

A Hand State Approach to Imitation with a Next-State-Planner for Industrial Manipulators

Alexander Skoglund*, Boyko Iliev*, and Rainer Palm†

Center for Applied Autonomous Sensor Systems,
Department of Technology, SE-701 82 Örebro University, Sweden

*{alexander.skoglund, boyko.iliev}@tech.oru.se

†rub.palm@t-online.de

Abstract—In this paper we present an approach to reproduce human demonstrations in a reach-to-grasp context. The demonstration is represented in hand state space. By using the distance to the target object as a scheduling variable, the way in which the robot approaches the object is controlled. The controller that we deploy to execute the motion is formulated as a next-state-planner. The planner produces an action from the current state instead of planning the whole trajectory in advance which can be error prone in non-static environments. The results have a direct application in Programming-by-Demonstration. It also contributes to cognitive systems since the ability to reach-to-grasp supports the development of cognitive abilities.

Index Terms—Programming-by-Demonstration, Hand State Approach, Next-State-Planner, Human Like Motion, Fuzzy Clustering.

I. INTRODUCTION

In this paper we present an approach to goal directed imitation of reaching and grasping, where the essential properties of the target object and the desired motion profile are obtained from demonstrations. Goal-directed motion primitives are needed in systems for Programming-by-Demonstration (PbD), where robots need to reproduce tasks defined by human demonstrations [11]. In the general case, it is not possible to create one-to-one mappings from human to robot motions due to the differences in body structures. Thus, the human demonstration of a grasping task cannot be used directly as a desired motion trajectory for the robot. Instead, it serves as a source of information about the target object as well as the basic properties of the arm and hand motions during reach and grasp. That is, the robot has to reproduce the task by using its own motion primitives.

Work by Lopes and Santos-Victor have addressed the issue of self explored motions (called sensory-motor maps) using a developmental approach in contrast to this we use the human demonstration to guide the robot from start [12].

To illustrate the approach, we consider a scenario where a human demonstrator has to teach an industrial robot equipped with a two finger gripper how to perform a reach-and-grasp-task. The demonstration is done with the teacher wearing a 6-D motion capturing device and a data-glove tracking hand and finger motions.

In our previous work [17] we have shown how a Pick&Place task is transformed from a human demonstration

into desired motion trajectories for the robot. However, the grasping part, i.e. how to pick up an object, was implemented in a fairly simple manner in order to match the limited capabilities of the robotic gripper. In this article, we suggest a general approach based on the concept of hand state, which is able to cope with complex grasping tasks and enables the robot to generalize experience from multiple demonstrations. Moreover, it addresses the problem of coordination of reaching and grasping motion primitives into a coherent goal-directed motion.

In our approach, we use the hand state hypothesis introduced by Oztop and Arbib in [14] as a part of the Mirror Neuron System model. According to the definition, the hand state trajectory describes the evolution of hand's pose and configuration in relation to the target object. That is, it encodes the goal-directed motion of the hand during reach and grasp. Thus, when the robot tries to imitate an observed grasp, it has to move its own hand in such a way that it follows a hand state trajectory similar to the one of the demonstration.

In this paper, we suggest a next-state-planner which is able to generate a hand state trajectory for the robot from a human demonstration, thus serving as a motion primitive. The planned trajectory preserves essential properties of the human motion, such as velocity profile and trade-off between speed and accuracy. The next-state-planner approach combines the planner and controller in one hybrid control architecture where the new next state is planned from the current state. This is in contrast to traditional approaches where the complete trajectory is pre-planned all the way to the target and then executed by the controller. Our use of a dynamical system differs to previous work (i.e., [6], [8] and [10]) in how we combine the demonstrated path with the robots own plan. The use of hand state trajectories distinguish our work from most previous work on imitation; most approaches seen in the literature use joint space for motion planning while some other approaches use the Cartesian space, see [8].

When a robot is required to imitate a demonstrated motion, problems and difficulties arise: 1) the recorded trajectory itself cannot be used since it contains noise and unwanted “wiggles” and unintended motions, 2) the robot may not start

from the same position as the human, that means the initial hand state is different and so will be the total trajectory, 3) the robot’s configuration differs from human configuration. From this follows that both amplitude and length of the trajectory will differ from the demonstrated motion, meaning that the generation of robot motion should be *similar* to the original motion - not the same.

In order to deal with the above problems we employ the hand state approach to imitation is developed using a so-called Next-State-Planner.

Grasping and manipulation are pointed out as a major problem towards the development of artificial cognitive systems. For example, basic reach and grasp skills are necessary for the robot to explore the environment and interact with it. The ability to interact with objects and learn about their properties plays a key role in the cognitive development of the robot since the perception and action are mutually dependent, see [19].

The rest of the paper is organized as follows: In Section II we review the hand state hypothesis and related work. Furthermore we describe methods for analyzing the demonstration. In Sec. III the trajectory generation based on the hand state-approach is presented. In Sec. IV we demonstrate the approach by experiments. Sec. V concludes the paper with a final discussion and future work.

