

Imitation Learning with Generalized Task Descriptions

Clemens Eppner

Jürgen Sturm

Maren Bennewitz

Cyrill Stachniss

Wolfram Burgard

Abstract—In this paper, we present an approach that allows a robot to observe, generalize, and reproduce tasks observed from multiple demonstrations. Motion capture data is recorded in which a human instructor manipulates a set of objects. In our approach, we learn relations between body parts of the demonstrator and objects in the scene. These relations result in a generalized task description. The problem of learning and reproducing human actions is formulated using a dynamic Bayesian network (DBN). The posteriors corresponding to the nodes of the DBN are estimated by observing objects in the scene and body parts of the demonstrator. To reproduce a task, we seek for the maximum-likelihood action sequence according to the DBN. We additionally show how further constraints can be incorporated online, for example, to robustly deal with unforeseen obstacles. Experiments carried out with a real 6-DoF robotic manipulator as well as in simulation show that our approach enables a robot to reproduce a task carried out by a human demonstrator. Our approach yields a high degree of generalization illustrated by performing a pick-and-place and a whiteboard cleaning task.

I. INTRODUCTION

Several techniques exist for transferring new skills to robots. A very promising technique is called “imitation learning”: Here, a robotic system observes an instructor while performing a task [4], [2]. From multiple demonstrations, the robot then needs to infer a generalized task description and reproduce it accordingly even under slightly modified conditions. As an example, consider Fig. 1. The task of cleaning a whiteboard is demonstrated by a human and transferred to a simulated as well as to a real robotic manipulator.

Teaching skills by direct demonstration is a very natural way of skill transfer in humans and animals. In the aim to create more versatile, adaptable, and sociable robotic platforms, research on the mechanisms of learning new behaviors by observation has a very high potential. Furthermore, such social learning can speed up learning complex behaviors enormously, as it provides strong prior information for the learning process, thereby scaling back the learning task for the robot. This can reduce the search space for traditional learning algorithms significantly, such that previously intractable tasks can be learned.

In this paper, we show that imitation learning is well suited as a user-friendly instruction method for manipulation tasks. Our approach uses motion capture data generated by a vision system to track body parts of the instructor

This work was partly supported by the DFG under contract number SFB/TR-8 as well as by the EC under contract number FP6-IST-045388-INDIGO

All authors are with the Computer Science Department of the University of Freiburg, Germany
{eppner, sturm, maren, stachniss, burgard}@informatik.uni-freiburg.de

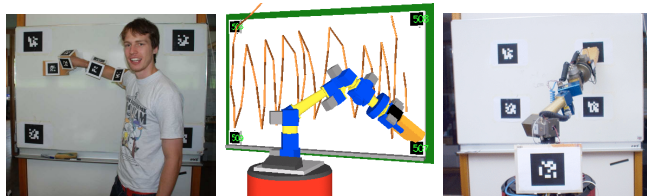


Fig. 1. Learning the whiteboard cleaning task. Left: The human instructor demonstrates how to clean the whiteboard using visual motion capturing. Middle: After the first demonstration, the robot can replay the recorded trajectory. Right: After several demonstrations, the robot can generalize the task and reproduce it under changed conditions, for example, on a whiteboard with different size and position.

and the 3D positions of relevant objects in the scene. The body configuration as well as the relations between objects and body parts of the demonstrator are in turn modeled as normally distributed observation nodes of a dynamic Bayesian network (DBN). For reproducing an observed skill, the network is evaluated at each time step in order to infer the most-likely action. Within our framework, new constraints can be dynamically added to the network, e.g., to incorporate collision avoidance during reproduction in order to deal with unforeseen obstacles.

Our relation-based approach extends the recent work of Calinon and Billard [6] by formalizing the problem by means of a DBN. We furthermore allow for incorporating additional constraints for modeling unexpected obstacles that should be considered during imitation.

The remainder of the paper is organized as follows: We briefly review related work on imitation learning in Sec. II. Our framework on imitation learning via inference in a DBN is introduced in Sec. III, followed by Sec. IV on learning the parameters of the DBN from motion capture data and Sec. V on reproducing the generalized skill. Experimental results obtained with a real 6-DoF robotic manipulator as well as in simulation are presented in Sec. VI.

II. RELATED WORK

In the past, various techniques have been used for transferring task knowledge to a robotic system. Initially, the required motion trajectories were hand-coded by an engineer [10], [19]. However, the more complex the task description becomes, the more difficult it is to create and maintain large controllers. Alternatively, the required joint angle trajectories of the robot can be shown by a single demonstration, for example, by a human teacher using a joystick, a motion capturing system, or kinesthetic training. The resulting sequence is recorded and can then be replayed by the robot. However, if the observations are noisy or

unpredicted disturbances in the task environment occur, simple playback of the recorded motion is in general not sufficient to reliably reproduce a given task. To deal with noise in the observations and to generalize over multiple demonstrations of the same tasks, several authors suggested hidden Markov models (HMM) to encode and reproduce a demonstrated action, e.g., [1], [7], [20].

