The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Imitating Object Movement Skills with Robots – A Task-Level Approach Exploiting Generalization and Invariance

Michael Gienger, Manuel Mühlig and Jochen J. Steil

Abstract— **This paper presents an architecture for learning and reproducing movements with a robot in interaction with a human teacher. We focus on the movement representation and propose three enhancements to increase generalization capabilities: Firstly, we introduce a flexible task-level movement representation that is based on neuropsychological findings. Movement is represented in task-oriented frames of reference, and generalizes to a variety of different situations. Secondly, we propose a mechanism to decouple the task descriptors from the perceived objects in the robot's environment. This allows to formulate a set of generic controllers, and to interactively create associations with perceived objects. Thirdly, we introduce a method to dynamically modify the system's body schema to account for structural changes such as having grasped a tool. The changes are consistently treated in the kinematics computations. This permits to generalize movements to be carried out in different ways, for instance with different hands or bi-manually. A set of experiments in an interactive imitation learning situation underline the capabilities of the proposed concepts.**

I. INTRODUCTION

The capability to acquire and imitate motor skills is one of the key aspects to achieve cognitive abilities in robotic systems. The recent years showed a trend from researching individual methods within this area towards combining them in more comprehensive architectures that comprise movement learning, representation, reproduction and planning capabilities. Such architectural approaches are important steps towards more autonomy and a fluent human-robot interaction. It is yet an unsolved question how to combine reactive low-level control schemes with more abstract representations up to a symbolic level. It seems that there is a significant dependency of higher level planning to the underlying movement representation. Particularly when it comes to applying acquired skills in novel situations, it is important to incorporate invariance and generalization capabilities into the movement representations.

A number of interesting architectures have been published in the recent time. A hybrid architecture to instruct a robot grasping tasks has been proposed in [1]. They incorporate active vision, gestural instruction and a dialog system and couple these elements with a hierarchical movement generation system. In [2], similar sub-systems are used, but integrated in a three-layered cognitive architecture. Movement is coordinated by a Petri-Net and distributed to the individual extremities of the robot by means of parallel interacting behavior-based components. A comprehensive architecture for grasp-oriented visual perception has been presented in [3]. They combine visual attention and different visual cues with grasp planning and inference strategies. Above approaches focus mainly on the architectural aspects, but incorporate rather simple movement generation schemes. More complex movement adaptation has been realized by combining planning methods and reactive movement control. While [4] proposes a scheme to couple sampling-based planners with local adaptations based on visual feedback, [5] learn a probabilistic representation and use optimal control techniques to adapt the trajectories to a given situation. A dynamical systems representation is proposed in [6]. Fast policy learning methods adapt the movement to disturbances.

Such trajectory adaptation mechanisms are capable to deal with dynamic environments. However, learning does usually not capture *how* (for instance with which effector) to execute the movement, and to which object to relate the movement. In many real-world problems, there exist several ways to solve a movement task, and it can be applied to various different objects. It is desirable to generalize learnt movements to different ways of reproducing them. This is related to the correspondence problem which is the transfer of movement skills to different embodiements. This has for instance been addressed in [7] by projecting the observed movements into the teachers frame of reference using a viewpoint transformation. However, movement is represented and reproduced using a fixed mapping of visual input to the degrees of freedom of the system. A joint-level movement representation according to a detailed kinematic model of a human has been presented in [8]. Movements are acquired using motion capture, and can be reproduced in different task-specific ways on a robot using a set of converter modules. An approach using dynamic Bayesian networks has been presented in [9]. The DBN is used to learning relations between the observed positions of the objects and body parts of the instructor. A number of approaches to learn and adapt the body schema have been proposed, for instance for proprioceptive models [10] and for models including tools [11], [12]. The concepts of body schema and body percept have been exploited in [13] to solve the correspondence problem. They define similarity of teacher and robot based on the effector movement in cartesian space, and apply their methods to imitation of hand writing and to more symbolic sequential tasks.

The architecture presented in this paper combines elements of prior work in the area of imitation learning [14], [15], movement control [16] and optimization [17]. While we

M. Gienger and M. Mühlig are with the Honda Research Institute Europe, Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany.

M. Mühlig and J. J. Steil are with the Research Institute for Cognition and Robotics, Bielefeld University, 33594 Bielefeld, Germany.

concentrate on the human-robot interaction in a related publication [18], this paper will emphasize the aspects of movement representation and generation. The objective is to provide methods to introduce generalization capabilities so that such architectures can perform a task in various ways and in different situations. We will focus on goal-directed movements with objects. In Section II we will give an overview on the architecture and its major elements. In Section III we will derive a flexible movement control scheme that comprises findings from neuropsychology. Section IV will introduce a mechanism to dynamically link perceptual information to the control system using an attention mechanism. This concept is enhanced in Section V by a method to dynamically modify the systems body schema at run-time. This allows to generalize task descriptors to a set of options how the movement can be executed, for instance one-handed or bi-manual. The proposed concepts have been verified in experiments with a humanoid robot, which will be explained in Section VI.

II. SYSTEM OVERVIEW

Figure 1 depicts the architecture underlying the presented system. It is divided into three layers, with an increasing level of abstraction from bottom to top. The left part of the figure refers to the perceptual functions, while the right side refers to the elements related to movement planning, prediction and control.

