Learning to Manipulate Articulated Objects in Unstructured Environments Using a Grounded Relational Representation

Dov Katz Yuri Pyuro Oliver Brock
Robotics and Biology Laboratory
University of Massachusetts Amherst

***Abstract*— We introduce a learning-based approach to manipulation in unstructured environments. This approach permits autonomous acquisition of manipulation expertise from interactions with the environment. The resulting expertise enables a robot to perform effective manipulation based on partial state information. The manipulation expertise is represented in a relational state representation and learned using relational reinforcement learning. The relational representation renders learning tractable by collapsing a large number of states onto a single, relational state. The relational state representation is carefully grounded in the perceptual and interaction skills of the robot. This ensures that symbolically learned knowledge remains meaningful in the physical world. We experimentally validate the proposed learning approach on the task of manipulating an articulated object to obtain a model of its kinematic structure. Our experiments demonstrate that the manipulation expertise acquired by the robot leads to substantial performance improvements. These improvements are maintained when experience is applied to previously unseen objects.**

I. Introduction

Autonomous manipulation remains one of the great challenges in robotics. The successful endowment of autonomous robots with robust manipulation skills will have substantial impact in many important application areas, ranging from personal and professional service robotics to flexible manufacturing and planetary exploration.

We view autonomous manipulation as the purposeful and deliberate change of the configuration of an object. The object's configuration uniquely describes the object's pose by specifying every degree of freedom of the object. An object can have extrinsic and intrinsic degrees of freedom. Extrinsic degrees of freedom describe the spatial relationship between the object and its environment. Intrinsic degrees of freedom describe the relationship among the rigid bodies of an articulated object and are often relevant to the object's intended function. Examples of objects with intrinsic degrees of freedom include tools (scissors, pliers, etc.), doors, door handles, books, or drawers. Successful manipulation must be informed by knowledge of the extrinsic and intrinsic degrees of freedom of an object.

In unstructured environments, a robot cannot rely on a detailed and accurate *a priori* model of the environment. It must therefore be able to acquire task-relevant information to inform its interactions with the objects in the environment. Based on this information, the robot must adapt its manipulation behavior to ensure successful task execution. Manipulation becomes a continuous and interactive process of acquiring information about the environment and subsequently adapting the interaction with the environment in response to this information.

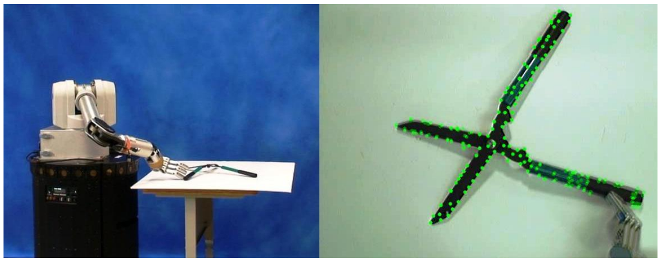


Fig. 1. UMan (UMass Mobile Manipulator) interacts with an articulated object to acquire information about the object's kinematic structure. The right image shows the scene as seen by the robot through an overhead camera; dots mark tracked visual features.

The contribution of this paper is a learning-based approach to manipulation in unstructured environments. This approach permits the robot to autonomously acquire manipulation expertise from its interactions with the environment. The resulting expertise enables the robot to select the most effective manipulation action based on partial state information. The manipulation expertise is learned in a relational state representation. This representation is essential, as it renders learning tractable by collapsing a large regions of the state space onto a single, task-relevant, relational state. The symbolic representation is carefully grounded in the perceptual and interaction skills of the robot to ensure that relationally learned knowledge remains applicable in the physical world.

Using this learning-based manipulation approach, we show how a robot can autonomously learn manipulation strategies to obtain a kinematic model of unknown articulated objects. The robot physically interacts with the object by pushing or pulling it and observes the object's motion (see Fig. 1). As these interactions create a change in the configuration of the object, the robot incrementally discovers the object's intrinsic and extrinsic degrees of freedom. The robot learns to select interactions that are most likely to reveal the maximum information about the kinematic structure. The acquired manipulation knowledge substantially reduces the number of interactions required to obtain an accurate kinematic model. Furthermore, the manipulation knowledge acquired with one object transfers to other objects, even if they have different kinematic structures.

In the following section we introduce the relational representations of kinematic structures that forms the basis of our learning-based approach to manipulation. We then describe how this representation can be grounded using the perceptual and interaction capabilities of the robot. We proceed to discuss the relational learning framework and how it can be grounded with respect to the relational representation. Finally, we demonstrate the effectiveness of our approach in manipulation experiments with articulated objects.

II. Related Work

We distinguish between three categories of approaches to manipulation [2, 15, 17]. The first category pertains to manipulation planning. Approaches in this category assume that an accurate geometric model of the manipulated object is available and devise manipulation plans based on this model. These manipulation planners address various flavors of manipulation, including grasping and in-hand manipulation [24], manipulation with sliding contacts [26], non-prehensile manipulation [1, 14], and gross motion planning for manipulation [21]. In contrast to these approaches, we focus on problems for which accurate models are not available.

