IEEE Copyright Statement:

Programming-by-Demonstration of Reaching Motions for Robot Grasping

Alexander Skoglund, Johan Tegin, Boyko Iliev and Rainer Palm

Abstract— This paper presents a novel approach to skill modeling acquired from human demonstration. The approach is based on fuzzy modeling and is using a planner for generating corresponding robot trajectories. One of the main challenges stems from the morphological differences between human and robot hand/arm structure, which makes direct copying of human motions impossible in the general case. Thus, the planner works in hand state space, which is defined such that it is perception-invariant and valid for both human and robot hand. We show that this representation simplifies task reconstruction and preserves the essential parts of the task as well as the coordination between reaching and grasping motion. We also show how our approach can generalize observed trajectories based on multiple demonstrations and that the robot can match a demonstrated behavior, despite morphological differences. To validate our approach we use a general-purpose robot manipulator equipped with an anthropomorphic three-fingered robot hand.

Index Terms— Programming-by-Demonstration, Hand State, Motion Planner, Fuzzy Modeling, Correspondence Problem.

I. INTRODUCTION

Programming-by-Demonstration (PbD) refers to variety of methods where the robot learns how to perform a task by observing a human teacher, which greatly simplifies the programming process [1], [2], [3] and [4]. One major scientific challenge in PbD is how to make the robot *capable* of imitating a human demonstration. Although the idea of copying human motion trajectories using a simple teaching-playback method seems straightforward, it is not realistic for several reasons. Firstly, there is a significant difference in morphology between the human and the robot, known as the correspondence problem in imitation [5]. The difference in the location of the human demonstrator and the robot might force the robot into unreachable parts of the workspace or singular arm configurations even if the demonstration is perfectly feasible from human viewpoint. Secondly, in grasping tasks the reproduction of human hand motions is not possible since even the most advanced robot hands cannot match neither the functionality of the human hand nor its sensing capabilities. However, robot hands capable of autonomous grasping can be used in PbD provided that the robot is able to generate an appropriate reaching motion towards the target object.

In this article, we present an approach to learning of reaching motions where the robot uses human demonstra-

tions in order to collect esential knowledge about the task. This knowledge, i.e., grasp-related object properties, hand-object relational trajectories, and coordination of reach and grasp motions is encoded and generalized in terms of *hand state space* trajectories. The hand state components are defined such that they are perception-invariant and defines the correspondence between the human and robot hand. The hand-state representation of the task is then embedded in a motion planner which enables the robot to perform reaching motions from an arbitrary robot configuration to the target object. The resulting reaching motion ensures that the robot hand will approach the object in such way that the probability for a successful grasp is maximized.

Four experiments describe how human demonstrations of goal-directed reach-to-grasp motions can be reproduced by a robot. Specifically, the generation of reaching and grasping motions in pick-and-place tasks is addressed. The first experiment is a simulation of an autonomous grasp performed from different poses in relation to the target. This will show how accurate the positioning of the end effector needs to be to execute a successful grasp. It is important to know when the end effector is in a position where the grasp execution can be started. The second experiment shows how the correspondence problem can be solved and how to generate a trajectory to executable on a real robot. The third experiment illustrates how the robot generalizes its knowledge for new positions of the object. It reproduces the demonstration regardless of the initial position of the robot and the position of the object. The goal of this experiment is to investigate how well each model can generalize across the workspace. This is related to the number of models needed for the robot to perform a successful reaching-to-grasp action; good generalization ability means that fewer models are needed. The fourth experiment is done to assess the reaching and grasping as an integrated process. A complete pick-and-place task is eventually demonstrated and executed by the robot.

The contributions of the work in this paper are as follows:
1) We introduce a novel approach using a *next-state-planner* based on the *fuzzy clustering* approach to encode human and robot trajectories.
2) We apply the *hand state concept* [6] to encode motions in hand state trajectories and apply this in PbD. The hand state description is the link between human and robot motions.
3) The combination of the next-state-planner and the hand state approach provides a tool to address the *correspondence problem* resulting from the different morphology

Alexander Skoglund, Boyko Iliev and Rainer Palm are with the Center for Applied Autonomous Sensor Systems, School of Science and Technology, Örebro University, Sweden; email: alexander.skoglund@oru.se, boyko.iliev@oru.se, rub.palm@t-online.de

Johan Tegin is with Mechatronics Laboratory, Machine Design, Royal Institute of Technology, Stockholm, Sweden; email: johant@md.kth.se

of the human and the robot. The experiments shows how the robot can generalize and use the demonstration despite its fundamentally different morphology.

One advantage of this approach over trajectory averaging (e.g., [1] or [7]) is that one of the human demonstrations is used instead of an average which might contain two essentially different trajectories [8]. By capturing a human demonstrating the task, the synchronization between the reach and the grasp is also captured, demonstrated in [9]. Other ways of capturing the human demonstrating, such as kinesthetics, cannot easily capture this synchronization.

