin intervals between 0 to 1. The magnitude of the velocity vector (4th bottom plot) peaks at ~5cm/sec at approximately 1/3 into the movement.

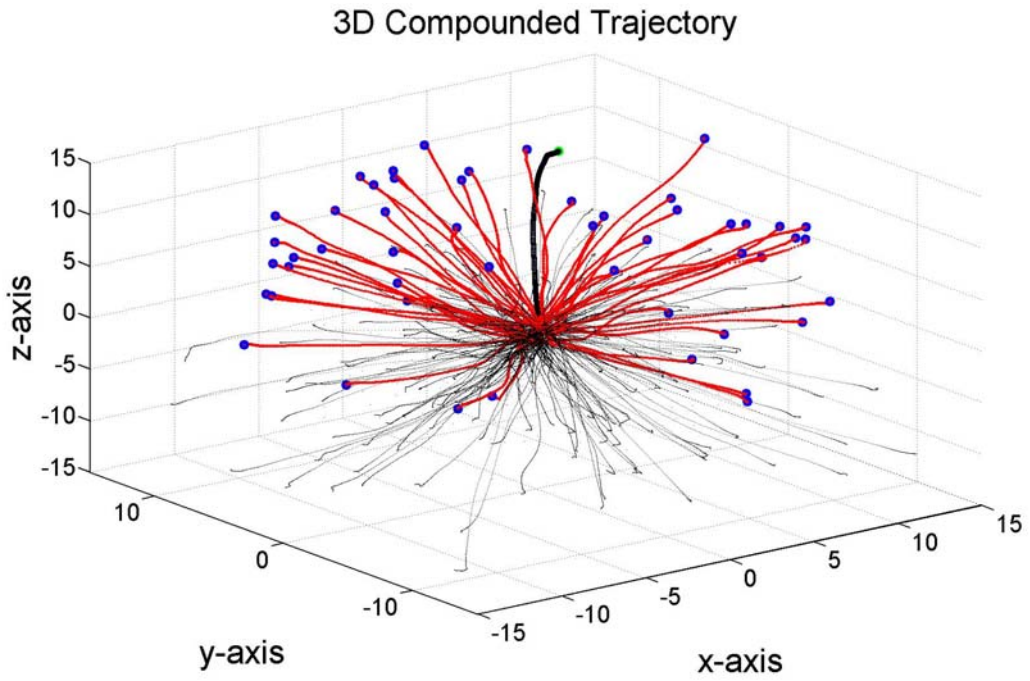

Fig. 6: Red lines are the 50 nearest neighbor trajectories to initial frame(green) found in step B-5. Thick black line is the controller's trajectory. Thin black lines are all the example trajectories.

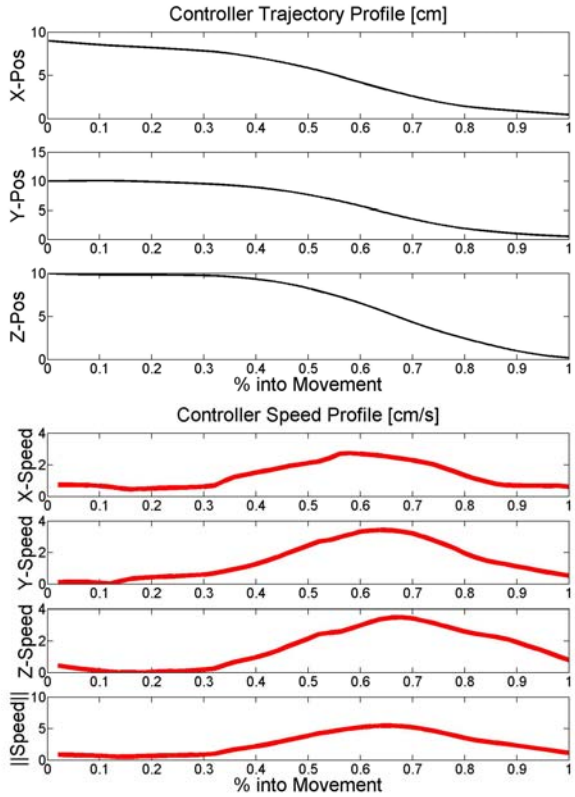

Fig. 7: Controller profile from sample starting location. (Top) Position profile in reference frame of target for reaching movement. (Bottom) Bell-curve speed profile for reaching movement.

