

# Compact Models of Human Reaching Motions for Robotic Control in Everyday Manipulation Tasks

Freek Stulp, Ingo Kresse, Alexis Maldonado, Federico Ruiz, Andreas Fedrizzi, Michael Beetz  
Intelligent Autonomous Systems Group, Technische Universität München, Germany

**Abstract**—Autonomous personal robots are currently being equipped with hands and arms that have kinematic redundancy similar to those of humans. Humans exploit the redundancy in their motor system by optimizing secondary criteria. Tasks which are executed repeatedly lead to movements that are highly optimized over time, which leads to stereotypical [25] and pre-planned [15] motion patterns. This stereotypical motion can be modeled well with compact models, as has been shown for locomotion [1]. In this paper, we determine compact models for human reaching and obstacle avoidance in everyday manipulation tasks, and port these models to an articulated robot.

We acquire compact models by analyzing human reaching data acquired with a magnetic motion tracker with dimensionality reduction and clustering methods. The stereotypical reaching trajectories so acquired are used to train a Dynamic Movement Primitive [12], which is executed on the robot. This enables the robot not only to follow these trajectories accurately, but also uses the compact model to predict and execute further human trajectories.

## I. INTRODUCTION

Autonomous personal robots are currently being equipped with hands and arms that are increasingly similar to those of humans, with respect to workspace, motion characteristics, and degrees of freedom [24], [16]. As the internal (joint) space of these robots is much higher than the external (task) space, the resulting kinematic redundancy is a valuable resource for optimizing robot motion. In recent years we have seen a number of powerful, high-performance motion planning approaches that are capable of searching the high-dimensional space of motion plans and can produce effective and often efficient plans, even in cluttered scenes [2].

Unlike these motion planning approaches that tend to generate specific motion plans for every manipulation task, humans usually perform these manipulation tasks with highly stereotyped movement patterns [25].

There are good reasons that the motions of robots for everyday manipulation tasks in the presence of, and in co-operation with, humans should be similar to human reaching behavior. First, robots with human-like motion will enable humans to more easily perform perspective taking and intention recognition [17]. This is necessary to enable implicit coordination (which humans use when coordinating their actions) in joint human-robot tasks. Second, we believe that robots acting in human environments have to learn advanced manipulation skills through imitation learning [21], [3]. The task of transferring observed reaching and grasping behavior into the robot's motion control system becomes much

easier if both motion control systems apply similar control strategies. Third, humans can sequence motion primitives seamlessly [7]. We expect that robots that have the same motion primitives as humans are able to achieve similar smooth execution of motion primitives [23].

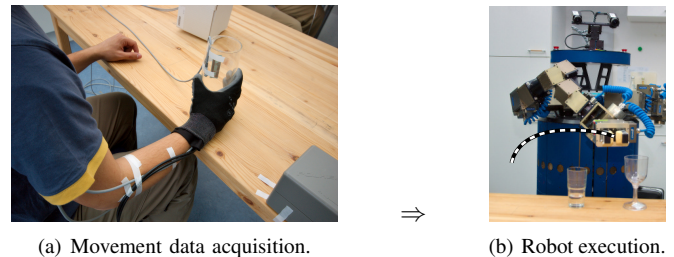


Fig. 1. Translating observed human movements to robot trajectories using compact models.

In this paper we investigate the human reaching behavior for object grasping in simple situations with one obstacle. The purpose of this investigation is the rational reconstruction of a computational model that can predict human reaching behavior by first predicting the motion strategy that the human will apply and then predicting the particular motion pattern that results from the application of the strategy to the particular reaching task. We implement the computational model and apply it to an autonomous robot with a 6-dof arm to produce very similar reaching behavior.

We consider the possibility of specifying compact control programs for robot reaching, that are capable of producing behavior that shows many of the advantages of human reaching behavior, a powerful computational mechanisms for the realization of robots that are to perform everyday manipulation tasks, as they are required in the course of housework, for example.

The key contributions of this paper are the following: 1) We derive from experimental data a computational model for human reaching behavior for objects standing on a plane with a single obstacle with varying location. This model consists of a strategy selection component that decides whether the obstacle can be ignored, or which strategy should be applied to reach around the obstacle. The second component predicts the motion trajectory for each strategy with high probability and accuracy. 2) As a proof of concept, we realize this compact model on an autonomous manipulation platform to produce similar reaching behavior.