II. INTERPRETATION OF HUMAN DEMONSTRATIONS IN HAND-STATE SPACE

Interpretation of human demonstrations is done on the basis of the following assumptions:

- The type of tasks and grasps that can be demonstrated are *a priori* known by the robot.
- We consider only demonstrations of power grasps (e.g., cylindrical and spherical grasps) which can be mapped to–and executed by–the robotic hand we use.

If hand motions with respect to a potential target object are associated with a particular grasp type G_i, it is assumed that there must be a target object that matches the observed grasp type. In other words, the object has certain grasp-related features, also called *affordances* [6], which makes this particular grasp type approapriate.

For each grasp type G_i, a subset of suitable object affordances is identified a priori and learned from a set of training data. In this way, the robot is able to associate observed grasp types G_i with a set of affordances A_i offered by the object to perform the observed grasp. Once the target object is known, the hand state can also be defined. According to Oztop [6], the hand state must contain components describing both the hand configuration and its spatial relation with respect to the affordances of the target object. Thus, the hand state is defined in the form:

$$H = \{h_1, h_2, \ldots h_{k-1}, h_k, \ldots h_p\} \qquad (1)$$

Here, $h_1 \ldots h_{k-1}$ are *hand-specific components* which describe the motion of the hand during grasping. The remaining components $h_k \ldots h_p$ describe the motion of the hand in relation to the object. Thus, a hand state trajectory contains a record of both the reaching and the grasping motions as well as their synchronization in time and space.

In the PbD framework, $h_1, \ldots h_k$ must be such that they can be recovered from both human demonstrations and the perception system of the robot. That is, the definition of H must be *perception invariant* and can be updated from arbitrary types of sensory information. Fig. 1 shows the definition of the hand state in this paper.

Let the human hand be at some initial state H_1. Then the hand moves along a certain path and reaches the final state H_f where the target object is held by the hand. That is, the recorded motion trajectory can be seen as a sequence of states, i.e.,

$$H_d(t) : H_1(t_1) \rightarrow H_2(t_2) \rightarrow \ldots \rightarrow H_f(t_f) \qquad (2)$$
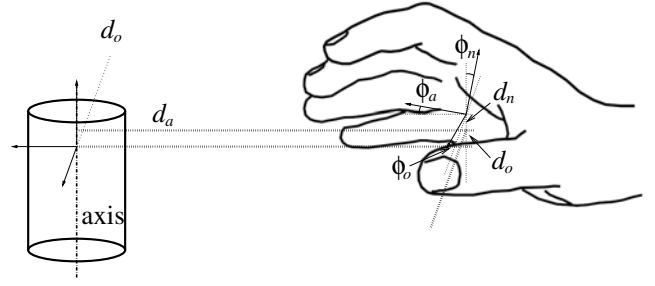

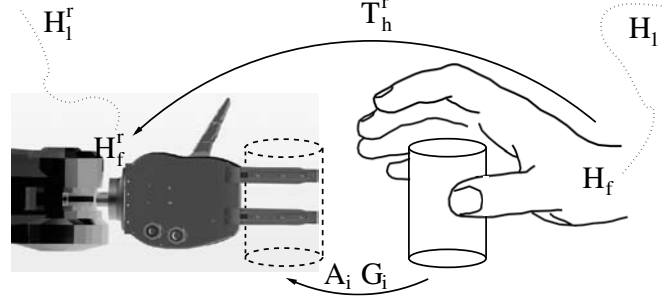Fig. 1. The hand state describes the relation between the hand pose and the object affordances.

Fig. 2. Mapping from human hand to robotic gripper.

Since $H_d(t)$ cannot be executed by the robot without modification in the general case, we have to construct the *robotic* version of $H_d(t)$, denoted by $H^r(t)$, see Fig. 2 for an illustration. To find $H^r(t)$ a mapping from the human grasp to the robot grasp is needed, denoted T_h^r. This mapping is created as follows. T_h^r is a transformation matrix which defines the spatial relation between the human hand and the robot hand while holding the object with a grasp corresponding to G_i. Thus, we can measure the pose of the demonstrator hand and the robot hand holding the same object at fixed position and obtain T_h^r as a static mapping between the two poses. It should be noted that this method is only suitable for power grasps. In the general case it might produce ambiguous results or rather inaccuarate mappings.