II. MOTION ANALYSIS OF HUMAN DEMONSTRATIONS

The concept of hand state stems from research in neurophysiology, more specifically schema theory and it is introduced by Oztop and Arbib [14]. They propose a functional model of the mechanisms behind primate grasping. Furthermore, they introduce the Mirror Neuron System model to explain the function of the Mirror Neuron System in Brodmann’s brain area F5, related to grasping. The model uses the notion of affordances, defined as *object features relevant to grasping*. This suggests that the F5 mirror neuron system uses an internal representation of the action, which is independent of the owner of the action. In accordance to that, the hand state is defined as a vector whose components represent the movement of the wrist relative to the location of the object and of the hand shape relative to object’s affordances. Consequently, grasp motions are modeled with hand-object relational trajectories as internal representations of actions.

If we apply these concepts to learning by imitation, our approach can be outlined as follows. The robot tries to learn how to reach and grasp by observing a human teacher and imitating the motions of the teacher, assuming that it knows the goal of these motions. For this purpose, the observed motions are mapped into hand state space. The initial imitations are not necessarily successful due to inability to reproduce the observed motions or because of wrong assumptions about the target object and its properties. The ultimate goal is that the robot learns from experience how to imitate the motions and how to grasp successfully. The reach and grasp motions are then synthesized in hand state space and executed with

the help of motion primitives, activated in such way that the hand follows the desired hand state trajectory.

A. Hand State definition

The hand state describes the configuration of the hand and its geometrical relation the affordances of the target object. Thus, the hand state \mathbf{H} and its components are defined as:

$$\mathbf{H} = [h_1 \dots h_{k-1} h_k \dots h_p] \quad (1)$$

where h_1 to h_{k-1} are hand-specific components, and h_k to h_p are components describing the hand pose with respect to the object.

This formulation allows the representation of motions in terms of hand state trajectories. Note that the hand state components are specific to the hand’s kinematic structure and the definition of the object affordances. This means that some hand state components will be defined differently for different grasp types since they involve different object affordances. In the case of robots imitating humans, we have to define \mathbf{H} in such way that it matches the capabilities of particular types of end-effectors, e.g. dexterous robot hands, two-finger grippers, as well as the human hand.

In this article we adopted a general formulation in the sense that it suits most industrial robot grippers. We formulate the hand state as:

$$\mathbf{H}(t) = [\phi_f(t) d_n(t) d_o(t) d_a(t) \phi_n(t) \phi_o(t) \phi_a(t)] \quad (2)$$

The individual components concern the position and orientation of the end-effector as well as the opening and closing of the gripper, reflecting the limited capabilities of the gripper. The first component is the only hand-specific one, describing the angle between the thumb and the index finger. The next three components, $d_n(t)$, $d_o(t)$ and $d_a(t)$, describe the distance from the object to the hand along the three axes n , o and a with the object as the base frame. The next three components, $\phi_n(t)$, $\phi_o(t)$ and $\phi_a(t)$, describe the rotation of the hand in relation to the object as around the three axes n , o and a . This notation implies that all measures of the object are expressed in the object frame. Our notation of the hand state is illustrated in Fig. 1.

To determine the hand state representation of a demonstration, the robot needs to have access to the complete motion trajectories of the teacher’s hand since the motion must be in relation to the target object. This means the the hand state trajectories can not be computed on-line unless the target object is known in advance.

Note that with this simplified definition of \mathbf{H} we cannot determine the actual human grasp type. This reflects the limited capabilities of the gripper we use in our experimental setup. However, the method can be applied in a straightforward way also to complex robotic hands capable of several grasp types, see [9] for details.

Furthermore, the configuration of the human hand is used for grasp classification, described in earlier work ([15], [16]), where the grasps are classified according to the taxonomy in

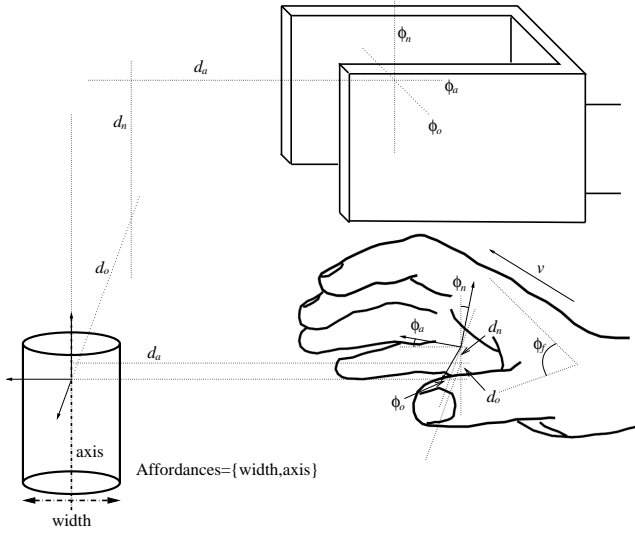


Fig. 1. The hand state is described in the object frame. This can only be determined after the demonstration is performed. A set of affordances are associated with each type of grasp; for the cylindrical grasp they are the center axis and the object width.