The rest of this article is structured as follows. In the next section, we discuss related work. In Section III we describe the experimental design for recording data of human reaching trajectories, and in Section IV we explain how stereotypical trajectories are derived from this data. How these stereotypical trajectories are ported to the robot is described in Section V. We conclude with a summary and outlook in Section VI.

## II. RELATED WORK

One inspiration for this work has been the work by Arechavaleta et al. [1] on modeling human locomotion trajectories with clothoids. With these compact models with only a few parameters, it is possible to describe well the large variation in walking to different positions, but also in between different subjects. In this paper, we apply a similar approach to the domain of reaching and manipulation, and port these compact models to a robot.

Considerable work on determining the influence of obstacles on reaching motion has been done in experimental psychology. In [5] for instance, two obstacles of with different heights are placed on 4 pairs of positions on a table (16 combinations in total). Subjects then reach for a target object at a fixed position, and the trajectories are recorded with two OPTOTRAK cameras. The results also show that certain obstacle positions have no influence on the trajectories, whereas others do have predictable effects. Similar studies use features of trajectories to determine if a trajectory is affected by an obstacle, such as lateral deviation from the default behavior in the  $xy$ -plane [5], or movement time, maximum grip aperture and maximum speed [15]. However, to derive compact models for robot control, we need to consider the trajectories as a whole, and cannot reduce it to only several features.

In contrast to our approach, which uses pre-planned stored trajectories, classical motion-planners such as Rapidly-exploring Random Trees (RRTs) rather perform a novel search for each situation by incrementally exploring the state space until a goal state is found. Although these planners can find solutions even in very cluttered environments, they often disregard previous search results and cannot guarantee smoothness or (even local) optimality. We believe that the two approaches complement each other well. Using search is appropriate when novel complex environments are encountered, but using a set of standard prototypical trajectories to solve standard situations which are frequently encountered is beneficial as 1) these trajectories can be smoothed and optimized incrementally with every execution, so as to achieve a level of performance and reliability in everyday activities which is hard to achieve with single-query approaches [25]; 2) a potentially expensive search is no longer required. Existing experiments on obstacle avoidance in humans support the hypothesis that obstacle avoidance is not performed on-line, but that reaching movements are rather pre-planned to take potential collisions with obstacles into account. For an overview, we refer to [15].

Imitating trajectories was first used in the context of industrial robots 30 years ago, in very constrained task contexts and with fixed goals. By using Dynamic Movement Primitives to model the trajectories, scaling to novel goals is possible. Furthermore, we relate external task relevant parameters (the position of the target object) to internal motor parameters, i.e. which trajectory to use to avoid the obstacle.

A related approach uses Gaussian mixture models to encode a set of trajectories [4]. One main difference to our approach is that Calinon et al. use kinesthetics (i.e. the human teacher moves the robot's actuators), whereas we use human motion data. As we strive for natural human-like motion, human motion data is essential to our approach. As to the methodology, Calinon et al. model the variance in the trajectory sets with Gaussian mixture models. One downside of this approach is that unwanted averaging effects may arise when multi-modal solutions exist. For instance, for several obstacle positions, the subject usually chooses avoid the obstacle by going around it on the left side, but sometimes also on the right side. The average, going in between, would lead to a certain collision. We deal with multi-modality by clustering the trajectories before processing them further.

## III. EXPERIMENTAL DESIGN AND DATA ACQUISITION

The goal of this experiment is to answer quantitatively 1) at which positions do obstacles lead to human reaching behavior which is different from the default behavior when no obstacles are present? 2) which reaching strategies do humans use to avoid the obstacle?

The reaching motions were captured with a Polhemus Liberty magnetic position/orientation tracker. One sensor was attached to the hand, as depicted in Figure 2, and another sensor was attached to the glass to measure the exact time when the lifting movement started. Before performing the experiment we used one sensor to measure the positions of the obstacle grid in the tracker's coordinate frame. All sensors are tracked with very high precision ( $<0.01\text{mm}$ ) at 240Hz.



Fig. 2. Experimental set-up (left), and the location of the marker (right).