The second category uses feedback control to achieve manipulation. Particularly relevant are approaches that use learning to design controllers [28]. These methods alleviate the difficulties that analytical methods for controller design encounter in the presence of modeling errors for systems with complex kinematics and dynamics. A different approach to learning-based controller design relies on memory-based learning [16]. Controllers can also be designed by searching in configuration space [22]. All methods in this category determine specific controllers that can serve to ground a relational state representation such as the one we use.

Whereas approaches in the previous category are concerned with individual controllers, approaches in the third category sequence [3, 18] or compose [20] multiple controllers to generate more complex manipulation behaviors. Composite controllers can be arranged into state transition diagrams to further increase robustness and versatility [19]. Most often, the necessary state transition diagrams are designed by the programmer, but they can also be learned using reinforcement learning [9]. Again, we view the resulting controllers as candidates to ground a relational state representation.

There are many other approaches that address aspects of manipulation but cannot easily be assigned to one of these three categories. We will discuss several with particular relevance to the work presented here. Christiansen et al. [4] learn manipulation strategies for a tray-tilting task in conjunction with a dynamic model of the domain. Edsinger and Kemp emphasize the importance of task-specific perceptual features that exploit common structural features of functionally related objects to facilitate manipulation in human environments [8]. Stoytchev presents an approach to learn tool affordances for robotic tool use [23]. He emphasizes the importance of grounding this representation in the robot's behavioral repertoire. This enables the immediate application of the robot's accumulated experience. In our work, we combine task-specificity for perception and grounding for action by requiring that an adequate grounding of our relational state representation has to rely on task-specific perception *and* task-specific behaviors.

III. Relational Representation

The relational representation is critical to the success of our learning-based approach to manipulation. Using a finite set of relations, we are able to describe an infinite number of states and actions. It thus becomes possible to represent and reason about situations that a propositional representation cannot handle. For example, a robot may encounter many types of scissors, varying in color, shape, and size. All scissors, however, have the same kinematic structure. A single relational formula can capture the kinematic structure of *all* scissors, irrespective of their other properties. Therefore, a single relational action can be applied to *all* objects. In contrast, a propositional representation would have to include a proposition for every encountered object and one for every action applicable to this object. The relational representation avoids this combinatorial explosion of actions and states, thus greatly reducing the state space and making learning possible.

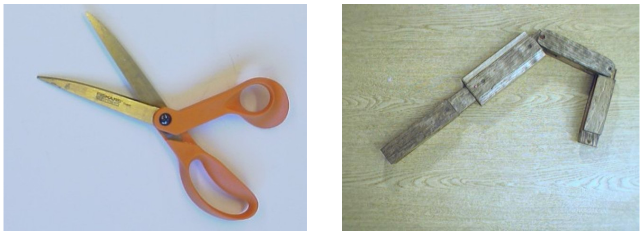


Fig. 2. Two examples of kinematic structures: scissors with a single revolute joint and a wooden toy with a prismatic joint and two revolute joints.

Our relational representation for kinematic models of articulated objects captures joint types, link properties, and kinematic relationships between links. Figure 2 shows two examples of planar kinematic structures. The scissors have a single revolute degree of freedom and the wooden toy is a serial kinematic chain with a prismatic joint (on the left of the figure) and two revolute joints. Our relational representation uses predicates $R(\cdot)$, $P(\cdot)$, and $D(\cdot)$ to describe that rigid bodies are connected by a revolute joint, a prismatic joint, or are disconnected, respectively. The predicates are n-ary, with $n \geq 2$, to capture branching kinematic structures. The rigid body passed as the first argument to the relation is the one in relationship with all other arguments. For example, $R(x, y, z)$ is equivalent to $R(x, y) \wedge R(x, z)$.

Using these relations, we can represent the kinematic structure of the scissors as

$$D(l_b, R(l_1, l_2)),$$

where l_1 and l_2 represent the two links of the scissors and l_b is a disconnected background link. The kinematic structure of the wooden toy can be represented as

$$D(l_b, R(l_4, R(l_3, P(l_1, l_2)))).$$

Note that this representation is not unique. The wooden toy could also be represented as

$$D(P(l_4, R(R(l_1, l_2), l_3)), l_b).$$

Which of these representations is used by the robot depends on the order of discovery of the links. The most deeply nested relation is discovered first.

Kinematic loops are represented by using the same link twice. A 5R kinematic loop is described by:

$$D(l_b, R(l_1, R(l_5, R(l_4, R(l_3, R(l_2, l_1)))))).$$

By extending our atomic representation of links to m-ary relations $L(\cdot)$, $m \geq 1$, we can include link properties in our description of kinematic chains. In this paper we will limit ourselves to a single property, namely the size of the link. The wooden toy can now be represented as

$$D(l_b, R(L(s, f_4), R(L(s, f_3), P(L(s, f_1), L(s, f_2))))),$$

where s stands for the property *small* and the f_i spatially identify links in the physical world (see section IV). The extension to an arbitrary number of link properties is straightforward.