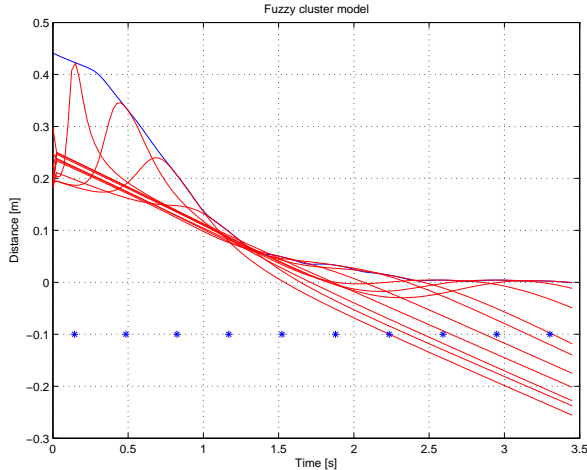


Fig. 2. Fuzzy clustering principle. The blue stars are the cluster centers positions, the lines are the output from each local linear model and the blue line is the total approximated model. The weights of each local model and the offsets are not shown.

[7]. The grasp type G , and its associated set of affordances A are used to provide the frame in which the hand state is described. The classification is out of scope of this paper, and we currently only investigated cylindrical grasps of one single item. Thus the grasp types will always be $G = \text{cylindrical}$; the affordances are, besides position, size and cylinder axis $A = \{\text{width}, \text{axis}\}$.

B. Modeling of Hand States

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. A TS fuzzy model is illustrated in Fig. 2.

There is evidence that the internal models of arm dynamics

found in biological systems state-dependent rather than time-dependent [2]. Therefore, when we transform human demonstrations into robot motions, we employ *distance to object*, d , as additional scheduling variable for hand state trajectories. However, to preserve the velocity profile from the human demonstration the distance to the target is modeled as a function of time using time clustering, see Fig. 3.

We apply TS fuzzy clustering on hand state trajectories, with d as input and $\mathbf{H}(d)$ as output:

$$\mathbf{H}(d) = [\phi_f(d) \ d_n(d) \ d_o(d) \ d_a(d) \ \phi_n(d) \ \phi_o(d) \ \phi_a(d)] \quad (3)$$

The role of the scheduling variable d is important since it expresses *when* to move to the next state. On the other hand, the hand state variables reflect *where* the hand should be. Thus, d synchronize *when* and *where*.

The TS fuzzy models are constructed from captured data from the end effector trajectory described by the nonlinear function:

$$\mathbf{x}(y) = \mathbf{f}(y) \quad (4)$$

where $\mathbf{x}(y) \in R^n$, $\mathbf{f} \in R^1$, and $y \in R^m$. The parameter y can be the time t or the distance d . Equation (4) is linearized at selected data points y_i with

$$\mathbf{x}(y) = \mathbf{x}(y_i) + \frac{\Delta \mathbf{f}(y)}{\Delta y} \Big|_{y_i} \cdot (y - y_i) \quad (5)$$

which results in a linear equation in y .

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

where $\mathbf{A}_i = \frac{\Delta \mathbf{f}(y)}{\Delta y} \Big|_{y_i} \in R^n$ and $\mathbf{a}_i = \mathbf{x}(y_i) - \frac{\Delta \mathbf{f}(y)}{\Delta y} \Big|_{y_i} \cdot y_i \in R^n$. Using (6) as a local linear model one can express (4) in terms of an interpolation between several local linear models by applying TS fuzzy modeling [18]:

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

$w_i(y) \in [0, 1]$ 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 $\sum_{i=1}^c w_i(y) = 1$.

The degree of membership $w_i(t)$ of an input data point y in an input cluster C_i is determined by

$$w_i(y) = \frac{1}{\sum_{j=1}^c \left(\frac{(y-y_i)^T M_{i\text{pro}}(y-y_i)}{(y-y_j)^T M_{j\text{pro}}(y-y_j)} \right)^{\frac{1}{\tilde{m}_{\text{pro}}-1}}} \quad (8)$$

The projected cluster centers y_i and the induced matrices $M_{i\text{pro}}$ define the input clusters C_i ($i = 1 \dots c$). The parameter $\tilde{m}_{\text{pro}} > 1$ determines the fuzziness of an individual cluster [5].

C. Adaptation of Takagi-Sugeno Models

From the demonstrations we model the motion as a TS fuzzy-model, where the distance is a function of time. However, the robot's motion time MT_{robot} will be differ from the demonstration since the initial conditions are different. MT_{robot} is computed at $t = 0$, before the robot starts executing the motion. To adapt the model to the robots initial condition, we need to perform three steps:

- 1) Normalize the cluster centers positions C and the offsets a with respect to the maximum cluster centers positions c_{max} , by:

$$\begin{aligned}\tilde{c}_i &= c_i/c_{max} \\ \tilde{a}_i &= a_i/c_{max}\end{aligned}\quad (9)$$

- 2) Compute the new cluster centers C_{new} and offsets a_{new} by multiplying them by MT_{robot} ,

$$\begin{aligned}c_i^{new} &= MT_{robot} * \tilde{c}_i \\ a_i^{new} &= MT_{robot} * \tilde{a}_i\end{aligned}\quad (10)$$

- 3) Compute a scaling factor k by:

$$\begin{aligned}y_0 &= \sum_{i=1}^c w_i(t_0) \cdot (\mathbf{A}_i \cdot t_0 + \mathbf{a}_i) \\ k &= \frac{MT_{robot}}{c_{max}} \cdot \frac{d(t_0)}{y_0}\end{aligned}\quad (11)$$

where y_0 is the output for t_0 using Eqn. 7, the first term in Eqn. 11 represents scaling along the input dimension (time), and the second scaling along the output dimension (distance).

