Robotic Playing for Hierarchical Complex Skill Learning

Simon Hangl, Emre Ugur, Sandor Szedmak and Justus Piater[1]

***Abstract*— In complex manipulation scenarios (e.g. tasks requiring complex interaction of two hands or in-hand manipulation), generalization is a hard problem. Current methods still either require a substantial amount of (supervised) training data and / or strong assumptions on both the environment and the task. In this paradigm, controllers solving these tasks tend to be complex. We propose a paradigm of maintaining simpler controllers solving the task in a small number of specific situations. In order to generalize to novel situations, the robot transforms the environment from novel situations into a situation where the solution of the task is already known. Our solution to this problem is to play with objects and use previously trained skills (basis skills). These skills can either be used for estimating or for changing the current state of the environment and are organized in skill hierarchies. The approach is evaluated in complex pick-and-place scenarios that involve complex manipulation. We further show that these skills can be learned by autonomous playing.**

I. Introduction

Complex object manipulation in uncontrolled environments is a hard and not yet completely solved problem in robotics. One of the major issues in this context is the problem of generalizing motor skills [1]–[4]. Much of it incorporates a paradigm where the aim is to adapt the controller itself to the changing environments. This increases the complexity of the manipulation controller, as it should deal with a wide range of different situations.

We propose to combine simpler and previously-learned skills in order to achieve more complex tasks. The aim is to exploit simple skills to transfer the environment into a state where simple controllers can achieve the desired complex task. This allows the complexity of the controllers to be reduced, as they do not have to deal with generalization.

Humans use similar behavioural patterns e.g. in sports such as golf. The player always tries to stand in the same position relative to the ball instead of adapting the swing itself in order to hit the ball from another position. Therefore, the player is able to execute the same (or very similar), previously-learned trajectories. This can highly reduce the training cost by constraining the search space. We emphasize that in most approaches in robotics, the robot would have to adapt the swing in order to hit the ball from many different positions. A similar strategy seems to be exploited by human infants. Piaget observed similar patterns in infant playing at the age between 8 and 12 months [5]. This stage in the life of infants is called the *coordination of secondary schemata* and Piaget calls it the stage of *first actually intelligent behaviours*. Infants use previously-learned skills to bring the objects into a state where they can perform an intended action (e.g. kicking an obstacle out of the way to grasp an object; pulling a string attached to an object to bring it within reach). An important property of this stage is that they do not predict the effects of these actions directly, but rather learn to compose previously-known skills to achieve a specific task. They do not have an understanding of what the effect of a manipulation is. However, they know that a composition of certain skills leads to a successful manipulation. In this paper we propose a method that follows a similar paradigm. The robot holds a set of *preparatory skills*. These are used to bring the system from an arbitrary state to a state where a desired complex skill can be executed with limited generalization demands. Let $\mathbf{s}$ be the complete (and unknown) physical state of the system the robot is located in. We use a set S of *sensing actions* $S_i \in S$ and the haptic sensor data $\mathbf{t}_{S_i}(\mathbf{s}) \propto p(\mathbf{t}_{S_i}|\mathbf{s})$ collected during the execution of S_i to gather task-relevant information from the environment. We use a classifier to estimate discrete sensing action-dependent state labels $E_{S_ij} \propto p(E_{S_i} = E_{S_ij}|\mathbf{t}_{S_i})$. In the remainder of the paper we will refer to these estimated labels E_{S_ij} as the *perceptual state* (given the sensing action S_i). These labels can be predefined by supervision or generated from experience. Given the observed state E_{S_ij} the robot will pick an appropriate preparatory skill. After preparation, the complex skill is executed in order to achieve a desired task. For clarity we will illustrate the single components in a tabletop book grasping scenario (Fig. 1). In this case the robot cannot perform conventional grasping strategies because it cannot get the fingers below book. The *complex skill* involves the coordination of two hands, where one hand prevents the book from sliding while the other presses against the binding of the book and tries to lift it. The second hand performs an in-hand manipulation to position a finger underneath the book in order to finally grasp it (Fig. 1). The relevant *perceptual state*, namely the robot-relative orientation of the book, can be determined by a sliding action (*sensing action*) along the book surface. The complex behaviour is shown for one specific orientation. Pushing controllers might be handy to prepare this orientation from an arbitrary orientation. Therefore, pushing controllers are good *preparatory skills* for this task.