Fig. 7 shows the position and velocity profiles of the controller from a sample start location in the reference frame of the target. The controller gives a very smooth trajectory to the target with a bell-shaped speed profile. The notable difference is the peak velocity magnitude of ~4cm/sec occurs at approximately 2/3 into the movement. These results do not change significantly as a function of starting location and target location. The controller of course performs increasingly worse in regions of the workspace less densely populated with example reaching trajectories. Of note, however, is that to achieve real-time continuous control with similar characteristics to human data in the majority of the workspace only 245 reaching examples were needed. Total data collection time was ~5minutes. How well this approach to imitation learning scales with task complexity remains to be determined.

B. Imitation Reach-to-Grasp

Fig.8 shows reach-to-grasp examples from one subject in the reference frame of the target and Fig. 9 shows the average speed profiles for these trajectories for the thumb and index finger. The typical bell-curve shaped profiles are observed and unlike the reaching experiments they do end with zero velocity. This implies that subjects are able to do more accurate reach to grasp movements than reaching movements in our VRE. Fig. 10 shows the average aperture as a function of percent of time into the movement, and reaches a maximum of 7cm at 50% of the movement. Fig. 11 shows a snapshot of step B-5 in Fig. 4. The first 100 nearest neighbors by distance of starting location of motion sequence to the target is found, and of these starting locations the 10 nearest neighbors in PCA space are then found. For the entire motion trajectory only these nearest neighbor motion sequences are used. For construction of the local velocity vector field the 500 nearest neighbor frames in PCA space are found and a first order least squares fit to this data is calculated for the thumb and index finger separately. The control/velocity for the current frame position is determined from this local least squares solution. Each control decision takes an average of .005 seconds on an Intel P3 1.4 GHZ with 512MB RAM, resulting in essentially real-time control.

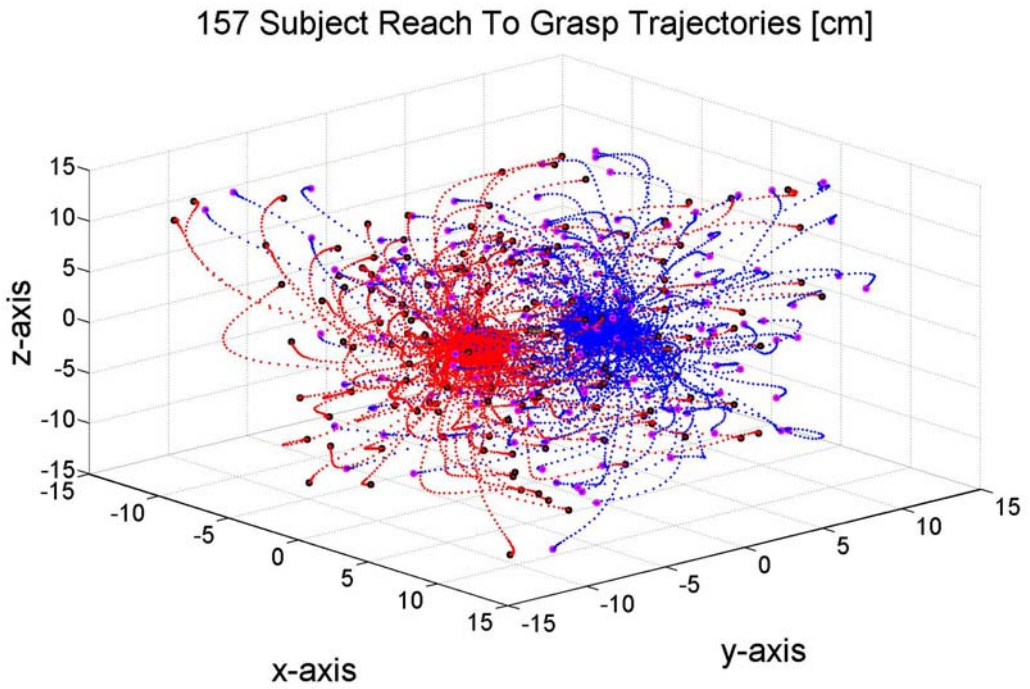

Fig. 8: 157 Reach to Grasp Trajectories from a single subject in the target reference frame. Red and blue lines show the thumb and index fingers trajectories, respectively.