With T_h^r defined, we can now create the robot version of H_d as follows. The current pose of the robot hand defines the initial state of H_1^r. The target state of H_f^r will be derived from the demonstration by mapping the goal configuration of the human hand H_f into a goal configuration for the robot hand H_f^r (see Fig. 2) using the transformation T_h^r:

$$H_f^r = T_h^r H_f \qquad (3)$$

For the power grasp the robot hand is positioned so the grasp is expected to be successful at H_f^r. Next the human hand position H_f in hand state space and the robot hand position H_f are use to compute the transformation T_h^r from human to robot obtained by:

$$T_h^r = H_f^r H_f^{-1} \qquad (4)$$

III. GENERATION OF ROBOT HAND STATE TRAJECTORIES FROM DEMONSTRATIONS

In this section we describe how robot reaching motions are generated based on generalization of human demonstration in hand state space.

Having the initial and the target states defined, we have to generate the trajectory between the two states. In principle, we could transform $H_d(t)$ using (3) in such way that it has its final state in H_f^r. Then, the robot starts at H_1^r, approaches the displaced demonstrated trajectory and tracks it until the target state. However, such approach would not take trajectory constraints into account. Thus, it is also necessary to specify exactly how to approach $H_d(t)$ and what segments must be tracked accurately. Moreover, $H^r(t)$ has to synchronize the reaching motion driving the arm with the grasping.

The workspace restrictions of the robot also have to be considered when creating trajectories. A trajectory might contain regions which are out of reach, or two connected points on the trajectory require different joint space solutions, thus, the robot cannot execute the trajectory. To avoid or remedy the effect from this problem the manipulator must be placed at a position/orientation with good reachability. Other solutions include a mobile platform, larger robot, or more degrees of freedom (DOF) to mimic the redundancy of the human arm.

A. Trajectory modeling using fuzzy clustering

By modeling the hand state trajectories recorded from the demonstration using Takagi-Sugeno (TS) fuzzy clustering we obtain three benefits: 1) a compact representation of the dynamic arm motion in form of cluster centers, 2) nonlinear filtering of noisy trajectories and 3) simple interpolation between data samples. Three types of models are needed: a model of the distance to the object and as a function of time; a model of the hand states a function of time; and a model of the hand state as a function of the distance. The TS fuzzy models are constructed from captured data described by the nonlinear function:

$$\mathbf{x}(y) = \mathbf{f}(y) \tag{5}$$

For fuzzy time clustering (see [10] for details) $\mathbf{x}(y) \in R^n$, where n is either 1 for distance and 6 for hand state, $\mathbf{f} \in R^1$, and $y \in R^+$. For distance clustering of the end effector pose $\mathbf{x}(y) \in R^6, \mathbf{f} \in R^6$, and $y \in R^+$. The parameter y can be the time or the distance. Equation (5) is linearized at selected data points which results in a linear equation in y.

$$\mathbf{x}(y) = \mathbf{A}_i \cdot y + \mathbf{a}_i \tag{6}$$

where $\mathbf{A}_i = \frac{\Delta \mathbf{f}(y)}{\Delta y}|_{y_i} \in R^n$ and $\mathbf{a}_i = \mathbf{x}(y_i) - \frac{\Delta \mathbf{f}(y)}{\Delta y}|_{y_i} \cdot y_i \in R^n$. Using (6) as a local linear model one can express (5) in terms of an interpolation between several local linear models by applying TS fuzzy modeling [11]:

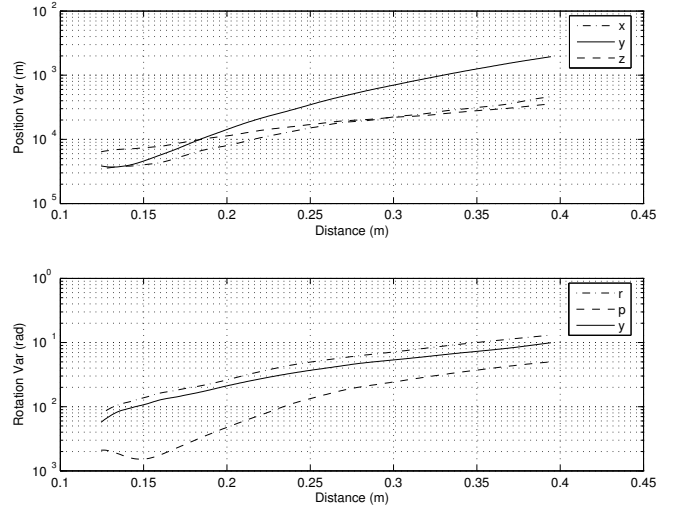$$\mathbf{x}(y) = \sum_{i=1}^{c} w_i(y) \cdot (\mathbf{A}_i \cdot y + \mathbf{a}_i) \tag{7}$$

Fig. 3. Position- and orientation-variance of the hand state trajectories as function of distance, across 21 demonstrations of a reaching motion to grasp a soda can.

$w_i(y) \in [0, 1]$ is the degree of membership of the data point y to a cluster with the cluster center y_i, c is number of clusters, and $\sum_{i=1}^{c} w_i(y) = 1$.