Fig. 1. Book manipulation: Infant vs. autonomous robot

The learning problem solved in this paper is how to autonomously learn to generalize complex skills shown by kinesthetic teaching (or hard-coding) within the paradigm described above. This involves the selection of the best sensing action and the best preparatory skill given a specific perceptual state in a so-called *playing phase*. We do this by using a reinforcement learning method called *projective simulation* (PS [6]) which is well suited for this type of learning problems. Further, we show how the sensor data gathered during the sensing action can be mapped to the discrete perceptual state by a data-driven classifier called *Maximum Margin*

[1]Department of Computer Science, University of Innsbruck, 6020 Innsbruck, Austria first.last@uibk.ac.at

Regression (MMR [7]). In the playing phase the data required to initialize the classifier is generated autonomously as well. We also show how the robot can create skill hierarchies by adding complex skills to its repertoire of preparatory skills. In this way, the robot can learn increasingly more complex tasks over time.

II. Related Work

Belief-space planning where the systems state is partially observed by sensors is similar in nature. In most belief-space planning methods a control policy is trained and at each time step the next commands are predicted [8], [9]. These commands are given in the action space of the robot, while we select complete controllers. This causes a significant reduction of the learning complexity. Such macro actions were used in a navigation task to reduce the dimensionality [10]. However, these macro actions were limited to the navigation domain (e.g. relocation primitives). In a manipulation scenario, this method would require pre-defined primitives, which is hard to achieve generally. Other related work uses haptic feedback to derive information about the environment. Robots can learn the meaning of haptic adjectives that were previously assigned to a set of objects by interacting (tap, squeeze, slide, ...) with them [11]. Similarly, interaction primitives were used to classify objects by using haptic feedback clustered with K-means [12]. Similar in spirit, these methods do only deal with the estimation of properties and not with manipulation. Associative skill memories on the other hand [13] assign typical task-specific force patterns to manipulations. The patterns are used to predict the success of manipulation during execution. Therefore the robot can react in time and change the trajectory accordingly. Jain et. al. transferred haptic time series collected from stereotypical tasks performed by humans to robotic manipulation [14]. The data was used categorize objects or detect anomalies during the execution. Manipulation primitives have been proposed in order to estimate object poses and further afforded actions [15]. Primitives are composed in order to transfer the object to a pose in which a task can be executed. Even though using similar ideas, our approach is not limited to object poses as state space. Vigorito et. al. [16] predicted manipulation effects (in contrast to our method) and composed skills by planning in state space by optimizing intrinsic reward. Another class of competitors are logic-based planning systems, e.g. STRIPS [17], where provably-correct plans are derived to achieve a certain goal by matching pre- and post-conditions of action primitives. This requires a higher level of abstraction and often prior knowledge (e.g., how abstract symbols are created from real-world data). In contrast, our method does not predict outcomes of actions, but learns successful sequences of actions in an open loop. Open-loop planners for grasping [18], [19] rearrange the objects in clutter in order to perform simpler grasps. However, this method is restricted to the grasping domain. Applications of deep learning to robotics (Levine et. al. [20]) are interesting in this context as they require a high autonomy because of the huge data demand. Large scale autonomous experiments were performed to train a CNN for grasp success prediction, which is used to servo the robot in a closed loop. In contrast to our work they do not need to design internal representations but require a huge amount of data.

III. System architecture

The method comprises two interrelated pathways, the *execution pathway* and the *playing pathway*. The *execution pathway* is used to execute a complex skill, i.e. to execute the sensing action, estimate the perceptual state from haptic data, perform the preparatory action and finally execute the complex skill. Initially, the system

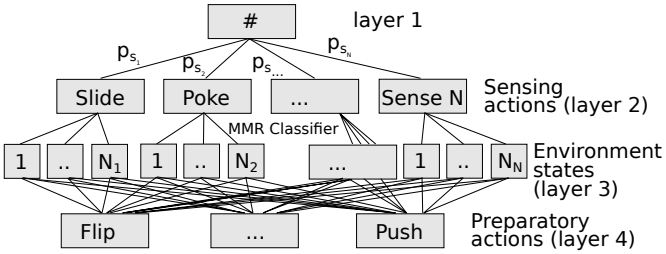


Fig. 2. ECM in the hierarchical skill learning scenario for one complex skill.