Reinforcement learning techniques have been successfully applied to learn controllers for an individual skill or for motor primitives [12], [3], [17]. As the size of the search space grows exponentially with the dimensionality of the learning problem, Ijspeert and Schaal [13], [18] proposed to learn parameterized controllers instead that are based on differential equations.

Pardowitz and Dillmann presented a system that generalizes over household tasks in a hierarchical manner [16]. Actions performed by the human demonstrator are recognized as a sequence of “elementary operators”, of which a graph-based task representation is learned. In this approach, the incrementally updated network topology reflects the learned temporal ordering of the individual actions.

Although symbolic representations are well suited for planning and reasoning, their limitation to higher-level skills renders them inapplicable in domains where a continuous motor control is required. By contrast, trajectory learning directly starts by encoding each demonstration by a sequence of continuous observations. Due to the massive amounts of captured data, dimensionality reduction techniques are often applied. Chalodhorn *et al.* [8] used principal component analysis (PCA) to reduce the high-dimensional motion capture data of a recorded human walk. While a direct playback of the human data on a humanoid robot would make the robot fall, the authors showed that after a few trials the robot was able to modify the imitated gait incrementally. There, a sensory-motor predictor was learned and used to produce dynamically stable actions. Similarly, Grimes *et al.* [11] also used PCA to reduce the high-dimensional configuration space and applied a DBN to infer dynamically stable imitative actions using constraint variables and a learned forward model of the robot dynamics.

In our approach, we also use a DBN to learn and reproduce tasks. By means of the DBN, relations between objects in the environments and body parts are learned and considered during reproduction. We show that by using this framework, it becomes also possible to incorporate additional constraints (such as collision avoidance) during skill reproduction.

We use the idea presented by Calinon and Billard [6] to describe actions using relations between objects. In contrast to their approach which is based on Gaussian mixture models, we use a kernel estimator to model the relations. This way, we can deal with a small number of demonstrations and avoid fitting Gaussian mixtures.

III. IMITATION LEARNING FRAMEWORK

As most techniques for generating imitative actions, our approach uses two steps. First, a set of repeatedly carried-out

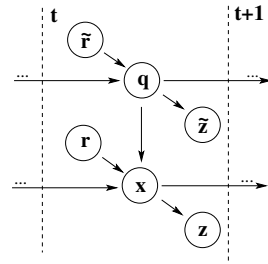


Fig. 2. Dynamic Bayesian network (DBN) illustrating the conditional independence assumptions used for learning and reproducing tasks. The arrows indicate conditional dependencies between variables. Here, \tilde{z} denotes the observation of the demonstrators body configuration which is represented by q (joint angles). z is the observation of the world state x that encodes the position of relevant object in the scene. The object-manipulator relations r as well as the constraints on joint angles \tilde{r} are used to model the action that should be learned and carried out by the robot. During learning, the distributions over the relations $p(\tilde{r})$ and $p(r)$ is determined. During reproduction, these relations are known and the most likely body configuration of the robot is to be estimated and executed by the robot.

actions of a human demonstrator is observed by the robot. The robot has to infer the relevant parts of the demonstrated task and to build an internal representation. This is done in the so-called learning step. Second, the robot must be able to reproduce an action to actually imitate the demonstrator. This step is called imitation or reproduction step.

From a more formal point of view, we treat the problems as a stochastic process that can be described by means of the dynamic Bayesian network depicted in Fig. 2. The DBN is an intuitive graphical description of the conditional independence assumptions made in the model. While the teacher demonstrates the task several times during the first phase of our imitation learning approach, we learn relevant relations, also called constraints, between n objects in the world and the manipulator as well as the joint angle constraints for each point in time. Thus, the constraints encode the actions necessary to carry out the task which is being demonstrated. The corresponding latent variables are denoted $r(t) \in \mathbb{R}^{3n}$ – the constraints between objects in world space, encoded as the 3D displacement from end-effector to each object – and $\tilde{r}(t) \in \mathbb{R}^m$ the constraints in configuration space – i.e., the joint angles of the m -DoF manipulator – respectively.

The relations $\tilde{r}(t)$ regarding the joint angles are only meaningful if the demonstrator and the imitator have a similar body structure. They are important in case actions should be imitated as precise as possible.