With this relational representation of kinematic structures, it becomes possible to reason and learn about objects based on their kinematic structure. All experience acquired by manipulating scissors can be applied to all other scissors, as long as they have an identical kinematic structure. If specific properties of the links of an object affect the desired manipulation behavior, we can add these properties to the relational description of the links. Our representation is then able to differentiate between identical kinematic structures based on link properties. All properties irrelevant to manipulation are ignored during learning. This reduction in state space makes the learning problem considerably easier.

We also use a relational representation for the actions performed by the robot. Actions apply pushing or pulling forces to one of the links. The forces can be applied along the major axes of the link or along a forty-five degree angle to the major axes. An action is represented as $A(L(\cdot), \alpha)$, where $L(\cdot)$ represents a link and $alpha$ is an atom describing one of the possible six pushing/pulling directions relative to the link.

IV. Grounding the Relational Representation

The relational representation described in the previous section can only support the learning of manipulation knowledge if it is grounded in the physical capabilities of the robot. Grounding bridges between the symbols of our representation and the physical, continuous world [10]. Grounding ensures that we can symbolically interpret the observations made by the robot in regards to its interactions with the world. At the same time, grounding ensures that the resulting symbolic manipulation knowledge maintains its relevance and predictive power for the robot's real-world interactions.

To ground our relational representation, we bind the relations $R(\cdot, \cdot)$, $P(\cdot, \cdot)$, and $D(\cdot, \cdot)$ as well as the link properties to real-world perceptual capabilities of the robot.

In prior work we developed a skill for the robust perception of kinematic degrees of freedom and link properties of planar articulated objects [12]. Figure 1 shows a real-world interactive experiment with garden shears. The robot interacts with the shears to determine the location of the revolute joint and the spatial extent of the links. The image on the right shows the robot's view of its own interaction with the shears. Dots indicate tracked visual features.

This skill provides adequate grounding for our relational representation of links and their kinematic relationship. It extracts the degrees of freedom of an object by tracking the motion of the visual features in the scene. Tracked features are clustered into links using a graph representation in which the features correspond to vertices. Two vertices are connected by an edge if the relative distance of the corresponding visual features does not change throughout the interaction with the object. By clustering the features, it is possible to identify the spatial location and extent of the links. The features associated with a single link are grouped into the sets f_i (see previous section). The relationship between different clusters (links) in the graph can be analyzed to reveal their kinematic relationship.

The robustness of this skill has been proven in dozens of real-world experiments. The skill does not require prior knowledge of the object, is insensitive to lighting conditions and specularities, succeeds irrespective of the texture and color of the object's parts, works reliably even with low-quality cameras, and at the same time is computationally efficient. As a consequence, it is ideally suited for the grounding of our relational representations of kinematic structures for unstructured environments. For a detailed description of this interactive perception skill the reader is referred to reference [12].

V. Learning Manipulation with Grounded Relational Reinforcement Learning

The grounded relational description of states is the basis for our learning framework. To learn manipulation knowledge from interactions with the environment, we cast the incremental acquisition of kinematic representations of objects as a relational reinforcement learning [7, 25, 27] problem.

In reinforcement learning, an agent learns an optimal policy for solving a task. This policy tells the robot which action to perform in a particular state. The robot acquires the policy incrementally, by performing experiments. In our case, an experiment consists of the robot pushing an object. For every action, the robot receives a reward. In our experiments, the robot receives a reward for every degree of freedom it discovers. Over the course of multiple experiments, the robot incorporates new experiences into its policy. As a result, our robot learns an effective policy for acquiring kinematic models of articulated objects.

We formalize this problem as a Relational Markov Decision Process (RMDP) [27] and then apply Q-learning [29] to find an optimal policy. A Markov Decision Process (MDP) is a tuple $M = (S, A, T, R)$, where S designates the set of possible states, A is the set of actions available to the robot,

$T : S \times A \rightarrow \Pi(S)$ specifies a state transition function to determine a probability distribution $\Pi(S)$ over S, indicating the probability of attaining a successor state when an action is performed in an initial state, and $R : S \times A \rightarrow \mathbb{R}$ is a function to determine the reward obtained by taking a particular action in a particular state. In our case, the description of states and actions is relational and therefore we have a relational MPD.

The relational description of states and actions of the RMDP was presented in Section III. We now describe the remaining components of the RMDP and how Q-learning is employed to determine an optimal policy π for manipulating articulated objects in unstructured environments.

A. Transition Function

The transition function captures the state transitions that occur in the physical world when an action is applied. We never explicitly represent this function. Instead, we rely on the real world and on our perceptual capabilities to determine the new state after the application of an action has been completed.

B. Reward Function

The reward function $R : S \times A \rightarrow \mathbb{Z}$ returns the number of links and joints that were discovered by performing an action in a particular state.