The new model becomes:

$$\mathbf{x}(y) = \frac{1}{k} \sum_{i=1}^{c_{new}} w_i(y) \cdot (\mathbf{A}_i \cdot y + \mathbf{a}_i^{new})\quad (12)$$

The original model, top graph in Fig. 3, is scaled using the above algorithm to a new duration shown in Fig. 3, bottom.

D. Variance in Hand State

From a number of demonstrations of the same motion it is possible to extract information on how exact it must be followed and what boundaries exist in Cartesian coordinates. By looking at the variance we can draw conclusions about the importance of each component of the hand state during reach-to-grasp. In other words, how much can the trajectory deviate from the demonstrated one at certain distance. The variance of the hand state at distance d to target is given by:

$$\text{var}(^k h(d)) = \frac{1}{n-1} \sum_{i=1}^n ({}^k h_i(d) - \text{mean}({}^k h(d)))^2\quad (13)$$

where d is the Euclidean distance to the target, h_i is the k :th hand state parameter of i :th demonstration (see Eqn. 2) and n is the number of demonstrations. Fig. 4 show how the variance increases as the distance to the object increases, which means that the hand's position and orientation is less

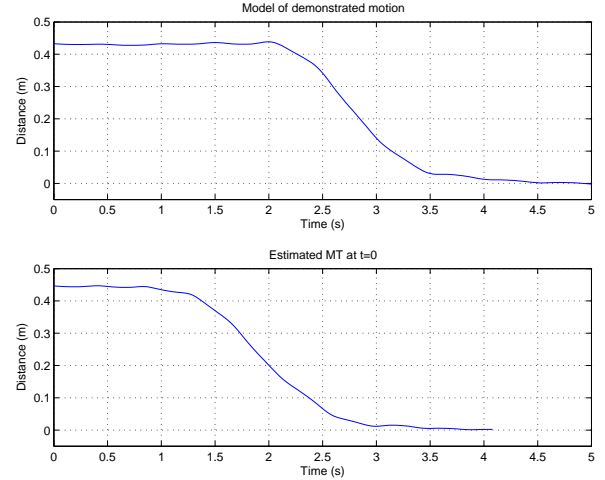


Fig. 3. The top graph shows the original model of the desired distance to the target as the time evolves. The bottom graph shows how the model is adapted to the robots domain, this estimate is made at the initial position, and might change during the execution.

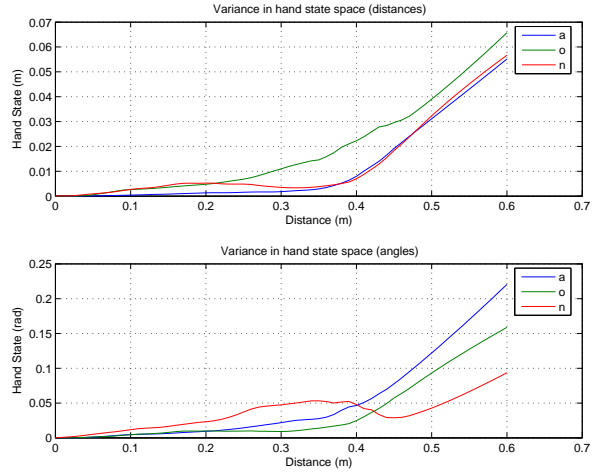


Fig. 4. Position- and orientation-variance of the hand state trajectories as function of distance, across 24 demonstrations of a reaching to grasp motion. Note that distances over 0.47 are extrapolations made by the clustering method.

relevant when the distance to the target increases. Recall that to compute the hand state the object's position must be known, so we use the hand's final position for this purpose. Therefore, the variance at zero distance is zero.

E. Characteristics of Human Motions

The time taken to reach a target is related to the movement's amplitude. In the case of reach to a target the amplitude is equal to the distance to the target at the start of the motion $A = d(t_0)$. The movement's duration is also related to the width of the target. This relation is formulated by Fitts' law [4] as:

$$MT = a + b \log_2 \left(\frac{2A}{W} \right)\quad (14)$$

where MT is the duration of the motion, A is the amplitude of the motion, equal to distance to the target at the start of the motion $d(t_0)$, W is the width of the object and a and

b are coefficients. Fitts' law describes the trade-off between speed and accuracy of the motion.

From a set of demonstrations where MT , $d(t_0)$ and W are known we can solve for the coefficients a and b . When a robot is instructed to imitate the human demonstration we cannot assume that the amplitude of the motion will be the same as in the demonstration. Thus we need to compute the desired duration of the robot's motion MT_{robot} , if we want to preserve the characteristics of the human motion. Since we know the width of the object, the initial distance to the object, and the coefficients a and b , we can apply Fitts' law (Eqn. 14) to compute the time MT_{robot} .

III. GENERATION OF ROBOT TRAJECTORIES

In this section we present the next-state planner, which generates a hand state trajectory for the robot using the TS fuzzy-model of the demonstration. It serves as a motion primitive for controlling the arm's reaching motion. Fig. 5 shows the architecture of the next-state-planner. The planner works as follows. Firstly, the distance to target is computed, $d(t_0) = A$, and the duration (MT) of motion is computed using Fitt's law (see Eqn 14). Since the position and affordances of the object are assumed to be known, the initial hand state of the robot can be computed for initialization. Secondly, the desired distance d^* is obtained from the TS fuzzy-model (Eqn. 12), and the desired hand state \mathbf{H}^* from the TS fuzzy-model (Eqn. 7).