B. Variance in hand state trajectories

In this section, we show how to generate an executable $H^r(t)$ by incorporating knowledge from previous demonstrations of similar tasks. We exploit the fact that when humans grasp one object several times they seem to repeat the same grasp type, which leads to similar approach motions. Based on that, multiple demonstrations of grasp type (G_i) using affordances A_i become very similar to each other the closer we get to the target state. This implies that successful grasping requires accurate positioning of the hand in some area near the object while the path towards this area has to satisfy less strict constraints. By looking at the variance in several demonstrations, the importance of each hand state component can be determined. The variance of the hand state at distance d to the target is given by:

$$var(^k h(d)) = \frac{1}{n-1} \sum_{i=1}^{n} (^k h_i(d) - mean(^k h(d)))^2 \tag{8}$$

where d is the Euclidean distance to the target, $^k h_i$ is the kth hand state parameter of ith demonstration (from Eqn. 1) and n is the number of demonstrations. Fig. 3 show how the variance decreases as the distance to the object decreases, which means that the position and orientation of the hand is less relevant when the distance to the target increases.

C. Generation of robot trajectories

In this section, the next-state planner is presented. The next-state planner generates a hand state trajectory for the robot using the TS fuzzy-model of a demonstration. As the resulting $H^r(t)$ is in Cartesian space we exploit the inverse kinematics provided by the controller for the robot arm. The TS fuzzy-model serves as a motion primitive for controlling the arm's reaching motion. The initial hand state of the robot

3

is determined from its current configuration and the position and orientation of the target object, since these are known at the end of the demonstration. Then, the desired hand state H_d^r is computed from the from the TS time cluster model (Eqn. 7).

The desired hand state H_d is fed to a hand state trajectory generator. This planner is inspired by the Vector-Integration-To-Endpoint (VITE) planner suggested by Bullock and Grossberg [12]. Instead of using a goal attractor as in VITE, we use the desired hand state trajectory as an attractor at each state. The system has the following dynamics:

$$\ddot{H} = \alpha(-\dot{H} + \gamma(H_d - H)) \qquad (9)$$

where H is the hand state, $\ddot{H}$ is the acceleration H_d is the desired hand state encoded in Eqn. 7, α is a positive constant, and γ is a positive weighting parameter for the tracking point. The weight γ reflects the importance of the path, acquired form variance, see Sec. III-B. We have empirically found γ to produce satisfying results at:

$$\gamma_{pos} = 0.3 \frac{1}{\sqrt{Var(H_{xyz}(d))}}$$

$$\gamma_{ori} = 5 \frac{1}{\sqrt{Var(H_{rpy}(d))}}$$

where γ_{pos} and γ_{ori} are the weights for position and orientation, respectively. $Var(H_{xyz}(d))$ and $Var(H_{rpy}(d))$ are the variance for the position and orientation respectively, from Eqn. 8, of the respective hand state component. α_{pos} and α_{ori} were fixed during our experiments at 8 and 10, respectively, with $dt = 0.01$. These gains were chosen to provide dynamic behavior similar to the demonstrated motions, but other criteria can also be used.

Analytically, the poles in Eqn. 9 are found from:

$$p_1, p_2 = -\frac{\alpha}{2} \pm \sqrt{\frac{\alpha^2}{4} - \alpha\gamma} \qquad (10)$$

so the real part of p_1 and p_2 will be ≤ 0, which will result in a stable system [13]. Moreover, $\alpha \nleq 4\gamma$ and $\alpha \geq 0$, $\gamma \geq 0$ will contribute to a critically damped system, which is fast and has small overshoot. Fig. 4 shows how different values γ affects the dynamics of the planner.

The next-state planner uses the demonstration to generate a similar hand state trajectory, using the distance as a scheduling variable. Hence, the closer to the object the robot is the more important it becomes to follow the demonstrated trajectory. This property is reflected by adding a higher weight to the trajectory-following dynamics when we get closer to the target; in reverse a long distance to the target leads to a lower weight to the trajectory following dynamics.