In the second phase, we use the learned relations to infer actions of the robot. At this time, the relation variables are observable variables in the DBN and generate the actions of the robot given the estimated world state.

For reasons of simplicity, let us consider first the DBN for one time step t (neglecting the index t). Let $q \in \mathbb{R}^m$ refer to the configuration of the m -DoF arm of the demonstrator (during learning) or the robot (during reproduction). Let $x \in \mathbb{R}^{3n+3}$ be the vector of the 3D positions of relevant objects in the scene

$$x = \{x_E, x_1, \dots, x_n\}, \quad (1)$$

where x_1, \dots, x_n are the positions of the n objects in

the scene. In the remainder of the paper, we use a robot manipulator for imitating humanoid arm movements and x_E is the position of the end effector. Note that any set of body parts can be used in case full body actions should be imitated without changing the math except for adding additional variables and indices (x_E would become x_{E_1}, \dots, x_{E_M}).

The observation of 3D poses x is called $z \in \mathbb{R}^{3n+3}$, and the observations of joint configurations $q \in \mathbb{R}^m$ will be referred to as $\tilde{z} \in \mathbb{R}^m$.

Our DBN depicted in Fig. 2 implies the following independence assumptions:

$$p(\tilde{r}, q, \tilde{z}, r, x, z) = \frac{p(z | x) \cdot p(\tilde{z} | q) \cdot p(\tilde{r}) \cdot p(r)}{p(x | q, r) \cdot p(q | \tilde{r})} \quad (2)$$

In our model, we assume the following distributions in the nodes of the DBN: The observation models $p(z | x)$ and $p(\tilde{z} | q)$ are assumed to be Gaussian distributions and so are the distributions over the relations $p(r)$ and $p(\tilde{r})$. Since we have no information about the distributions over relations in the beginning, we set their variance to infinity.

The posterior about the objects in the scene can be written as follows:

$$p(x | q, r) = p(x_E | q) \cdot p(x_1, \dots, x_n | x_E, r) \quad (3)$$

Eq. (3) is obtained by applying the product rule and by assuming that the poses of objects are independent from the joint configuration given the position of the end effector. By further applying the product rule, we obtain:

$$p(x | q, r) = p(x_E | q) \cdot p(x_1 | x_E, r) \cdot p(x_2, \dots, x_n | x_E, r) \quad (4)$$

$$= p(x_E | q) \cdot \prod_{i=1}^n p(x_i | x_E, r_i) \quad (5)$$

$$\approx p(x_E | q) \cdot \prod_{i=1}^n \mathcal{N}_{r_i}(x_i - x_E; \Sigma_i) \quad (6)$$

Eq. (4) is obtained by assuming that given the relations r as well as x_E , the positions of two objects in the scene are independent. By applying this assumption n times, we obtain Eq. (5).

We additionally assume that also the posterior about the pose of the end effector $p(x_E | q)$ is Gaussian, which is a reasonable assumption for the robots equipped with accurate joints (such as Schunk modules in our case). $p(x_E | q)$ then corresponds to the kinematic function.

Finally, we make the assumption that the differences between actions carried out during the individual demonstrations can be described by Gaussian distributions. Thus, the individual $r_i \in r$ are represented by a mean and a variance for the three dimensions. This leads to Eq. (6). Similarly, the individual joint constraints $r_j \in \tilde{r}$ are represented by Gaussians. Thus, $p(q | \tilde{r})$ can also be computed by a product as in Eq. (6).

IV. LEARNING PHASE

In the first phase, the robot observes a person that repeatedly carries out the task the robot has to perform. Given the DBN structure explained above, the key challenge of this learning phase is to learn the object-manipulator relations (r) and – if needed – the joint constraints (\tilde{r}).

A. Motion Capturing and Object Pose Estimation

To estimate the motion trajectories of the human demonstrator while executing a task and the 3D position of relevant objects in the scene, we use passive markers and images of a monocular camera. In particular, we use the ARToolKit [14], which is a software library providing the means to extract the 6D pose (orientations and position) of fiducial markers (black squares on a white background) from single camera images.

We attach compounds of 4 markers around the teacher’s arm (see left image of Fig. 1) to bypass the problem of occlusions. In most cases, not more than one marker of the same compound is visible. To deal with the case that two markers are visible simultaneously (sensing more markers at the same time is impossible due to their mutual orthogonality within one compound), we perform a linear interpolation between their poses depending on the degree of visibility of the markers. Finally, we apply a Kalman filter to track the 3D marker position estimates over time. To derive the demonstrator’s joint angles from marker poses, we use an anthropomorphic arm model and apply straightforward geometric operations. As a result, we are able to reliably estimate the probability distributions over q and x during the demonstrations.

