

Project no.: 027657  
 Project full title: Perception, Action & Cognition through Learning of Object-Action Complexes  
 Project Acronym: PACO-PLUS  
 Deliverable no.: D2.3.2

**Title of the deliverable: Technical report on "Goal-directed action learning by imitation and coaching"**

Contractual Date of Delivery to the CEC:	31. January 2010
Actual Date of Delivery to the CEC	12. February 2010
Organisation name of lead contractor for this deliverable:	JSI
Authors:	Aleš Ude, Andrej Gams, Tamim Asfour, and Rüdiger Dillmann
Participants:	JSI, UniKarl
Work package contributing to the deliverable	WP1, WP2, WP8
Nature:	R & D
Version	1.0
Total number of pages:	4
Start date of project:	1 <sup>st</sup> Feb. 2006
Duration:	52 months

**Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)**

**Dissemination Level**

<b>PU</b>	Public	<b>X</b>
<b>PP</b>	Restricted to other programme participants (including the Commission Services)	
<b>RE</b>	Restricted to a group specified by the consortium (including the Commission Services)	
<b>CO</b>	Confidential, only for members of the consortium (including the Commission Services)	

**Abstract:**

In this deliverable we present our final report on action learning by imitation and coaching. We describe how to relate the perceptual input describing the relevant information about objects in the environment to the motor knowledge of the robot, thus enabling the generalization of the available data to new situations. Motor knowledge is represented by a library of trajectories, and each of them can solve the desired task in a particular situation. These trajectories are obtained using methods such as kinesthetic guiding, observation of human motion, and on-line coaching. The generalization process results in a new control policy, which is tuned to the current perceptual input. Nonlinear dynamic systems have been selected to represent the control policies because they support the utilization of on-line perceptual feedback. The proposed methodology can be used with both periodic and discrete movements. By connecting object information with the robot's motor knowledge, we form early object-action complexes at the sensorimotor level, which can be executed by a robot. The applicability of the proposed methodology is demonstrated in several real-world experiments.

**Keyword list:** Nonlinear dynamic systems, statistical learning, perception-action coupling.

## Table of Contents

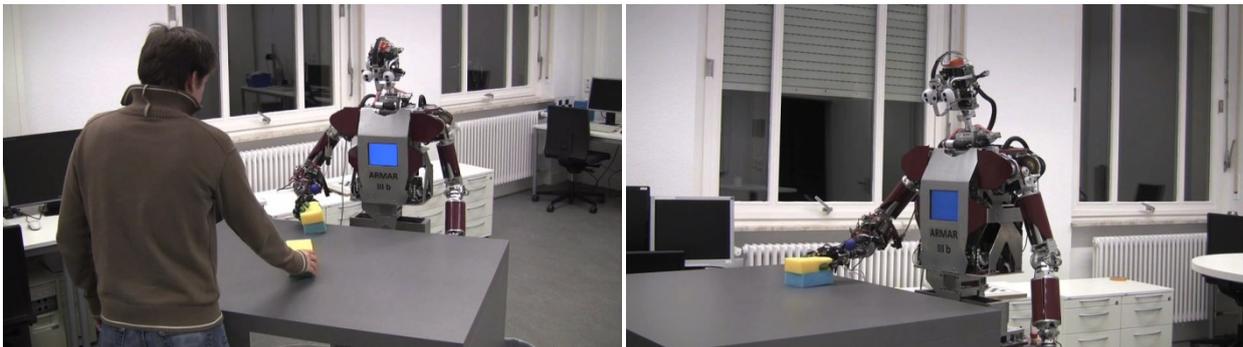
1. EXECUTIVE SUMMARY .....	3
2. PUBLICATIONS ASSOCIATED WITH D2.3.2 .....	4

## 1. Executive Summary

Direct imitation has been applied successfully to learn complex movements on humanoid robots, e.g. dancing, which would be difficult to program manually. Direct imitation and movement reproduction are, however, not useful for problems that involve the manipulation of objects because in such tasks the observed movements need to be adapted to the current state of the 3-D world. It is impossible to generate trajectory libraries that contain solutions for every possible object configuration. Instead, a cognitive robot must be able to generate new movements based on the available motor knowledge and the state of the objects in the environment. In other words, a connection between objects and motor programs need to be established, which can be understood as early object-action complexes at the sensorimotor level.

Solutions such as constraint specification or the adaptation of equations describing the underlying control policies to account for perceptual feedback can be effective, but they require an expert to specify the appropriate mathematical equations. While this can be easily achieved for some types of positional constraints, which often appear in tasks associated with reaching and require only the robot's kinematics to be known, such constraints become increasingly more complex for dynamic tasks, which can often be described only by complicated physical models. In the last year of the project we further developed and tested our approach to action generalization, which does not require any physical models of the task to be available; generalization is performed in a statistical way. A person interacting with the robot must specify only the appropriate parameters that need to be provided as perceptual feedback. Both periodic and discrete movements are supported now. As demonstrated in the examples described in the attached paper, this parameter space can normally be specified in an intuitive way.

In a nutshell, the proposed methodology takes the perceptual input describing the relevant parameters of the task, and statistically generates a new control policy using the provided percepts as query points into the trajectory library. Nonlinear dynamic systems are used to represent both periodic and discrete movements. In this way the robot can adapt the previously recorded trajectories to new situations that were not observed during training. The robot establishes the connection between object and its own movements without needing to understand the complete physics of the task. Only such an approach is feasible for cognitive robots that need to operate in unknown and dynamic environments. A competent behaviours in such environments can only be achieved if the robot can accumulate new knowledge in an open-ended manner and extrapolate the available knowledge to new situations, as realized in the proposed approach.



**Figure 1** Coaching and execution of periodic wiping movements

The observation of human demonstrations and coaching were used to acquire the initial sensorimotor experiences throughout the project. In deliverable D8.2.1 we presented our work on robot coaching based on the verbal input of the user, which was successfully used to acquire discrete movements. In the final year of the project we developed a new approach for teaching periodic movements, where the coach

teaches motion trajectories by adapting his own movements based on his own observation of the robot's motion. The coach stops the demonstration once he / she is satisfied with the robot's performance, i.e. once the robot behaves as desired. Thus instead of providing verbal feedback to adapt the robot motion, the coach attempts to demonstrate such movements that are suitable for the capabilities of the robot. Suitable algorithms that allow on-line estimation of periodic movements were developed. We successfully tested this approach for teaching drumming and wiping movements. Another important topics in this area in the last year of the project was to extend the master motor map interface, which was developed to transfer the available motor knowledge to different robots. The master motor map was enhanced with additional physical parameters and can now describe both the kinematics and dynamics of the encoded motion. Such a representation can better facilitate the transferability of the available trajectories to new robots.

## **2. Publications Associated with D2.3.2**

- A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives", conditionally accepted in IEEE Transactions on Robotics, 2010.

**Abstract:** Acquiring new sensorimotor knowledge by imitation is a promising paradigm for robot learning. To be effective, action learning should not be limited to direct replication of movements obtained during training, but must enable also the generation of actions in situations that a robot has never encountered before. This paper describes a methodology that enables the generalization of the available motor knowledge. New actions are synthesized by applying statistical methods, where the goal and other characteristics of the action are utilized as queries to create an optimal control policy with respect to the current state of the world. Nonlinear dynamic systems are employed as an underlying motor representation. The proposed approach enables the generation of a wide range of policies without requiring an expert to modify the underlying representations to account for different task-specific features and perceptual feedback.

The paper also demonstrates that the proposed methodology can be integrated with an active vision system of a humanoid robot. 3-D vision data is used to provide query points for statistical generalization. While 3-D vision on humanoid robots with complex oculomotor systems is often difficult due to the modeling uncertainties, we show that these uncertainties can be accounted for by the proposed approach.

- S. Gärtner, M. Do, C. Simonidis, T. Asfour, W. Seemann and R. Dillmann, "Generation of Human-like Motion for Humanoid Robots Based on Marker-based Motion Capture Data", 41st International Symposium on Robotics, Munich, Germany.

In order to efficiently reuse captured movements on various robots, an intermediate model is needed, decoupling representation of a motion, which can be stored in a motion repository, from its execution on an actual robot. On the contrary, there exist numerous human motion capture systems that produce output in terms of different models stored in different formats. To overcome this problem, the Master Motor Map (MMM) presents an appropriate interface based on a unified model. In this paper, we will further propose an extension of this model by adding certain anthropomorphic properties, such as mass distribution, segment length, moment of inertia, etc. Such an anthropomorphic model of the segmented body is of use in terms of determining forward and inverse dynamics as well as motion synthesis and retargeting.

# Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives

Aleš Ude, Andrej Gams, Tamim Asfour, and Jun Morimoto

**Abstract**—Acquiring new sensorimotor knowledge by imitation is a promising paradigm for robot learning. To be effective, action learning should not be limited to direct replication of movements obtained during training, but must enable also the generation of actions in situations that a robot has never encountered before. This paper describes a methodology that enables the generalization of the available motor knowledge. New actions are synthesized by applying statistical methods, where the goal and other characteristics of an action are utilized as queries to create an optimal control policy with respect to the current state of the world. Nonlinear dynamic systems are employed as an underlying motor representation. The proposed approach enables the generation of a wide range of policies without requiring an expert to modify the underlying representations to account for different task-specific features and perceptual feedback.

The paper also demonstrates that the proposed methodology can be integrated with an active vision system of a humanoid robot. 3-D vision data is used to provide query points for statistical generalization. While 3-D vision on humanoid robots with complex oculomotor systems is often difficult due to the modeling uncertainties, we show that these uncertainties can be accounted for by the proposed approach.

**Index Terms**—Learning and Adaptive Systems, Humanoid Robots, Imitation Learning, Active Vision on Humanoid Robots

## I. INTRODUCTION

**L**EARNING of behaviors that can be applied to solve a given task regardless of the current configuration of the external world is a difficult problem because the search space that needs to be explored is potentially huge [1]. The size of the search space depends both on the number of degrees of freedom of the robot and on the objects involved in the action. Furthermore, external objects can affect the search space indirectly. To overcome problems arising from high dimensional and continuous perception-action spaces, it is necessary to guide the search process. One of the most successful paradigms that can be used for this purpose is imitation or robot programming by demonstration [1], [2]. Direct imitation has been applied successfully to learn complex movements on humanoid robots such as dancing, which would be difficult to program manually [3], [4], [5]. Alternatively, the robot can be physically guided through the desired trajectory, which requires that the robot is back-drivable [6] or even force controlled [7]. Direct imitation and movement reproduction are,

however, not useful for problems that involve the manipulation of objects because in such tasks the observed movements need to be adapted to the current state of the 3-D world. Already the early work of Kuniyoshi et al. [8] stressed the importance of task segmentation and extraction of meaningful action units. It is highly unlikely that an appropriate movement would be observed in advance and included in the library of trained movements for any given situation.

It should be noted that the initial action knowledge does not necessarily need to be acquired by imitation or kinesthetic guiding. The field of developmental robotics [9] stresses the role of autonomous exploration. For example, forward models can be acquired through an exploration process, similar to the motor babbling stage of infants [10].

The main interest of this paper is in the generalization of the available action knowledge regardless of how this knowledge was initially acquired. A methodology that can adapt single trajectories obtained by imitation was proposed by Miyamoto et al. [11]. They developed a new representation for the desired trajectory, which they referred to as via-points. By monitoring its performance, the robot was able to continuously adapt the via points until it could play a fairly difficult Japanese game kendama or execute tennis serves. This type of approaches have later been studied in the frame of modern reinforcement learning theories [12]. They enable the robot to autonomously improve its performance by practicing until it can solve the task in one particular configuration of the external world. Note that this includes a lengthy learning process and therefore cannot be applied on-line.