In the experiment, the subject sits at a table, and is asked to repeatedly reach for, grasp, and lift a target glass. The hand always starts in the black square in Figure 3. Before each reaching motion, an obstacle glass is placed on different positions on a grid on the table. The grid is 40x80cm. In Figure 3 for example, the obstacle glass is at position  $D6$ . For convenience, we sometimes informally refer to 'the trajectories that arose from the reaching movement

that was performed when the obstacle was placed at position *D6*<sup>1</sup> as ‘the *D6*-trajectories’. The target glass is always at position *B4*. The obstacle glass was placed 10 times on each of the 29 positions. Furthermore, 30 reaching motions were performed without any obstacle glass. These are the ‘*default*-trajectories’. The total number of reaching motions is therefore  $29 \times 10 + 30 = 320$ . The order of obstacle placement was random, to avoid learning effects. Please also see the accompanying video showing the experimental setup.

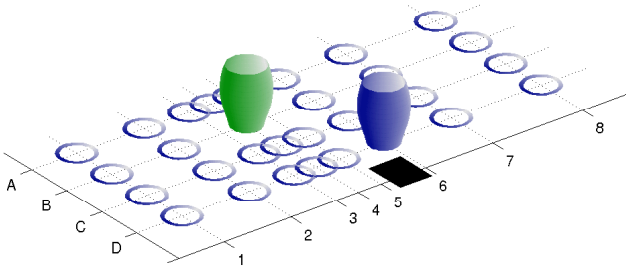


Fig. 3. Positions of the obstacles on the table. The green glass is the target glass, which is always at position *B4*. The blue glass is an example obstacle at position *D6*. The flat black region is the initial location of the fingers.

Several post-processing steps were needed to prepare the data for analysis. First of all, the trajectories are automatically cropped so that they only contain the relevant reaching trajectory, i.e. from the initial movement of the hand until the movement of the glass (which was also tracked). To compensate for slightly different target glass positions, the trajectories are then translated so that the center of the base of the initial glass position is at (0,0,0). The position of the hand is then rotated and scaled such that it always starts at (0,-42,0). This common frame of reference facilitates comparison. The scaling changes the velocity profile of the trajectory, but since we do not compare velocities, this is acceptable. Finally, all trajectories are resampled using spline interpolation so that they contain 100 samples.

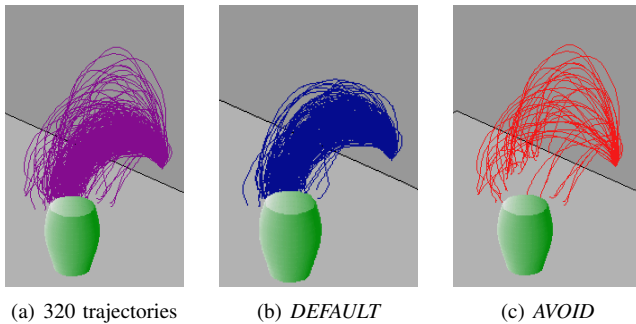


Fig. 4. All trajectories (left) segmented to according whether they are influenced by the obstacle (AVOID) or not (DEFAULT).

#### IV. DETERMINING COMPACT MODELS

Compact models are determined in two steps. First, we determine when the presence of the obstacle influences the reaching trajectory, by comparing sets of trajectories in a

lower-dimensional PCA space. Then we perform a clustering on the trajectories that are influenced by the obstacle, to determine different strategies for avoiding the obstacle.

##### A. Discerning between affected and unaffected trajectory sets

If obstacles are far away from the target glass (e.g. at positions *A1* or *C8*), we expect them not to have an influence on the reaching motion. Therefore, we expect the *A1*- and *C8*-trajectories to be very similar to the *default*-trajectories. In this section, we describe a distance measure between sets of trajectories, and use it to discern between affected and unaffected sets.

We used the trajectory comparison approach described Roduit et al. [19]. Here, the difference measure between two sets of trajectories is computed by 1) computing a point distribution model of the two sets of trajectories 2) taking only the first  $n$  components, by inspecting the eigenvalues of the covariance matrix of the merged trajectories 3) computing the Mahalanobis distance between the coefficients of the two sets of trajectories. A more detailed explanation of this method can be found in [19].