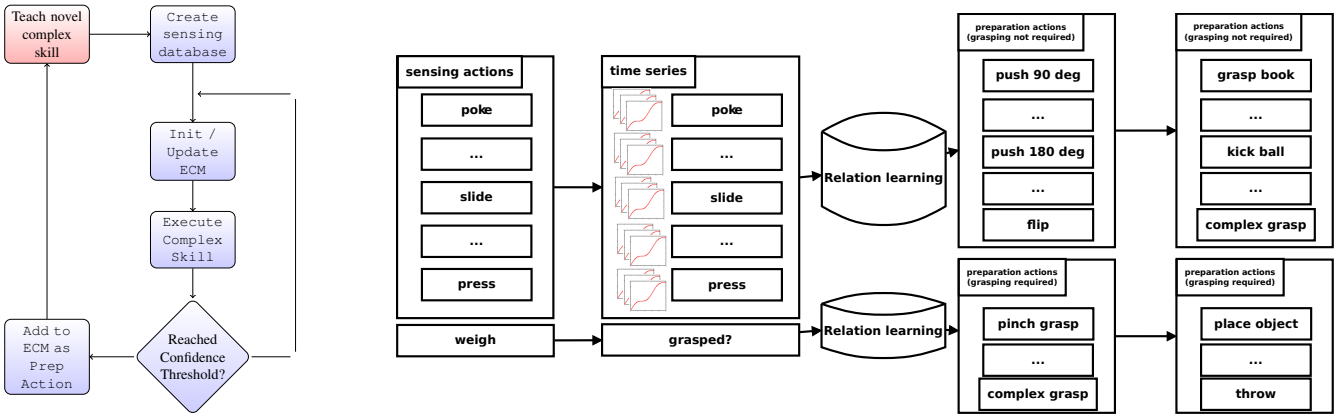
does not know which sensing action and which preparatory skills are required to achieve a certain task. It also has no information about what haptic feedback corresponds to which predefined (or automatically generated) discrete perceptual states. The *playing pathway* is used to acquire this information by playing with the object. In order to train a novel skill, the robot needs to gather haptic information about the perceptual states it might observe (e.g. the rotation of a book or information whether a box is opened or closed) and explores the environment with its sensing actions. Each action S_i is assumed to leave the perceptual states E_{S_ij} unchanged (e.g., they do not change the rotation of a book) and can therefore be performed multiple times. This is important in order to create a haptic database for each state E_{S_ij}. After creating this database the system is trained to *select a sensing action* and to *pick the preparation skill* that ensures the successful execution of the novel skill given an estimated perceptual state. This is achieved by a reinforcement learning method called *projective simulation* (PS) [6]. In each roll-out (skill execution, reward collection and model update) the execution pathway is performed and the reward is measured until the success rate reaches a certain threshold. When the success rate for the novel skill is high enough, it is added to the set of preparatory skills. This way, the construction of skill hierarchies is possible, as a complex skill can be used as a preparatory action for another complex skill. For example, placing a book on a shelf requires grasping it first. If the robot only knows how to push objects, it will not be able to perform the complex placing action. However, as soon as it has learned how to properly grasp a book, it can do the placement by using grasping as a preparatory skill.

A. Execution pathway

In order to execute a novel complex skill that was trained by playing, the following steps are executed (Fig. 3(b)):

- *Select a sensing action* in order to collect data for perceptual state estimation.
- *Perform the sensing action* and measure haptic data.
- *Estimate the perceptual state* by classifying the haptic data.
- *Select and execute a preparatory skill* to transform the environment into a state in which the complex action can be executed successfully.
- *Execute the complex skill*, e.g. by replay of trained trajectories or execution of hard-coded controllers.

1) relational model for sensing / preparation pairs: The relational model is the heart of the method and is used for two essential tasks: (1) selecting the best sensing action given a desired task, and (2) selecting the correct preparatory action given the estimated perceptual state. We use projective simulation (PS) for skill execution and skill learning. PS is a reinforcement learning method that consists of an *episodic and compositional memory*



(a) Overview of *playing pathway*: Training of the haptic properties, selection of the best sensing actions and preparatory skills by reinforcement learning

(b) Schematic sketch of the *execution pathway*. The robot senses the task-relevant perceptual state by haptic exploration and classifies the measured data. The estimated haptic class is mapped to the preparatory action by the relation learning module. After preparing the environment with the preparatory action, the complex skill can be executed.

Fig. 3. Schematic sketch of the two parts of the proposed method: the *playing pathway* (left) and the *execution pathway*.

(ECM) which is a network of so-called *clips*, i.e., episodic memory fragments that include percepts and actions. Each complex skill is encoded by an ECM with the novel, fixed, layered structure proposed in this work (Fig. 2). A complex skill is executed by a *random walk* between clips through the ECM. A transition probability $p\left(c_j|c_i\right)$ is assigned to each pair of clips c_i, c_j that is connected in Fig. 2 with

$$p\left(c_j|c_i\right) = \frac{h\left(c_i, c_j\right)}{\sum_k h\left(c_i, c_k\right)} \tag{1}$$