The dynamical system we use as hand state controller is similar the VITE controller suggested by Bullock and Grossberg [1] and the dynamical systems used by Ijspeert [8]. But instead of only having a goal attractor as in VITE, or use internal dynamic variables as Ijspeert et. al., we use a second additional attractor: the desired hand-state trajectory. The system has the following dynamics:

$$\ddot{\mathbf{H}}_{t+1} = \alpha(-\dot{\mathbf{H}}_t + \beta(\mathbf{H}_G - \mathbf{H}_t) + \gamma(\mathbf{H}^*(d) - \mathbf{H}_t)) \quad (15)$$

$$\dot{\mathbf{H}}_{t+1} = \dot{\mathbf{H}}_t + \ddot{\mathbf{H}}_{t+1} \quad (16)$$

$$\mathbf{H}_{t+1} = \mathbf{H}_t + \dot{\mathbf{H}}_{t+1} \quad (17)$$

where \mathbf{H}_G is the goal, $\mathbf{H}^*(d)$ the desired value at distance d , \mathbf{H}_t is the state at time t , $\dot{\mathbf{H}}$ and $\ddot{\mathbf{H}}$ are the velocity and acceleration, α is a positive constant and β , γ are positive weighting parameters where β weights the goal and γ tracking point. The goal hand state \mathbf{H}_G is obtained from the fuzzy clustering model at the estimated final time step. The desired hand state value $\mathbf{H}^*(d)$ at distance d is computed using the desired distance at the current time step and use the TS-model at that distance. The weights β and γ are variables reflecting the importance of the goal versus the path, acquired from variance, see Sec. II-D.

The next-state planner uses the models built from the demonstration to generate a hand state trajectory similar to the demonstrated one. The hand state trajectory for a reaching motion is modeled as a function of the distance to the object. Hence, the closer to the object we are the more important it becomes to follow the desired trajectory. On the

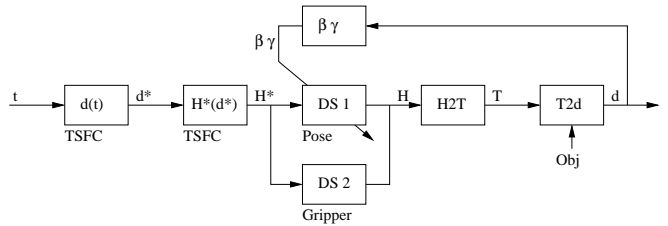


Fig. 5. Controller architecture. t is the time, d^* is the desired distance to target computed from the TS-model, $\mathbf{H}^*(d)$ is the desired hand state computed from the TS-model, “DS1” and “DS2” are dynamical system controlling the arm’s pose and the gripper’s state respectively, H2T transforms the hand state into Cartesian coordinates, “d2T” computes the Euclidean distance to the target and “ $\beta \gamma$ ” computes β and γ from d .

basis of the variance in the hand state (see Sec. II-D), we can conclude that a shorter distance to the target decreases the variance in hand state, approximately following a square curve. This property is reflected in the dynamical system by adding a higher weight to the trajectory-following dynamics the closer we get to the target; in reverse a long distance to target leads to a higher to the goal directed dynamics, and low weight it follow some specified path to the goal. The outcome from these weighting parameters is that at some distance to the target the goal is to reach the target, without any trajectory constrains. The closer we get to the target the goal becomes less important and the path becomes more important. The final position of the path is the same point as the goal position, thus the point of attraction becomes the same at the end of the motion.

The controller has an open loop structure as in Fig. 5. The reason for this structure is that a commercial manipulator usually has a closed architecture, where the controller is embedded into the system. For this kind of manipulators a fixed trajectory is usually pre-loaded. Therefore, we generate the trajectories in batch mode for this platform. Since our approach is general, for a given different robot platform with hetroceptive sensors (e.g., vision) our method can be implemented in an online mode.

IV. EXPERIMENTAL EVALUATION

To illustrate our approach we performed demonstrations of a reaching motion. Simulations were made in MATLAB using “Robotic Toolbox” [3]. We performed two experiments:

- 1) generate a new trajectory based on one of the models from the recorded trajectories but in the robots work (covered in Sec. IV-C),
- 2) generate trajectories from the robot’s home position to 100 randomly placed objects (covered in Sec. IV-D).

As experimental platform we used one motion capturing system for the human demonstration and an industrial serial manipulator (ABB IRB140) equipped with a two finger gripper. In these experiments we only simulated the manipulator.

A. Data capturing of Human Demonstration

The experimental platform we use consists of Immersion CyberGlove with 18 sensors to capture the hand configuration (only yaw and pitch data are needed to position the grasp center point; the remaining sensors are used for



Fig. 6. The experimental setup with the data glove and the 6D tracker used for data collection.