The work of Miyamoto et al. [11] shows the importance of a proper representation for the control policy. Their representation is essentially a spline-based representation. Splines have been used in robotics in many contexts [13], but their explicit dependency on time can be cumbersome [14]. Hidden Markov models (HMMs) are another popular methodology to encode and generalize the observed trajectories [15], [16], [17]. It has been shown that HMMs can be used effectively for motion and situation recognition [17] and to determine which control variables should be imitated and how [16]. Kulić et al. [18] extended these works by showing how HMMs can be used to automatically cluster large databases of movements into the constituent primitives. Gaussian mixture models [19] provide another probabilistic representation with some advantages over HMMs when reproducing the learned movements. This work also demonstrates how movements can be generalized by taking into account different analytically specified constraints between hands and objects.

A fundamentally different approach to motion representation based on nonlinear dynamic systems as policy primitives

Aleš Ude and Andrej Gams are with the Department of Automatics, Biocybernetics, and Robotics, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia and with the ATR Computational Neuroscience Laboratories, 2-2-2 Hikaridai, Soraku-gun, Kyoto 619-0288, Japan.

Tamim Asfour is with the University of Karlsruhe, Kaiserstr. 12, 76131 Karlsruhe, Germany.

Jun Morimoto is with the ATR Computational Neuroscience Laboratories, 2-2-2 Hikaridai, Soraku-gun, Kyoto 619-0288, Japan.

was proposed in [20], [21]. The resulting control policies were termed as *dynamic movement primitives* (DMPs). DMPs are based on systems of second-order differential equations, which encode the properties of the desired motion. Ijspeert et al. [20], [21] proposed equations for periodic and discrete movements and demonstrated that they can be used to learn tasks such as tennis strokes and drumming. One of the most important advantages of DMPs is that they remove the direct dependency of the control policy on time. As explained in [14], explicit timing is cumbersome because it adds an additional level of complexity needed for aborting, halting, or resetting the clock when unforeseen disturbances happen during movement execution. Another approach based on dynamic systems is described in [6], where the final trajectory is generated as a linear combination of attractor dynamics and Gaussian mixture models learned from example trajectories.

### A. Contribution of this Paper

The main purpose of this paper is to propose and experimentally evaluate a methodology for generalizing example trajectories to new situations that were not observed during training. To this end every example trajectory is associated with parameters that describe the characteristics of an action, typically its goal, and serve as query points into the example database. Our work is thus situated at a higher abstraction level than direct imitation and variants of reinforcement learning (as in [11]), but below action sequencing and task planning.

We utilize nonlinear dynamic systems as a basic motor representation. Other researchers have shown that by changing the underlying differential equations, DMPs can be modified in several ways to account for various perturbations that might occur during the execution of the task. For example, Pastor et al. [22], [23] proposed to add terms that enable obstacle avoidance. Such modifications, however, are necessarily designed for one particular issue only. For each new problem, an engineer must redesign the underlying dynamic system, which is unsatisfactory for robots that need to operate in natural environments and have to solve new problems every day. The approach proposed in this paper enables the generalization of DMPs to new situations based on the available training data and relevant characteristics of the task, which can be supplied in a natural way. Since the basic differential equations of DMPs remain unchanged, we are still able to use any analytic modifications of the underlying dynamic system. Such modifications can be very useful to account for various unforeseen perturbations that can occur during on-line execution and are typically not part of the training data. Section IV-B demonstrates that the proposed approach is suitable for on-line integration with the active vision of a humanoid robot.

## II. CONTROL POLICIES AS DYNAMIC SYSTEMS

We start by a quick overview of the theoretical fundamentals of the motor representation used in this work. Ijspeert et al. [20], [21] proposed to describe a control policy by a set of nonlinear differential equations with well-defined attractor dynamics. For a single degree of freedom denoted by  $y$ , which can either be one of the internal joint angles or one

of the external task space coordinates, the following system of linear differential equations with constant coefficients has been proposed as a basis for motion specification [14]

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z), \quad (1)$$

$$\tau \dot{y} = z. \quad (2)$$

Provided that the parameters  $\alpha_z$ ,  $\beta_z$ , and  $\tau > 0$  are selected appropriately, e.g.  $\alpha_z = 4\beta_z$ , this system has a unique attractor point at  $y = g$ ,  $z = 0$ .

Differential equations (1)–(2) ensure that  $y$  converges to  $g$  and can therefore be used to realize discrete point-to-point movements. To increase a rather limited set of trajectories that can be encoded by (1)–(2) and thus enable the approximation of general point-to-point movements, Eq. (1) needs to be modified. In the case of discrete movements, one can add a linear combination of radial basis functions to (1) [14]<sup>1</sup>

$$f(x) = \frac{\sum_{i=1}^N w_i \Psi_i(x)}{\sum_{i=1}^N \Psi_i(x)} x, \quad \Psi_i(x) = \exp(-h_i(x - c_i)^2), \quad (3)$$

where  $c_i$  are the centers of radial basis function distributed along the trajectory and  $h_i > 0$ . A phase variable  $x$  is used in (3) instead of time to avoid direct dependency of  $f$  on time. Its dynamics is defined by

$$\tau \dot{x} = -\alpha_x x, \quad (4)$$

with initial value  $x(0) = 1$ . A solution to (4) is given by  $\exp(-\alpha_x t / \tau)$ , thus  $x$  tends to 0 as time increases. This results in the following system of differential equations

$$\tau \dot{z} = \alpha_z(\beta_z(g - y) - z) + f(x), \quad (5)$$

$$\tau \dot{y} = z, \quad (6)$$

which can be used to approximate discrete movements of various shapes. Since  $x$  tends to zero, the influence of the nonlinear term  $f(x)$  decreases with time and system (5)–(6) converges to  $[0, g]^T$  just like (1)–(2). The other role of  $x$  is to localize the radial basis functions along the trajectory that needs to be approximated. The control policy specified by variable  $y$  defines what is called a *dynamic movement primitive* (DMP).

In the case of periodic movements, the following linear combination of periodic functions can be used to change the dynamics of the basic second order system [14]

$$f(\phi) = \frac{\sum_{i=1}^N w_i \Gamma_i(\phi)}{\sum_{i=1}^N \Gamma_i(\phi)} r, \quad \Gamma_i(\phi) = \exp(h_i(\cos(\phi - c_i) - 1)), \quad (7)$$

where  $r$  is the amplitude of the oscillator and  $h_i > 0$ . Eq. (5) and (6) are replaced by

$$\dot{z} = \Omega(\alpha_z(\beta_z(g - y) - z) + f(\phi)), \quad (8)$$

$$\dot{y} = \Omega z. \quad (9)$$

<sup>1</sup> $f$  defined in [14] is scaled by  $g - y_0$ , i.e.  $f(x) = \frac{\sum_{i=1}^N w_i \Psi_i(x)}{\sum_{i=1}^N \Psi_i(x)} x(g - y_0)$ ,  $y_0 = y(0)$ . Thus when the goal configuration  $g$  changes, the encoded movement gets scaled. We omit this scaling factor because we are not interested in automatic scaling, which is achieved differently in our approach. If  $g$  is kept constant, the scaling factor has no effect.

The phase variable  $\phi$  has been introduced in this case to avoid the explicit dependency on time. The phase is assumed to move with constant speed

$$\dot{\phi} = \Omega, \quad (10)$$

where  $\Omega$  is the frequency of oscillation,  $\tau = 1/\Omega$ .

### III. ACTION GENERALIZATION USING DYNAMIC MOVEMENT PRIMITIVES

In this section we describe the core algorithms developed in the paper. Section III-A first explains the generation of DMPs using only one training trajectory. Based on these results we then describe in Sections III-B–III-D our approach to action generalization.

#### A. Reproduction from Single Demonstration

The trajectory of any smooth movement can be estimated by adapting the parameters  $w_i$  of Eq. (3) and (7). We start from sampled desired trajectory positions, velocities, and accelerations  $\{y_d(t_j), \dot{y}_d(t_j), \ddot{y}_d(t_j)\}$ ,  $j = 1, \dots, T$ , where  $t_j$  are the sampling times. This data can be obtained either by kinesthetic guiding or from a human demonstration. It can be given either in the joint or in the Cartesian space. Each degree of freedom is described by its own dynamic system. The system of two first order linear equations (5)–(6) in the case of discrete movements and (8)–(9) in the case of periodic movements can be rewritten into one second order equation by replacing  $z$  with  $y$  in Eq. (5) and (8), respectively

$$\tau^2 \ddot{y} + \alpha_z \tau \dot{y} - \alpha_z \beta_z (g - y) = f, \quad (11)$$

with  $f$  defined as in Eq. (3) and (7). The formula  $\tau = 1/\Omega$  is used in the case of periodic movements. Writing

$$F(t_j) = \tau^2 \ddot{y}_d(t_j) + \alpha_z \tau \dot{y}_d(t_j) - \alpha_z \beta_z (g - y_d(t_j)), \quad (12)$$

$$\mathbf{f} = \begin{bmatrix} F(t_1) \\ \dots \\ F(t_T) \end{bmatrix}, \quad \mathbf{w} = \begin{bmatrix} w_1 \\ \dots \\ w_N \end{bmatrix},$$

we obtain the following set of linear equations

$$\mathbf{X}\mathbf{w} = \mathbf{f}, \quad (13)$$

which needs to be solved to estimate the DMP describing the desired motion. In case of discrete movements we have

$$\mathbf{X} = \begin{bmatrix} \frac{\Psi_1(x_1)}{\sum_{i=1}^N \Psi_i(x_1)} x_1 & \dots & \frac{\Psi_N(x_1)}{\sum_{i=1}^N \Psi_i(x_1)} x_1 \\ \dots & \dots & \dots \\ \frac{\Psi_1(x_T)}{\sum_{i=1}^N \Psi_i(x_T)} x_T & \dots & \frac{\Psi_N(x_T)}{\sum_{i=1}^N \Psi_i(x_T)} x_T \end{bmatrix}$$

and in case of periodic movements

$$\mathbf{X} = r \begin{bmatrix} \frac{\Gamma_1(\phi_1)}{\sum_{i=1}^N \Gamma_i(\phi_1)} & \dots & \frac{\Gamma_N(\phi_1)}{\sum_{i=1}^N \Gamma_i(\phi_1)} \\ \dots & \dots & \dots \\ \frac{\Gamma_1(\phi_T)}{\sum_{i=1}^N \Gamma_i(\phi_T)} & \dots & \frac{\Gamma_N(\phi_T)}{\sum_{i=1}^N \Gamma_i(\phi_T)} \end{bmatrix}.$$

$x_i$  and  $\phi_i$  are obtained by respectively integrating (4) and (10).