B. Multiple Demonstrations

Our approach relies on demonstrating the same task multiple times in order to achieve a good generalization. One problem when generalizing task descriptions from multiple demonstrations is the fact that the repeatedly observed actions are not time-synchronized, even though the different demonstrations typically do not vary largely. To deal with varying movement velocity profiles, we apply derivative dynamic time warping (DDTW) [15] which is able to account for local distortions in the time domain by computing a nonlinear transformation of the time axis of the individual demonstrations. Based on the aligned demonstrations, we can derive the relations r and \tilde{r} which encode the action to imitate.

C. Deriving Relations for Generalized Task Descriptions

By assuming that all relations in r and \tilde{r} are (individual) Gaussians, we can directly infer a mean and a variance estimate for the individual relations for each point in time given the estimates for x and q . Especially the variance is a key element for the tasks descriptions since it describes how accurately the demonstrator enforced this relation during the demonstrations.

Formally, a relation r_i is fully described by a 3D mean μ_i and 3D variance σ_i^2 vector (assuming the relations in the three dimensions to be independent).

In theory, we could compute μ_i and σ_i^2 directly from the estimates for x and q during learning. In practice, however, we typically have to deal with a rather small number of demonstrations and therefore rather rough and non-smooth estimates are obtained if the values are computed directly. To overcome this problem, we apply a Parzen window kernel estimator for computing smooth function approximations.

This is a non-parametric technique that allows for estimating a function μ based on a set of sample points. To compute estimates of the relations between objects in the scene r_i and the end effector, we compute relation samples consisting of time and position $\{t, l(t, d, i)\}$ with $l(t, d, i) = x_i^d(t) - x_E^d(t)$, $t = 1, \dots, T$ and $d = 1, \dots, D$ where D are the number of demonstrations. Then, we obtain

$$\mu_i(t') = \frac{\sum_{d=1}^D \sum_{i=1}^T K\left(\frac{t'-t}{h}\right) l(t, d, i)}{\sum_{d=1}^D \sum_{i=1}^T K\left(\frac{t'-t}{h}\right)}, \quad (7)$$

where h is the Parzen window size (empirically determined, $h = 0.2$ s) and K is a kernel function. We use the standard choice for the K , namely the squared exponential kernel

$$K(x) = \exp\left(-\frac{1}{2}\|x\|^2\right). \quad (8)$$

The variance given the sample points can be estimated similarly as

$$\sigma_i^2(t') = \frac{\sum_{d=1}^D \sum_{i=1}^T K\left(\frac{t'-t}{h}\right) (l(t, d, i) - \mu(t))^2}{\sum_{d=1}^D \sum_{i=1}^T K\left(\frac{t'-t}{h}\right)}. \quad (9)$$

This procedure is carried for each object in the scene and accordingly done for the joint relations \tilde{r} .

V. REPRODUCTION PHASE

The goal of the reproduction or imitation phase is to carry out the demonstrated task to achieve the same result. Given the DBN, we can seek for the configuration of joints q^* that maximizes the likelihood given the demonstrations.

A. Incremental Optimization

If we consider only one time step during the optimization, we seek for the configuration q^* that maximizes the joint probability. During reproduction, r and \tilde{r} are known. Furthermore, the robot can control its body/manipulator by specifying a joint configuration and does not have to rely on noisy marker observations of its body. Therefore, the maximization turns into

$$q^* = \underset{q}{\operatorname{argmax}} p(q | \tilde{r}) p(x | r, q) p(z | x). \quad (10)$$

As discussed before in Sec. III, the posteriors $p(q | \tilde{r})$, $p(x | r, q)$, and $p(z | x)$ are basically products of Gaussians and lead to a Gaussian distribution again. Thus, to maximize the joint probability, we need to determine the mean of that Gaussian. To do so, we proceed as follows. Consider that we are currently at time step t and want to seek for the joint configuration that maximizes Eq. (10) at $t + 1$. Each relation between x_i and x_E generates a relative displacement vector Δ_i :

$$\Delta_i(t+1) = x_i(t) - x_E(t) - \mu_i(t+1) \quad (11)$$

Since we want to compute a new joint configuration for the robot, we need to convert the constraints expressed in

world coordinates in joint space. We achieve this by applying a variant of the damped-least squares method described by Buss [5]. This approximative technique performs a linearization of the kinematic function. According to this method, a desired movement in world coordinates (Δ) is transformed to an executable movement in joint space ($\tilde{\Delta}$) by

$$\tilde{\Delta}_i(t+1) = J(JJ^T + \lambda^2 I)^{-1} \Delta_i(t+1) \quad (12)$$

$$\tilde{\Sigma}_i(t+1) = \left(J(JJ^T + \lambda^2 I)^{-1} \right) \Sigma_i(t+1) \left(J(JJ^T + \lambda^2 I)^{-1} \right)^T \quad (13)$$

where λ is the so-called damping factor and J refers to the Jacobian. $\Sigma_i(t+1)$ is a 3×3 matrix encoding the variances from the relation r_i with the corresponding variances for dimension x , y , and z on the diagonal. Due to the linear mapping, we obtain also a Gaussian in configuration space.