Suppose we have 2 sets, each containing 10 trajectories in 3D space. First, these trajectories are merged into one matrix  $\tau$  of size  $300 \times 20$ , where the columns are the concatenation of the  $x$ ,  $y$  and  $z$  coordinates of 100 samples along the trajectory, and each row represents one such trajectory. The next step is to compute  $\mathbf{P}$ , which is the matrix of eigenvectors of the covariance matrix of  $\tau$ . Given  $\mathbf{P}$ , we can decompose each trajectory  $\tau_k$  in the set into the mean trajectory and a linear combination of the columns of  $\mathbf{P}$  (called deformation modes) as follows  $\tau_k = \bar{\tau} + \mathbf{P} \cdot B_k$ . This procedure is called point distribution model analysis.

$B$  is then split into the coefficients for the original 2 sets of trajectories:  $B_{1..10}$  and  $B_{11..20}$ . By inspecting the eigenvalues of the covariance matrix of  $\tau$ , we determined that the first 5 components suffice to reconstruct and describe the trajectories well, so only the first 5 deformation modes are used. This reduces the high dimensionality of the initial 300D trajectory sets substantially, and facilitates comparison.

Finally, the distance measure  $d$  is the Mahalanobis distance between  $B_{1..10}$  and  $B_{11..20}$ . Details on this method can be found in [19].

This distance measure  $d$  is computed between all sets of trajectories  $A..D1..8$ , and the *default*-trajectories. The height of the glasses in Figure 5 represents  $d$  for the set of reaching motions when the obstacle was at that position. We automatically determine an appropriate threshold on  $d$  (called  $d_{thres}$ ) by determining the valley point of the histogram of the  $d$  values<sup>1</sup>. The distance  $d$  between the *C5*-, *D6*-, *D5*-, and *C4*-trajectories and the *default*-trajectories is higher than this threshold. These four positions are depicted as red glasses in Figure 5.

<sup>1</sup>The distance measure  $d$  can be linked to the Hotelling's  $T^2$  statistic. Because the variances of different trajectory sets were quite different in our data, we could not use the two-sample Hotelling's  $T^2$  statistic, as assumes equal variances. That is why we are required to determine a threshold.



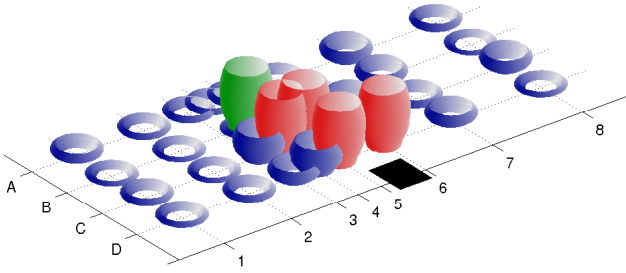


Fig. 5. The height of the glass represents  $d$  for that obstacle position.  $d_{thres}$  is 16.4 for this graph. The red glasses (at C5, D6, D5, C4) lie above this threshold.

We now create two large sets of trajectories. The *DEFAULT*-trajectories contain the *default*-trajectories, and also all trajectories for which  $d < d_{thres}$ . The *AVOID*-trajectories contains the remaining 40 trajectories (C5, D6, D5, C4). Both sets are depicted in Figure 4.

#### B. Determining prototypical trajectories with clustering

The next step is determining prototypical trajectories that represent qualitatively different strategies for avoiding the obstacle. The first strategy is the *DEFAULT* strategy, which is the mean of the all the trajectories in the *DEFAULT* set, i.e. those that were very similar to the trajectories that arose when no obstacle was present. For the *AVOID*-trajectories, we perform a  $k$ -means clustering on the 40 trajectories in the set<sup>2</sup>. As Jenkins et al. [13], we perform the clustering using several different spaces, and compare the results. We use

- a 300D space, in which the 100  $x$ ,  $y$  and  $z$  coordinates are simply concatenated for each trajectory.
- a 3D PCA space, being the first three deformation modes as defined in the previous section.
- a 3D space computed with Local Linear Embedding (LLE) [20] from the 300D space (using 10 neighbors).

In each space, the distance between two trajectories is determined by the angle between the two  $n$ -dimensional vectors representing the trajectories. The number of clusters is set to 3 manually. The clustering algorithm in the 300D space yields the three clusters depicted in Figure 6.