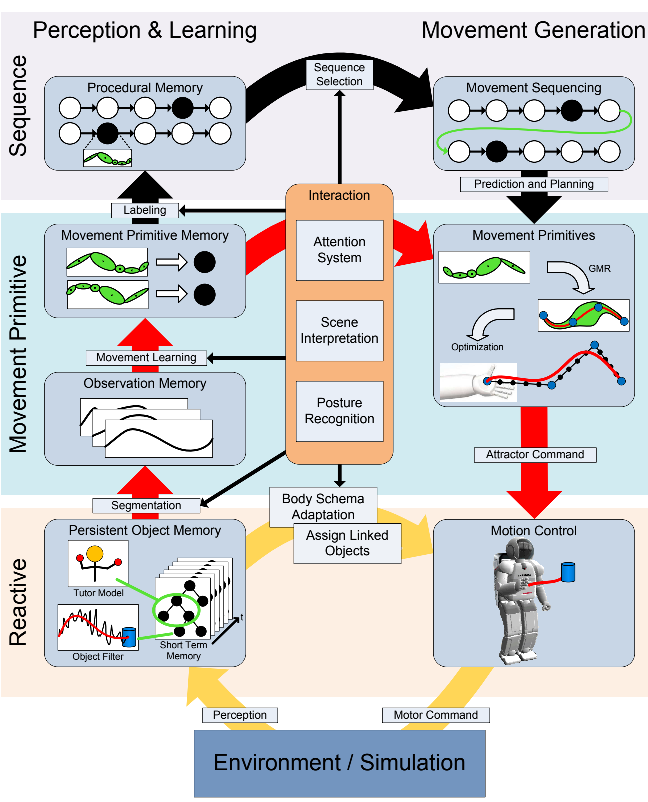

Fig. 1. System architecture

A. Perception and Learning

A fundamental element of this system is the *Persistent Object Memory* (POM). It is the interface between the system and the real world and can be interpreted as a simple working memory. The POM comprises a kinematic model of the robot as well as the spatial and geometrical information of the objects in its surrounding. Further, it maintains a model of a human tutor whose pose is determined with the robot's sensors. Each perceived entity is associated with a confidence value that determines the quality of the perception, for instance if objects are occluded or invisible. A short-term memory maintains a history of the recent past.

The *Interaction System* is connected to the POM and computes a set of cues. An important cue to structure the learning process is detecting distinct poses or gestures of the tutor. We utilize this cue to indicate the robot what to do, or to re-confirm actions the robot is about to take. Another important cue determines time events to segment object trajectories for imitation learning. It is based on the temporal coherence of hand- and object movements and explained in [18] in more detail. The system is additionally equipped with an attention system. Each object represented in the POM is associated with a saliency value. This value can be modified by the robot's interaction partner by shaking the object or pointing to it.

With the above elements, the tutor can show object movements to the robot. They are automatically segmented and stored as time series of object transformations in the *Observation Memory*. Having acquired a set of such observations, the tutor can instruct the robot to learn a movement representation for them. The employed learning scheme exploits the statistical characteristics of the observations and is presented in detail in [14]. We'll briefly recapitulate its fundamental capabilities: First, the observations are projected into a task space. While we introduced a scheme to automatically detect the most feasible task spaces in [15], a simplified scheme is used here. We relate the relative coordinates of the most salient object with respect to the second most salient object. This allows the tutor to interactively indicate which objects are important before demonstrating the movement to the robot. The observations are then temporally normalized using a Dynamic Time Warping algorithm. A Gaussian Mixture Model is then fitted to the normalized data, so that the movement is statistically represented by a mean and a variance. It is stored in the *Movement Primitive Memory*. The granularity of the primitives is on the level of simple reaching or stacking movements.

On the highest level, the system is equipped with a *Procedural memory*. This memory maintains sequences of movement primitives as discrete nodes. The nodes can correspond to learnt movements, or pre-defined ones. The sequence is represented as a hierarchical state chart. The nodes are connected by one or several transitions. We currently use a pre-defined set of sequences, which are for instance preparatory movements to grasp an object. Learnt movement primitives are currently embedded at pre-defined locations within the sequence. This allows us to perform the required preparatory movements for instance to grasp an object, before applying the learnt skill to it.

B. Movement generation

On the lowest level, movement is generated with a redundant whole body controller [16], [19]. It is based on the kinematic control scheme proposed by [20] and integrated with a walking and balancing control system [21]. The scheme allows to augment a task vector that is composed of a set of independent task descriptors. The coupling to the real world is realized through the POM and explained in Section III.

On the movement primitive layer we apply a linear attractor system to the selected task descriptors. This leads to a smooth, human-like movement that converges the robot's pose to the attractor target values. We developed movement primitives with different levels of complexity: Simple primitives converge the robot's trajectory reactively, with mechanisms to locally avoid joint limits or collisions. Sequential primitives are composed of several attractor vectors and with an optimization method to adapt them with respect to a set of criteria [17]. It is local in space, but anticipates a future time horizon. In decently complex environments, it allows to generate movements that are collision-free and optimal in other respects. Imitation primitives additionally incorporate the statistical information of the demonstrated movement. We developed a criterion that describes the similarity of the observed movement to the movement of the robot. The similarity is weighted with the variance at the corresponding time point. This results in imitated movements that reflect the tutor's characteristics precisely in phases with low variance, while phases with higher variance weight the other criteria stronger.