The constraints defined by $\tilde{r}(t+1)$ can be easily used to compute a desired movement

$$\tilde{\Delta}_{\tilde{r}}(t+1) = \tilde{r}(t+1) - q_t \quad (14)$$

while the variances $\tilde{\Sigma}(t+1)$ do not need to be transformed.

All constraints resulting from the observation of the demonstrator's joint angle configurations or from the arrangement of objects in the scene are expressed in terms of updates in joint space. This allows us to combine them by multiplying the normal distributions as it has also been done by Calinon and Billard [6]. The resulting distribution is the product over the $N + 1$ Gaussians resulting from the N task space relations plus the joint space relations. We can use it to directly obtain the next configuration $q^*(t+1)$ according to Eq. (10) by selecting the mean from this combined Gaussian, as given by

$$\hat{q}(t+1) = q_t + \tilde{\Sigma}(t+1) \left((\tilde{\Sigma}_{\tilde{r}}(t+1))^{-1} \tilde{\Delta}_{\tilde{r}}(t+1) + \sum_{i=1}^n (\tilde{\Sigma}_i(t+1))^{-1} \tilde{\Delta}_i(t+1) \right), \quad (15)$$

with

$$\hat{\Sigma}(t+1) = \left(\sum_{i=1}^n (\tilde{\Sigma}_i(t+1))^{-1} + (\tilde{\Sigma}_{\tilde{r}}(t+1))^{-1} \right)^{-1} \quad (16)$$

The mean of that distribution defines the configuration of the robot at the next time step that maximizes the probability distribution specified in Eq. (10).

B. Local Optimization with Obstacles

The technique described in the previous section can directly be applied to deal with unforeseen obstacles in the scene during reproduction. Consider that the robot observes an obstacle during the imitation that was not there during the demonstrations. To avoid this obstacle during the reproduction task, we can add additional constraints between the observed obstacle and the closest point on the robot's body, as used in approaches based on potential fields for collision avoidance.

Without changing the framework described above, the robot can reactively introduce constraints for avoiding obstacles while carrying out its task as close a possible to the human demonstrator. Let x_O be the position of the obstacle. Instead of adding a repellent force, we add an attractor at the opposite side of the end-effector,

$$\Delta_O = -\alpha \frac{x_E - x_O}{\|x_E - x_O\|} \quad (17)$$

$$\Sigma_O = \beta \cdot \exp\left(\|x_E - x_O\|^2\right) \cdot I, \quad (18)$$

where α determines the desired distance to the obstacle and β gives the relative importance with respect to the other constraints.

It should be noted that this technique works well for small or rather simple structured obstacles added to the scene. In case complex or, for example, U-shaped obstacles are found in the environment, this approach is likely to suffer from local minima caused by contradictory constraints.

C. Global Optimization

The problem of local minima, however, can be avoided by not incrementally optimizing the joint probability distribution of the DBN for the upcoming time step but optimizing it over all time steps $1 \dots T$ of the task sequence at once:

$$q_{1:T}^* = \underset{q_{1:T}}{\operatorname{argmax}} p(q_{1:T}, x_{1:T}, \tilde{r}_{1:T}, r_{1:T}, z_{1:t}) \quad (19)$$

Note that at a particular time step t , only the first $1 \dots t$ observations $z_{1:t}$ are already available and can be included for planning. Doing this optimization on a global level, however, comes with significantly increased computational cost due to the high dimensionality of $q_{1:T}^*$.

One way of rather efficiently estimating $q_{1:T}^*$ is to make use of probabilistic roadmaps or rapidly-exploring random trees (RRTs) [9], and find the shortest path using A^* on the sampled set of nodes since we are only interested in the most-likely imitation sequence. Given that we properly encode the likelihoods of all constraints in the cost function later used by A^* , the solution of the planner will approximate Eq. (19) well.