The integration of Eq. (8)–(10), which are used for periodic movements, requires that the frequency of movement  $\Omega$  is known. To automatically determine the frequency during

demonstration, we applied the system of adaptive frequency oscillators proposed in [24], [25]. The authors proposed to replace the constant speed assumption (10) by a system

$$\dot{\phi}_i = \Omega_i - K e(t) \sin(\phi_i), \quad (14)$$

$$\dot{\Omega}_i = -K e(t) \sin(\phi_i), \quad (15)$$

$$\dot{\alpha}_i = \eta \cos(\phi_i) e(t), \quad (16)$$

where  $e(t) = y_d(t) - \hat{y}(t)$  and  $\hat{y}(t) = \sum_{i=1}^L \alpha_i \cos(\phi_i)$ . Note that if  $e(t) = 0$ , the system (14)–(16) becomes equivalent to (10). It has been shown that by integrating this system, the frequencies  $\Omega_i$  contained in the desired motion trajectory can be estimated. The most significant frequency is selected as the base or fundamental frequency  $\Omega$  for the DMP (see [24] for details) and this frequency is used to estimate the form parameters  $w_i$ . This is accomplished by incrementally solving the equation system (13) using recursive least squares

$$\mathbf{P}_j = \mathbf{P}_{j-1} - \frac{\mathbf{P}_{j-1} \mathbf{x}_j \mathbf{x}_j^T \mathbf{P}_{j-1}}{1 + \mathbf{x}_j^T \mathbf{P}_{j-1} \mathbf{x}_j}, \quad (17)$$

$$\mathbf{w}_j = \mathbf{w}_{j-1} + (f_j - \mathbf{x}_j^T \mathbf{w}_{j-1}) \mathbf{P}_j \mathbf{x}_j, \quad (18)$$

where  $\mathbf{P}_0 = \sigma^2 \mathbf{I}$ ,  $\mathbf{w}_0 = 0$ ,  $f_j = F(t_j)$ ,  $\mathbf{x}_j$  is the  $M$  dimensional column vector associated with the corresponding row of the system matrix  $\mathbf{X}$  and the final optimal weights are given as  $\mathbf{w} = \mathbf{w}_T$ . The estimation of the frequency  $\Omega$  can be done simultaneously with the recursive estimation of the form parameters  $\mathbf{w}$ ; at each time step  $t_j$  first the current  $\Omega$  is estimated by integrating (14)–(16). The current frequency is then used to calculate target values  $F(t_j)$  of Eq. (12), where as usually  $\tau = 1/\Omega$ . Finally, the new estimate  $\mathbf{w}_j$  is calculated using the recursion (17)–(18).

In the equations above,  $\alpha_x$ ,  $\alpha_z$ , and  $\beta_z$  are constant. In case of discrete movements,  $g$  is set to the desired final position while  $\tau$  is the time duration of the example movement. In case of periodic movements, the scaling factor  $r$  is set to 1 while the frequency  $\Omega$  is calculated as explained above. The parameters  $c_i$  and  $h_i$  are determined by setting a distribution pattern and increasing the number of parameters until the desired reconstruction accuracy on all example trajectories is achieved. See [14] for other approaches. The recursive least squares method has proved to be useful for incremental learning of periodic movements [21], while in case of discrete movements we directly solve Eq. (13). Unlike earlier works that estimated parameters  $w_i$  separately using locally weighted regression (LWR), we apply a full linear system (13) to estimate the parameters  $\mathbf{w}$ , which allows us to approximate trajectories with a smaller number of basis functions. While a separate estimation of  $\{w_i\}$  has advantages when  $\{w_i\}$  are used for classification [21], here the primary interest is in reconstruction and full regression can increase the accuracy. It is possible to formulate our approach within the LWR framework as well.

#### B. Generalization of Discrete Movements

The main aim of this paper is to generalize DMPs to situations that are not part of the example database. Lets assume that we have a number of example trajectories

$$\{y_d^k(t_{k,j}), \dot{y}_d^k(t_{k,j}), \ddot{y}_d^k(t_{k,j}) | k = 1, \dots, M, j = 1, \dots, T_k\}, \quad (19)$$

and time constants  $\tau_k$  that solve the task in a given situation.  $M$  is the number of example trajectories and  $T_k$  the number of sampling points on example trajectory  $k$ . Each situation is characterized by some parameters  $\mathbf{q}_k \in \mathbb{R}^n$ , which are often related to the goal of an action and provide query points into the database. In the case of reaching movements, the goal of an action is just the final reaching destination  $\mathbf{q}_k = (g_{k,1}, \dots, g_{k,n})$ . In other cases, the goal might not be directly associated with the final configuration on the trajectory. For example, we also studied ball throwing, where the goal is specified by the position of the basket into which the ball should be thrown. Even in the case of reaching movements, the query points can be given in Cartesian space, while the DMPs might be encoded in the joint space. The issue is how to generate a DMP representing a new movement for every query  $\mathbf{q}$ , which in general will not be one of the examples  $\mathbf{q}_k$ .

In one-shot learning of Section III-A, all data is relevant when estimating the trajectory parameters  $\mathbf{w}$ . Such global optimization approaches do not make sense for generalization from multiple examples because in general no global task models are available. Furthermore, control policies that solve tasks, which are very different than the desired one, do not carry much information about it. If we take the example of ball throwing, it is reasonable to assume that trajectories associated with basket positions close to the current target are more relevant than examples associated with more distant targets.

Locally weighted regression (LWR) is a method that fits local models to nearby data [26]. It is a form of lazy learning where the computational cost of training is minimal; it simply consists of storing examples in the database. For a given query point  $\mathbf{q}$ , the optimal parameters are calculated directly from the available data by weighting the objective function

$$C(\mathbf{q}) = \sum_k L(\Xi(\mathbf{q}_k, \mathbf{w}), \mathbf{f}_k) K(d(\mathbf{q}, \mathbf{q}_k)) \quad (20)$$

Based on Eq. (13), local models are characterized by

$$L(\Xi(\mathbf{q}_k, \mathbf{w}), \mathbf{f}_k) = \|\mathbf{X}_k \mathbf{w} - \mathbf{f}_k\|^2 \quad (21)$$

in our system. We need to minimize the objective function

$$\sum_{k=1}^M \|\mathbf{X}_k \mathbf{w} - \mathbf{f}_k\|^2 K(d(\mathbf{q}, \mathbf{q}_k)), \quad (22)$$

with respect to  $\mathbf{w}$ . Here  $K$  is the kernel function and  $d$  is the metrics in the space of query points  $\mathbf{q}$ .

There are many possibilities to select the weighting kernel  $K$  [26]. We chose the tricube kernel

$$K(d) = \begin{cases} (1 - |d|^3)^3 & \text{if } |d| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

because this kernel has finite extent and a continuous first and second derivative, which means that the first two derivatives of the prediction (as a function of query points) are also continuous. It is important to ensure continuity when changing the parameters on-line. The final support of  $K$  reduces the computational complexity of the optimization problem (22) because the examples trajectories for which  $K$  vanishes do not influence generalization. This reduces the size of the system

matrix associated with the objective function (22). As discussed in [26], [27], the choice of weighting function is rarely critical for the performance of locally weighted regression. The selected kernel performed well in our experiments.

$K$  and distance  $d$  in the space of query points determine how much influence each of the example movements has on the final estimate of the control policy. The influence of each example movement diminishes with the distance of the query point  $\mathbf{q}$  from the data point  $\mathbf{q}_k$ . A standard weighted Euclidean distance can be used when query points are given in Euclidean space

$$d(\mathbf{q}, \mathbf{q}_k) = \|\mathbf{D}(\mathbf{q} - \mathbf{q}_k)\|, \quad \mathbf{D} = \text{diag}(a_i), \quad a_i > 0. \quad (24)$$

In our experiments we selected  $a_i$  so that at least one or two examples in each direction of the query point space were relevant for the calculation of the new DMP. Thus the number of considered examples was between  $2^n$  and  $4^n$ . Other metrics could be applied if query points were given in different spaces such as for example special rotation group. If the data is not distributed uniformly, then it is necessary to associate local bandwidth with each training query point to assure that the desired number of local models is used to compute new parameters  $\mathbf{w}$

$$\sum_{k=1}^M \|\mathbf{X}_k \mathbf{w} - \mathbf{f}_k\|^2 K\left(\frac{d(\mathbf{q}, \mathbf{q}_k)}{s_k}\right), \quad s_k > 0. \quad (25)$$

In our experiments, data was distributed uniformly along each dimension and  $s_k$  could be set to 1.

The computational complexity of solving the least squares system (25) is  $\mathcal{O}(N^2T)$ ,  $T \leq \sum_{k=1}^M T_k$ , and thus increases linearly with the number of data points and quadratically with the number of radial basis functions used in (3) and (7), respectively. Due to our choice of weighting kernel  $K$ , we normally have  $K(d(\mathbf{q}, \mathbf{q}_k)) = 0$  for many  $k$ . Moreover, by cutting the support of basis functions (3) and (7) once their value falls below a certain threshold, matrices  $\mathbf{X}_k$  become sparse as well. The quadratic dependence on the number of basis functions is not a problem because this number is generally much lower than the number of data points. There were around 10000 ~ 50000 data points in our experiments and at most 25 basis functions for DMPs. These facts make computational complexity low enough to allow us to resolve the least-squares problem (25) using standard methods from sparse matrix algebra. Especially in tasks such as ball throwing (see Section IV-D), where accurate reproduction is paramount, it is important to solve full systems (13) and (25) because not considering the correlations between the neighboring basis functions (3) and (7) could reduce the accuracy. The linear computational complexity in the number of data points was one of the reasons for selecting LWR when calculating the parameters  $\mathbf{w}$ . On the other hand, Gaussian process regression was used to estimate the parameters that depend only on the number of example trajectories (see Section III-D).

The proposed approach is appropriate only if example trajectories smoothly transition as a function of query points. Otherwise nearby data does not provide information about the movement associated with the new query point  $\mathbf{q}$ . Besides  $\mathbf{w}$

### procedure GeneralizePeriodicMotion

```

P =  $\sigma^2 \mathbf{I}$ , w = 0;
 $\forall k, \phi_k = t_1 \Omega_k, j_k = 1$ ;
while not all data points processed
   $k' = \arg \min_k \{\phi_{k,j_k}\}$ ;
  using  $\{y_d^{k'}(t_{k',j}), \dot{y}_d^{k'}(t_{k',j}), \ddot{y}_d^{k'}(t_{k',j})\}, \phi_{k'}$  calculate
     $\mathbf{x} = \mathbf{x}_{k'} K(d(\mathbf{q}, \mathbf{q}_{k'})), f = F(t_{k',j}) K(d(\mathbf{q}, \mathbf{q}_{k'}))$ ;
  update P and w using Eq. (17)–(18) with x and f;
   $j_{k'} = j_{k'} + 1$ ;
end

```

Fig. 1. Procedure for the generalization of periodic movements

we also estimate the parameters  $\tau$  and  $\mathbf{g}$  (see Section III-D). This means that the function

$$\mathbf{G} : \mathbf{q} \rightarrow [\mathbf{w}^T, \tau, \mathbf{g}^T]^T, \quad (26)$$

which is in general unknown, needs to be smooth. This is normally the case if the robot uses the same strategy to solve the task in different situations and we provide a number of examples in Section IV.

### C. Generalization of Periodic Movements

In Section III-A we described a method that estimates the parameters of a DMP for reproduction of the sampled periodic motion. The simultaneous, incremental estimation of frequency  $\Omega$  and form parameters  $\mathbf{w}$  allow the robot to reproduce the demonstrated motion immediately. Thus the human instructor can observe the performance of the robot during training and stop demonstrating the trajectory once the robot's performance is satisfactory. This is similar to the idea described in [28] where the transfer of human motor skills to the robot is supported by a training system that keeps the human instructor and the robot in a real-time control loop. To generalize the learned movements we store the estimated frequency and the sampled trajectory from the last few periods that resulted in good movement reproduction as judged by the teacher. In our experiments we used the last five periods. Note that estimating the frequency is essential to ensure that the data from same number of periods is used in all training examples.