where $h(c_i, c_j)$ is called the *transition weight*. In the first step (transition from layer 1 to layer 2), a sensing action S_i is selected and executed. A time series $t_{S_i} = \{(t_{S_ik}, \mathbf{F}_{S_ik}, \mathbf{T}_{S_ik}, \mathbf{P}_{S_ik})\}$ of haptic data is measured, where t_{S_ik} is the k-th time step of the observed time series, $\mathbf{F}_{S_ik}$ is the Cartesian force, $\mathbf{T}_{S_ik}$ is the torque and $\mathbf{P}_{S_ik}$ is the Cartesian end-effector position. From this data, the perceptual state is estimated using a multi-class classifier, namely *maximum margin regression* (MMR) [7]. The multi-dimensional input is simply a concatenated vector $\mathbf{t}_{S_i} = (\mathbf{F}_{S_i0}, \mathbf{T}_{S_i0}, \mathbf{P}_{S_i0}, \ldots, \mathbf{F}_{S_iT}, \mathbf{T}_{S_iT}, \mathbf{P}_{S_iT})$. The output is an N_{S_i}-tuple, where N_{S_i} is the number of classes for sensing action S_i. If the time series belongs to class E_{S_ij}, only the j-th entry is nonzero. MMR is simple to use, the code is available online[1], and it was shown to perform well on different complex tasks [21]. In this step, the walk is not random but a transition from clip S_i to the j-th child is performed according to the classifier output. The transition between layers 3 and 4 and therefore the selection of the preparatory skill is again random according to equation 1, and the corresponding preparatory skill is executed. Finally, the robot performs the desired complex skill.

Complex skills that require an object to be grasped are treated separately. The only such sensing action is *weighing*. If the external force $F = 0$, the object is not grasped yet, and only preparatory actions that result in a grasp are considered. Otherwise (i.e., the object is already grasped) the novel complex skill can be executed directly. If the complex skill does not require a grasp, then only the preparatory actions that do not result in a grasp are considered (Fig. 3(b)).

[1]https://iis.uibk.ac.at/software/mmr_mmmvr

B. Playing pathway

A novel complex skill (e.g. shown by hard-coding a controller, kinesthetic teaching or providing more complex controllers with some limited amount of generalization) is learned by playing with the object (Fig. 3(a)). The purpose is to learn the best transition weights $h(c_m, c_n)$ and to create a haptic database for state estimation.

1) Creation of the haptic database: The haptic database is required to initialize the MMR classifier for state estimation. The perceptual state is a discrete class that determines some aspect of the environment. It can, but does not have to, have a semantic meaning (e.g. whether a box is open or not, or the pose of an object). The haptic time series are labelled with the corresponding state E_{S_ij} they were measured in for all sensing actions S_i. In order to do so, each perceptual state E_{S_ij} has to be created. This can be done in a supervised or unsupervised manner. In other words, either a supervisor prepares the state (e.g. a human shows the robot a box when it is open or closed) or the robot tries to prepare the states by itself. If no supervisor is available to play with the robot, it uses the preparatory skills to change the state of the environment (e.g. rotating a book by 90 degrees in each iteration). After each execution the system is assumed to be in a novel state and the haptic time series are collected.

Not all sensing actions have the same discriminative power for every task / object (e.g., while a sliding action is good for deciding the orientation of a book, a poking action is not helpful in that case). Therefore, for each complex skill and object class we assign a *discrimination score* D_i to each sensing action S_i. It is computed by performing cross-validation for the time series classifier. With an average success rate of s_i, the discrimination score D_i is given by

$$D_i = \exp(\alpha s_i) \tag{2}$$

with a fixed *stretch factor* α. The higher the stretch factor, the more strongly slight differences in the average success rate influence the discrimination score. The score D_i determines how well the sensing action S_i can distinguish its states E_{S_ij}.

2) Initialization of the ECM: After the classifier is trained, the robot can start with the exploration of sensing and preparation actions. The ECM has to be initialized, setting the transition weights

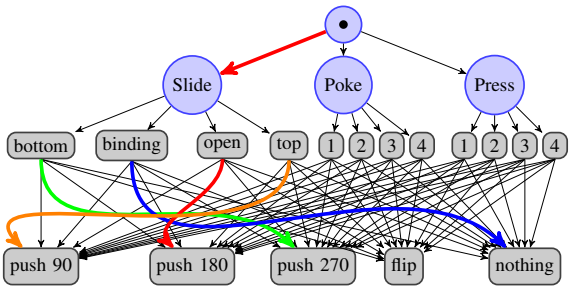


Fig. 4. Qualitative sketch of the episodic memory after learning how to grasp a book. Coloured lines indicate a high probability of the transition. Semantic labels are assigned to perceptual states if available.