Clustering in the three spaces yield almost exactly the same clusters (the three spaces only disagree on the categorization of 3 trajectories). This implies that 1) these clusters are good stereotypes, and do not just depend on the clustering space or method; 2) only a very compact representation in 3 dimensions are needed to determine these stereotypes.

<sup>2</sup>The reason why we do not include all 320 trajectories in the clustering, is because it might be biased towards default behavior. For instance, suppose the table would have been 10m by 10m, and we had placed obstacles at only a few (e.g. 4) positions would affect reaching behavior. Including the unaffected trajectories for the other 9.996 positions in the clustering would lead to an over-representation of unaffected trajectories, and hence a bias. Therefore, we first split the sets of trajectories *DEFAULT* and *AVOID*, and perform clustering only on *AVOID*.

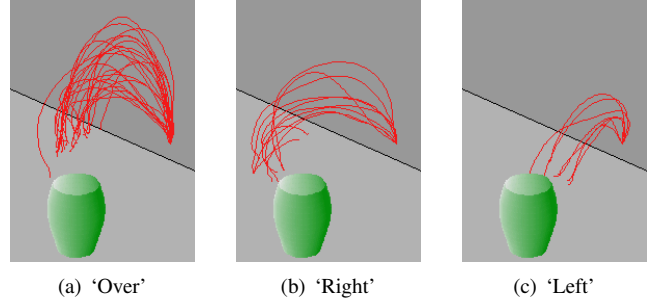


Fig. 6. The three clusters within the *AVOID*-trajectories.

The average trajectories for these sets are plotted in Figure 7. Clustering the *AVOID*-trajectories yields stereotypical reaching movements, which we label 'over', 'left', and 'right', denoting the direction in which the obstacle is avoided. We call these the 'principal trajectories'. What is very interesting about these trajectories is that they are not qualitatively different from each other, but are rather variations of the default behavior. From the top view for instance, it is apparent that the 'over' strategy almost perfectly follows the default behavior in the  $xy$ -plane, and is therefore simply a version of the default behavior, scaled in the  $z$ -plane. Similarly, 'left' and 'right' strategies hardly vary from the default behavior in the  $z$ -plane.

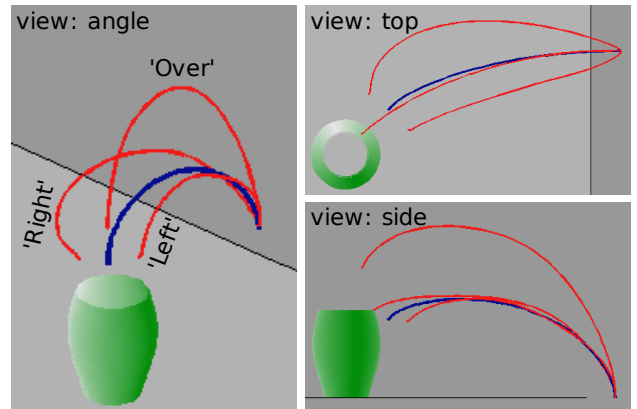


Fig. 7. The averages of the *DEFAULT*-trajectories (dark blue) and the three clusters in the *AVOID*-trajectories (bright red) from different views.

In the next section, we investigate whether these principal trajectories, and interpolations between them, predict observed behavior well. We will also use them to learn Dynamic Movement Primitives [12] on a robot. Therefore, the principal trajectories are a compact model that is not only used to explain human behavior, as in [1], but also as a means of parameterizing a controller, as in [8].

#### V. TRAJECTORY IMITATION BY AN ARTICULATED ROBOT

We now show as a proof of concept that the compact model can be realized on an autonomous manipulation platform to reproduce the principal trajectories, and interpolate between

them. The platform used is a B21 autonomous robot with 6 DOF Amtec Powercube arm shown in Figure 11. Please also see the accompanying video showing two example trajectories.

In the control system, each principal trajectory is represented by a Dynamic Movement Primitive (DMP) [12]. One DMP was trained for each principal trajectory with the regression learning algorithm described in [11]. Some advantages of DMPs are 1) within a certain range, they generalize to other goal locations 2) compliance 3) convergence to the goal location is guaranteed.