IV. EXPERIMENTS

We recorded human demonstrations of a pick and place task with two different subjects, using the PhaseSpace Impulse motion capturing system. The data are collected at a sampling rate of 120 Hz, using nine LEDs located a different
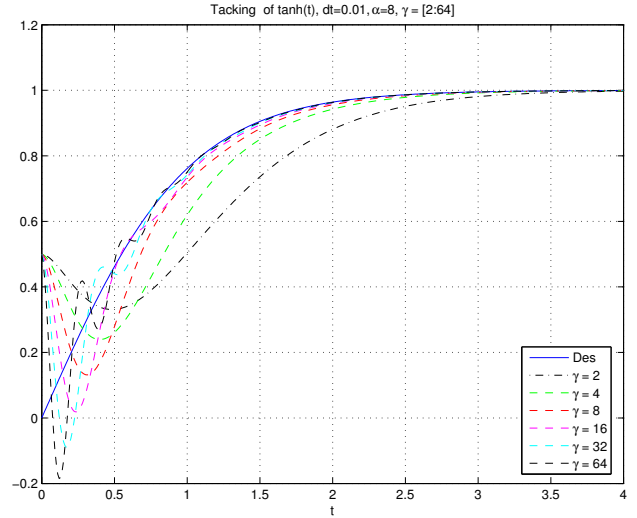

Fig. 4. The dynamics of the planner for six different values of γ. The tracking point is $tanh(t)$, with $dt = 0.01$ and α is fixed at 8. A low value on $\gamma = 2$ produces slow dynamics (black dot-dashed line), while a high value $\gamma = 64$ is fast but overshoots the tracking point (black dashed line).

point on the hand and one LED attached to the object. The experimental setup can be seen in Fig. 10. The motions are automatically segmented into reach and retract motions using the velocity profile and distance to the object. The robot used in the experiments is the industrial manipulator ABB IRB140. The anthropomorphic gripper is the KTHand, described in detail in [14].

The hand state is defied for these experiments to contain six hand-object relation components: displacement x, y and z direction and rotation around the three axes: roll r, pitch p and yaw y, see Fig. 1.

A. Experiment 1 – Gripper pose variation

To investige how end effector position–and hence the approach trajectory–affect grasp succes, we have dynamically simulated grasping an orange using different hand positions across a 3D-grid. The joint space is linearly transformed to enable control of the total grasping force using the sum from all tactile force sensors. The relative positions of the fingers are position controlled. The hybrid force/position controller is applied to the Barrett hand model from grasp initiation until the grasp is completed. See [14] for a detailed description of the robotic hand and the hybrid force/position controller. A grasp is considered failed if no force closure grasp was reached during grasp formation. Fig. 5 shows the results from such simulations.

The required accuracy in position of the end effector is in the centimeter range. The required accuracy of the reaching motion depends of abilities of the gripper; an autonomous gripper like the Barrett hand or the KTHand impose a looser constraint on the reach motion than a parallel gripper, which requires much higher accuracy. For fully autonomous execution of a grasp learnt using the suggested approach, we must also consider uncertainties with respect to object position, orientation, and in the object model itself.
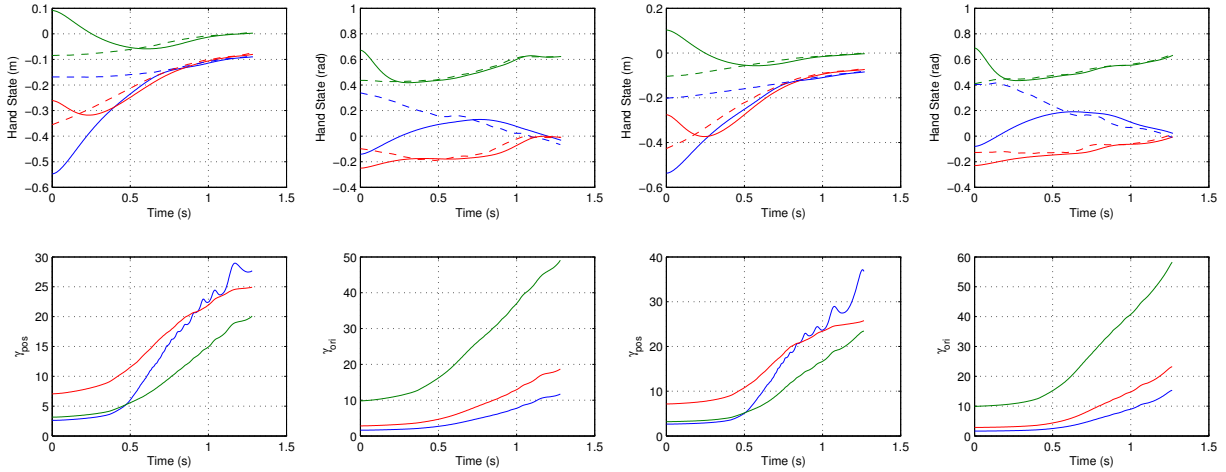
Fig. 6. Two sample demonstrations, and the corresponding imitations. The solid lines are the trajectories produced by the controller, dashed lines are the recorded from the demonstration.
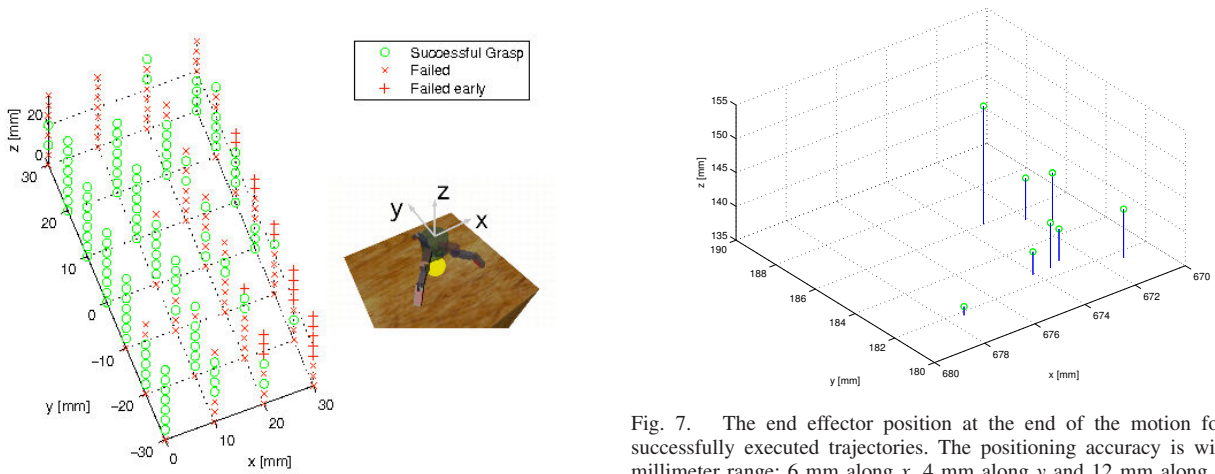