$h(c_i, c_j)$. Sensing actions that can discriminate well between their states should be preferred, and the discrimination score D_i can be used as an initial transition weight between the #-clip (starting clip of each random walk) and clip S_i with $p_{S_i} \propto h_{S_i} = D_i$. The transition probabilities between layers 2 and 3 are given by the time series classifier, where only the transition to the predicted state has nonzero probability. The weights from layer 3 to layer 4 are initialized with the constant value h_{init} (uniform distribution).

3) Relation learning: In order to learn which sensing actions are discriminative for a given task and which preparatory skills should be used in an observed state, PS provides a way to update the transition weights by using external rewards. The update is done with a modified version of the original PS update rules [6]. Random-walk paths should be more likely in future situations if the action taken was rewarded, i.e. the complex action succeeded after performing the preparation action, and should be less likely otherwise. Let $\{s = c_1 \to c_2 \to \cdots \to c_K = a\}$ be a random walk path that received a reward $\lambda^{(t)} \in \mathbb{R}$. The weights are updated by

$$\underbrace{h_{ij}^{t+1}}_{\text{next weight}} = \max(1.0, \underbrace{h_{ij}^{t}}_{\text{current weight}} \underbrace{-\gamma\left(h_{ij}^{t} - 1\right)}_{\text{damping}} + \rho_{ij} \underbrace{\lambda^{(t)}}_{\text{reward}}) \quad (3)$$

where $h_{ij}^t = h^t(c_i, c_j)$. ρ_{ij} is 1 if the path consists of a transition $c_i \to c_j$ and 0 otherwise. The forgetting factor γ defines how quickly the model forgets previously-achieved rewards for a given path. It should only be nonzero if the robot is placed in an environment where the behaviour of the objects changes slowly over time (e.g., the object can break and change its physical properties after manipulating it for a few hours).

4) Building skill hierarchies: If complex skill A can be executed with a certain confidence, it is added to the ECM of another complex skill B by connecting it to each clip in layer 2 with the initial weight h_{init}. The robot then goes back to the playing phase for skill B. If B already has a high confidence, the transition weights to certain preparatory skills will be high compared to h_{init} and the probability of exploring the new preparatory skill A is low but nonzero. If the confidence is low, all weights will be low and the robot will start exploring the novel preparatory skill. Thus, PS is well suited for constructing skill hierarchies.

IV. Evaluation

For evaluation we apply our approach to a *complex book grasping* task in an autonomous playing scenario. In a placement task a skill hierarchy using the grasping skill is learned. We use statistics (e.g. experimental success rates of skills) of the book grasping scenario to *simulate the convergence behaviour* of the same setting in case more preparatory skills are used. We further show that the same sensing actions and preparatory skills can be used for the different problem of placing an object into a (closed) box.

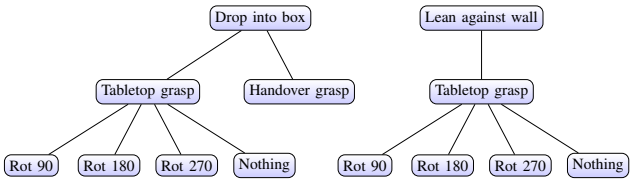


Fig. 6. Skill hierarchies for given complex skills (sub-skills with very low usage probability are omitted). For leaning vertically the correct orientation of the book matters. For the tabletop grasp the book is always grasped at the binding side, whereas in the handover grasp this is not the case (handover grasp is omitted). For dropping the book orientation does not matter and the hand-over grasp is considered.

A. Applicability to real-world tasks

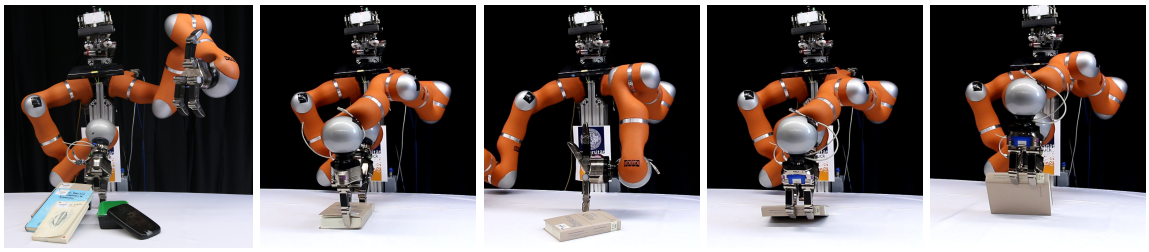
We apply our method to a book grasping task (see video [2]). The main challenge is to get a finger underneath the book in order to grasp it. The book is grasped by squeezing it between both hands, lifting it on the binding side and then using in-hand manipulation to wrap the fingers around it. It is easy to teach this skill for a single specific situation (e.g. by kinesthetic teaching) but hard to generalize to arbitrary situations because of the complex interplay of two arms and the book in several different orientations.