The above training process makes the following data available for generalization purposes: trajectory data points within last few periods of motion  $\{y_d^k(t_{k,j}), \dot{y}_d^k(t_{k,j}), \ddot{y}_d^k(t_{k,j}) | k = 1, \dots, M, j = 1, \dots, T_k\}$  and the associated frequencies  $\Omega_k$ . Each situation is characterized by some parameters  $\mathbf{q}_k \in \mathbb{R}^n$ . In our practical experiments with drumming, we used drums mounted at different heights and the relative difference in drum height was applied as query point. Given a new query point  $\mathbf{q}$ , we could now directly minimize objective function (25) to calculate the new parameters  $\mathbf{w}$ . However, since the data was acquired by incremental learning, it is better to generalize incrementally as well. For this purpose we need to parse all of the available trajectory data points in proper order. To achieve this in the case of movements with differing frequencies, we need to maintain separate phase information for all trajectories. The procedure shown in Fig. 1 ensures that the data from all trajectories is parsed and discounted uniformly.

### D. Estimation of Attractor Points, Timing, and Frequencies

Unlike  $\alpha_x, \alpha_z, \beta_z, N, c_i$  and  $h_i$ , which are kept constant across the example trajectories during generalization, time constant  $\tau$  in case of discrete movements and frequency  $\Omega$  in the case of periodic patterns as well as the attractor points  $g_i, i = 1, \dots, D_g$ , where  $D_g$  is the dimension of the space in which motion is specified, change from example to example. All of them can be viewed as a function of query points  $\mathbf{q}$ . For instance, in the case of reaching movements encoded in the joint space with the destination position determined by vision in Cartesian space, the function that maps  $\mathbf{q}$  to  $\{g_i\}_{i=1}^{D_g}$  is the inverse kinematics of the robot. In the case of throwing movements, the query points are given as target basket position and the relationship becomes more complex. This relationship needs to be estimated from the data.

Among various paradigms that could be used for this purpose, Gaussian process regression (GPR) has proved to be very effective. It can be considered as a Bayesian regression method and provides a predictive distribution. Therefore, GPR can show good generalization performance and the predictive distribution can be used to measure the uncertainty of the estimated function. It has been demonstrated that this technique outperforms other regression methods on problems such as estimating inverse dynamics of a seven degrees of freedom robot arm [29]. A Gaussian process is defined as

$$g(\mathbf{q}) \sim \mathcal{GP}(m(\mathbf{q}), k(\mathbf{q}, \mathbf{q}')), \quad (27)$$

where  $m(\mathbf{q}) = \mathbb{E}(g(\mathbf{q}))$  is the mean function and  $k(\mathbf{q}, \mathbf{q}') = \mathbb{E}((g(\mathbf{q}) - m(\mathbf{q}))(g(\mathbf{q}') - m(\mathbf{q}')))$  the covariance function of the process. Lets further assume that we have a set of noisy observations  $\{(\mathbf{q}_k, y_k) | k = 1, \dots, M\}$ ,  $y_k = g(\mathbf{q}_k) + \epsilon$ ,  $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ . Subtracting the mean from the training data we can assume that  $m(\mathbf{q}) = 0$ . If we are given a set of query points  $g(\mathbf{q}^*)$ , then the joint distribution of all outputs is given as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma_n^2 \mathbf{I} & \mathbf{K}(\mathbf{Q}, \mathbf{Q}^*) \\ \mathbf{K}(\mathbf{Q}^*, \mathbf{Q}) & \mathbf{K}(\mathbf{Q}^*, \mathbf{Q}^*) \end{bmatrix}\right), \quad (28)$$

where  $\mathbf{Q}, \mathbf{Q}^*, \mathbf{y}, \mathbf{y}^*$  respectively combine all inputs and outputs and  $\mathbf{K}(\cdot, \cdot)$  are the associated joint covariance matrices calculated according to (27). It can be shown [29] that the expected value  $\bar{\mathbf{y}}^*$  is given by

$$\bar{\mathbf{y}}^* = \mathbb{E}(\mathbf{y}^* | \mathbf{Q}, \mathbf{y}, \mathbf{Q}^*) = \mathbf{K}(\mathbf{Q}^*, \mathbf{Q}) [\mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{y}, \quad (29)$$

with the following estimate for the covariance of the prediction

$$\text{cov}(\mathbf{g}^*) = \mathbf{K}(\mathbf{Q}^*, \mathbf{Q}^*) - \mathbf{K}(\mathbf{Q}^*, \mathbf{Q}) [\mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{K}(\mathbf{Q}, \mathbf{Q}^*).$$

One commonly used covariance function is

$$k(\mathbf{q}, \mathbf{q}') = \sigma_f^2 \sum_{i=1}^n \exp\left(-\frac{1}{2} \frac{(q_i - q'_i)^2}{l_i^2}\right), \quad (30)$$

which results in a Bayesian regression model with an infinite number of basis functions.  $n$  denotes the dimension of the query point space. See [29] for more details.

Each dimension of every example movement in the library is described by the sampled trajectory points, the attractor

### procedure CollectTrainingData

- acquire** trajectory points  $\{y_d^k(t_{k,j}), \dot{y}_d^k(t_{k,j}), \ddot{y}_d^k(t_{k,j})\}$   $k = 1, \dots, M, j = 1, \dots, T_k\}$  by kinesthetic guiding or imitation;
- extract** the attractor points  $\mathbf{g} = \{g_k\}$  and time constants  $\tau = \{\tau_k\}$  or frequencies  $\Omega = \{\Omega_k\}, k = 1, \dots, M$ ;
- associate** the acquired trajectories with query points  $\mathbf{Q} = \{\mathbf{q}_k\}$ ;
- calculate** the Cholesky decomposition of covariance matrices  $\mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma_n^2 \mathbf{I}$ ;

### procedure GeneralizeTrajectory

- for the given query point  $\mathbf{Q}^* = \mathbf{q}^*$  and using Gaussian process regression (29), **estimate** the attractor point  $g^*$  and time constant  $\tau^*$  or frequency  $\Omega^*$  (with  $\mathbf{y} = \mathbf{g}, \tau, \Omega$ , respectively);
- minimize** (25) to estimate the parameters  $\mathbf{w}_i \in \mathbb{R}^N$  specifying the optimal DMP for each query point  $\mathbf{q}_i^*$ ;

Fig. 2. Training and generalization of goal-directed actions (for one dimension of the space in which trajectories are defined)

point  $g$  and the time constant  $\tau$  or frequency  $\Omega$  and the associated query points  $\mathbf{q}$ . Note that  $\tau$  and  $\Omega$  remain constant across dimensions whereas  $g$  changes. A number of examples such as reaching with grasping, ball throwing, and drumming, which show that query points can be defined in a natural way, have already been mentioned. All this data can easily be acquired when collecting the movements (in case of discrete movements,  $g$  is simply the final position on the trajectory and  $\tau$  is the movement duration, while in case of periodic movements,  $g$  is the mean of the sampled trajectory points and  $\Omega$  is the estimated frequency). Based on this information we can calculate the Cholesky decomposition of covariance matrices  $[\mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma_n^2 \mathbf{I}]^{-1}$ , which depend only on training data. Note that by writing  $\mathbf{z} = [\mathbf{K}(\mathbf{Q}, \mathbf{Q}) + \sigma_n^2 \mathbf{I}]^{-1} \mathbf{g}$ , the new attractor points  $\bar{g}^*$  (and similarly  $\tau^*$  or  $\Omega^*$ ) associated with the query  $\mathbf{Q}^* = \mathbf{q}^*$  can be written as

$$\bar{g}^* = \sum_{k=1}^M k(\mathbf{q}^*, \mathbf{q}_k) z_k. \quad (31)$$

Thus similarly as in LWR process (25), the data is weighted based on the distance of the training query points from the current query point. To generate a new action, the robot is first given a new query point. Using GPR, the optimal  $g$  and  $\tau$  (or  $\Omega$ ) for this query are calculated. The optimal parameters  $\mathbf{w}$  are estimated using the procedure of Section III-B and III-C. The sketch of the complete procedure is given in Fig. 2.

## IV. EXPERIMENTS

We conducted six sets of experiments to demonstrate the usefulness of the proposed approach. The following tests were done for discrete movements: (1) a real-world reaching study, (2) a study that connects the proposed system with grasping and active vision, (3) a simulated reaching study to evaluate the accuracy of reproduction when learning discrete movements, and (4) a simulated ball throwing study that demonstrates



Fig. 3. Image sequence showing the teaching of the trajectories to HOAP-3 robot with kinesthetic demonstration. Teaching of reaching movements to CB-i was done in the same way.

how more dynamic tasks can be generalized. To validate the generalization of periodic movements, we conducted (1) a simulated periodic pattern movement study, and (2) a real-world drumming experiment.

Two humanoids were used to validate the proposed approach; a small humanoid robot HOAP-3 built by Fujitsu Automation and a full-size humanoid robot CB-i built in collaboration between Sarcos, ATR, and JST [30]. Both robots enable the acquisition of trajectories via kinesthetic guiding. CB-i is, however, a much more capable system with which we could study the learning of control policies integrated with full-fledged active vision.

### A. Real-World Reaching

The first set of experiments demonstrates the generalization of reaching movements on a humanoid robot HOAP-3. By means of kinesthetic guiding (see Fig. 3) we recorded a set of 25 reaching movements, which were used to build a library of example movements. The example trajectories in task space and the associated query points are presented in Fig. 4. All the movements roughly originated from the same starting position ( $P_0$ ) and ended on a grid roughly in the coronal plane of the robot,  $\sim 0.15$  m in front of it.

The robot's forward kinematics was used to calculate the final end-effector's positions in Cartesian space on the trajectory. Gaussian process regression (see Section III-D) was

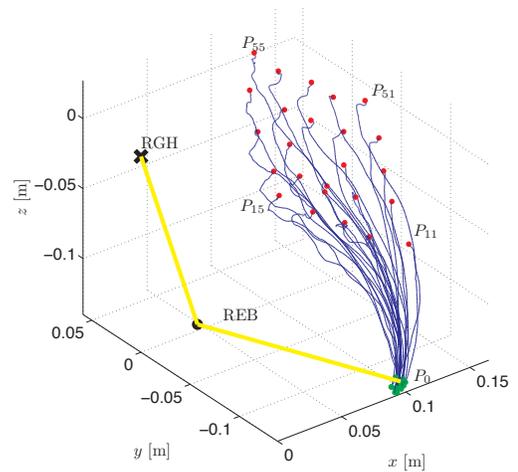


Fig. 4. Query points and trajectories for the HOAP-3 robot reaching experiment. The goals for the demonstration movements are marked with red dots. The starting positions are marked with green dots and the trajectories with blue lines. The robot's arm is in yellow. The shoulder (RGH) and elbow (REB) joints are marked.