It is worth noting that the compact models contain only kinematic information, more specifically the coordinates from the hand in Euclidean space. This is easily observable from human subjects, in contrast to internal dynamic states like forces and torques. We rely on inverse kinematic algorithms and low-level controllers for successfully tracking the generated trajectories with the robot.

The trajectory generated by the DMP module is taken and fed to a work space single point attractor, which takes the next intermediate point in the trajectory and pulls the end-effector to this intermediate goal until it is reached. The output of the single point attractor is the desired velocity vector which is given to the velocity based inverse kinematics controller which generates the velocities in joint space. We use the damped least squares inverse kinematics algorithm from [14] as implemented in the Orocos-KDL library [22] which achieves more stable behavior around singularities.

#### A. Results

Figure 10 depicts the robot reproducing the default reaching behavior. In this section, we analyze the accuracy of executing principal trajectories, and interpolations between these trajectories. Furthermore, we determine the relation between the compact model and trajectories that were actually executed by the human subject.

**Accuracy of principal trajectory following.** A comparison between the human principal trajectories and the motion reproduced by the robot with its learned DMPs is depicted in Figure 8(a). Visual inspection shows that they coincide almost perfectly.

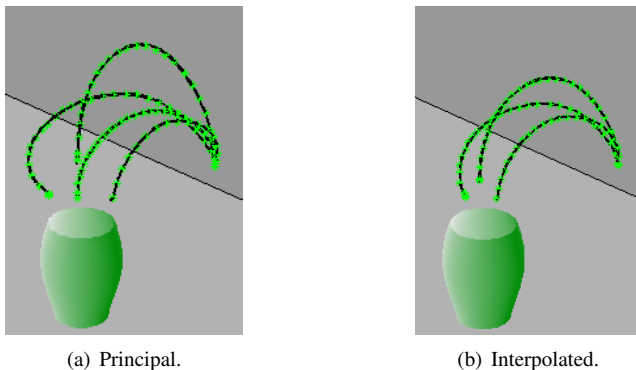


Fig. 8. Comparison between human (black line) and robot (green markers) reaching trajectories.

**Generating novel trajectories by principal trajectory interpolation.** The robot can combine principal trajectories to generate novel trajectories. In Figure 8(b), three trajectories are depicted. These are simply linear interpolations between the default behavior, and the three avoidance strategies. Composing the trajectory in this way is consistent with neurophysiological findings, where it was found that leg movements of frogs are linear combinations of several convergent force fields [10]. From Figure 8(b), we conclude that the robot also follows these trajectories very accurately.

**Observed trajectories similar to principal and interpolated trajectories exist.** In Figure 9, the robot's trajectories are bright green. From the 320 human trajectories, the one that is most similar to the robot trajectory is included in dark blue. The similarity measure is again the angle between the two 300-dimensional vectors representing the 2 trajectories, as in Section IV-B.

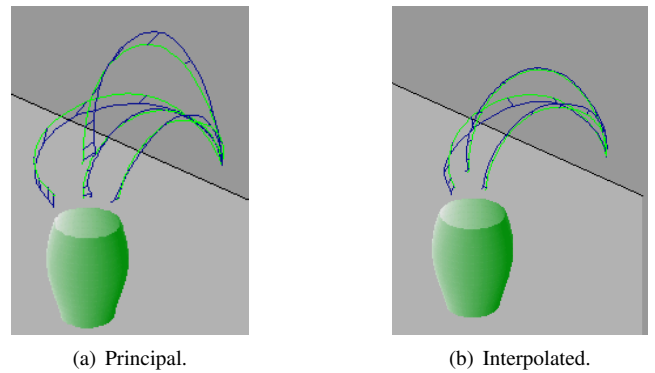


Fig. 9. Robot trajectories (light green), and their most similar human trajectories (dark blue).

Apparently, the robot trajectories have human counterparts which are quite similar qualitatively. Some interesting conclusions from this observation are: 1) the principal trajectories are not merely theoretical 'platonic' idealizations, but are actually observed in human behavior as well. 2) Using the principal trajectories as a compact model enables us to predict other trajectories which are also observed in human behavior as well. We therefore conclude that the compactness of the model has not reduced its explanatory power. Arechavaleta et al. have made similar conclusions for compact models of human locomotion [1].