On the highest level, the movement primitives are linked to the discrete nodes of the *Procedural memory* and connected by transitions. The states correspond to the movement primitives, while the transitions correspond to sensory events, such as "Target reached, contact detected". They are all evaluated through the POM. Switching from one node to another is carried out in the following way: When entering the new node, the currently active task descriptors are replaced with those associated with the new primitive, and initialized with the current robots state. The new attractor targets are initialized from the given primitive. We also allow to leave the target unspecified. In this case, the target is set to the current value of the respective task descriptor. This way, we achieve invariance of a movement primitive with respect to different situations, for instance when the target of a task descriptor cannot be associated with a specific movement primitive, but depends on the prior movements of the system.

Each node is also equipped with an internal simulation of the controller. This permits to predict the future state of the robot based on the assigned targets. We currently use this prediction mechanism for two cases: Firstly, if a critical situation is predicted, the system will ask for a confirmation before it continues. Secondly, we use it to parallelize the learning and optimization steps with the movement of the robot. In this way, the system is able to already learn and optimize movements while it is carrying out the related preparatory movements.

III. TASK-LEVEL CONTROL

Biological findings suggest that human movement is encoded in a variety of action-oriented reference frames. For instance [22], [23] distinguish between egocentric and allocentric reference frames, and give evidence from neuropsychological studies. Egocentric frames are placed relative to the human, and comprise head-, arm-, gaze- and grasp-centered ones. Allocentric frames are represented in environmental coordinates, such as room- or object centered ones. In this section, we exploit these findings and derive task descriptors that relate the movement of one body with respect to any other body. This allows for instance to describe the position of one end effector with respect to the other, the orientation of the camera to the body, etc.

The robot's kinematics is described in the form of a tree structure depicted in Figure 2. The individual links are connected by degrees of freedom (joints) or fixed transformations. The tree may also comprise objects from the environment. To mathematically formalize this concept, we

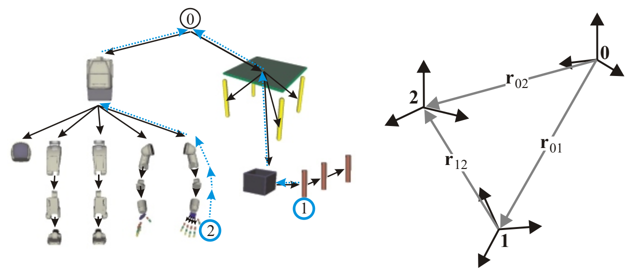

Fig. 2. Left: Kinematic tree - Right: Relative body coordinates

look at the relative kinematics of an articulated chain, such as depicted in Figure 2 right. Coordinate frame 0 denotes its root. Frame 1 is an arbitrary body which is connected to 0 through a set of joints. Body 2 shall be represented relative to body 1 with vector $\boldsymbol{r}_{12}$. We now can write the kinematic equations as follows:

$$\boldsymbol{r}_{12} = \boldsymbol{r}_{02} - \boldsymbol{r}_{01} \qquad \dot{\boldsymbol{r}}_{12} = \dot{\boldsymbol{r}}_{02} - \dot{\boldsymbol{r}}_{01} + \boldsymbol{\omega}_1 \times \boldsymbol{r}_{12} \quad . \tag{1}$$

The outer product term of eq. (1) right is due to the angular velocity $\boldsymbol{\omega}_1$ of body 1. Introducing the coordinate system in which the respective vector is represented as the left sub-index and projecting the velocities into the state space with the respective translational $\dot{\boldsymbol{r}}_i = \boldsymbol{J}_{T,i}\,\dot{\boldsymbol{q}}$ and rotational Jacobians $\boldsymbol{\omega}_i = \boldsymbol{J}_{R,i}\,\dot{\boldsymbol{q}}$, the differential kinematics gets

$${}_1\dot{\boldsymbol{r}}_{12} = \boldsymbol{A}_{10}\left({}_0\boldsymbol{J}_{T,2} - {}_0\boldsymbol{J}_{T,1} + {}_0\tilde{\boldsymbol{r}}_{12}^T\,{}_0\boldsymbol{J}_{R,1}\right)\dot{\boldsymbol{q}} = {}_1\boldsymbol{J}_{T,rel}\,\dot{\boldsymbol{q}} \tag{2}$$

with $\tilde{\boldsymbol{r}} = (\boldsymbol{r}\times)$ being a skew-symmetric matrix representing the outer product, and $\boldsymbol{A}_{10}$ being a rotation matrix from frame 0 to frame 1. If the reference ("1") body corresponds to a fixed frame, it has no velocity and the corresponding

Jacobian is zero. In this case, we get the differential end effector kinematics with respect to an inertial (world-fixed) coordinate system.

The task descriptors for a segments spatial orientation can be computed for instance in Euler (3d) or Spherical angles (2d), or as the inclination of one body axis with respect to any other (1d). It needs to be mentioned that the mapping from a rotation matrix to a serial angle representation is not unique. We therefore compute the differential kinematics in terms of the (unique) angular velocities

$$ {}_1\boldsymbol{\omega}_{12} = \boldsymbol{A}_{10}\left({}_0\boldsymbol{J}_{R,2} - {}_0\boldsymbol{J}_{R,1}\right)\dot{\boldsymbol{q}} = {}_1\boldsymbol{J}_{R,rel}\,\dot{\boldsymbol{q}}\,. \tag{3} $$

and compute the feedback term $\Delta\boldsymbol{e}$ in eq. (4) with some unique representation, such as the CLIK formulation of [24] for Euler angles, or our formulation [16] for Spherical angles. We also use task descriptors for the linear and angular momentum, or individual joint angles, which are skipped for brevity.