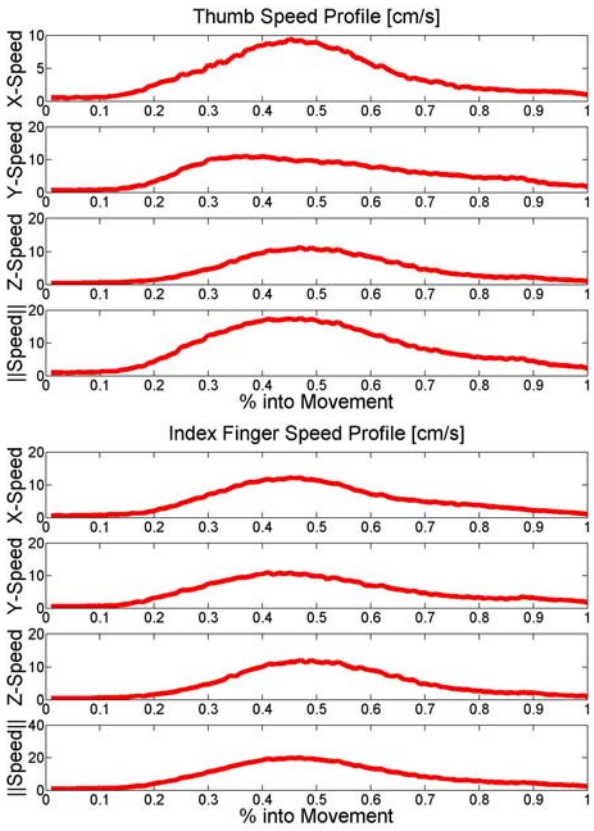

Fig. 9: Average bell-curved speed profiles for reach to grasp movements time-normalized in 5% bin intervals between 0 to 1. The magnitude of the velocity vector (4th bottom plot) peaks at ~20cm/sec at approximately 50% into the movement.

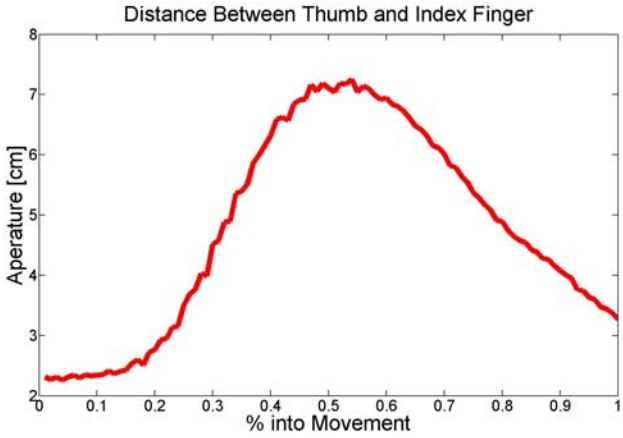

Fig. 10: Average distance between the thumb and index finger. Maximum aperture of 7 cm occurs at 50% of movement.