## VI. CONCLUSION

For standard everyday manipulation tasks, humans use standard pre-planned [15] stereotypical [25] reaching motions which have been optimized with respect to various criteria. This optimization-induced standardization allows human reaching trajectories to be represented very compactly, as has been shown for locomotion [1].

In this paper, we derive a compact model from tracked human reaching trajectories, and port it to an articulated robot. We have shown that the robot is able to follow principal trajectories from the compact model, and that it can generate

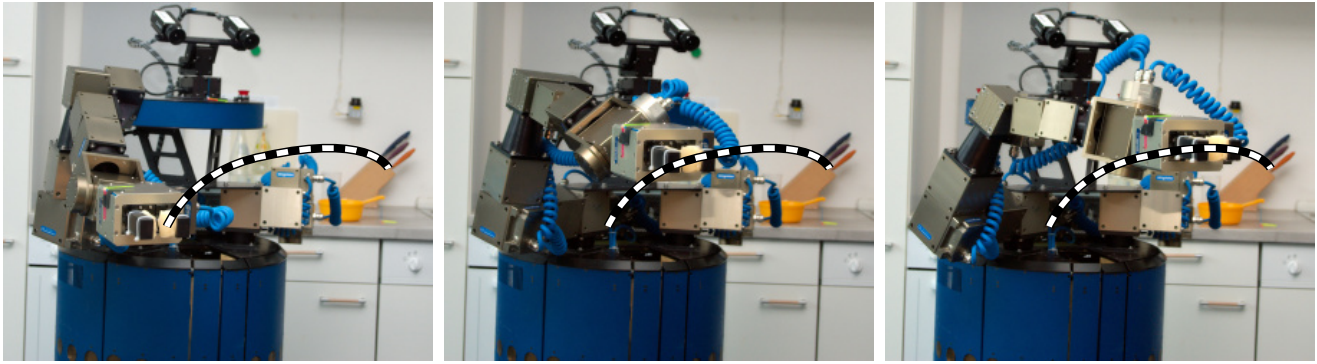


Fig. 10. The B21 robot reproducing a principal trajectory with its PowerCube arm. Please also see the accompanying video.

novel trajectories through interpolation. The compact model representation is validated by the observation that trajectories executed by the robot have similar counterparts amongst the trajectories executed by the human.

We are currently extending the work described in this paper in several ways. First of all, we are conducting further experiments with 5 more subjects. In these experiments, we will only place obstacles in the area where obstacles had an effect on the trajectories, allowing a more dense obstacle distribution. In a further follow-up experiment, we also intend to use obstacles of different sizes.

The advantages of DMPs have not yet been fully exploited in this article. For instance, A DMP can take goal locations other than that with which it was trained, and still generate a qualitatively similar trajectory. Preliminary work, depicted in Figure 11, shows that this is indeed the case for our robot. In future work, we want to compare these trajectories to those of humans, if the target glass is placed at another position, e.g. B6. Can DMPs extrapolate and explain this behavior for which they were not trained as well? Also, we intend to encode the four principal trajectories and their in one single DMP, which takes parameters that define how much of each principal trajectories is involved in generating the motion. Appropriate parameters for a given task context then depend on the size and location of the obstacle. We see this as an alternative to on-line obstacle avoidance with potential fields [18].

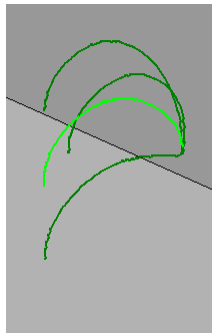


Fig. 11. Even though a DMP is trained with a principal trajectory with only one specific goal position, it generalizes to other goal positions, whilst retaining the general shape of the trajectory. In this image, the default principal trajectory (light green) was modified by the DMP to reach goals that are respectively 10cm over, under, and behind the original goal (dark green).

#### ACKNOWLEDGEMENTS

We are grateful to Heiko Hoffmann and Pierre Roduit for providing us with the Matlab code described in [11]

and [19] respectively. The research described in this article is funded by the CoTeSys cluster of excellence (Cognition for Technical Systems, <http://www.cotesys.org>), part of the Excellence Initiative of the DFG.