With these equations, we can formulate task descriptors that relate any segment of the tree to any other. Further, it is possible to compute these descriptors element-wise, such as "position of body 2 with respect to body 1 in x-direction", or "Euler α angle of body 2 with respect to body 0".

For a set of task descriptors, we augment an overall task Jacobian, and compute the joint rates with an inverse kinematics scheme based on the concept presented in [20]

$$ \delta\boldsymbol{q} = \boldsymbol{J}^{\#}\,\Delta\boldsymbol{e} - \alpha\,(\boldsymbol{I} - \boldsymbol{J}^{\#}\boldsymbol{J})\boldsymbol{W}^{-1}\left(\frac{\partial H}{\partial \boldsymbol{q}}\right)^{T} \tag{4} $$

where $\boldsymbol{J}^{\#}$ is a $\boldsymbol{W}$-weighted Pseudo-Inverse of the augmented task Jacobian, $\Delta\boldsymbol{e}$ is the feedback term of the task coordinates, and H is a secondary objective whose gradient is projected into the null space of the movement through the right term of eq. (4). We utilize terms to avoid joint limits and proximities to obstacles as described in [17].

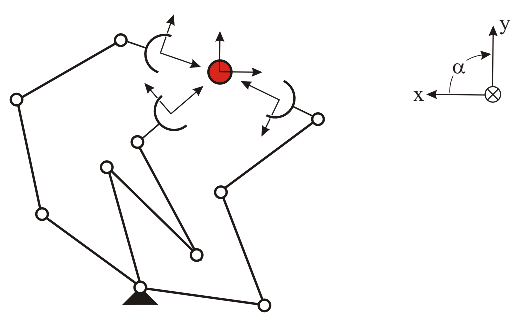

Fig. 3. Different robot postures according to the same task represented in effector coordinates $(x\ y\ \alpha)^T$

The choice of the order of the relative coordinates yields some interesting aspects. This is illustrated in Figure 3 for a simple planar redundant system controlled with task variables $(x\ y\ \alpha)$. If the task variables are represented in the objects frame of reference, different values are needed to realize the depicted poses. If, like depicted, the orientation between object and end effector is not important, it may be more advantageous to represent the task variables in the effector's frame of reference. In that case, all three poses can be realized with the same values. This task description introduces an invariance with respect to the relative pose between effector and object. Its null space comprises the relative pose between effector and object. When resolving redundancies with eq. (4), the achieved pose will correspond to a (local) optimum with regard to the cost function H.

An important property of this concept is the decoupling from the task description from the absolute or world coordinates. When for instance representing the left hand's transformation in the frame of reference of the right hand, the world coordinate trajectories emerge from the secondary objectives in eq. (4). Both hands absolute transform will vary over time according to the secondary objective, while their relative coordinates track the task variables. The absolute coordinates are in that way resolved in the null space of the movement. There are many other examples, such as representing a gazing controller as an object in head-centered coordinates which is "pointed" to by the focal axis, or a pointing controller in a similar way.

IV. Linked Objects

In the previous section, we presented a scheme to derive task descriptors that can comprise robot - object or object - object relations. In dynamic environments, the number of objects as well as their identities and geometrical shapes are not known in advance. In order to decouple the task descriptors from a concrete object identity and such to decrease the number of required task descriptors, we introduce the concept of *linked objects*. This is a way to dynamically couple salient objects to the bodies associated with the task descriptors introduced in the previous section. A linked object may be associated with a perceived object within the POM, or directly refer to the world reference. Linked objects are the entities on which the object-related task descriptors are formulated, and which are constituting to the objects to be considered in the collision avoidance.

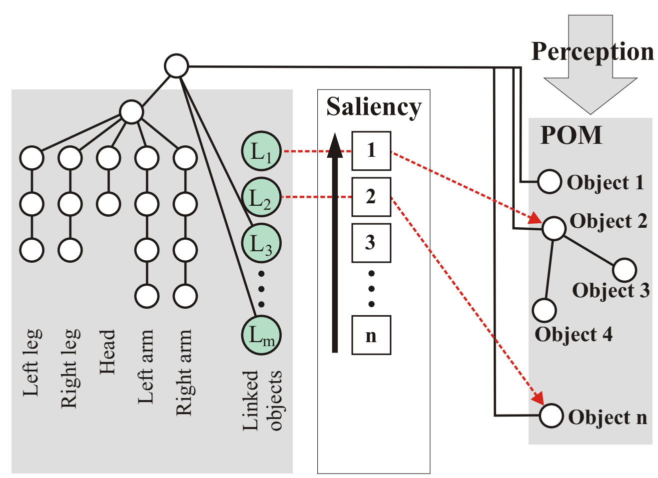

Fig. 4. Linked objects

In order to create the links, we use an attention mechanism. This mechanism organizes an ordered list of salient objects. We directly assign the salient objects of this list to the linked objects in their order: The linked object one refers to the most salient object, linked object two to the second most salient object etc. If an object's saliency is below a threshold, the association to a linked object is deleted, and the linked object refers to the world reference. This is depicted in Figure 4. Linked object L_1 is associated with object 2 and has the highest saliency. L_2 is associated with object n.