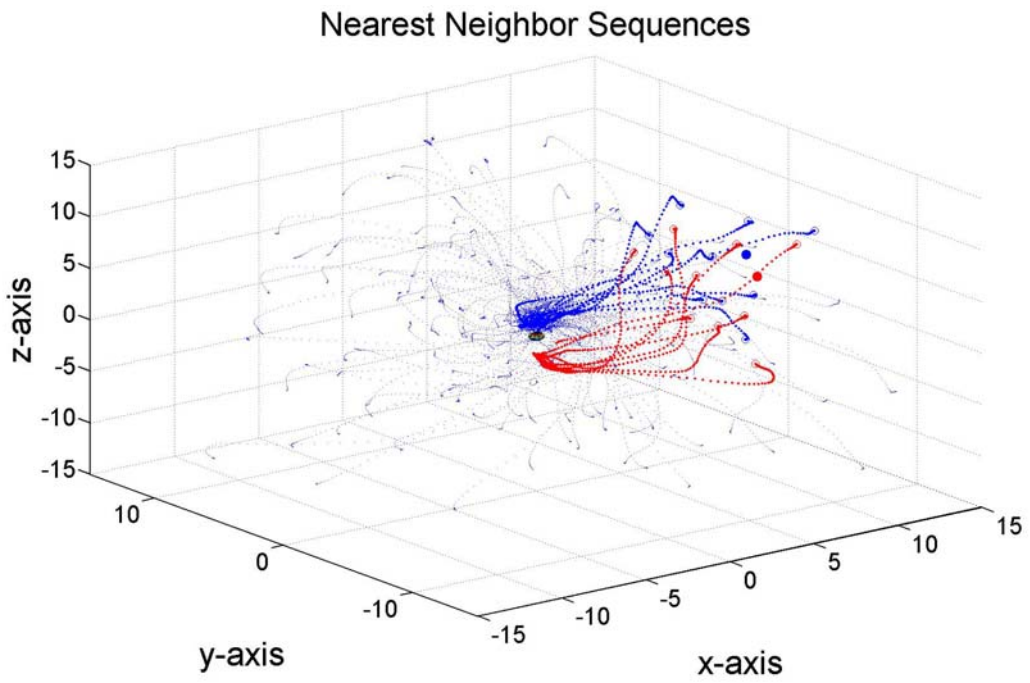

Fig. 11: 10 nearest neighbor trajectories to initial frame (large red and blue dots) found in step B-5. Red and blue dotted lines show the thumb and index fingers trajectories, respectively.

Fig. 12 and 13 show the reach to grasp trajectory of the controller from a sample start location in the reference frame of the target. The controller gives a very smooth trajectory to the target with roughly a bell-shaped speed profile, Fig. 14. Similar to the subject data, the peak velocity magnitude of ~18cm/sec occurs at approximately 1/2 into the movement. These results do not change significantly as a function of starting location and target location. The suspicious y-component of the index finger speed that goes to zero does this because the trajectory is briefly perpendicular to the y-axis as seen in Fig. 13. The aperture profile in Fig. 16 compares very closely with the teacher's examples with a maximum aperture ~7.5cm at 50% of the movement.

Like with the reaching controller, these results do not change significantly as a function of starting and target location and again it performs increasingly worse in regions of the workspace less densely populated with example reaching trajectories. Of note, however, is that to achieve real-time continuous control with similar characteristics to human data in the majority of the workspace only 157 reach-to-grasp examples were needed. Total data collection time was ~5minutes. How well this approach to imitation learning scales with task complexity remains to be determined.

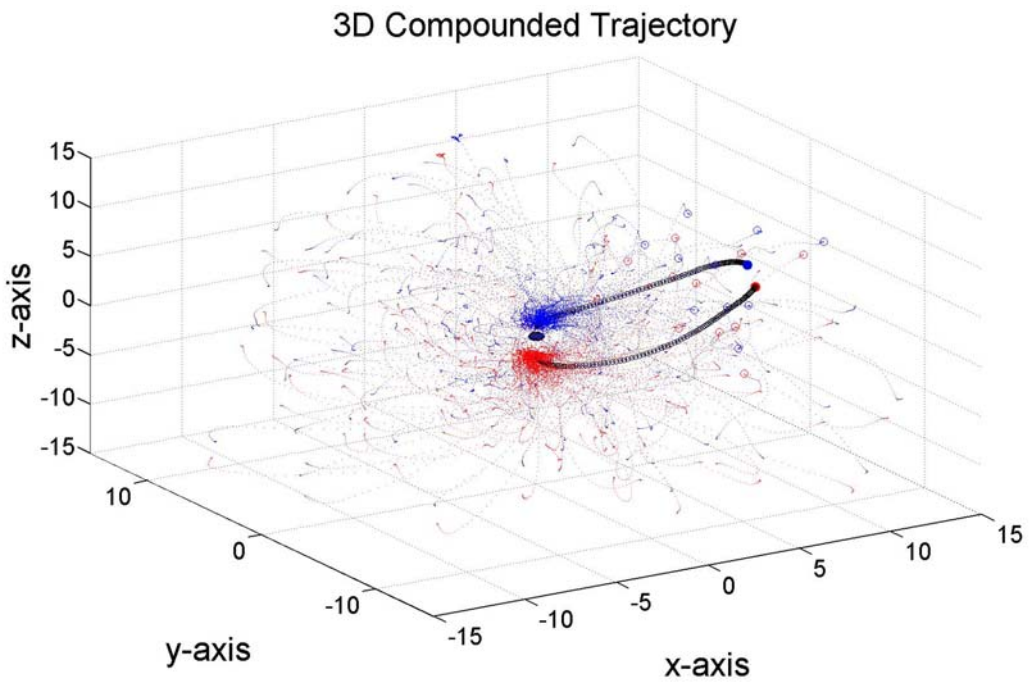

Fig. 12: 3D reach to grasp sample trajectory for imitation controller in target reference frame. The line originating from the large red dot is the thumb, and the line originating from the large blue dot is the index finger.