C. Q-Learning

Q-learning [29] determines a policy $\pi : S \rightarrow A$ for selecting actions based on the current state. To determine this policy, our goal is to learn the Q-value function $Q : S \times A \rightarrow \mathbb{R}$ by performing a series of experiments, each of which reveals how much reward a particular action can obtain in a particular state. The Q-value function accumulates information about the total expected reward for an entire trial. The policy defined by the Q-value function is given by $\pi(s) = \arg\max_a Q(s, a)$.

As the robot performs actions in its environment and receives the resulting rewards, the Q function is updated according to the following rule:

$$Q(s_t, a_t) = (1-\alpha)\, Q(s_t, a_t) + \alpha \left(r_{r+1} + \gamma \max_a Q(s_{t+1}, a) \right),$$

where α is the learning rate, γ is the discount factor, and r_t is the reward received at time t.

D. Representation of Q-Value Function

Q-learning requires an adequate representation for the Q-value function. In our case, this representation is instance-based [6]. The robot stores each of its experiences as a tuple of state, action, and the Q-value obtained when performing the action in that state. Because states and actions are relational and stored uninstantiated, every stored experience is applicable to a possibly infinite number of situations.

Given the current state, the robot has to retrieve estimates of Q-values for actions from its experience. This is particularly important when the robot has not previously visited the current state. By doing so, the robot is able to leverage relevant prior experience in a new situation, thereby improving its learning performance.

To identify the experience most relevant to a state, we need a similarity measure for states. Similarity is affected by the state's kinematic structure and by the properties of the links in that structure. Neither of these aspects have to match perfectly for the robot to retrieve relevant experience. We first describe how unification is used to match properties of individual links between the kinematic structure of the current state and the state stored in the Q-value function. We then explain how similarity between kinematic structures can be identified.

Let us assume the robot at time t has uncovered the existence of three links (large, small, large), connected into a serial chain by revolute joints; the corresponding relational state description is

$$s_t = R(R(L(l, f_1), L(s, f_2)), L(l, f_3)),$$

ignoring the background for simplicity. Further assume that the Q-value function representation contains a single experience with a structure/action/reward tuple

$$(s, a, r) = (R(R(L(s, v_1), L(s, v_2)), L(s, v_3)), A(v_3, 45^\circ), 1.6)\,.$$

The state s represents a serial chain with two revolute joints and three small links. Note that the Q-value function does not store the sets of features f_i for each link but instead includes a variable v_i. This variable can now be instantiated by unifying the memorized state s with the current state representation s_t. Due to the different instantiation of link size, however, unification fails in this case. We can still retrieve somewhat less relevant experience by ignoring the link size. The resulting unification leads to a binding of $v_3 \leftarrow f_3$. This instantiates the action to $A(f_3, 45^\circ)$, telling the robot to push on the link described by the visual features in f_3 from 45° angle relative to the principal axes of the feature set.

This example illustrated how the unification process progressively ignores the least discriminative property of links until unification succeeds. We now explain how similar kinematic structures can be mapped onto each other to retrieve relevant experience.

We saw in Section III that the relational representation of kinematic structures is not unique. State s_t, for example, is equivalent to $R(L(l, f_3), R(L(s, f_2), L(l, f_3)))$. We would like to retrieve relevant experience in the presence of this ambiguity. Furthermore, for a state $s_t = R(L(l, f_1), L(l, f_2))$ we would like to be able to leverage our experience by realizing that $L(l, f_1)$ in s_t could represent $R(L(l, f_1), L(s, f_2))$ in s before the additional degree of freedom was discovered.

To identify closely related kinematic structures, we represent a relational state as an undirected, labeled graph $G = (V, E)$. A vertex $v \in V$ corresponds to a link. An edge $e \in E$ is labeled as either prismatic or revolute, corresponding to the kinematic relationship between two links.

This graph representation naturally supports the desired ability to retrieve relevant experience from the Q-value function, even for structure-preserving re-orderings of the relational representation as well as for super or sub-structures of the current state. Given two graphs G_t and G corresponding to s_t and s, we check for graph isomorphism to find exact structural matches and sub-matches, even when the relational descriptions of the underlying structure vary. To determine partial matches, we check for subgraph monomorphism between G_t and G. In contrast to subgraph isomorphism, which is a bijection, subgraph monomorphism is an injection, thus the match is one-to-one but not onto.

When one or multiple graph matches exist, the robot retrieves the experience associated with the closest match. When no graph match can be established or the action stored with the matching state cannot be instantiated based on the match, the robot is unable to retrieve relevant experience from the Q-value function.

Each time the robot performs an action and receives a reward, we store this experience in the instance-based Q-value function. If an exact graph match exists between the current state and a previously encountered state (graph isomorphism), we update the existing memory entry with the new experience. Otherwise, we add this experience as a new instance to the representation of the Q-value function.

Subgraph monomorphism is an NP-complete problem. However, efficient algorithms for small graphs exist [5]. Since most real-world articulated objects posses a small number of links, the theoretical computational complexity does not impose any practical limitations on our approach.