Fig. 5. Grasp results from dynamic simulations of different initial hand positions. Grid spacing is 10 mm in the xy-plane and 5 mm along the z-axis. Please note that the coordinate system orientation in this figure is different from that elsewhere in this paper.

Fig. 7. The end effector position at the end of the motion for the 8 successfully executed trajectories. The positioning accuracy is within the millimeter range; 6 mm along *x*, 4 mm along *y* and 12 mm along *z*.

B. Experiment 2 – Learning from demonstration

For this experiment 26 demonstrations of a pick and place task were performed. A soda can was grasped with a spherical grasp. To make the scenario more realistic the object is placed with respect to what is convenient for the human and what *seems* to be feasible for the robot.

Five of the 26 demonstrations were discarded in the segmentation (see [3]) and modeling process for reasons such as failure to segment the demonstrations into three distinct motions (approach, transport and retract) or the amount of data were not enough for modeling because of occlusions. Only the reach to grasp phase is considered in this experiment. All 21 demonstrations were used for trajectory generation and to compute the variance, shown in Fig. 3. Moreover, the variance is used to compute the γ-gain, which determines how much the robot can deviate from the followed trajectory. The trajectory generator produced 21 reaching motions, one from each demonstration, which are loaded to the robot controller and executed. Note that by using each demonstrated trajectory as the desired trajectory H_d instead of an average

we avoid fusing trajectories which are essentially different into an incoherent trajectory. Large differences will instead affect the variance, resulting in a small γ-gain. In eight attempts, the execution succeeded while 13 attempts failed because of unreachable configurations in joint space. This could be prevented by placing the robot at a different location with better reachability. Moreover, providing the robot with more demonstrations, with higher variations in the path, will lead to fewer constraints. Two sample hand state trajectories of the successfully generated ones are shown in Fig. 6. In the top graphs it is shown how the generated trajectory converges towards the desired trajectory, after the initially different locations. The bottom graphs shows how γ varies over time, to make the generated trajectory H^r follow the desired H_d.

In the eight successfully executed reaching motions we measured the variation in position of the gripper, shown in Fig. 7, which is within the millimeter range. This means that the positioning is accurate enough to enable successful grasping using an autonomous gripper, such as the Barrett hand [15] or the KTHand.

C. Experiment 3 – Generalizations in work space

In this experiment, the method is tested on how well it generalizes by examining if feasible trajectories will be
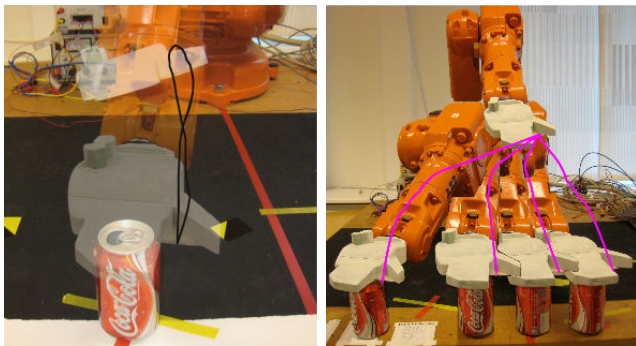
Fig. 8. Left:A trajectory generated when the initial position is the same as the desired final position, showing that the method generate trajectories as similar to the demonstration as possible based on the distance. Right: The object is placed at four new locations within the workspace.

generated when the object is placed at arbitrary locations and when the initial configuration of the manipulator is very different from the demonstration. This will determine how the trajectory planner handles the correspondence problem in terms of morphological differences.