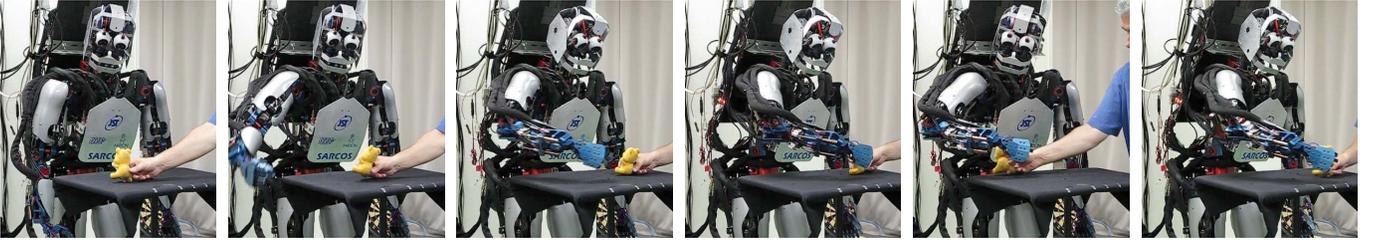


Fig. 6. Image sequence showing the robot reaching over the edge of the table towards a target. The robot tracks the target with its own eyes and determines the target’s location, which is used to generate a new attractor point  $g$  for the DMP using GPR. This is done in real-time. The initially open-loop movement that avoids the table thus gradually transforms into a closed loop movement that follows the object. Note the different head and eyes postures in the images.

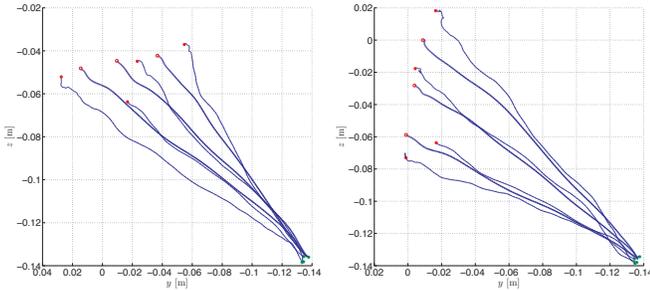


Fig. 5. Evaluation of the real-world reaching experiment. Both graphs show the results of generalization when query points smoothly change along a dimension of the query point space. The generalized joint trajectories (bold) resemble the trajectories of the near demonstrations (thin). The left graph shows changing of the query point in  $y$  direction and the right graph in  $z$  direction. Note that the  $y$  axis is reversed to preserve similarity to Fig. 4. Query points associated with demonstration trajectories are presented with full circles and the query points for generalized movements with empty circles.

applied to estimate the mapping from these positions, which served as query points, to the corresponding joint angles at the final configuration on the trajectory. Thus, in this case GPR learns a local approximation for the inverse kinematics. Using GPR we also estimated the mapping from query points to the time duration of the trajectory. Four joints (shoulder and elbow) were used in this task.

Fig. 5 shows the generalized trajectories projected onto the  $yz$  plane. The new trajectories, presented in bold, resemble the trajectories of reaching associated with the near demonstrations, presented by thin solid lines. The generalization procedure continuously transitions between example trajectories when changing the query points in both  $y$  and  $z$  direction, while  $x$  remains roughly the same, just as in the demonstrations. The accuracy of reproduction is further studied in Section IV-C.

### B. Integration with Active Vision and Grasping

In our next experiment we focused on the integration of the proposed approach with active vision. The role of active vision is to provide parameters, i.e. query points, characterizing the task. In the case of reaching and grasping, the task is characterized by the position of the object to be grasped. The use of active vision is essential if the robot is to find and grasp objects in a natural way. In this experiment we used a full-size humanoid robot CB-i for testing. Like in the case of HOAP-3, the training data was obtained by guiding the robot through 25 example trajectories. Besides reaching towards the desired position, the acquired trajectories also avoid the table. Unlike in the case of HOAP-3, the positions of the target object were

acquired by the robot’s own visual system.

CB-i’s oculomotor system has 7 degrees of freedom (three in the neck and two in each eye), which ensures great flexibility when the robot needs to find and direct its view towards objects in the scene. To compute 3-D data in body coordinates by stereo vision, such a robot must continuously update the position and orientation of the cameras in the robot body coordinate frame, which is very difficult to accomplish with high precision. Both camera locations depend on the current joint configuration of the robot. The appropriate estimation and calibration procedures are described in [31]. This work showed that even with carefully designed calibration procedures, it is difficult to estimate all necessary coordinate frame transformations with high accuracy. As a consequence, we cannot rely on a very accurate 3-D vision on full-size humanoid robots having oculomotor systems with many degrees of freedom.

The core part that integrates active vision with the developed generalization system is the estimation of the mapping from the desired Cartesian positions (query points) to the final joint positions (DMP attractor points) on the sampled trajectories. This mapping is obtained by Gaussian process regression of Section III-D. Note that although the 3-D pose estimation in the robot base frame is not very precise due to many transformations involved in pose calculation, this does not affect the accuracy of the generated control policy because the system directly relates the measured poses to the joint configurations on the trajectory. The inaccuracies of pose estimation are thus taken into account by GPR, the system only needs to be repeatable.

As shown in Fig. 6, the robot needs to avoid the table while reaching toward the object. Based on the training trajectories that avoid the table, the system generates a new trajectory that also avoids the table, but at the same time ends at the desired Cartesian position. Without estimating the shape parameters  $w$ , the robot’s arm would collide with the table. The first three images of Fig. 6 and video **discrete-reach-open.mov**, which is attached to this paper, show the successful open-loop reaching with obstacle avoidance. Once the robot’s arm reaches over the table, we estimate the new attractor point  $g$  using the results of active vision and GPR. No additional programming was needed to create the closed-loop behavior, just active vision was allowed to update the attractor points. Theoretically it is possible to update the shape parameters  $w$  as well, but depending on the amount of data, this calculation may require a few hundred milliseconds and is therefore unsuitable for feedback control. Once the time duration exceeds  $\tau$ , parameters  $w$  do not influence the movement any more. The last three images

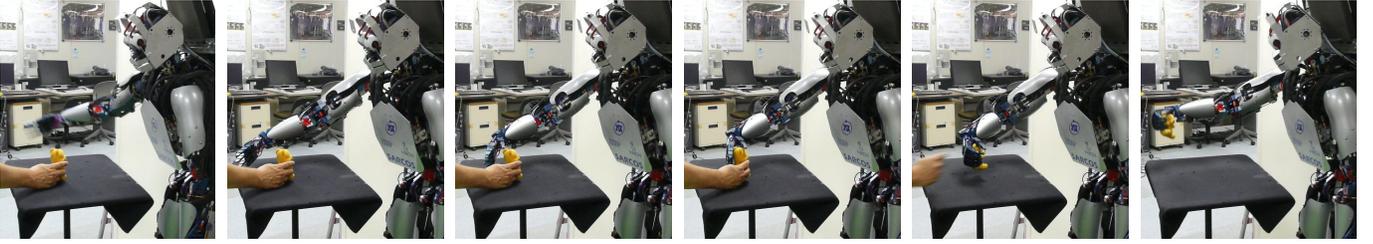


Fig. 7. Image sequence illustrating the grasping process. The head and eyes actively follow the object. To pick up an object, the robot applied power grasps.

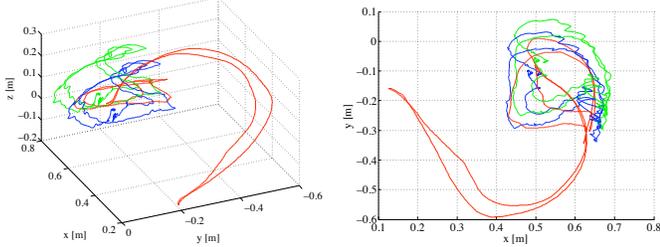


Fig. 8. The left image shows the 3-D path of the robot's hand (calculated by forward kinematics) while reaching and tracking an object (red). The object positions estimated by active vision are shown in green, while the blue curve shows the estimated object positions with corrected mean value so that it coincides with the mean of the hand motion. The right image shows the same data projected onto a plane, which is very roughly parallel to the table. The proposed approach is successful at correcting systematic vision errors.

of Fig. 6 and the attached video **discrete-reach-closed.mov** show that the robot can successfully follow a moving object, thus demonstrating the power of the DMP representation when reacting to the external feedback.

The resulting robot hand path while following the object in a closed-loop is depicted in Fig. 8. The mean value of robot hand motion while following the object moving on the table and the mean value of object motion differ by  $[1.6, 4.2, 7.6]$  cm. Although the modeling errors are nonlinear and not a simple displacement, we can assume that the average modeling error would be at least that much if we just used the estimated object's positions and converted them into joint angles by means of inverse kinematics. Such an error would be too large for grasping. In our system GPR successfully corrects at least part of the modeling errors and the resulting attractor points are accurate enough for grasping.

The realized grasping behavior is shown in Fig. 7 and in the attached video **discrete-grasping.mov**. Active vision detects when the object stops moving and initiates the open loop reaching. The attractor points for the initial reaching trajectory are generated by suitably displacing the estimated query points (based on the desired approach direction). After the initial reaching movements has finished, the vision system starts supplying the current object position and the second order attractor dynamics automatically generates a closed-loop approach motion for grasping. The system is accurate enough to pick an object from a hand of a person. Note that without active eye and head joints, the robot would not be able to follow the object, especially when it comes close to the body, and the behavior could not be generated.

### C. Simulated Reaching Study

In a computational study we examined how well Cartesian minimum jerk trajectories can be generalized. Minimum jerk

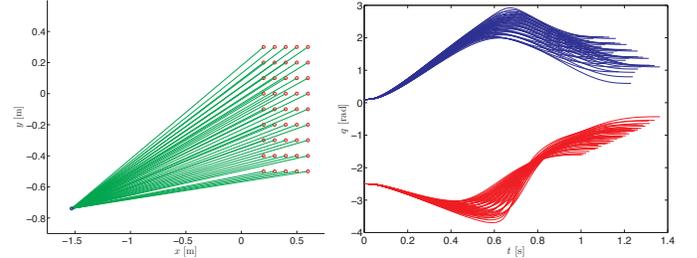


Fig. 9. 45 example minimum jerk trajectories in Cartesian space (left) and the associated joint space trajectories (right). In the figure left red circles depict the final reaching positions that were used as query points for LWR. The sum of limb lengths was 1.31 meters.

trajectories are often used in robotics because they resemble human reaching trajectories [32]. For training we generated 45 Cartesian minimum jerk trajectories, which were converted into joint space of a planar 2R robot (see Fig. 9). In Cartesian space these trajectories correspond to straight lines. The final end-effector positions on the trajectories were used as query points. The query points were distributed uniformly with spacing of 0.1 meters in a rectangular area with corners at locations  $(0.2, -0.5)$  and  $(0.6, 0.3)$  meters. Joint velocities and accelerations were computed analytically.

As usually we applied GPR to learn the mapping from the desired goal position in Cartesian space to the final joint configuration and time duration. In simulation, time duration was decided based on the distance of the end effector's final position from its initial position on the trajectory.