The experiments were performed with two KUKA LWR 4+ robotic arms with Schunk SDH grippers (Fig. 5(a)). Different types of books (soft-cover, hard-cover, varying sizes) were used to train the haptic database. All experiments were performed with the built-in impedance mode of the KUKA arms, which allows books of different dimensions to be handled without explicitly coding them into the skills.

Three different sensing actions were used: Sliding (finger slides along the edge that is parallel to the table edges, while the second hand keeps the object in place), Poking (the object is poked from the top) and Pressing (the book is squeezed between the 2 hands). As preparatory skills we used a discretised version (90, 180 and 270 degrees) of a rotation controller that rotates the object by an arbitrary angle. We also used a flipping controller (flipping the book upside down). The reward was estimated automatically by measuring the force on the end-effector. After rewarding the book was dropped onto the table and a random rotation action was selected to prepare another random starting state. The learning parameters of the PS model were set to $\lambda_{\text{succ}} = 1000$ (successful roll-outs), $\lambda_{\text{fail}} = -30$, $h_{\text{init}} = 200$ and $\gamma = 0$ (no forgetting).

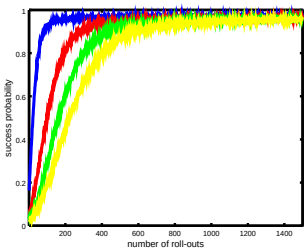
One of the challenges was to design robust controllers for autonomous play. All the controllers and machine learning techniques were developed within the *kukadu* framework[3][4]. The code is available online and free to use. The robot was made to play for 100 roll-outs. For creation of the haptic database the book was pushed clockwise by 90 degrees in order prepare the perceptual states autonomously. At each rotation, 50 samples per sensing controller were collected. A qualitative sketch of the learned ECM is shown in Fig. 4. The thick, coloured lines correspond to transitions with high weights and high probability. The dominant sensing action is the sliding action, and its child states have a semantic meaning, i.e., the orientation of the book. The transitions between the state clips and the preparatory skills match the ground truth. The

[2] https://iis.uibk.ac.at/public/shangl/iros2016/iros.mpg
[3] https://github.com/shangl/kukadu
[4] https://github.com/shangl/iros2016

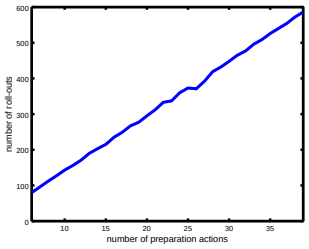


(a) (b) (c) (d) (e)

Fig. 5. Robot setting (Fig. 5(a)) used for the experiments; Figs. 5(b) to 5(e) show a sample execution of the book grasping skill. In Fig. 5(b) the sliding sensing action is visualized; Afterwards, the book is rotated by 90 degrees (Fig. 5(c)) and lifted (Figs. 5(d), 5(e)).



(a) Evolution of average success rate for $N_p = 6$ (blue), $N_p = 20$ (red), $N_p = 30$ (green), $N_p = 39$ (yellow)



(b) Number of roll-outs N_r required to reach a success rate of 0.9 for different N_p.

Fig. 7. Convergence behaviour of the proposed hierarchical skill learning model

execution sequence for a specific perceptual state after the playing phase is shown in Figs. 5(c)–5(e). After training, the tabletop grasping strategy was added to the set of preparatory skills, and two placement strategies (drop into box, lean against a wall) were shown by kinesthetic teaching. The robot was able to learn that it has to grasp the book first. In that scenario, the robot constructed the skill hierarchies shown in Fig. 6.

B. Convergence simulation

An important property is the success rate convergence. For the book picking task the ground truth is known, as the perceptual states of the sliding action are semantically meaningful, i.e. the orientation of the book. Success rate statistics for the implemented controllers were measured during the real experiment and were used to simulate the convergence behaviour if some useless preparatory skills were added. The probability of correct classification was given by $p_{\text{slide}} = 0.93$ (sliding action), $p_{\text{poke}} = 0.27$ (poking action), $p_{\text{press}} = 0.4$ (pressing action). The success probability of the grasping action was $p_p = 0.98$ (given that the correct perceptual state was estimated). As the used projective simulation model is based on a stochastic process, $N = 10000$ agents (each agent is a simulated separate robot) were executed over $t = 1500$ roll-outs for different numbers of preparatory actions N_p. The success in each time step was averaged over all agents (Fig. 7(a)). Fig. 7(b) shows how many roll-outs were required to reach a success rate $p_{\text{succ}} = 0.9$ for different N_p. For the special case of $N_p = 6$ the number of roll-outs was determined with $N_r = 80$. A trivial system that tries every combination of $N_s = 3$ sensing actions, $N_o = 4$ sensing outcomes per action and $N_p = 6$ preparation actions would require $N_r = N_s N_o N_p = 72$ roll-outs to observe every combination only once. However, from this information it is difficult to infer knowledge about a certain sensing / preparation skill combination (e.g., success / failure could be caused by noise, unexpected temporary circumstances, etc.). Our method only requires 80 roll-outs and focusses on regions in the exploration space that are interesting for the problem and is able to handle such kind of noise.