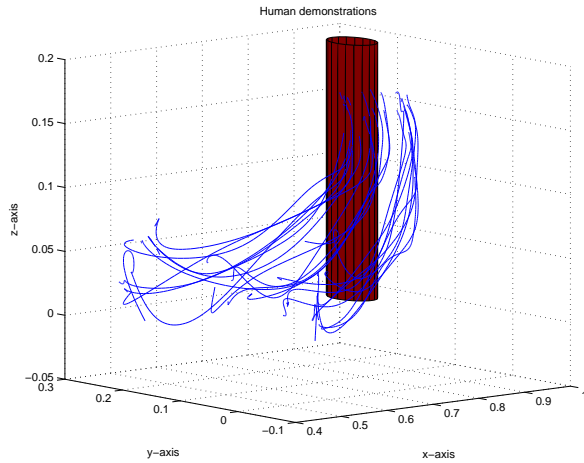


Fig. 7. Trajectory of a reaching and grasp task. 24 demonstration reaching for a cylindrical grasp.

measuring finger joint angles) and a Polhemus FASTRAK 6 DOF magnetic tracker that captures the wrist's position and orientation. Fig 6 shows the experimental setup. Compared to vision systems the use of wearable sensors are somewhat impeding for the user but provide several important benefits. They are generally more accurate and do not suffer from occlusion. Moreover they provide fast and simple computation by placing sensors directly on the measure points avoiding ambiguities. In our experiments, a demonstrator performs a set of grasps of a cylindrical object (bottle) located at a fixed position (C1). The grasp begins from an arbitrary position in the work-frame. The demonstrator's arm/hand moves toward the object and the grasp ends when the object is firmly caught in the hand. Then, the hand moves away from the object and rests before repeating the task from another initial position. The motions are automatically segmented into reach and retract motions using the velocity profile, reach motions are used for these experiments. The data representing the motion of the hand toward the object while performing the grasp is collected. A set of paths for hand motions are shown in Fig. 7.

B. Generation of Hand State Trajectories

From the recorded demonstration we reconstruct the end effector trajectory as it moves, represented by a time dependent homogeneous matrix $T_{ee}(t)$. Each element is represented by the matrix:

$$T_{ee} = \begin{pmatrix} \mathbf{N}_{ee} & \mathbf{O}_{ee} & \mathbf{A}_{ee} & \mathbf{Q}_{ee} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (18)$$

We can also extract affordances of the grasped object. When the cylindrical grasp is recognized using grasp recognition (see [15]) we extract the affordances associated with a cylinder. These are position $\mathbf{C} = [C_x C_y C_z]^T$, radius R and cylinder axis $\mathbf{P} = [P_x P_y P_z]^T$ (height is not possible to extract using our current platform). The position \mathbf{C} is determined by the $T_{ee}(t_f)$, where t_f is the time at the final position. The cylinder axis is determined by the orientation vector \mathbf{O}_{ee} of T_{ee} .

Now, recall the definition of the hand state from Eqn. 1, which is defined in relation to the object C , thus T_{ee} is represented in the object frame T_{ob} , which is $T_{ee}(t_f)$. The transformation to the end effector frame relative to the object frame is expressed by:

$${}^{ob}T_{ee} = T_{ob}^{-1} \cdot T_{ee} \quad (19)$$

where T_{ee} is the hand frame and T_{ob} is the object frame. These transformations are illustrated in Fig. 8.

The hand state representation (Eqn. 2) is invariant with respect to the actual location and orientation of the target object. Thus, grasp demonstrations of objects at different locations and initial conditions can be represented in a common domain. The object domain is both the strength and weakness of the hand state approach; a change in Cartesian space does not result in a change in hand state space, since it describes motions in relation to objects; when an object is firmly grasped the hand state is fixed and will not capture a change in Cartesian space. So for object handling and manipulation the use of hand state space is limited.

The input to the fuzzy clustering are the hand state parameters and the Euclidean distance at each instance.

$$d_E = \sqrt{(\mathbf{Q}_{ee} - \mathbf{C})^2} \quad (20)$$

where \mathbf{Q}_{ee} and \mathbf{C} are the end effector position and object position respectively.

C. Experiment 1

In our first experiment we generate motions that imitates one of the demonstrations, but statistics from all demonstrations are used. That is, one of the trajectories, $\mathbf{H}(t)$ (shown in Fig. 7) is used to generate a robot trajectory. The final position of the path is the same for the robot as in the demonstration, but it starts from a different initial position. In Fig. 9 the hand state trajectories from four demonstrations are displayed together with the corresponding generated trajectory for the robot. Despite the different initial conditions the generated trajectory converges towards the imitated path.