The errors in Tab. I and II were calculated by integrating equation system (4)–(6) to obtain joint positions  $\tilde{\mathbf{y}}(t_j)$ , and comparing the result to the ideal minimum jerk trajectory  $\mathbf{y}(t_j)$  expressed in the robot joint space. These ideal trajectories were computed using the same formulas as when generating the training examples. Both average (32) and maximum error (33) on the trajectory were estimated

$$\text{error}_{\text{average}} = \frac{1}{T} \sum_{j=1}^T \|\tilde{\mathbf{y}}(t_j) - \mathbf{y}(t_j)\|, \quad (32)$$

$$\text{error}_{\text{max}} = \max_{j=1, \dots, T} \|\tilde{\mathbf{y}}(t_j) - \mathbf{y}(t_j)\|. \quad (33)$$

Results in Tab. I demonstrate that reaching movements can be generalized with high precision. The resulting trajectories accurately generalize the spatial course of movement, its dynamics, and the final configuration. Since it can be expected that the errors will be larger on the boundary of query points used for training, we estimated the error both within the full rectangular area enclosed by all query points of Fig. 9, and in the reduced area enclosed by query points situated at least one query point away from the boundary points. As expected,

TABLE I

ERRORS IN REACHING MOVEMENTS (IN CENTIMETERS AND DEGREES, RESPECTIVELY) SYNTHESIZED BY LWR AND GPR.

Training	Joint space (across trajectory)		Cartesian space (final position error)		Grid size (centimeters)
	Full	Reduced	Full	Reduced	
Average error	0.24	0.19	0.12	0.09	10 × 10
Max. error	0.97	0.46	0.47	0.30	10 × 10

TABLE II

ERRORS IN REACHING MOVEMENTS (IN CENTIMETERS AND DEGREES, RESPECTIVELY) GENERATED BY A SINGLE DMP, WHICH WAS TRAINED TO REPRODUCE ONE OF THE EXAMPLE TRAJECTORIES.

Training	Joint space (across trajectory)		Cartesian space (final position error)		Grid size (centimeters)
	Full	Reduced	Full	Reduced	
Average error	8.85	5.62	0.43	0.32	10 × 10
Max. error	22.47	13.88	0.97	0.77	10 × 10

the errors are smaller for the internal query points.

Tab. II shows that representation with only one DMP is too rough for precise movement reproduction. While the final position could be reached accurately due to the properties of DMPs, the trajectory reproduction accuracy (32) is worse by an order of magnitude compared to the precision of the proposed approach. In Section IV-B we presented an example where the robot could not avoid the table if it did not reproduce the shape of motion. These results confirm that the proposed approach preserves the shape of the trajectory. The columns describing the Cartesian space error in both tables show that GPR is successful at estimating the inverse kinematics of the robot. The error is larger in Tab. II than in I because the execution was stopped at time  $\tau$ . The DMP obtained by generalization converges to the goal faster than the DMP that was trained to reproduce one of the example trajectories.

#### D. Ball Throwing

In the second simulation study we considered a task where the goal is less directly linked to DMP parameters than in the case of reaching. We studied the problem of throwing a ball into a basket, which is a dynamic task dependent not only on the positional part of the movement, but also on velocities. It is well known that the trajectory of the ball after the release is fully specified by the position and velocity at the release time

$$x = x_0 + v_0 t \cos(\alpha), \quad y = y_0 + v_0 t \sin(\alpha) - \frac{gt^2}{2}, \quad (34)$$

where  $(x_0, y_0)$  is the release point,  $v_0$  is the linear velocity of the ball at release time and  $\alpha$  is the initial angle of the throw path. We considered the problem where the target basket is placed in  $xy$ -plane. Since the robot can turn towards the basket, solving this problem enables the robot to throw the ball to any target position in space. The understanding of the physics of the task allows us to compare the movement generalization results with an ideal system.

The basket positions, i. e. the positions where the ball is supposed to land, were used as query points. They were uniformly

TABLE III

ERRORS IN THE SYNTHESIZED BALL THROWS (DISTANCE FROM THE TARGET IN CENTIMETERS)

Training area	Adams-Bashforth-Moulton integration		Euler integration		Grid size (centimeters)
	Full	Reduced	Full	Reduced	
Average error	2.82	2.00	4.29	4.01	50 × 50
Max. error	17.44	6.62	17.26	9.86	50 × 50

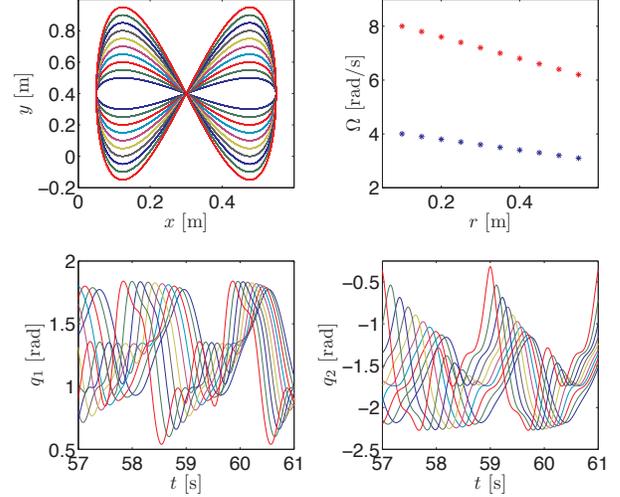


Fig. 10. *Top left:* Library of example figure-8 trajectories in Cartesian space. *Top right:* Frequencies of example movements,  $x$  direction in blue and  $y$  direction in red. *Bottom left:* Trajectories of joint one of the 2R planar robot. *Bottom right:* Trajectories of joint two of the 2R planar robot.

distributed with spacing of 0.5 meters within a rectangular area with corners at (1.2, 0.1) and (5.2, 2.1) meters. Example movements with proper position and velocity at the release time for the given basket positions were generated. Instead of the complete time duration, we used the timing at the release point as time constant  $\tau$ . GPR was used to approximate the mappings from a query point (basket position) to the DMP attractor point and to the timing at the release point. In this task the proper estimation of timing is much more important than when estimating the time duration of reaching trajectories.

As can be seen in Tab. III, we can generate throws to any target in space with the average accuracy of about 2 cm for the inner query point area where a sufficient amount of example movements is available. These results also show that the accuracy is higher when a more advanced integration technique such as Adams-Bashforth-Moulton method [33] is used to integrate the system (4)–(6). Errors are significantly larger when a simpler Euler’s method is applied. This demonstrates that while generalized DMPs approximate the desired trajectories and their dynamics with high precision, care must be taken to achieve a good enough accuracy for tasks that depend on accurate reproduction, such as the presented ball throwing example. This study shows that our approach can generalize not only the shape of motion but also its dynamics.

#### E. Simulated Periodic Pattern Study

Similarly to our analysis in the case of discrete movements, we conducted a simulation experiment to examine how well

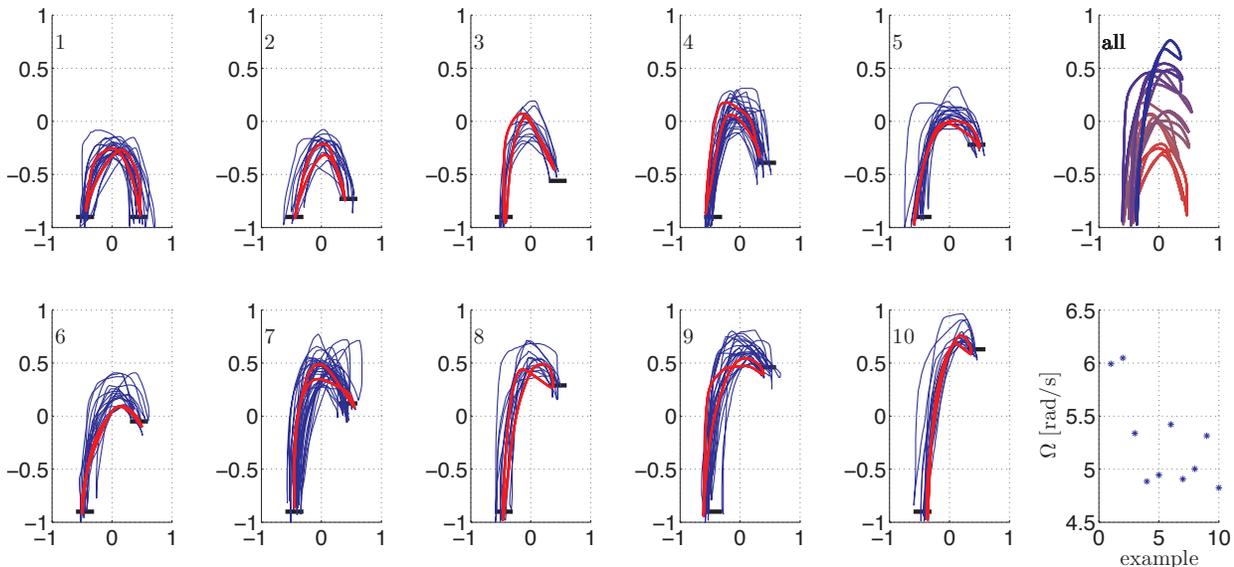


Fig. 12. Example periodic trajectories in Cartesian space. The blue lines show the input into the learning and the red line the final demonstration trajectory. To transform the demonstrated screen trajectories into Cartesian space, the values have to be multiplied by 0.2 m. All the trajectories are in the same plot in the top-right plot. The frequencies are in the bottom-right plot.

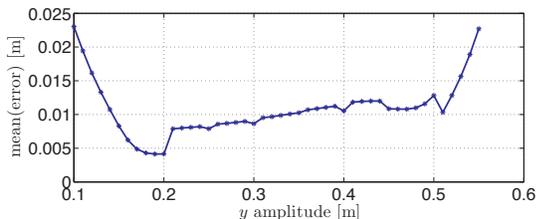


Fig. 11. Mean error of movement reproduction using movement generalization over the entire scope of the library of example movements with the step of 1 cm.

we can approximate periodic Cartesian trajectories of a robot end-effector. Just like in the simulated reaching study, we used a planar 2R robot for simulation.

As an example periodic motion we chose a Cartesian figure-8 trajectory with a varying amplitude and frequency. Several reasons speak for this, namely we can analytically generate a signal with an arbitrary amplitude for comparison, the motion frequencies of the two separate dimensions in Cartesian space are different, and the trajectories in joint space are highly nonlinear. While the primary aim of the periodic movement generalization algorithm is to produce DMP movements that cannot be attained by simple modulation, for the purpose of evaluation this section uses an example that could be easily attained by modulating the amplitude control parameter of the periodic DMP. The next section gives an example where generalization could not be achieved by simple modulation.

A set of example figure-8s in Cartesian space and the trajectories in joint space are depicted in Fig. 10. Cartesian space trajectories are distributed evenly from the smallest trajectory at  $x = 0.3 + 0.2 \cos(\Omega t)$  m and  $y = 0.4 + 0.1 \sin(2\Omega t)$  m, to the largest at  $y = 0.4 + 0.55 \sin(2\Omega t)$  m with a step of 5 cm ( $x$  does not change). The sampled joint space motion is used to form the library of example movements.

Using the amplitude in the  $y$  dimension as the query point parameter, we generalize the example joint trajectories, i.e.

calculate the parameters  $w$  of a periodic DMP, to the new amplitude as described in Section III-C. GPR is applied to estimate the frequency of the generalized joint space trajectory as a function of query points. Different frequencies in each dimension of the example demonstration movements (see Fig. 10) show that we can successfully estimate the periodic movements onto a single period. Once the new movement is generated, frequencies can easily be modulated with the periodic DMP frequency control parameter, just like it is done in a standard DMP approach (see [24] for details).

Fig. 11 shows the mean error of movement generalization over the entire scope of the library of example movements with the distance  $a$  set to 0.15. As expected, the distance of the generalized trajectory from the analytically computed trajectory increases at the edge of the database, as there example movements are available only on one side of the query point. The accuracy of generalization is slightly decreasing, which is the result of the increasing amplitude. The error in the generalized movements at query points that exactly coincide with the trained movements is clearly smaller than the error between them.