During interaction, we can now use the attention mechanism to indicate the important objects to the system. The tutor can increase an object's saliency by shaking or pointing to it, and decrease it by hiding it. If an object's saliency exceeds another body's saliency, the list will be sorted, and the link associations will be updated. The saliency computation includes a small hysteresis so that reorganizing the links is insensitive to sensor noise.

This is utilized both in learning movements, and in movement generation:

- Movement learning
  In this work, we assume that the demonstrated movement is mainly characterized by the relative movement of the objects. Before computing a generic movement representation based on a set of observations, the trajectories of linked object 1 and 2 are therefore projected into the space of relative object coordinates. This means that the movement of the most salient object is represented in the coordinates of the second most salient object. If only one object is salient, its trajectories are represented in world coordinates.
- Movement generation
  This is the more interesting case. Formulating a task descriptor that relates to linked objects now gives us the flexibility to interactively change the robot's behaviour. If we for instance formulate a task descriptor to gaze at linked object one, the robot will always track the most salient object in the scene. The same applies for reaching or approaching an object. If we would for instance like to reach for object three, we simply point to it before we make the robot perform the reaching movement with a task descriptor relating the hand position to the linked body one.

V. BODY SCHEMA ADAPTATION

Kinematic structures as depicted in Figure 2 represent a parent-child hierarchy: The movement of a segment will affect the movement of its children. In many practical situations, changes to this kinematic configuration occur. An example is a robot grasping an object and putting it at a different position. Another example is to put an object from a table on a tray which is placed on the table. A common approach to deal with such changes is to keep the kinematic configuration, but to compute the robots end effector coordinates based on the desired object transformation. This way, the movement can be controlled in end-effector coordinates, and collisions can be taken into account by applying avoidance strategies based on the transformed object geometry.

We propose to address this problem by adapting the body schema, which commonly refers to the perception of a humans physical appearance, or the interpretation of the body by the brain. In the following, we assume the geometric properties of our system to be known, and rather focus on dealing with structural changes during interaction with the environment. We argue that kinematic structure modifications can be modeled in a higher abstraction of the movement generation system, such as in actions or in action sequences. For instance if a robot "grasps" an object, it is either known (or it can be reconfirmed by tactile or visual feedback) that the grasp is successful and the object is held by the robot's end effector.

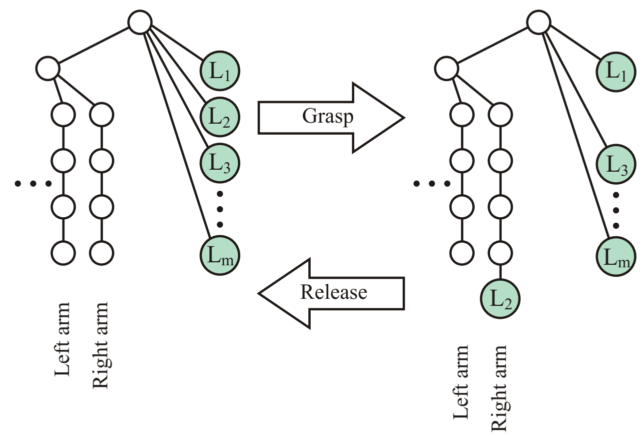

Fig. 5. Adaptation of the kinematic chain according to performed action. The linked objects are denoted with L_i, index i being the saliency index.

We suggest to exploit this knowledge and apply such structural modifications based on actions like grasping or releasing an object. This is depicted in Figure 5. Applying an action that grasps linked object 2 will modify its connectivity so that it is connected to the grasping hand of the system. The relative transformation between linked object and hand has to be computed according to the robots state at the time the structural change occurs, so that the alignment is consistent with the perception. It should be noted that this also accounts for the case where the linked object refers to a parent-child structure like object 2 in Figure 4. An example would be to grasp a tray on which two objects are placed. In the same way, releasing the linked object can be associated with connecting it to the worlds frame of reference, or any other object at which it is positioned.

This approach is beneficial, since firstly, an abstraction of the embodiment is introduced. Object movements are generic, while the movement of an end effector always incorporates the knowledge about a specific embodiment. Secondly, representing movements in object coordinates allows to introduce invariance in the same line of argument as discussed in Figure 3: Stacking a cylinder on top of another can be described by aligning the cylinders symmetry axis, while it is rather difficult to find a general end-effector object relation.

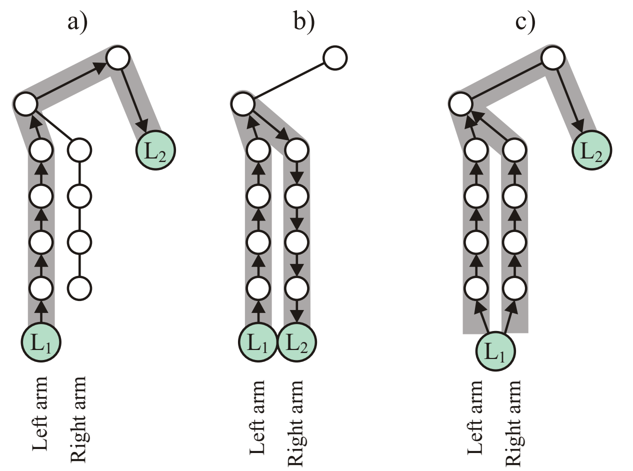

Fig. 6. Kinematic chains for different body schemas. The gray lines cover the joints and transformations that are involved in the movements.