Similar to other memory-based approaches to learning, our approach may require large amounts of memory. Several methods to remedy this problem have been proposed, specifically in the context of relational reinforcement learning [6]. We believe that the consolidation of experiences based on domain-specific generalization is an important issue for future research. Ultimately, we expect to apply unification and graph matching to the obtained experience in order to generate general manipulation rules, greatly reducing the memory requirement of our instance-based representation for the Q-value function.

E. Action Selection: Balancing Exploration and Exploitation

To learn an optimal policy, the robot has to balance exploration and exploitation. Exploration refers to the execution of an action to improve the Q-value function's estimate of the associated reward. Exploitation, in contrast, refers to action selection based on maximizing reward. If the robot explores too much, it will learn slowly. If it exploits too early, it will perform poorly because it has not gathered enough experience. We complete the description of our approach by explaining how action selection during learning balances exploration and exploitation.

When selecting an action for the current state, the robot can either perform exploration by selecting a new action, or it can use its experience with previously performed actions. In the latter case, the robot again chooses between exploration and exploitation. It can either perform exploitation by choosing the most promising action based on its current estimates of Q-values, or perform exploration in an attempt to improve the current estimates of Q-values.

To decide if a new action should be executed (the first trade-off), we compute the fraction ϕ of actions for which the robot already has gathered experience. If a number drawn uniformly at random from the interval $[0, 1]$ is smaller than $e^{-\beta\phi}$, the robot performs exploration ($\beta = 2$ in our experiments). Otherwise it selects one of the actions associated with state s. The selection among those actions represents the second trade-off. We perform it using Interval Estimation (IE) [11]. Intuitively, IE picks the action that still has the highest potential to perform well. More advanced alternatives to IE guarantee polynomial bounds on the resources required to achieve near-optimal return [13].

VI. Grounding Relational Reinforcement Learning

The learning framework described in this paper is entirely symbolic. To ground this framework, we have to link updates to relational state descriptions to the perceptual capabilities of the robot and actions performed by the robot to the relational description of actions in the learning framework.

The state description is grounded using the perceptual skill described in Section IV. When the perceptual discovers a new link, observes internal motion of a link, or observes a different kinematic relationship than the one represented in the state, the relational state representation is updated. The state description is also updated when new properties of links are perceived.

An action is grounded using the set of visual features f_i in the robot's perceptual space. The relational action can be translated into a force-controlled physical action that establishes contact with the table on the appropriate side of the point cloud and then performs a compliant motion until contact with the object is made and the desired motion is observed.

The task-specific grounding of state updates and the action executions closes the loop between the physical world and the learning framework. It ensures that the learned manipulation experience is physically meaningful and can be translated back into a useful physical action.

VII. Experimental Validation

To demonstrate the effectiveness of our learning-based approach to manipulation in unstructured environments, we perform two types of experiments. First, we show that our approach permits the learning of manipulation knowledge from experience. Second, we show that the acquired experiences transfer to previously unseen objects.

A. Experimental Setup

Our experimental evaluation requires a large number of experiments. For practical reasons, we performed these experiments in a simulated environment. Due to the robustness of the perceptual skill described in Section IV and due to the simplicity of force guided pushing required for our experiments, we argue that our results remain valid in real-world experiments. Our simulation environment is based on the Open Dynamics Engine (ODE), a dynamics simulator. The simulation includes gravity, friction, and non-determinacy.

In each experiment, the robot interacts with an articulated object to extract its kinematic structure. Example objects are given in Figures 3 and 4. Revolute joints are shown as red cylinders, prismatic joints are represented by green boxes, and links are shown in blue. Currently, our approach is limited to planar objects. We also restrict our experimentation to serial chains, even though our implementation handles branching mechanisms and loops. Perceptual information about the manipulated objects is obtained from a simulation of the perceptual skill described in Section IV [12]. We do not use the simulator's internal object representation to obtain information of the object.

Each experiment consists of a sequence trials. For each trial we report the average over 10 independent experiments. A trial consists of a number of steps; in each step, the robot applies a pushing action to the the articulated object. The trial ends when an external observer signals that the obtained model accurately reflects the kinematic structure of the articulated object. The number of steps required to uncover the correct kinematic structure measures the effectiveness with which the robot accomplishes the task.

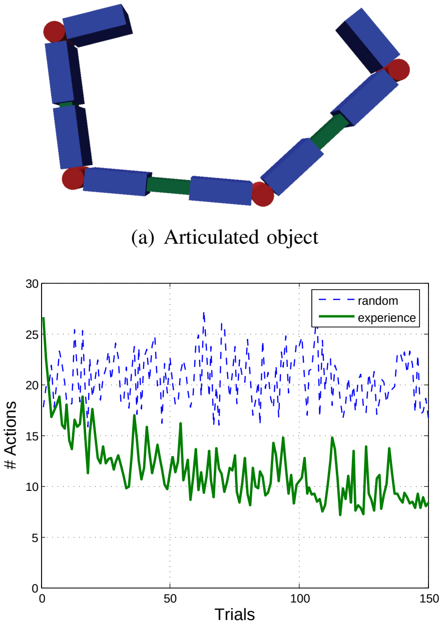
Each step of a trial can be divided into three phases. In the first phase, the robot selects an action and a link with which it wants to interact. The action is instantiated using the current state and the experience stored in the representation of the Q-value function. In the second phase, the selected action is applied to the link, and the ODE simulator generates the resulting object motion. The trajectories of the visual features tracked by the perception skill are reported to the robot. In the last phase, the robot analyzes the motion of visual features and determines the kinematic properties of the rigid bodies observed so far. These properties are then incorporated into the robot's current state representation. With each step, the robot accumulates manipulation experiences that improves its performance over time.