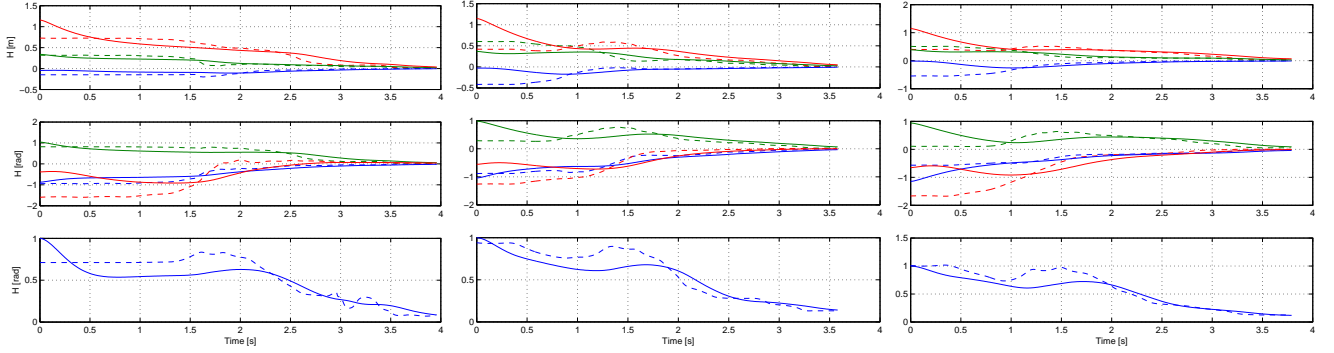


Fig. 9. Four sample demonstrations, and the corresponding imitations. The solid lines are the trajectories produced by the controller, dashed lines are the recorded from the demonstration. In the top and middled graphs, blue is distance and rotations in n -axis, red o -axis and green a -axis. In the bottom graphs grippers hand state is shown. Notice how the gripper's hand state is synchronized with the other hand states.

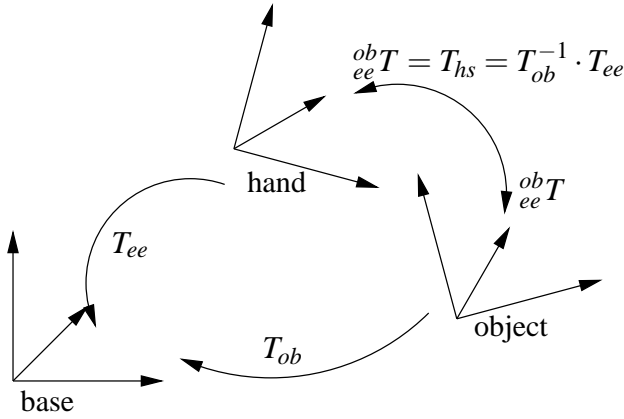


Fig. 8. The transformation between different frames. T_{ee} is the end effector transformation, T_{ob} is the object transformation and T_{hs} is the transformation from object to hand, which describes the a part of the hand state.

Fig. 10 shows the generated and demonstrated reaching motions towards objects for four recorded trajectories.

D. Experiment 2

In our second experiment trajectories are generated toward objects placed at random positions within the reachable area of the work space. We assume that these objects have observable affordances acquired by some other means than the demonstration (e.g., vision, intelligent environment, CAD models).

In total, 1425 of 2200 trail trajectories have been classified as successful. A success is defined as reaching a target in less than 1.5 times the estimated time *and* the distance to the object less than 0.01 m from the intended distance.

No.	1	2	3	4	5	6	7	8
SR	100%	99%	99%	99%	99%	31%	99%	99%
Var	0.08	0.06	0.16	0.03	0.11	0.77	0.22	0.11
No.	9	10	11	12	13	14	15	16
SR	8%	7%	42%	99%	99%	94%	87%	5%
Var	0.30	0.95	0.45	0.02	0.12	0.20	0.12	1.08
No.	17	18	19	20	21	22		
SR	1%	13%	79%	15%	99%	52%		
Var	0.00	0.48	0.17	0.18	0.42	0.30		

“SR” means success rate and “Var” is the variance of movement duration. Model 1-5, 7, 8, 12 and 13 succeeded in reaching the target 99-100% of the time and with a low

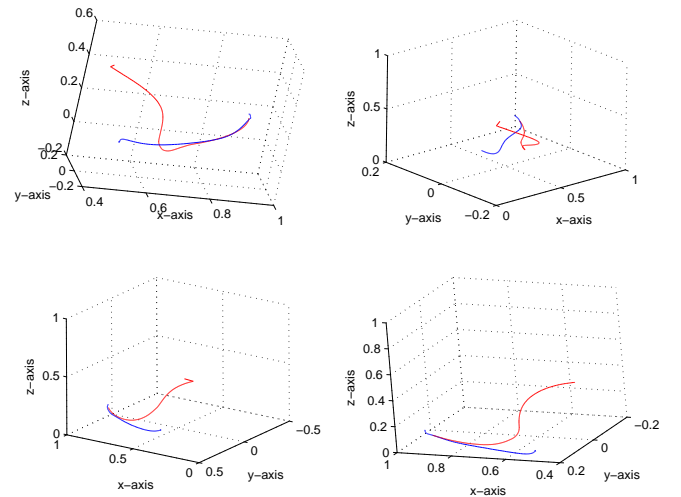


Fig. 10. Three sample trajectories. The blue line is trajectory executed by the robot, the red the demonstrated trajectory to follow. (The object is removed to not occlude the path.)

variance. On the other hand, model number 9, 10, 16 and 17 succeeded in less than 10% of the trials and have a high variance (a “normal” reaching lasts approximately 2 s). Model number 21 have a high success rate but also a high variance. The variance in hand state over the 1425 trials is shown in Fig. 11.

By testing each model we can decide if it performs satisfactory or not. In future an automatic performance measure will enable a developmental aspect of reaching motions (see [13] for a review), e.g., a reinforcement learning framework.