#### F. Real-World Drumming Experiment

In the final experiment we used the proposed approach to realize drumming on the humanoid robot CB-i. Of the total 39 DOFs, we used all 7 of the right arm for the execution of the movement. To demonstrate generalization, we trained the robot to perform one-handed drumming on two drums, which were mounted at different heights. Such drum placements – although with more drums – are common for drummers. Fig. 12 shows a set of example periodic trajectories in Cartesian space, which were recorded with a sampling rate of 100 Hz. The trajectories were scaled to the Cartesian space and mapped onto the robot’s joint angles via inverse kinematics in real-time. The trainer was modifying his motion based on the real-

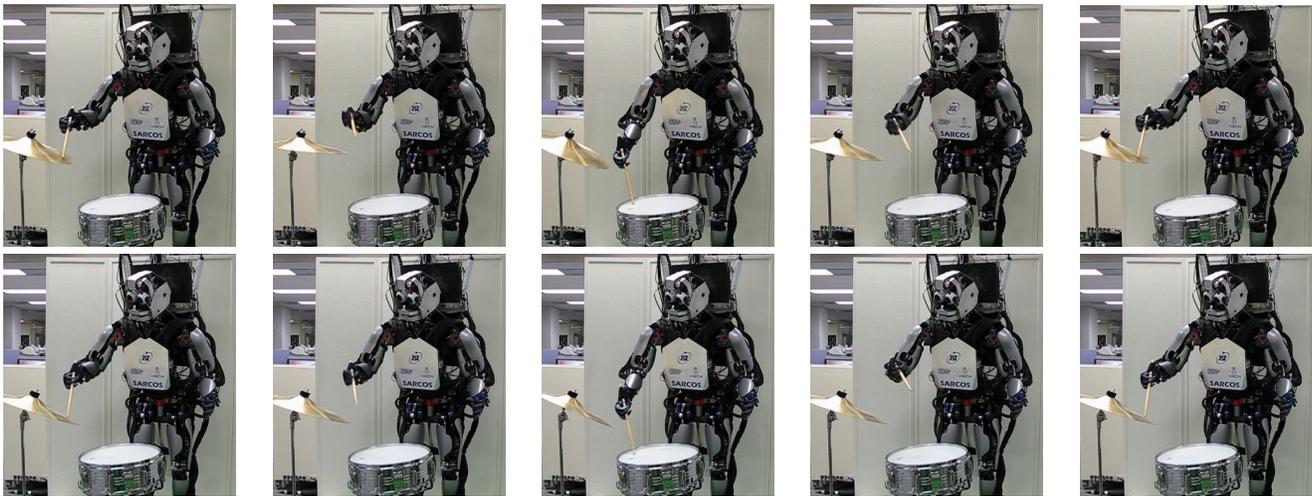


Fig. 13. Image sequence showing the drumming at two different heights. The cymbal is higher in the top row.

time feedback coming from the robot.<sup>2</sup> The left drum was placed at  $-0.162$  m. For training we mounted the right drum at  $[-0.177, -0.149, -0.091, -0.068, -0.049, -0.020, 0.017, 0.054, 0.079, 0.117]$  m. Thus the maximum height difference was about 30 cm. Video **periodic-examples.mov** shows the final example trajectories used for generalization.

Fig. 13 and videos **periodic-generalized1.mov** and **periodic-generalized2.mov** show the ability of the system to generalize the trained periodic demonstration trajectories. The resulting movement cannot be attained through simple modulation of a periodic DMP. Such generalization could be achieved only by modifying the underlying DMP equations [24], which would for the given example require a sophisticated set-up to modify the movement on only one side. The proposed approach can by itself generate appropriate movements directly from the collected data. The query point for generalization is the height difference between the two drums. While the height difference could be estimated by vision, we simplified this experiment by measuring the difference with a ruler and providing it to the algorithm.

Fig. 14 presents the results of generalization for four randomly selected query points. The figure shows the resulting periodic trajectories (blue) and the trajectories above (red) and below (green) the query point, which were used for generalization. Due to the limited support of the weighting kernel (23), the resulting trajectories are generated only from the four plotted example trajectories. As the trajectories were generated by human demonstration, they are quite different amongst themselves. Nevertheless, the algorithm was able to generate a trajectory similar to the example trajectories. If the demonstrations were more uniform, the generalized trajectories would be as well.

## V. FINAL DISCUSSION AND CONCLUSION

The action generalization approach proposed in this paper is realized using locally weighted regression and Gaussian process regression. The computing time needed to generate

<sup>2</sup>The attached video **periodic-learning-reproduction.mov** shows the training process and the direct reproduction of the trained movements for a standard two-handed drumming task.

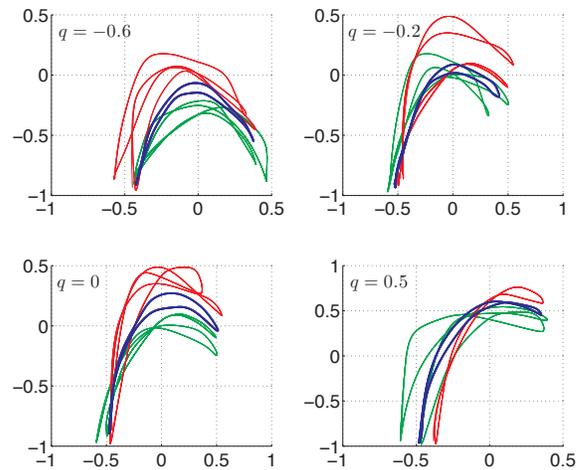


Fig. 14. The result of generalization for four randomly chosen query points. The figure shows the resulting trajectories in blue. The trajectories above the query point are shown in red and the ones below are shown in green. Among the 10 example trajectories, only the red and green trajectories were used for the generation of the generalized movement.

a DMP given a particular query point linearly depends on the number of data points sampled from the example trajectories. Thus the increase in computing time required for generalization compared to the computing time for standard one-shot learning of discrete DMPs is only linear in the number of considered examples. Nevertheless, an efficient implementation is important if DMPs are to be generated on-line using the perceptual input. The relatively low computational complexity of optimization problem (25) allows us to generate new control policies directly from the sampled data. In this way we avoid the pitfalls associated with the projection of the available data into lower dimensional parameter spaces (sometimes called latent spaces), which can lead to over-smoothing and therefore losing important details of the task.

Other researchers proposed ways to generalize example movements to new situations. For example, in [16] it is suggested to appropriately weight different constraints, like the fidelity of movement reproduction and the end-effector Cartesian positions, with respect to their importance to the task. If the initial positions on the example trajectories are

found to be less important, the robot can modify its motion to reach the final position from a different initial position at the expense of fidelity of reproduction. Systems like this, however, require an analytical model to describe the constraints. The same is true with approaches that modify the underlying differential equations of the DMPs, e. g. [22]. While this can be easily achieved for some types of positional constraints, which often appear in tasks associated with reaching and require only the robot's kinematics to be known, such constraints become increasingly more complex for dynamic tasks such as drumming or ball throwing, which in real-world implementations might require more complex models [34] than what is given in (34). The approach proposed in this paper does not require any physical models of the task to be available, generalization is done in a purely statistical way. Only the appropriate parameter space needs to be known. As demonstrated in our examples, this parameter space can normally be specified in an intuitive way. In case of unforeseen perturbations we exploit the basic properties of DMPs for on-line modifications. In Section IV-B we have shown that the developed system can be used in an active feedback loop, even correcting some of the modeling errors causing uncertainties in 3-D vision data.

As described in Section III-B, the proposed generalization strategy only makes sense for problems with example movements that transition smoothly as a function of query points. There exist tasks where this is not the case. Consider for example reaching movements that need to avoid an obstacle before arriving to the final configuration. If there are two sets of example movements, each avoiding the obstacle from a different side, then example movements that avoid the obstacle from different sides should not be blended together. The proposed approach could still be used, but it would need to be supplemented by a suitable clustering procedure that determines sets of trajectories suitable for generalization.

## REFERENCES

- [1] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in Cognitive Sciences*, vol. 3, no. 6, pp. 233–242, 1999.
- [2] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 109–116, 2004.
- [3] M. Riley, A. Ude, and C. G. Atkeson, "Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching," in *Proc. 2000 Workshop on Interactive Robotics and Entertainment*, Pittsburgh, Pennsylvania, April/May 2000, pp. 35–42.
- [4] A. Ude, C. G. Atkeson, and M. Riley, "Programming full-body movements for humanoid robots by observation," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 93–108, 2004.
- [5] M. Ruchanurucks, S. Nakaoka, S. Kudoh, and K. Ikeuchi, "Humanoid robot motion generation with sequential physical constraints," in *Proc. IEEE Int. Conf. Robotics and Automation*, Orlando, Florida, 2006, pp. 2649–2654.
- [6] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *IEEE Trans. Robotics*, vol. 24, no. 6, pp. 1463–1467, 2008.
- [7] S.-H. Hyon, J. G. Hale, and G. Cheng, "Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces," *IEEE Trans. Robotics*, vol. 23, no. 5, pp. 884–898, 2007.
- [8] Y. Kuniyoshi, M. Inaba, and H. Inoue, "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," *IEEE Trans. Robotics Automat.*, vol. 10, no. 6, pp. 799–822, December 1994.
- [9] M. Lungarella, G. Metta, R. Pfeifer, and G. Sandini, "Developmental robotics: a survey," *Connection Science*, vol. 15, no. 4, pp. 151–190, 2003.
- [10] Y. Demiris and A. Meltzoff, "The robot in the crib: A developmental analysis of imitation skills in infants and robots," *Inf. Child Dev.*, vol. 17, pp. 43–53, 2008.
- [11] H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato, "A kendama learning robot based on bi-directional theory," *Neural Networks*, vol. 9, no. 8, pp. 1281–1302, 1996.
- [12] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, pp. 682–697, 2008.
- [13] L. Sciavicco and B. Siciliano, *Modeling and control of robot manipulators*. London: Springer, 2000.
- [14] S. Schaal, P. Mohajerian, and A. Ijspeert, "Dynamics systems vs. optimal control – a unifying view," *Progress in Brain Research*, vol. 165, no. 6, pp. 425–445, 2007.
- [15] T. Asfour, F. Gyarfas, P. Azad, and R. Dilmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," *Int. J. Humanoid Robotics*, vol. 5, no. 2, pp. 183–202, 2008.
- [16] A. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, pp. 370–384, 2006.
- [17] T. Inamura, I. Toshima, H. Tanie, and Y. Nakamura, "Embodied symbol emergence based on mimesis theory," *Int. J. Robotics Research*, vol. 23, no. 4-5, pp. 363–377, 2004.
- [18] D. Kulić, W. Takano, and Y. Nakamura, "Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden Markov chains," *Int. J. Robotics Research*, vol. 27, no. 7, pp. 761–784, 2008.
- [19] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 32, no. 2, pp. 286–298, 2007.
- [20] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, Washington, DC, 2002, pp. 1398–1403.
- [21] —, "Learning rhythmic movements by demonstration using nonlinear oscillators," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Lausanne, Switzerland, 2002, pp. 958–963.
- [22] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Proc. IEEE Int. Conf. Robotics and Automation*, Kobe, Japan, 2009, pp. 763–769.
- [23] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Daejeon, Korea, 2008, pp. 91–98.
- [24] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarčič, "On-line learning and modulation of periodic movements with nonlinear dynamical systems," *Autonomous Robots*, vol. 27, no. 1, pp. 3–23, 2009.
- [25] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic Hebbian learning in adaptive frequency oscillators," *Physica D*, vol. 216, pp. 269–281, 2006.
- [26] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally weighted learning," *AI Review*, vol. 11, pp. 11–73, 1997.
- [27] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York, NY: Wiley, 1992.
- [28] E. Oztop, J. Babić, J. G. Hale, G. Cheng, and M. Kawato, "From biologically realistic imitation to robot teaching via human motor learning," in *Neural Information Processing: 14th International Conference*, ser. Lecture Notes in Computer Science. Berlin / Heidelberg: Springer, 2008, pp. 214–222.
- [29] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press, 2006.
- [30] G. Cheng, S.-H. Hyon, J. Morimoto, A. Ude, J. G. Hale, G. Colvin, W. Scroggin, and S. C. Jacobsen, "CB: a humanoid research platform for exploring neuroscience," *Advanced Robotics*, vol. 21, no. 10, pp. 1097–1114, 2007.
- [31] A. Ude and E. Oztop, "Active 3-D vision on a humanoid head," in *Proc. 14th Int. Conf. Advanced Robotics*, Munich, Germany, June 2009.
- [32] T. Flash and N. Hogan, "The coordination of arm movements: an experimentally confirmed mathematical model," *The Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, 1985.
- [33] J. D. Hoffman, *Numerical Methods for Engineers and Scientists*, 2nd ed. CRC Press, 2001.
- [34] U. Frese, B. Bäuml, G. Schreiber, I. Schaefer, M. Hähnle, and G. Hirzinger, "Off-the-shelf vision for a robotic ball catcher," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Maui, Hawaii, 2001, pp. 1623–1629.