A trial ends when the kinematic model obtained by the robot corresponds to the structure of the articulated object. In our simulation experiments, an external supervisor issues a special reward signal to end the particular trial. Note that such a supervisor is not required for real-world experiments. The robot can decide to perform manipulation based on incomplete available information. If new kinematic information is discovered during manipulation, the robot simply updates its kinematic model and revises its manipulation strategy accordingly.

B. Learning Manipulation Knowledge

To demonstrate the ability of the proposed learning framework to acquire relevant manipulation knowledge, we observe the number of actions required to discover a kinematic structure. We compare the performance of the proposed grounded relational reinforcement learning approach to a random action selection strategy, using an object with seven degrees of freedom and eight links (Fig. 3(a)). The resulting learning curve is shown in Figure 3(b). Random action selection, as to be expected, does not improve its performance with additional trials. In contrast, action selection based on the proposed relational reinforcement learning approach results in a substantial reduction of the number of actions required to correctly identify the kinematic structure. This improvement already becomes apparent after about 20 trials. Using the learning-based strategy, an average of 8 pushing actions is required to extract the complete kinematic model, compared to the approximately 20 pushing actions required with random action selection. This corresponds to an improvement of 60%.

This first experiment demonstrates that our approach to manipulation enables robots to acquire manipulation knowledge and to apply this knowledge to improve manipulation performance.