#### REFERENCES

- [1] G. Arechavaleta, J-P. Laumond, H. Hicheur, and A. Berthoz. The nonholonomic nature of human locomotion: a modeling study. In *IEEE International Conference on Biomedical Robotics and Biomechatronics*, 2006.
- [2] Dmitry Berenson, Rosen Diankov, Koichi Nishiwaki, Satoshi Kagami, and James Kuffner. Grasp planning in complex scenes. In *IEEE-RAS International Conference on Humanoid Robots*, 2007.
- [3] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. *Springer Handbook of Robotics*, chapter 59. Robot programming by demonstration. Springer, 2008.
- [4] 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, Special issue on robot learning by observation, demonstration and imitation*, 37(2):286–298, 2007.
- [5] Craig S. Chapman and Melvyn A. Goodale. Missing in action: the effect of obstacle position and size on avoidance while reaching. *Experimental Brain Research*, 2008.
- [6] Paul Fitzpatrick, Giorgio Metta, and Lorenzo Natale. Towards long-lived robot genes. *Robotics and Autonomous Systems*, 56(1):29–45, 2008.
- [7] T. Flash and B. Hochner. Motor primitives in vertebrates and invertebrates. *Current Opinion in Neurobiology*, 15:660–666, 2005.
- [8] Ajo Fod, Maja J Mataric, and Odest Chadwicke Jenkins. Automated derivation of primitives for movement classification. *Autonomous Robots*, 12(1):39–54, 1 2002.
- [9] Brian Gerkey, Richard T. Vaughan, and Andrew Howard. The Player/Stage Project: Tools for multi-robot and distributed sensor systems. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, pages 317–323, 2003.
- [10] S. Giszter, F. Mussa-Ivaldi, and E. Bizzi. Convergent force fields organized in the frog's spinal cord. *Journal of Neuroscience*, 13(2):467–491, February 1993.
- [11] Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamic movement primitives for movement generation motivated by convergent force fields in frog. In Roy Ritzmann and Robert Quinn, editors, *Fourth International Symposium on Adaptive Motion of Animals and Machines*, Case Western Reserve University, Cleveland, OH, 2008.
- [12] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *International Conference on Robotics and Automation (ICRA2002)*, 2002.
- [13] O. Jenkins, R. Bodenheimer, and R. Peters. Manipulation manifolds: Explorations into uncovering manifolds in sensory-motor spaces. In *International Conference on Development and Learning (ICDL)*, 2006.
- [14] A. A. Maciejewski and C. A. Klein. The singular value decomposition: Computation and applications to robotics. *International Journal of Robotics Research*, 8(6):63–79, 1989.

- [15] Mark Mon-Williams, James R. Tresilian, Vanessa L. Coppard, and Richard G. Carson. The effect of obstacle position on reach-to-grasp movements. *Experimental Brain Research*, 137:497–501, 2001.
- [16] A. Morales, T. Asfour, P. Azad, S. Knoop, and R. Dillmann. Integrated grasp planning and visual object localization for a humanoid robot with five-fingered hands. In *IROS*, 2006.
- [17] E. Oztop, D.W. Franklin, T. Chaminade, and G. Cheng. Human-humanoid interaction: Is a humanoid robot perceived as a human? *International Journal of Humanoid Robotics*, 2(4):537–559, 2005.
- [18] Dae-Hyung Park, Heiko Hoffmann, and Stefan Schaal. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *International Conference on Humanoid Robots*, 2008.
- [19] Pierre Roudit, Alcherio Martinoli, and Jacques Jacot. A quantitative method for comparing trajectories of mobile robots using point distribution models. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2441–2448, 2007.
- [20] Sam Roweis and Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [21] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.
- [22] Ruben Smits, Tinne De Laet, Kasper Claes, Peter Soetens, Joris De Schutter, and Herman Bruyninckx. Orocos: A software framework for complex sensor-driven robot tasks. *IEEE Robotics and Automation Magazine*, 2008.
- [23] Freek Stulp and Michael Beetz. Refining the execution of abstract actions with learned action models. *Journal of Artificial Intelligence Research (JAIR)*, 32, June 2008.
- [24] T. Wimbock, C. Ott, and G. Hirzinger. Impedance behaviors for two-handed manipulation: Design and experiments. In *ICRA*, 2007.
- [25] Daniel Wolpert and Zoubin Ghahramani. Computational principles of movement neuroscience. *Nature Neuroscience Supplement*, 3:1212–1217, 2000.