Figure 6 illustrates this for three examples. Let's assume a task descriptor that relates the transformation of linked object L_1 to the transformation of linked object L_2. The target values are determined to put L_1 on top of L_2. In example a), L_1 is connected to the left hand, while L_2 has a fixed transformation in world coordinates. The system will generate a trajectory moving the grasped L_1 on L_2 with its left arm. In case b), both L_1 and L_2 have been grasped. The result is a coordinated bi-manual movement, L_1 is put on L_2 which is held with the right hand. In case c), L_2 has again a fixed transformation in world coordinates, and L_1 has been grasped with both hands. In this case, the system will put L_1 on top of the static object L_2, but this time generating a coordinated bi-manual trajectory with the grasped object L_1.

The examples illustrate how to generalize movement represented in relative object coordinates to different body schemata. Casting the overall movement into an optimization problem such as for instance presented in [17] additionally adds the capability to adapt the resulting trajectories, for instance to avoid collisions or other limits.

VI. EXPERIMENTS

We conducted a set of experiments to validate the proposed concepts. The setup is depicted in Figure 7. A tutor is sitting at a table and demonstrates a task to a humanoid robot several times. The robot perceives the scene with its on-board cameras and determines the object's transformation based on a color and depth cue. Object rotations are currently extracted in the camera plane only. The tutor's pose is estimated by projecting the 3d position of the detected skin color blobs to the head and hands of a kinematic tutor model.

The tutor indicates the interesting objects with the attention mechanism by pointing to them. This results in associating the most salient object with L_1, and the second most salient object with L_2. After this, the tutor demonstrates the task to the robot. In this phase, the robot will cut the observations into segments based on the coherence of the hand and object movement, and store the segmented object trajectories in the *Observation Memory*. Now the tutor can

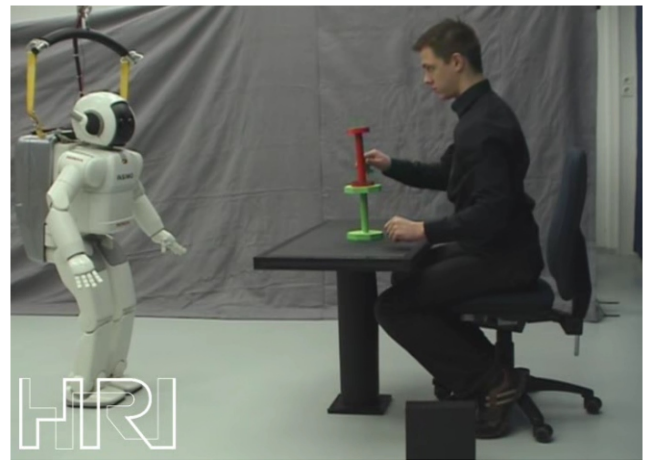

Fig. 7. Experimental setup

instruct the robot to reproduce the movement with a gesture. If no movement primitive has yet been learnt from the observations, this will be done first. Otherwise, the system will start the action sequence associated with the tutor's gesture. We prepared a set of action sequences that allow the robot to imitate the learnt movement in different ways, for instance performing the task with the left hand or with both hands. The sequences comprise some preparatory movements as well as the learnt ones. The preparatory movements have been designed to generalize for different situations. They are hierarchically organized in form of a state chart so that they can easily be reused in other situations. Figure 8 shows the execution flow. Throughout the sequence, the robot will gaze at the most salient object. Currently, we freeze the object's location once the robot starts to walk to the table. This is due to the limited field of view of the used cameras, in future it is planned to update the object's location steadily to account for more dynamic scenarios. Further, we pre-define the location where the object is to be grasped, and the set of task descriptors to be used. In this scenario, we grasp symmetrical objects and therefore select task descriptors that relate the hand position and polar angles to the objects transformation.

A. *Stacking with one hand*

Experiment a) in Figure 8 shows the robot performing a stacking task using its left hand. It will first approach the most salient object so that it can conveniently be grasped. Then, the hand will move towards a pregrasp pose. In this phase, the relative position of hand and object is controlled in hand coordinates, and the hand inclination is aligned with the object's handle. This task description is invariant against the object's position and orientation (see Figure 3). Once the hand reached the target coordinates, the fingers will be closed to a power grasp. At this point, we modify the systems body schema and attach the grasped object L_1 to the hand.