C. Task diversity

In Section IV-A, a specific example for the applicability of the method was shown. We now show that the same setting can be used to learn a diverse set of skills.

In the book grasping scenario the orientation of the book was estimated by sliding. However, our method does not assume such semantic categories and is not bound to any specific property (e.g. pose) of the object. To illustrate this, the system with the same sensing and preparatory controllers was confronted with a small rectangular food-box with a removable cover (Fig. 5(a)). The task was to place an object inside the box. By kinesthetic teaching the robot was taught to grasp an object and place it inside the open box. For the generation of the haptic database, the box was placed in front of the robot in the open and closed configurations. Poking was determined to be the best sensing action. From a human perspective it is reasonable to poke the top of a box to determine whether it is open or not. The flipping skill of the book picking scenario turned out to be able to remove the cover from the box as a preparatory action to place something inside the box (for comparison see Fig. 8).

We emphasize that the robot does not have any notion of the semantic meaning of the actions it performed. Still it came up with a semantically meaningful selection of skills and was able to achieve the task with the same skill set as in the book picking scenario.

D. Limitations

The design choices responsible for the high degree of autonomy and fast convergence, of course, come with some disadvantages and limitations. The strongest restriction comes from the open-loop nature. No effects are predicted, and in the current version there is no explicit way to check whether the task is still going well during execution. However, implicit error handling mechanisms can be hidden inside the preparatory controllers, which puts high demands on them. Further, complex planning outside of the skill hierarchies is not possible as there is no environment model. This is a disadvantage in case of multi-object tasks. Another restriction arises from the loose coupling of controllers in very dynamic tasks. For example, a ball might roll away between preparation and execution of the complex skill.

V. Conclusions

We introduced a novel method for robotic object manipulation. The key idea is to teach novel skills achieving complex tasks with limited generalisation capabilities. Previously-trained skills are then

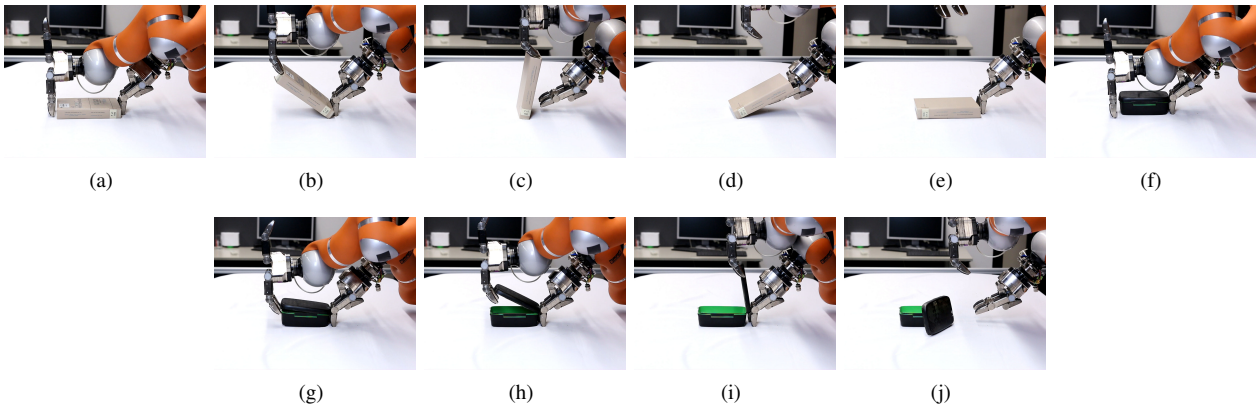


Fig. 8. Flipping skill performed in two different tasks – providing two different semantic effects. In the book picking scenario, the book got flipped. In the box opening scenario, the box got opened.

used to prepare the environment such that the limited controllers can still succeed. This way, the teaching of novel skills is simple. In this framework, learning can be mostly autonomous. Further, complex skill hierarchies can be constructed by adding learned complex skills to the set of preparatory skills.