V. DISCUSSION AND CONCLUSIONS

In this paper we have presented an approach to encoding reach-to grasp trajectories, obtained from demonstration, based on fuzzy clustering. The method implements a combined hybrid control architecture operating in hand state space. The three main features of the approach are:

- 1) a dynamical system is used to generate goal directed motions balanced between following a demonstrated trajectory and fast approach of the target;
- 2) trajectories are *indirectly* time depended since we encode distance to the object as a time variable;

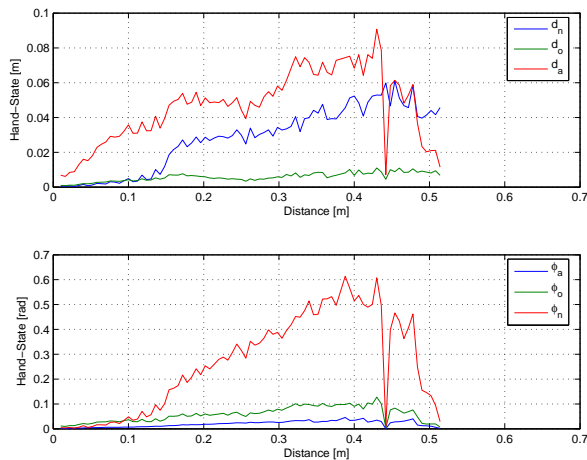


Fig. 11. The variance over 1425 trajectories where the models succeeded in reach-to-grasp.

3) Modeling in hand state space results in position- and orientation-invariant representation of trajectories. Moreover, it provides a natural way of synchronization of reach and grasp.

The fuzzy time-clustering techniques has the following advantages:

1) a compact representation by cluster centers, gains, weight and offsets and a simple linear interpolation between two cluster centers;

2) the simple algorithm in which cluster centers can be modified to encode a desired trajectory *similar* to the original demonstration.

The results of this work have direct application in Programming-by-Demonstration of robots. They also contribute to the fields of humanoid robotics and cognitive systems since the ability to reach to grasp enables interaction with the environment and supports the development of cognitive abilities.

As a part of our future work, we will test the hand state approach on different type of objects. The aim is to expand the approach for the most common human grasp types and find appropriate definitions of the corresponding affordances. Furthermore, more experimental data will also test our hypothesis on using the Euclidian distance to the object as a state parameter; e.g., circular motions will need additional parameters. Future work also include the implementation on a real industrial robot system, where possible weaknesses of the approach would be seen.

The ability to forget a model that performed badly or produce a very similar trajectory compared to other models will also be part of our future work. A pruning ability will prevent the database of models from becoming too large. Another direction for future research is to include robot's own perception in the learning process, e.g. to recognize and track objects and their affordances. In this way the robot would not need to blindly rely on human demonstrations and would be able to work in dynamic environments. In this respect, we are currently investigating the capability of this

approach for motion prediction from observations. This will incorporate intention recognition into our framework, which is a key capability for cognitive robotic systems.

REFERENCES

- [1] D. Bullock and S. Grossberg, "VITE and FLETE: Neural modules for trajectory formation and postural control," in *Volitional Action*, W. A. Hershberger, Ed. Elsevier Science Publishing B.V. (North-Holland), 1989, pp. 253–297.
- [2] M. Condit and F. A. Mussa-Ivaldi, "Central representation of time during motor learning," in *Proc. Natl. Acad. Sci. USA*, vol. 96, September 1999, pp. 11 625–11 630.
- [3] P. Corke, "A robotics toolbox for MATLAB," *IEEE Robotics and Automation Magazine*, vol. 3, no. 1, pp. 24–32, Mar. 1996.
- [4] P. M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology*, vol. 47, no. 6, pp. 381–391, June 1954.
- [5] D. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," *Proceedings of the 1979 IEEE CDC*, pp. 761–766, 1979.
- [6] M. Hersch and A. G. Billard, "A biologically-inspired controller for reaching movements," in *Proc. IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics*, Pisa, 2006, pp. 1067–1071.
- [7] T. Iberall, "Human prehension and dexterous robot hands," *The International Journal of Robotics Research*, vol. 16, pp. 285–299, 1997.
- [8] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, 2002, pp. 1398–1403.
- [9] B. Iliev, B. Kadmiry, and R. Palm, "Interpretation of human demonstrations using mirror neuron system principles," in *Proceedings of the 6th IEEE International Conference on Development and Learning*, Imperial College London, 11-13 July 2007.
- [10] I. Iossifidis and G. Schoner, "Autonomous reaching and obstacle avoidance with the anthropomorphic arm of a robotic assistant using the attractor dynamics approach," in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, April 2004.
- [11] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 799–822, 1994.
- [12] M. Lopes and J. Santos-Victor, "A developmental roadmap for learning by imitation in robots," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 37, no. 2, pp. 308–321, April 2007.
- [13] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, no. 4, pp. 151–190, December 2003.
- [14] 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.
- [15] 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: IEEE, July 16-21 2006, pp. 291–298.
- [16] —, "Segmentation and recognition of human grasps for programming-by-demonstration using time-clustering and fuzzy modeling," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, London, UK, July 23-26 2007.
- [17] 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.
- [18] T. Takagi and M. Sugeno, "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.
- [19] D. Vernon, "D2.1 a roadmap for the development of cognitive capabilities in humanoid robots," Project deliverable, EU-IST, Project No. 004370, 2007.