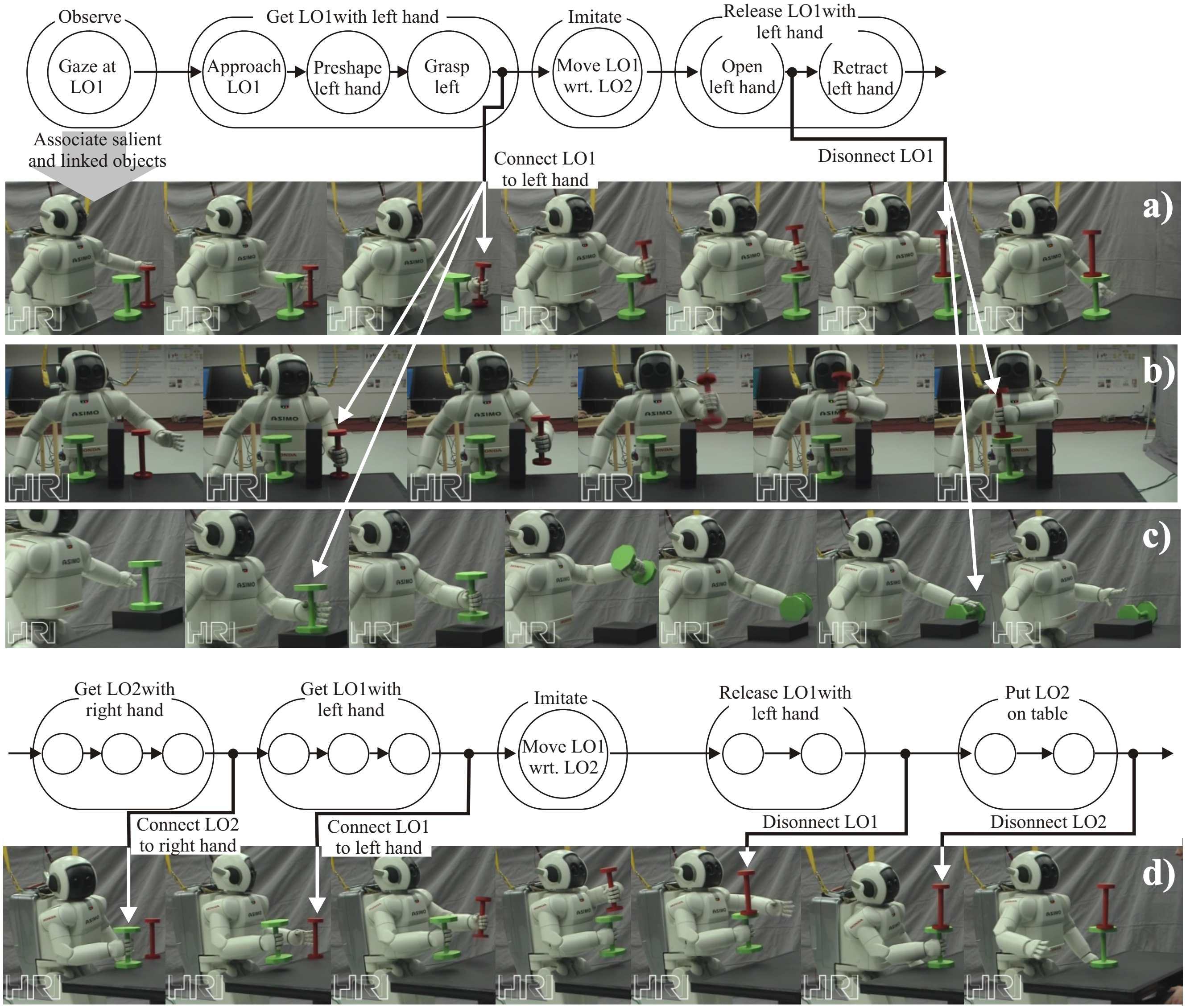

Fig. 8. Experiments

We currently rely on the precision of the system and don't incorporate additional tactile or visual information. Once the preparatory movement has been carried out, the system will imitate what it has learnt. The chosen task descriptor is the movement of L_1 with respect to L_2, which will result in the robot moving L_1 on top L_2. Once the movement is finished, the robot releases the object, retracts the hand relative to the object and walks back.

B. Stacking with obstacle avoidance

In the second experiment (Figure 8 b) we apply the same learnt movement as in a), but put an obstacle between the objects. The proximities between the obstacle and the linked objects are formulated as an optimization criterion, so that the trajectory will be adapted to avoid these collisions. The image sequence illustrates how the system modifies the movement to account for the new situation, but still preserves the important characteristics.

C. Rotating an object

The same sequence is also applied in the third experiment (Figure 8 c), but this time using another learnt movement that rotates L_1 (this time linked to the green object) and puts it next to L_2 (the box). This experiment shows the invariance of the task description used in the preparatory movements. We can apply the same task descriptors to retract the hand, even though the object is rotated after performing the task.

D. Bi-manual stacking

In the last experiment (Figure 8 d), we reuse some actions and add a preparatory movement to get L_2 with the right hand, and to put it on the table. After grasping L_2, we connect it with the right hand, and after putting it on the table, we disconnect it. These modifications allow us to perform the same learnt skill with a coordinated bi-manual movement as depicted in the last image sequence.

VII. CONCLUSION

We presented an architecture for interactive movement learning and generation of robots with a human tutor. We particularly focused on the topic of generalization and invariance of the underlying movement representation. Learnt movement is represented by the relation of object trajectories, and incorporates statistical information of the demonstrations. The robot's embodiment is not part of the representation. The association of the movement representation with a concrete situation is created when the movement is reproduced. This is achieved in interaction with a human. The major novelties of the contribution can be summarized as follows:

- We achieve flexibility by providing the system the capabilities to learn new movement primitives in interaction. Newly learnt and pre-defined primitives can be combined in a consistent scheme.
- We achieve invariance by describing the task in coordinates frames that generalize to different situations, such as robot-object or object-object relations.
- We employ an attention mechanism to associate salient objects with the movement representation. This allows the tutor to instruct the system to reproduce a movement with a variety of different objects.
- We introduced a method to dynamically modify the systems body schema. This allows to carry out learnt skills in different ways, for instance with different effectors, or even bi-manually.
- We achieve robustness by applying prediction and optimization methods, allowing the system to adapt its movement according to the current situation.