We propose to base the cost function on the Mahalanobis distance to the combined Gaussian $\mathcal{N}(\hat{q}(t); \hat{\Sigma}(t+1))$ computed in (15) and (16), as this Gaussian already incorporates all constraints r , \tilde{r} and the obstacle constraints in a time-dependent way. For a configuration q at time t , we define the cost as the likelihood with respect to the previously computed combined Gaussian, i.e.,

$$\operatorname{cost}(q, t) = (q - \hat{q}(t))^T \hat{\Sigma}(t+1)^{-1} (q - \hat{q}(t)). \quad (20)$$

Then, finding the cost-optimal sequence of configurations $q_{1:T}^*$ is equivalent to maximizing the likelihood of the trajectory $q_{1:T}^*$ in (19).

VI. EXPERIMENTS

We carried out a set of experiments to analyze our approach. We always observed a human demonstrator equipped with markers of the ARToolKit, see Fig. 3. We used this



Fig. 3. Photos of the demonstrated pick-and-place task. The trajectory is recorded both, in task and joint space.

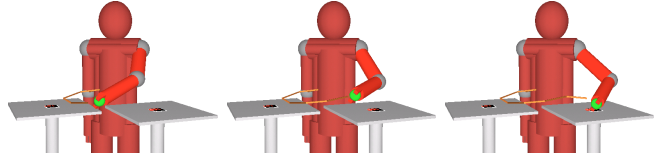


Fig. 4. Reproduction of the pick-and-place task by a human-like manipulator using both, task and joint space constraints.

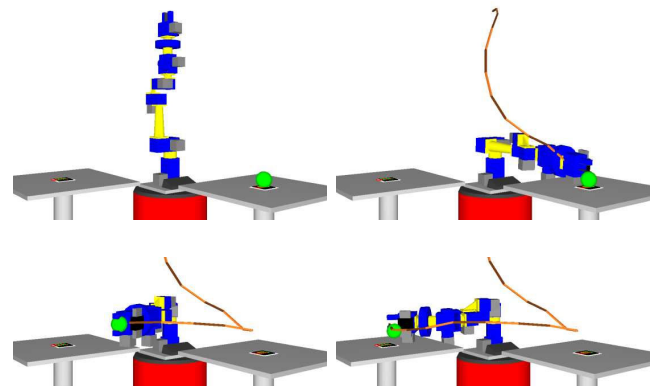


Fig. 5. Reproduction of the same pick-and-place task by our 6-DoF manipulator. Note that the robot successfully generalized the task, as the target and source location have been swapped.

data to learn the relations and thus constraints for the task reproduction online. The motions were sampled at a rate of 5 Hz. We segmented the training trajectories manually.

A. Imitating Human Actions

To imitate the observed behavior, we reproduced the tasks using a real robot equipped with a manipulator and two simulated robots, one with a manipulator and one with a human-like arm.

Fig. 4 shows the reproduction of the pick-and-place task after being demonstrated four times (see Fig. 3). The human-like simulated robot considers both, the joint and the task constraints, which leads to the fluent, human-like movement.

In Fig. 5, the same task is reproduced by our robotic manipulator. As the demonstrator and the imitator have a significantly different body scheme, the joint constraints \tilde{r} have been disabled. As can be seen, the robot is able to reproduce the task even though the setup between demonstration and imitation has been changed by setting a different target location.

We furthermore analyzed the number of demonstrations needed until a task could be reproduced reliable. For this analysis, a teacher picked up a cup and placed it at a distance of 1 m. As can be seen in Fig. 6, our approach converged after 4 iterations.

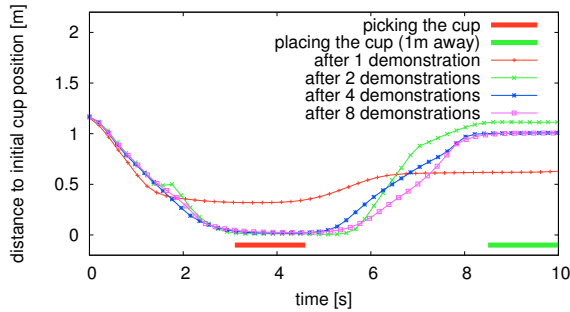


Fig. 6. Illustration how the approach converges to an action after different number of demonstrations. The demonstrated task was to pick up a cup and place it 1 m away. As can be seen, after 4 demonstrations, our approach converged and the robot was able to pick and place the cup.

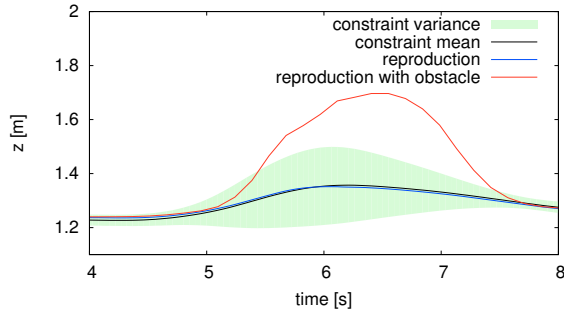


Fig. 7. Illustration of a constraint including the variance as well as two trajectories of reproductions – one for the obstacle free case and one in case an obstacle blocks the trajectory.