(a) Articulated object

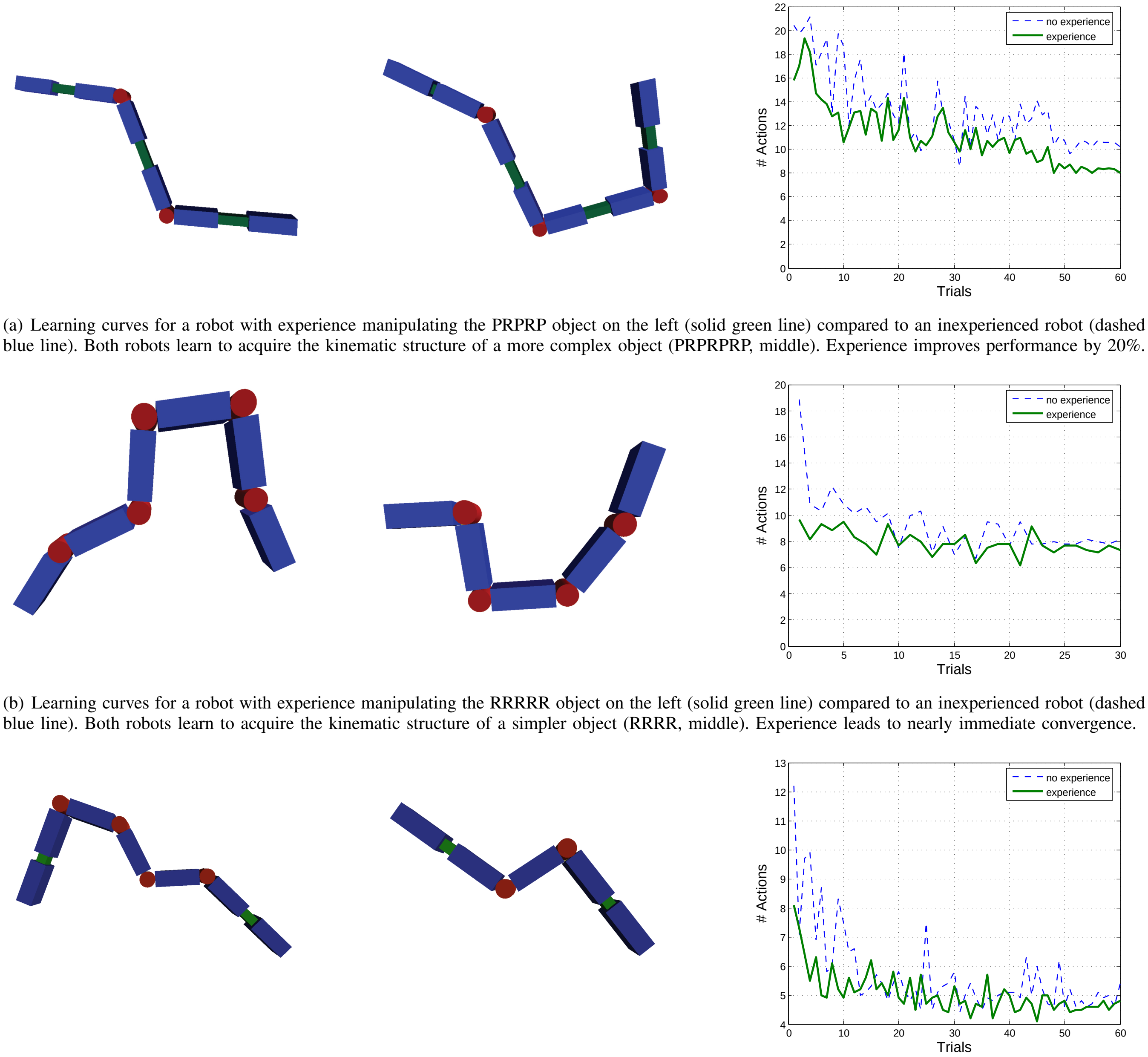
(b) Performance compared to random

Fig. 3. Experiments with a planar kinematic structure with seven degrees of freedom (RPRPRPR, R = revolute, P = prismatic). The learning curve for our learning-based approach to manipulation (green solid line) converges to eight required actions with a decreasing variance, representing an improvement of 60% over the random strategy (blue dashed line).

C. Transferring Manipulation Knowledge

To demonstrate that the manipulation experience acquired with one object transfers to other objects, we observe the number of actions required to discover a kinematic structure with and without prior experience.

In the first transfer experiment, the robot gathers experience with an articulated object with 5 degrees of freedom (see Fig. 4(a)). After 50 trials, the robot is given a more complex object with two additional degrees of freedom. The simple structure is a sub-structure of the more complex one. We compare the robot's performance with that of a robot without prior experience (see Fig. 4(a)). The robot with prior experience consistently outperforms the robot without experience. Over the first ten trials, this performance improvement is approximately 20%. In trials 10 to 40, the performance improvement is much smaller. Interestingly, as variance decreases (the robot decreases its exploration rate), the performance of the robot with experience again achieves a 20% performance



(a) Learning curves for a robot with experience manipulating the PRPRP object on the left (solid green line) compared to an inexperienced robot (dashed blue line). Both robots learn to acquire the kinematic structure of a more complex object (PRPRPRP, middle). Experience improves performance by 20%.

(b) Learning curves for a robot with experience manipulating the RRRRR object on the left (solid green line) compared to an inexperienced robot (dashed blue line). Both robots learn to acquire the kinematic structure of a simpler object (RRRR, middle). Experience leads to nearly immediate convergence.

(c) Learning curves for a robot with experience manipulating the PRRRRP object on the left (solid green line) compared to an inexperienced robot (dashed blue line). Both robots learn to acquire the kinematic structure of a simpler object (PRRP, middle). The simpler object is **not** a sub-structure of the complex object. With experience, convergence is achieved in about five trials.

Fig. 4. Experimental validation of transfer of manipulation experience between different articulated objects.

improvement. We attribute this to the fact that some useful manipulation strategies can more easily be discovered in the smaller state space of the simpler structure.

In the second experiment, the robot learns to manipulate a complex articulated object with 5 revolute joints. After 50 trials, the robot is given a slightly simpler structure that only possesses four revolute joints. Again, the simpler structure is a sub-structure of the more complex one. We compare the robot's performance after these initial 50 trials to another robot's performance without prior experience (see Fig. 4(b)). Given prior experience, the robot achieves convergence almost immediately. This corresponds to a performance improvement of about 50% in the first trial, relative to the robot without experience. After about ten trials, both robots converge to approximately the same performance, which is to be expected for simple structures that exclusively consist of revolute joints.

In the third experiment, the robot learns to manipulate an articulated object with 6 degrees of freedom (see Fig. 4(c)). After 50 trials, the robot is given a different structure that is not a substructure of the other. We compare the robot's performance after these initial 50 trials to another robot's performance without prior experience (see Fig. 4(c)). Again, experience results in a much faster convergence (after only five trials) towards about five required interactions. In addition, the variance of successive trials is reduced. After about 15 trials, both robots converge towards the same number of interactions.

Our experimental results provide strong evidence that learning from past experience can significantly improve manipulation performance. We attribute the effectiveness of our approach leverages to the proper, task-specific grounding of our relational representation.

VIII. Conclusion

We proposed a learning-based approach for manipulation in unstructured environments. We provide experimental evidence that this approach enables robots to autonomously acquire manipulation expertise by interacting with the environment. This expertise transfers across different instances of the manipulation task and substantially improves manipulation performance.

Learning and generalization of manipulation knowledge becomes possible due to a relational representation of states and actions. This representation reduces the state space and renders relational reinforcement learning tractable, even in complex manipulation domains. The power of this symbolic representation is leveraged in the real world through careful grounding of the symbols in the robot's perceptual and interactive capabilities.