We conducted a set of experiments in an interactive imitation learning scenario with a humanoid robot to verify the proposed concepts.

The proposed methods assume that movement can exclusively be represented by the object trajectories, which is feasible for a certain class of problems only. Further, it is assumed that it is known how to modify the body schema of the robot when grasping an object. In more complex scenarios, this is not trivial. Future work will focus on these issues.

ACKNOWLEDGEMENT

The authors thank the members of Honda's robotics research and development teams in Japan for their support.

REFERENCES

[1] P. McGuire, J. Fritsch, H. Ritter, J. J. Steil, F. Roethling, G. A. Fink, S. Wachsmuth, and G. Sagerer, "Multi-modal human-machine communication for instructing robot grasping tasks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS)*, 2002, pp. 1082–1089.

[2] C. Burghart, R. Mikut, R. Stiefelhagen, T. Asfour, H. Holzapfel, P. Steinhaus, and R. Dillmann, "A cognitive architecture for a humanoid robot: a first approach," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, December 2005, pp. 357–362.

[3] J. Bohg, C. Barck-Holst, K. Huebner, M. Ralph, B. Rasolzadeh, D. Song, and D. Kragic, "Towards grasp-oriented visual perception for humanoid robots," *International Journal of Humanoid Robotics*, vol. 6, no. 3, pp. 387–434, 2009.

[4] N. Vahrenkamp, A. Barski, T. Asfour, and R. Dillmann, "Planning and execution of grasping motions on a humanoid robot," in *IEEE-RAS International Conference on Humanoid Robots*, 2009.

[5] S. Calinon, F. Guenter, and A. Billard, "On Learning, Representing and Generalizing a Task in a Humanoid Robot," *IEEE Transactions on Systems, Man and Cybernetics, Part B.*, vol. 37, no. 2, pp. 286–298, 2007.

[6] J. Peters, "Machine learning for motor skills in robotics," *Künstliche Intelligenz*, vol. 2008, no. 4, pp. 41–43, 11 2008.

[7] M. Lopes and J. Santos-Victor, "Visual learning by imitation with motor representations," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 35, no. 3, pp. 438–449, 2005.

[8] P. Azad, T. Asfour, and R. Dillmann, "Toward an unified representation for imitation of human motion on humanoids," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, Rome, Italy, April 2007.

[9] C. Eppner, J. Sturm, M. Bennewitz, C. Stachniss, and W. Burgard, "Imitation learning with generalized task descriptions," in *Proceedings of the IEEE/RSJ International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[10] M. Hersch, E. Sauser, and A. Billard, "Online learning of the body schema," *International Journal of Humanoid Robotics*, vol. 5, no. 2, pp. 161–181, 2008.

[11] A. Stoytchev, "Computational model for an extendable robot body schema," Georgia Institute of Technology, College of Computing, Tech. Rep. GIT-CC-03-44, October 2003.

[12] C. Nabeshima, Y. Kuniyoshi, and M. Lungarella, "Adaptive body schema for robotic tool-use," *Advanced Robotics*, vol. 20, no. 10, pp. 1105–1126, 2006.

[13] C. A. Acosta-Calderon and H. Hu, "Robot imitation: Body schema and body percept," *Journal of Applied Bionics and Biomechanics*, vol. 2, no. 3-4, 2004.

[14] M. Mühlig, M. Gienger, S. Hellbach, J. Steil, and C. Goerick, "Task-level imitation learning using variance-based movement optimization," in *International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, 2009.

[15] M. Mühlig, M. Gienger, J. J. Steil, and C. Goerick, "Automatic selection of task spaces for imitation learning," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA, 2009.

[16] M. Gienger, H. Janssen, and C. Goerick, "Task-oriented whole body motion for humanoid robots," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, Los Angeles, USA, December 2005.

[17] M. Toussaint, M. Gienger, and C. Goerick, "Optimization of sequential attractor-based movement for compact movement representation," in *Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots*, Pittsburgh, USA, December 2007.

[18] M. Mühlig, M. Gienger, and J. J. Steil, "Human-robot interaction for learning and adaptation of object movements," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (submitted)*, Taipeh, Taiwan, 2010.

[19] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, "Real-time collision avoidance with whole body motion control for humanoid robots," in *IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007.

[20] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-7 no. 12, December 1977.

[21] M. Hirose, Y. Haikawa, T. Takenaka, and K. Hirai, "Development of humanoid robot Asimo," in *IEEE/RSJ International Conference on Intelligent Robots and Systems – Workshop 2*, 2001.

[22] T. Flash and T. J. Sejnowski, "Computational approaches to motor control." *Current Opinion in Neurobiology*, vol. 11, 2001.

[23] C. L. Colby, "Action-oriented spatial reference frames in cortex." *Neuron*, vol. 20, no. 1, pp. 15–24, January 1998. [Online]. Available: http://view.ncbi.nlm.nih.gov/pubmed/9459438

[24] S. Chiaverini, "Singularity-robust task priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, June 1997.