B. Dealing with Obstacles during Imitation

The additional obstacle constraints described in Sec. V-B allow the robot to deal with unforeseen obstacles during the task execution. The obstacle constraints act similar to a potential field pushing the robot away from obstacles, which are labeled by predefined markers.

Fig. 7 illustrates an example for a constraint in a pick and place task. The figure shows a reproduced trajectory for the obstacle free case and a trajectory that was selected in the presence of an obstacle. As can be seen, the robot moved its arm over the obstacle in order to avoid a collision.

We carried out a whiteboard cleaning task that nicely illustrates the properties of the presented methods. First, a human repeatedly cleaned a whiteboard in an area bounded by 4 markers with the same number of ups and downs (see left image of Fig. 1). Then, we attached a sponge to the robot and let it perform the demonstrated task. In the first experiment, we modified the size of the area to clean for illustrating the capabilities of generalization. Photos from this experiment can be seen in Fig. 8. Note that in case the area to clean is much larger than during learning, the whiteboard may not be cleaned well. The reason for that is that our approach then scales the trajectory to reproduce and thus there might be parts that will not be covered by the sponge.

In the second experiment, we showed the robot during reproduction phase an obstacle marker that was not there

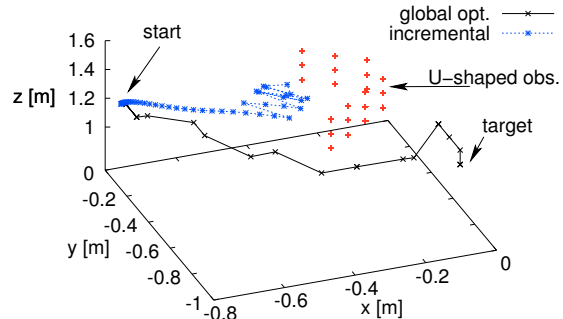


Fig. 10. The plot shows the end effector position of the robot over time during two experiments. When applying the incremental method, the end effector gets stuck in a U-shaped obstacle while the global method solves the task and the end effector reaches the target location.

during the learning phase, see first image of Fig. 9. Then, the robot had to clean the whiteboard while avoiding obstacles (and thus not cleaning the area of the marker). For reasons of illustration, we removed the marker during the experiment but kept it in the internal memory of the robot. In this way, the reader can see that the robot did not clean the corresponding area. Eight photos were taken during the reproduction and are depicted in Fig. 9. To avoid the area marked as an obstacle area, the robot lifts the sponge away from the whiteboard (in the direction of the observing camera).

C. Imitation by Planning

The experiments presented above used the reproduction by means of incrementally computing the maximum-likelihood configuration of the DBN. This can be done efficiently online, but the approach can suffer from local minima, for example, in the presence of U-shaped obstacles. Such an example is presented in Fig. 10 where the robot gets stuck.

If one applies the global optimization technique described in Sec. V-C, one can overcome this problem since the optimal solution over all time steps is computed. Thus, the robot is able to reproduce the task including the avoidance of the U-shaped obstacle. This global method, however, comes with a significantly increased computational load.

VII. CONCLUSION

In this paper, we presented an approach to imitation learning that extends a recently published method of Calinon and Billard. It enables a robot to observe, generalize, and reproduce tasks from observing a human demonstrator. We formalized the problem using a dynamic Bayesian network that is used for learning relations between the observed positions of the objects and body parts of the instructor. Additional constraints, for example, to avoid unforeseen obstacles can be added online. To imitate the action of a human, we estimate the actions that maximize the joint probability distribution represented by the DBN. We evaluated the approach and showed that a real robot equipped with a manipulator can learn and reproduce demonstrated actions. Based on a pick-and-place and a whiteboard cleaning task,



Fig. 8. The reproduction of the board cleaning task by our robot. It imitates the zig-zag movement for cleaning the board with the sponge. Note that the learned task representation allows for cleaning differently sized surfaces based on the markers.