The approach was evaluated in a complex pick-and-place task, where a book can be placed in several different ways. The system was able to learn that it has to grasp the book before it can be placed. Further it learns that the book has to be rotated to the right orientation such that it can be grasped from a table with a complex sequence of in-hand manipulations. Additionally, the same set of sensing and preparatory skills can be applied to a wide set of different problems. This was shown by using the same preparatory skill set for putting an object inside a closed box. It succeeded to open the box by the flipping action in order to place an object inside. Further, the convergence of the learning approach was evaluated in simulation when more preparatory actions are used. It was found that during the exploration phase, the robot focusses on regions that are of particular interest to the problem. Therefore it can learn a skill with high confidence with just a slightly higher number of roll-outs compared to executing every combination only once.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Communitys Seventh Framework Programme FP7/20072013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 610532, Squirrel. Special thanks to Hans J. Briegel and Vedran Dunjko for the very helpful and fruitful discussions and guidance.

REFERENCES

[1] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, May 2009.

[2] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic, "Dual arm manipulation – a survey," *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1340–1353, 2012.

[3] K. Mülling, J. Kober, O. Kroemer, and J. Peters, "Learning to select and generalize striking movements in robot table tennis," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 263–279, 2013.

[4] S. Hangl, E. Ugur, S. Szedmak, A. Ude, and J. Piater, "Reactive, Task-specific Object Manipulation by Metric Reinforcement Learning," in *17th International Conference on Advanced Robotics*, 7 2015.

[5] J. Piaget, *The Origins of Intelligence in Children*. New York: Norton, 1952.

[6] H. J. Briegel and G. De las Cuevas, "Projective simulation for artificial intelligence," *Scientific Reports*, vol. 2, pp. 400 EP –, May 2012.

[7] S. Szedmak, J. Shawe-Taylor, and E. Parado-Hernandez, "Learning via linear operators: Maximum margin regression," in *PASCAL Research Reports, http://eprints.pascal-network.org/*, 2005.

[8] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *Int. J. Rob. Res.*, vol. 31, no. 11, pp. 1263–1278, Sept. 2012.

[9] N. Du Toit and J. Burdick, "Robot motion planning in dynamic, uncertain environments," *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 101–115, Feb 2012.

[10] G. Theocharous and L. P. Kaelbling, "Approximate planning in POMDPs with macro-actions," in *Advances in Neural Information Processing Systems 16 (NIPS03)*, 2004.

[11] V. Chu, I. Mcmahon, L. Riano, C. G. Mcdonald, Q. He, J. M. Perez-tejada, M. Arrigo, N. Fitter, T. Darrell, and K. J. Kuchenbecker, "Using robotic exploratory procedures to learn the meaning of haptic adjectives," in *ICRA*. IEEE, 2013.

[12] A. Schneider, J. Sturm, C. Stachniss, M. Reisert, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, Oct 2009, pp. 243–248.

[13] P. Pastor, M. Kalakrishnan, F. Meier, F. Stulp, J. Buchli, E. Theodorou, and S. Schaal, "From dynamic movement primitives to associative skill memories," *Robotics and Autonomous Systems*, 2012.

[14] A. Jain and C. Kemp, "Improving robot manipulation with data-driven object-centric models of everyday forces," *Autonomous Robots*, vol. 35, no. 2-3, pp. 143–159, 2013.

[15] S. Sen, G. Sherrick, D. Ruiken, and R. Grupen, "Choosing informative actions for manipulation tasks," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, Oct 2011, pp. 721–726.

[16] C. M. Vigorito and A. G. Barto, "Intrinsically motivated hierarchical skill learning in structured environments," *IEEE T. Autonomous Mental Development*, vol. 2, pp. 132–143, 2010.

[17] R. E. Fikes and N. J. Nilsson, "Strips: A new approach to the application of theorem proving to problem solving," in *Proceedings of the 2Nd International Joint Conference on Artificial Intelligence*, ser. IJCAI'71. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1971, pp. 608–620.

[18] M. Dogar and S. Srinivasa, "A framework for push-grasping in clutter," in *Robotics: Science and Systems VII*, N. R. Hugh Durrant-Whyte and P. Abbeel, Eds. MIT Press, July 2011.

[19] D. Omrcen, C. Boge, T. Asfour, A. Ude, and R. Dillmann, "Autonomous acquisition of pushing actions to support object grasping with a humanoid robot," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, Dec 2009, pp. 277–283.

[20] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *arXiv preprint arXiv:1603.02199*, 2016.

[21] W. Mustafa, H. Xiong, D. Kraft, S. Szedmak, J. Piater, and N. Krüger, "Multi-Label Object Categorization Using Histograms of Global Relations," in *International Conference on 3D Vision*. IEEE, 10 2015.