Submission (please tick):

Poster

Paper

# Generation of Human-like Motion for Humanoid Robots Based on Marker-based Motion Capture Data

Stefan Gärtner<sup>1</sup>, Martin Do<sup>1</sup>, Christian Simonidis<sup>2</sup>, Tamim Asfour<sup>1</sup>, Wolfgang Seemann<sup>2</sup> and Rüdiger Dillmann<sup>1</sup>

<sup>1</sup> Karlsruhe Institute of Technology (KIT), Institute for Anthropomatics - IAIM, 76131, Karlsruhe, {stefan.gaertner | martin.do | tamim.asfour | ruediger.dillmann}@kit.edu

<sup>2</sup> Karlsruhe Institute of Technology (KIT), Institute of Engineering Mechanics, 76131, Karlsruhe, {christian.simonidis | wolfgang.seemann}@kit.edu

To increase acceptance of humanoids as part of our everyday lives, it is essential that motions of humanoids become more realistic and human-like. A proper approach to achieve this requirement will be introduced within the scope of this paper by adopting marker-based human motion capture. For this purpose, constraining and mapping of prerecorded motions will be applied since robots may have different degrees of freedom (DoFs) as well as a different kinematic structure than a human. Regarding this challenge, the motion must be adapted to a given robot while preserving important human-like characteristics of the recorded motion.

In order to efficiently reuse captured movements on various robots, an intermediate model is needed decoupling representation of a motion, which can be stored in a motion repository, from its execution on an actual robot. On the contrary, there exist numerous human motion capture systems that produce output in terms of different models stored in different formats. To overcome this problem, the Master Motor Map (MMM), firstly introduced in [1], presents an appropriate interface based on a unified model. An overview of the proposed system is illustrated in Figure 1. In this paper, we will further propose an extension of this model by adding certain anthropomorphic properties, such as mass distribution, segment length, moment of inertia, etc. Such an anthropomorphic model of the segmented body is of use in terms of determining forward and inverse dynamics as well as motion synthesis and retargeting.

Over the last decades, a lot of attempts have been made to develop sufficient dynamic models for simulating and analyzing complex motions of the human. Various biomechanical models are thoroughly reviewed in [2]. In order to calculate forward and inverse dynamics, knowledge of body segment properties reported in [3, 4, 5, 6], are required. Since the effort is very high to create model for each subject individually, a unified whole-body model is used instead that can be scaled in terms of body weight and height. Linear scaling equations are therefore commonly used due to their expediency.

The MMM is defined as a three-dimensional reference kinematic model enriched with proper body segment properties. The strategy with respect to the kinematic model is to define the maximum number of DoFs that might be used by any applied module. The kinematic model of the MMM including DoFs and the Euler angle conventions is shown in [1]. The linear equations published in [6] are applied to our model as they represent the most complete and practical series of predictive equations providing all frontal, sagittal, and horizontal moments of inertia. The body segment properties are adjusted with respect to the kinematics of the MMM and listed in Table 1.

Our approach of adapting movements consists of two constrained large-scale non-linear optimizations covering different requirements as illustrated in Figure 1. The used objective functions should maintain desirable properties of the motion, such as characteristic oscillations or particular configurations, and should refuse undesirable artefacts leading to unnaturalness. In general, constraints are associated with anatomic, mechanical, and motor task limitations. These are required to be able to determine a unique configuration that fits best with the given motion data and meets predefined requirements corresponding to the observed environment. To solve the mentioned optimization problems, sequential quadratic programming (SQP) is applied.

Our first optimization adapts a motion, represented through three-dimensional marker trajectories that can be captured with sophisticated marker-based system such as Vicon [7], to the articulated MMM model. The applied marker set is shown in Figure 2. Several approaches [8, 9, 10] have been proposed in order to compute feasible joint angle trajectories applying non-linear optimization. The construction of a sufficient objective function based on minimization of the sum of the squared distance between precaptured and virtual markers will be shown in this paper. Within this scope, virtual markers are defined as fixed points on the surface of the voluminous anthropomorphic model which have to be set up in advance.

To finally execute movements on the robot ARMAR-III [11], we will show the required transformation from MMM to ARMAR-III including another constrained non-linear optimization, as firstly proposed in [12]. The method has been further enhanced by adding appropriate spacetime constraints, introduced in [13], and additional constraints covering dynamic requirements. Spacetime constraints are required in order to satisfy certain task-related constraints on a motion while minimizing the changes of the captured motion. We will adapt various pick-and-place, passing over, and pouring movements, captured with a Vicon human motion capture system, to our robot ARMAR-III.

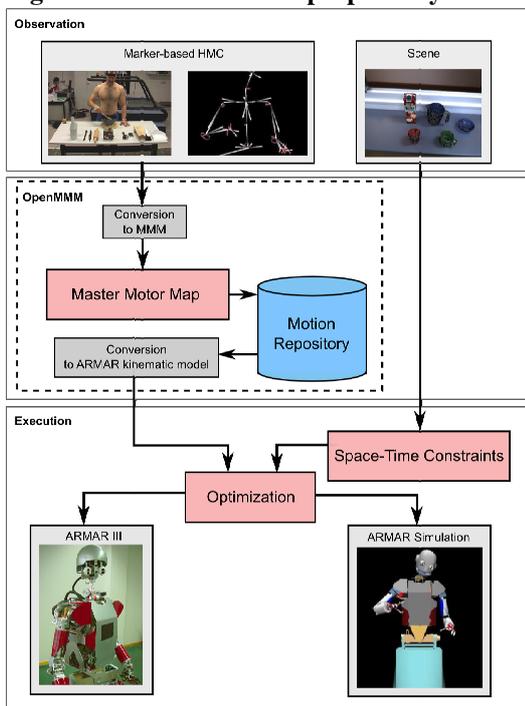
## Literature

- [1] P. Azad, T. Asfour, and R. Dillmann, *Toward an unified representation for imitation of human motion on humanoid-ids*, Robotics and Automation, 2007 IEEE International Conference on, 2007, pp. 2558 - 2563.
- [2] A. I. King, *A review of biomechanical models*, Journal of biomechanical engineering, vol. 106, 1984, pp. 97–104.
- [3] J. T. Barter, *Estimation of the mass of body segments*, Wright Air Development Center, Wright-Patterson Air Force Base, Ohio, WADC Technical Report 57-260, 1957.
- [4] R. Chandler, C. Clauser, J. Mc Conville, H. Renolds, and J. Young, *Investigation of the inertial properties of the human body*, National Technical Information Service, Virginia, Technical Report, 1975.
- [5] H. Hatze, *A mathematical model for the computational determination of parameter values of anthropomorphic segments*, Journal of biomechanical engineering, vol. 13, 1980, pp. 833–843.
- [6] P. de Leva, *Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters*, Journal of Biomechanics, vol. 29, no. 9, 1996, pp. 1223–1230.
- [7] Vicon Motion Systems. The Vicon Manual, 2002.
- [8] M. Riley, A. Ude, and C. Atkeson, *Methods for motion generation and interaction with a humanoid robot: Case studies of dancing and catching*, AAI and CMU Workshop on Interactive Robotics and Entertainment, 2000.
- [9] A. Safonova, N. S. Pollard, and J. K. Hodgins, *Optimizing human motion for the control of a humanoid robot*, Adaptive Motion of Animals and Machines, 2nd International Symposium on, 2003.
- [10] T.-W. Lu and J.J. O'Connor, *Bone position estimation from skin marker co-ordinates using global optimisation with joint constraints*, Journal of Biomechanics, vol. 32(2), 1999, pp 129-134.
- [11] T. Asfour, K. Regenstein, P. Azad, J. Schroeder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, *Armar-III: An integrated humanoid platform for sensory-motor control*, IEEE-RAS International Conference on Humanoid Robots, 2006.
- [12] M. Do, P. Azad, T. Asfour, and R. Dillmann, *Imitation of human motion on a humanoid robot using nonlinear optimization*, In International Conference on Humanoid Robots, 2008.
- [13] M. Gleicher, *Retargetting motion to new characters*, International Conference on Computer Graphics and Interactive Techniques, 1998, pp. 33- 42.

**Table 1: Adjusted body segment properties for the MMM model. Segment masses are relative to body masses; segment lengths are relative to body heights. Both segment center of mass and radii of gyration are relative to the respective segment lengths.**

Segment	Segment Length/ Total Body Height	Segment Weight/ Total Body Weight	Center of Mass/ Segment Length [x,y,z]	Radius of Gyration/ Segment Length [rxx, ryy, rzz]
Hip	0.26	0.11	[0 4 0]	[38 36,5 34]
Spine	0.10	0.10	[4 46 0]	[32 26 28,6]
Chest	0.18	0.17	[0 46 0]	[35 28,5 31,3]
Neck	0.05	0.024	[0 20 0]	[31,6 22 31,6]
Head	0.13	0.07	[12 13 0]	[31 26 30]
Shoulder R/L	0.10	0.021	[66 0 0]	[12 26 16]
Upper Arm R/L	0.16	0.027	[0 -57,3 0]	[26,8 15,7 28,4]
Lower Arm R/L	0.13	0.016	[0 -53,3 0]	[31 14 32]
Hand R/L	0.11	0.006	[0 -36 0]	[23,5 18 29]
Thigh R/L	0.25	0.14	[0 -33 0]	[25 11,4 25]
Shank R/L	0.23	0.04	[0 -44 0]	[25,4 10,5 26,4]
Foot R/L	0.15	0.013	[0 -6- 39]	[21 19,5 12]

**Figure 1: Overview of the proposed system.**



**Figure 2: Applied marker set for capturing whole-body motions of a human.**