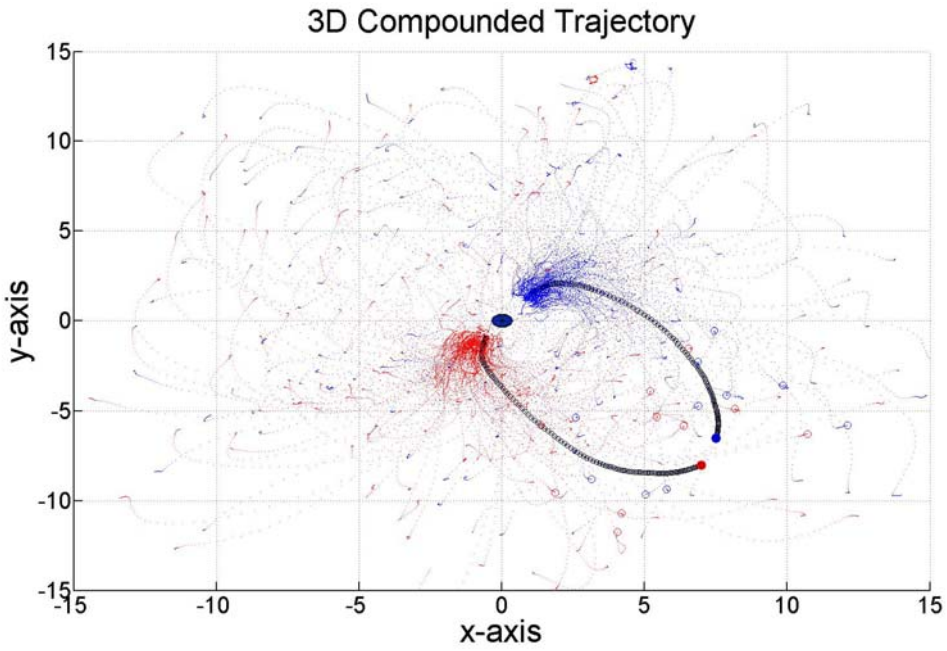

Fig. 13: 2D view of reach to grasp sample trajectory. Note close to the end of the movement, the index finger is perpendicular to the y-axis for a moment.

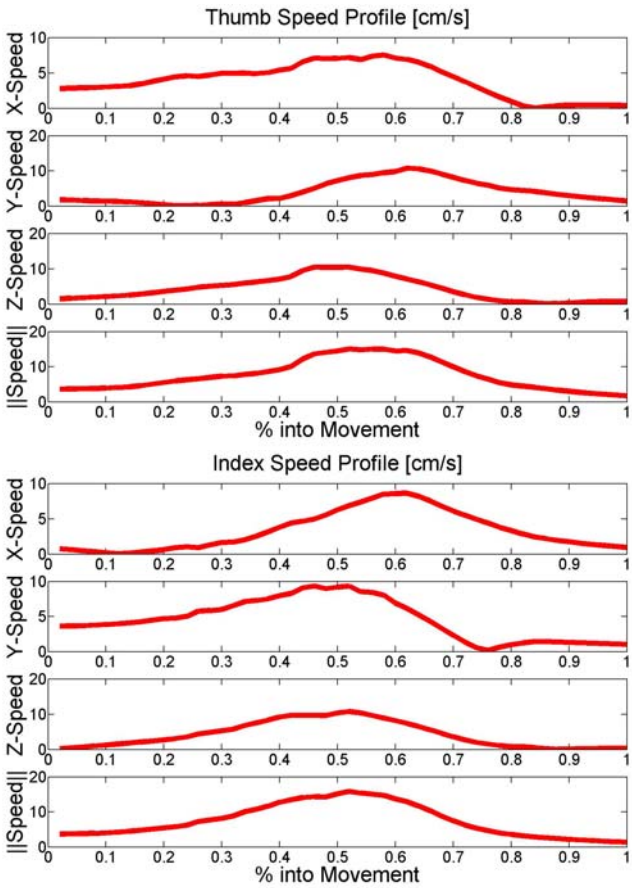

Fig. 14: Speed profile for the sample reach to grasp movement shows roughly a bell-curve shape. The y-component of the index finger speed goes to zero when the trajectory is perpendicular to the y-axis briefly.