If the initial distance between the end effector and the target is outside the data range, the TS-models must be extrapolated, a risky strategy for longer distances. Another approach is to apply a different control scheme for this region, such as the VITE strategy (see [12]) and when the distance is within the data range the proposed trajectory generator takes over. Three tests were preformed to evaluate the trajectory generator in different parts of the workspace.

First, trajectories are generated when the manipulator's end effector starts directly above the object at the desired final position with the desired orientation, i.e., $H_1^r = H_f^r$. The resulting trajectory is shown to the left in Fig. 8. Four additional case is also tested displacing the end effector by 50 mm in +x, -y, +y, and +z direction, all with very similar results (from the robot's view: +x is forward, +y left and +z up).

Second, the object is placed at four different locations within the robot's workspace; displaced 100 mm along the x-axis, and -100 mm, +100 mm, +200 mm, and +300 mm along the y-axis, seen to the right in Fig. 8. The initial pose of the manipulator is the same in all reaching tasks. The planner successfully produces four executable trajectories to the respective object position.

Third, we tested reaching the object at a fixed position from a random initial configuration. Fig. 9 shows the result from two random initial positions where one trajectory is successfully followed and the other one fails. The failure is a result of operation in hand state space instead of in joint space, and it might therefore have a tendency to go onto unreachable joint space configurations, as seen in the right column of Fig. 9. To prevent this it is possible to combine two controllers: one operating in joint space and the other in hand state space, similar to the approach suggested in [16], but at the price of violating the demonstration constraints.

The conclusion from this experiment is that the method generalizes well in the tested scenarios, thus adequately

Fig. 9. A trajectory generated from random initial positions reaching for the same object. In the left column, a successful reaching motion is generated where the final position is on top of the can. The right column shows a case where the robot reaches an unreachable joint configuration and cannot move along the trajectory.

addressing the correspondence problem. However, the unreachability problem has to be addressed in future research to investigate how the robot should balance the two contacting goal: reaching an object in its own way, with the risk of collision, and reaching an object as the demonstrator showed.

D. Experiment 4 – A complete task

To test the approach on an integrated system the KTHand is mounted on the ABB manipulator and a pick-and-place task is executed, guided by a demonstration showing pick-and-place task of a box ($110 \times 56 \times 72$ mm). The reaching motion and the grasp are executed as described in the previous experiments in this Section. The synchronization between reach and grasp can be performed by a simple finite state machine or by using the hand specific components of the hand state for automatic synchronization [9]. After the grasp is executed, the motion to the placing point is performed by following the demonstrated trajectory. Since the robot grasp pose corresponds approximately to the human grasp pose it is possible for the planner to reproduce the human trajectory almost exactly. This *does not* mean that the robot actually can *execute* the trajectory, due to workspace
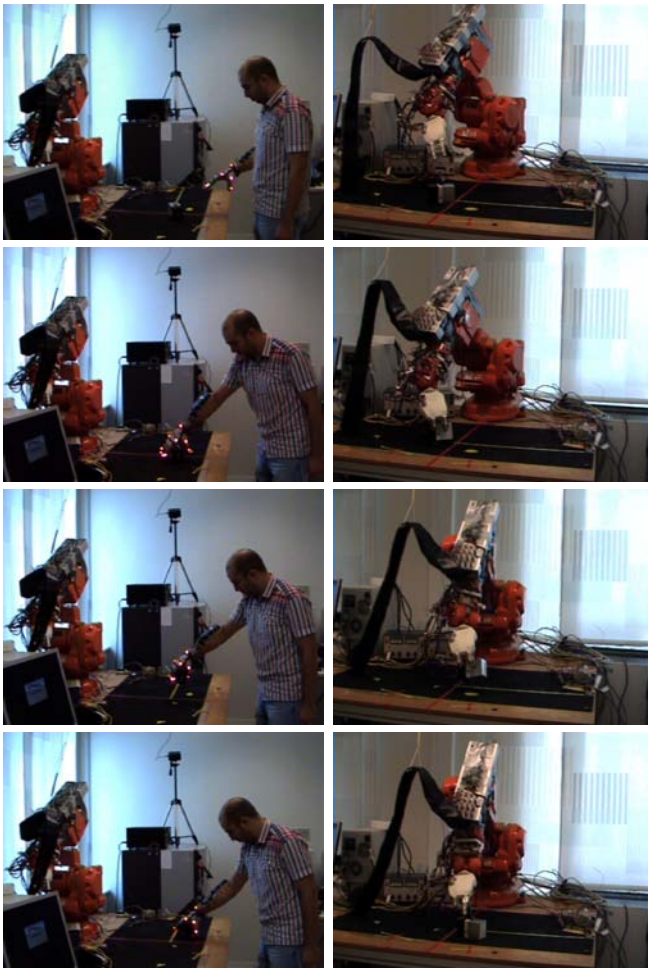
Fig. 10. Industrial manipulator programmed using a demonstration.

constrains. The retraction phase follows the same strategy as the reaching motion. Fig. 10 shows the complete task learned from demonstration.