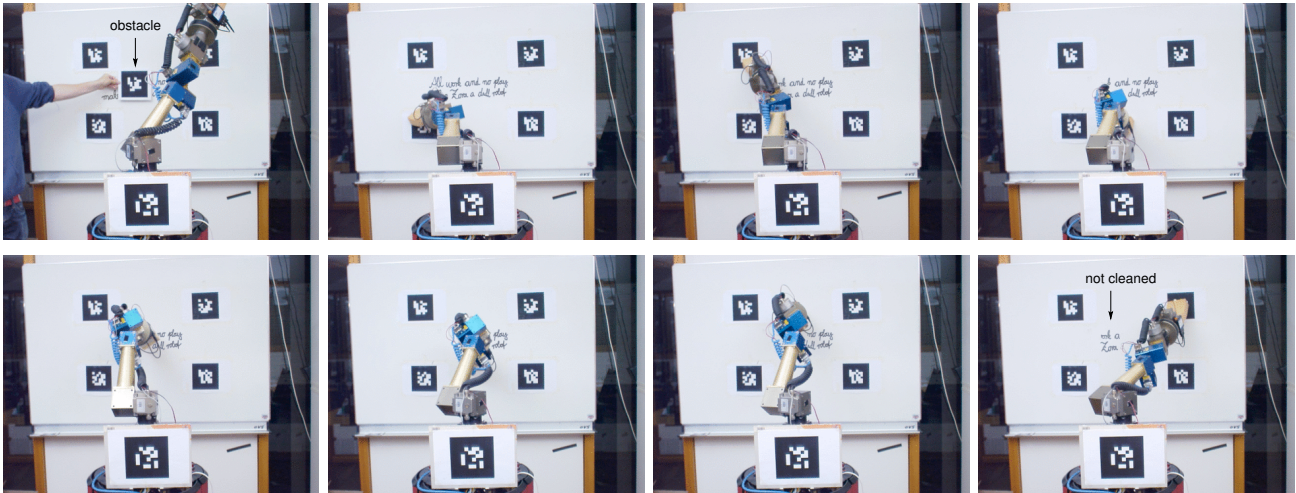


Fig. 9. The board cleaning task with an obstacle reproduced by our 6-DoF manipulator. In the first frame, the position of the obstacle is shown to the system via a marker. Next, the manipulator cleans the whiteboard similar to the previous experiment shown in Fig. 8 but additionally avoids the obstacle. In our current implementation, the obstacle is supposed to have the size and position of the corresponding marker that can be perceived using the ARToolkit system. As can be seen in the last frame, part of the text in the area of the obstacle marker was not wiped.

we illustrated the flexibility of the method to generalize over different spatial setups.

REFERENCES

- [1] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2006.
- [2] P. Bakker and Y. Kuniyoshi, "Robot see, robot do: An overview of robot imitation," in *In AISB96 Workshop on Learning in Robots and Animals*, 1996, pp. 3–11.
- [3] D. Bentivegna, C. G. Atkeson, and G. Cheng, "Learning tasks from observation and practice," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 163–169, 2004.
- [4] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, in press.
- [5] S. R. Buss and J. su Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics Tools*, vol. 10, pp. 37–49, 2004.
- [6] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling skill constraints in joint space and task space," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, September 2008.
- [7] S. Calinon, F. Guenter, and A. Billard, "Goal-directed imitation in a humanoid robot," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005.
- [8] R. Chalodhorn, D. B. Grimes, and R. P. N. Rao, "Learning to walk through imitation," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. San Mateo, CA: Morgan Kaufmann, 2007.
- [9] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, June 2005.
- [10] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [11] D. Grimes, R. Chalodhorn, and R. Rao, "Dynamic imitation in a humanoid robot through nonparametric probabilistic inference," in *Proc. of Robotics: Science and Systems (RSS)*, 2006.
- [12] R. Hafner and M. Riedmiller, "Neural reinforcement learning controllers for a real robot application," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007, pp. 2098–2103.
- [13] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," in *NIPS*, 2002, pp. 1523–1530.
- [14] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. of the 2nd Int. Workshop on Augmented Reality (IWAR)*, 1999.
- [15] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *In Proc. of the 1st SIAM Int. Conf. on Data Mining (SDM)*, 2001, pp. 5–7.
- [16] M. Pardowitz and R. Dillmann, "Towards life-long learning in household robots: The piagetian approach," in *Proc. 6th IEEE International Conference on Development and Learning, London, UK*, 2007.
- [17] J. Peters, S. Vijayakumar, and S. Schaal, "Reinforcement learning for humanoid robotics," in *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2003.
- [18] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert, "Learning Movement Primitives," in *International Symposium on Robotics Research (ISRR2003)*, 2003.
- [19] L. Sciacivco and B. Siciliano, *Modelling and Control of Robot Manipulators*. Springer, January 2000.
- [20] S. Tso and K. Liu, "Hidden markov model for intelligent extraction of robot trajectory command from demonstrated trajectories," in *Proceedings of The IEEE International Conference on Industrial Technology (ICIT)*, 1996.