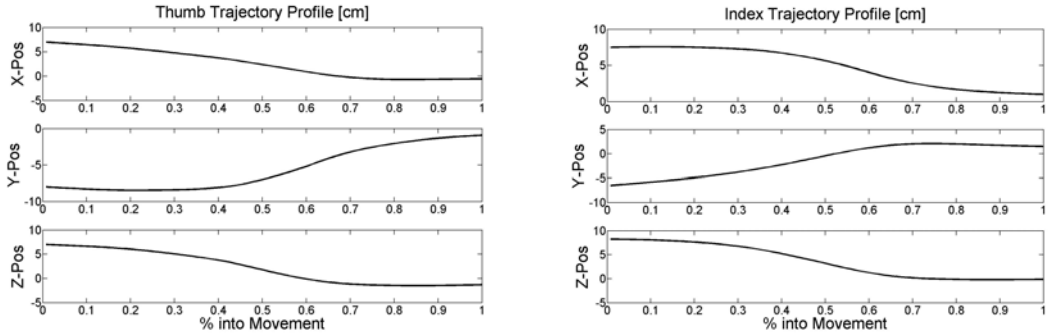

Fig. 15: Controller trajectory profile for the thumb and index finger from sample starting location.

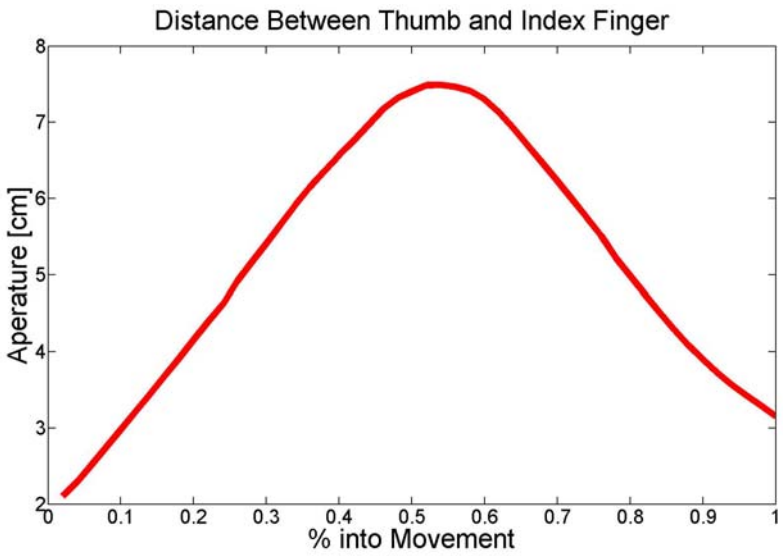

Fig. 16: Controller distance between the thumb and index finger. Maximum aperture of 7.5 cm occurs at 50% of movement.

V. Discussion

The results suggest that the imitation learning approach in this study is useful when designing reaching and reach to grasp controllers. We have demonstrated a real-time algorithm for constructing localized velocity vector fields around finger locations. The controller simply follows the unforced dynamics of this system towards the origin since the coordinate system is shifted to the reference frame of the target. Unlike previous models of reaching, no contact points have to be specified. Every initial starting location results in a natural contact location on the object.

How well this approach scales with task complexity is yet to be determined. Future studies will involve experiments where target orientation and contact locations are less trivial in the case of non-spherical targets. It is promising that a controller with the design presented here should be able to solve these tasks. However, it is not clear whether these approaches will work for the transport of a grasped object or a dynamic task such as bouncing a ball.

In this research we take a bottom-up approach of seeing what may work for an actual application. The tasks so far have been simple and a simple linear fit to the examples was sufficient. Future tasks might require more complex algorithms such as reinforcement learning. Before diving into these more powerful techniques it is first important to know how far simple approaches will take us and as we increase task complexity see where these techniques fail and why.

Models learned from imitation learning could be sufficient for learning simple tasks and a good initialization to speed up reinforcement learning algorithms for complex tasks. This is similar superficially to how humans learn from examples - by first observing teacher generated examples and then using these examples as a starting point for trial and error learning. In this research I ignore but recognize the difficulty of using vision to observe teacher generated examples. This is a problem beyond the scope of this research, but ignoring it does not prevent us from making progress in building a successful engineering application.