V. CONCLUSIONS AND FUTURE WORK

In this article, we present a method for programming-by-demonstration of reaching motions for robotic grasping tasks. Hand state representation is employed to create the mapping between the human and the robot hand which allows the robot to interpret the human motions as its own. It is shown that the suggested method can generate executable robot trajectories based on current and past human demonstrations despite morphological differences. The generalization abilities of the trajectory planner are illustrated by several experiments where an industrial robot arm executes various reaching motions and performs power grasping with a three-fingered hand.

One disadvantage of the method is related to the use of Cartesian space trajectories which may lead to unreachable joint space trajectories. Another shortcoming is the absence of obstacle avoidance ability, i.e., the user has this respon-

sibility. One possible solution is to incorporate an obstacle avoiding component in the dynamics of the planner.

In our future work we plan to extend the theoretical and experimental work to include all feasible grasp types of the KTHand. To remedy the effect of the small workspace of the robot a different workspace configuration will be used. Furthermore, the robot's own perception will be incorporated into the loop to enable the robot to learn from its own experience.

REFERENCES

[1] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 286–298, April 2007.

[2] M. Pardowitz, S. Knoop, R. Dillmann, and R. D. Zöllner, "Incremental learning of tasks from user demonstrations, past experiences, and vocal comments," *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 37, no. 2, pp. 322–332, April 2007.

[3] A. Skoglund, B. Iliev, B. Kadmiry, and R. Palm, "Programming by demonstration of pick-and-place tasks for industrial manipulators using task primitives," in *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Jacksonville, Florida, June 20-23 2007, pp. 368–373.

[4] J. Takamatsu, K. Ogawara, H. Kimura, and K. Ikeuchi, "Recognizing assembly tasks through human demonstration," *International Journal of Robotics Research*, vol. 26, no. 7, pp. 641–659, 2007.

[5] C. L. Nehaniv and K. Dautenhahn, "The correspondence problem," in *Imitation in Animals and Artifacts*, K. Dautenhahn and C. Nehaniv, Eds. Cambridge, MA: The MIT Press, 2002, pp. 41–61.

[6] E. Oztop and M. A. Arbib, "Schema design and implementation of the grasp-related mirror neurons," *Biological Cybernetics*, vol. 87, no. 2, pp. 116–140, 2002.

[7] N. Delson and H. West, "Robot programming by human demonstration: Adaptation and inconsistency in constrained motion," in *IEEE International Conference on Robotics and Automation*, 1996, pp. 30–36.

[8] J. Aleotti and S. Caselli, "Robust trajectory learning and approximation for robot programming," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 409–413, 2006.

[9] A. Skoglund, B. Iliev, and R. Palm, "A hand state approach to imitation with a next-state-planner for industrial manipulators," in *Proceedings of the 2008 International Conference on Cognitive Systems*, University of Karlsruhe, Karlsruhe, Germany, April 2-4 2008, pp. 130–137.

[10] R. Palm and B. Iliev, "Learning of grasp behaviors for an artificial hand by time clustering and Takagi-Sugeno modeling," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, Vancouver, BC, Canada, July 16-21 2006, pp. 291–298.

[11] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control." *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, January/February 1985.

[12] D. Bullock and S. Grossberg, "VITE and FLETE: Neural modules for trajectory formation and postural control," in *Volitonal Action*, W. A. Hershberger, Ed. Elsevier Science Publishing B.V. (North-Holland), 1989, pp. 253–297.

[13] W. S. Levine, "The root locus plot," in *The Control Handbook*, W. S. Levine, Ed. CRC Press, 1996, ch. 10.4, pp. 192–198.

[14] J. Tegin, J. Wikander, and B. Iliev, "A sub €1000 robot hand for grasping – design, simulation and evaluation," in *International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, Coimbra, Portugal, Sep 2008.

[15] J. Tegin, S. Ekvall, D. Kragic, B. Iliev, and J. Wikander, "Demonstration based learning and control for automatic grasping," in *International Conference on Advanced Robotics*, Jeju, Korea, Aug 2007.

[16] M. Hersch and A. G. Billard, "A biologically-inspired controller for reaching movements," in *Proc. IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, Pisa, 2006, pp. 1067–1071.