We validate the proposed approach in the context of extracting kinematic models of articulated objects. This is an important enabling skill for general manipulation in unstructured environments, as all manipulation tasks require a deliberate and purposeful change in the configuration of an object and therefore knowledge of the kinematic model of an object. We demonstrate that grounded relational reinforcement learning substantially improves the robot's performance in this task. Our experiments show that appropriately grounded relational reinforcement learning is a promising approach towards endowing robots with manipulation skills adequate for unstructured environments.

Acknowledgments

We gratefully acknowledge support by the National Science Foundation (NSF) under grants CNS-0454074, IIS-0545934, CNS-0552319, CNS-0647132, and CNS-0812986. We also thank Nick Roy for insightful suggestions and discussions.

References

[1] J. D. Bernheisel and K. M. Lynch. Stable transport of assemblies by pushing. *IEEE Transactions on Robotics*, 22(4):740–750, 2006.

[2] A. Bicchi and V. Kumar. Robotics grasping and contact: A review. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, 2000.

[3] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *International Journal of Robotics Research*, 18(6):534–555, 1999.

[4] A. D. Christiansen, M. T. Mason, and T. M. Mitchell. Learning reliable manipulation strategies without initial physical models. *Robotics and Autonomous Systems*, 8(1-2):7–18, 1991.

[5] L. P. Cordella, P. Foggia, C. Sandone, and M. Vento. Performance evaluation of the VF graph matching algorithm. In *Proceedings of the 10th ICIAP IEEE Computer Society Press*, volume 2, pages 1038–1041, 1999. http://amalfi.dis.unina.it/graph/db/vflib-2.0/doc/vflib.html.

[6] K. Driessens and J. Ramon. Relational instance based regression for relational reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2003.

[7] S. Džeroski, L. de Raedt, and K. Driessens. Relational reinforcement learning. *Machine Learning*, 43(1-3):7–52, 2001.

[8] A. Edsinger and C. C. Kemp. Manipulation in human environments. In *Proceedings of the IEEE International Conference on Humanoid Robots*, 2006.

[9] R. Grupen and M. Huber. A framework for the development of robot behavior. In *Proceedings of the 2005 AAAI Spring Symposium Series: Developmental Robotics*, Stanford, USA, 2005.

[10] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[11] L. P. Kaelbling. *Learning in Embedded Systems*. MIT Press, 1993.

[12] D. Katz and O. Brock. Manipulating articulated objects with interactive perception. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, USA, 2008.

[13] M. Kearns and S. Singh. Near-optimal reinforcement learning in polynomial time. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 260–268. Morgan Kaufmann, San Francisco, CA, 1998.

[14] K. M. Lynch and M. T. Mason. Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *International Journal of Robotics Research*, 18(1):64–92, 1999.

[15] M. T. Mason. *Mechanics of Robotic Manipulation*. MIT Press, 2001.

[16] A. W. Moore. Efficient memory-based learning for robot control. Technical Report UCAM-CL-TR-209, Computer Laboratory, University of Cambridge, 1990.

[17] A. M. Okamura, N. Smaby, and M. R. Cutkosky. An overview of dexterous manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 255–262, San Franscisco, USA, 2000.

[18] R. Platt, R. Burridge, M. Diftler, J. Graf, M. Goza, E. Huber, and O. Brock. Humanoid mobile manipulation using controller refinement. In *Proceedings of the IEEE International Conference on Humanoid Robots*, Genova, Italy, December 2006.

[19] R. Platt, A. H. Fagg, and R. A. Grupen. Manipulation gaits: Sequences of grasp control tasks. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, USA, 2004.

[20] L. Sentis and O. Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robots*, 2(4):505–518, 2005.

[21] T. Siméon, J.-P. Laumonde, J. Cortés, and A. Sahbani. Manipulation planning with probabilistic roadmaps. *International Journal of Robotics Research*, 23(7-8):729–746, 2004.

[22] S. S. Srinivasan, M. A. Erdmann, and M. T. Mason. Control synthesis for dynamic contact manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2523–2528, 2005.

[23] A. Stoytchev. Behavior-grounded representation of tool affordances. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3071–3076, Barcelona, Spain, 2005.

[24] A. Sudsang and J. Ponce. In-hand manipulation: Geometry and algorithms. *Algorithmica*, 26(4):466–493, 2000.

[25] P. Tadepalli, R. Givan, and K. Driessens. Relational reinforcement learning: An overview. In *Proceedings of the Workshop on Relational Reinfocement Learning at ICML '04*, Banff, Canada, 2004.

[26] J. C. Trinkle and R. P. Paul. Planning for dexterous manipulation with sliding contacts. *International Journal of Robotics Research*, 9(3):24–48, 1990.

[27] M. van Otterlo. A survey of reinforcement learning in relational domains. Technical Report TR-CTIT-05-31, Department of Computer Science, University of Twente, The Netherlands, July 2005.

[28] S. Vijayakumar, A. D'Souza, T. Shibata, J. Conradt, and S. Schaal. Statistical learning for humaonid robots. *Autonomous Robots*, 12(1):59–72, 2002.

[29] C. J. C. H. Watkins and P. Dayan. *Q*-learning. *Machine Learning*, 8(3):279–292, 1992.