Interactive virtual environments with simulated physics have not been significantly studied for imitation learning of motor control tasks. These environments allow the experimenter to focus on developing controllers without worrying about correspondence and real-world uncertainty problems. Hopefully this research will give some insight into the human control strategy which will lead to building controllers for real environments. However, just because these strategies may work in simulated environments does not mean that it is trivial to translate them to the real-world. It is unclear how to solve the correspondence problem and how to handle real-world uncertainties in general. The approach presented here could, however, be directly applied to computer graphics for rapid generation of novel motion sequences in continuous time and space.

ACKNOWLEDGMENT

This work was supported by NSF grants DGE-0333451 and ECS-0524761.

REFERENCES

[1] Bertsekas D P, & Tsitsliklis J N (1996). Neuro-Dynamic Programming. Athena Scientific.

[2] Billard A, and Mataric M, **Learning human arm movements by imitation: Evaluation of a biologically-inspired connectionist architecture**. Robotics & Autonomous Systems 941 (2001), 1-16.

[3] Doya K. (2000). **Reinforcement learning in continuous time and space**. Neural Computation, 12, 219-245.

[4] Flash T, Hogan N (1985). **The coordination of arm movements: An experimentally confirmed mathematical model**. Journal of Neuroscience, 5, 1688-1703.

[5] Grochow K, Martin S L, Hertzmann A, Popović Z. **Style-based Inverse Kinematics**. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2004).*

[6] Jeannerod M. **The timing of natural prehension movements**. J. Mot. Behav. 1984; 16:235-54

[7] Jeannerod M (1981). **Intersegmental coordination during reaching at natural visual objects**. In J. Long & A. Baddeley (Eds.), Attention and performance IX (pp. 153-169). Hillsdale, NJ: Erlbaum.

[8] Kuhlen T, Kraiss K F, and Steffan R. **How VR-based reach-to-grasp experiments can help to understand movement organization within the human brain**. *Presence*, 9(4), 2000.

[9] Lee J, Chai J, Reitsma P, Hodgins J, and Pollard N, **Interactive Control of Avatars Animated with Human Motion Data**, ACM Transactions on Graphics (*SIGGRAPH 2002)*, volume 21, number 3, 491-500, July 2002.

[10] Paulignan Y, Jeannerod M. "Prehension movements: the visuomotor channels hypothesis revisited". In: Wing AM, Haggard P, Flanagan R, editors. Hand and brain: neurophysiology and psychology of hand movements. Orlando: Academic Press, 1996:265_ 82.

[11] Santello, M., Soechting, J. F. **Gradual molding of the hand to object contours**. J. Neurophysiol. 79: 1307-1320, 1998.

[12] Santello, M., Soechting, J. F. **Matching object size by controlling finger span and hand shape**. *Somatosens. Mot. Res.* 14: 203-212, 1997.

[13] Schaal S, Peters J, Nakanishi J, Ijspeert A, **Learning Movement Primitives**. International Symposium on Robotics Research (ISRR2003), Springer Tracts in Advanced Robotics. Ciena, Italy: Springer 2004.

[14] Schaal S, Ijspeert A, Billard A (2003). **Computational approaches to motor learning by imitation.** Philosophical Transaction of the Royal Society of London: Series B, Biological Sciences 358: 537-547.

[15] Simons, D. J. (2000), **Current Approaches to Change Blindness**. Visual Cognition, 7, 1-15.

[16] Smeets JBJ, Brenner E, Biegstraaten M (2002) **Independent control of the digits predicts an apparent hierarchy of visuomotor channels in grasping**. Behavioural Brain Research, 136:427-432

[17] Smeets JBJ, Brenner E. **A new view on grasping**. Motor Control 1999; 3:237-71.

[18] Sutton, R.S., Barto, A.G. (1998). Reinforcement Learning: An Introduction. MIT Press.

[19] Tomohiko Mukai, Shigeru Kuriyama. **Geostatistical Motion Interpolation**, ACM Transactions on Graphics, vol.24, issue.3, pp.1062-1070, 2005.7.

[20] Wing AM, Haggard P, Flanagan JR (Eds) (1996). Hand and Brain: Neurophysiology and Psychology of Hand Movement, Academic Press: San Diego