Active Exploration of Joint Dependency Structures

Johannes Kulick   Stefan Otte   Marc Toussaint

Abstract— **Being able to manipulate degrees of freedom of the environment, such as doors or drawers, is a requirement for most tasks a robot is supposed to perform. Often these external degrees of freedom depend on other ones, e.g., a drawer can only be opened if the lock is not locking the joint. We propose an approach to autonomously and efficiently explore and uncover joint dependency structures. We develop a probabilistic model for joint dependency structures which is the basis for active learning. Discontinuities in the dynamics of the joint, which often indicate key points of the joint, are used to segment the joint space into meaningful segments which then allows efficient exploration with the developed *maximum cross-entropy (MaxCE)* exploration strategy. Experiments in a simulated environment and on a real PR2 suggest that the proposed approach yields efficient exploration of joint dependency structures.**

I. Introduction

Robots acting in the real world have to manipulate various joints to achieve their goals – they manipulate the *environment's degrees of freedom*. Opening doors and drawers, turning keys, and pushing buttons are some examples of typical manipulations of the environment's degrees of freedom. In the spirit of the *physical exploration challenge* [11] we want to enable the robot to explore the environment and reduce uncertainty over the properties of the world.

Whereas existing work deals with (a) handling and controlling known mechanisms [10], [12], [7], (b) estimating joint types and parameters [16], [9] from given data, (c) autonomous exploration to estimate joint types and parameters [11], or (d) autonomous exploration to distinguish between pre-defined models [1], in this paper we focus on autonomous exploration of mechanisms with complex joint dependency structures, i.e., mechanisms where certain parts can only be articulated if the joints are in a specific configuration. Specifically, in this paper we model the dependency between joints, e.g., we can only open the drawer if the key unlocked it, and we take the control and the parameter estimation of mechanisms as given.

The difficulty of this task lies in the complexity of the continuous combinatorial space of dependent mechanisms. Each joint can potentially lock another one at each position. We overcome this problem by using the following insight. Most complex mechanisms are designed to be used by humans and thus give feedback of various kinds (e.g., sound, light, force feedback by rasters or joint limits) to signal the key points in the joint configuration space. These key points often unlock/lock other joints. We design a probabilistic graphical model capturing the above insight and develop efficient active learning strategies enabling robots to autonomously explore and uncover complex joint structures.

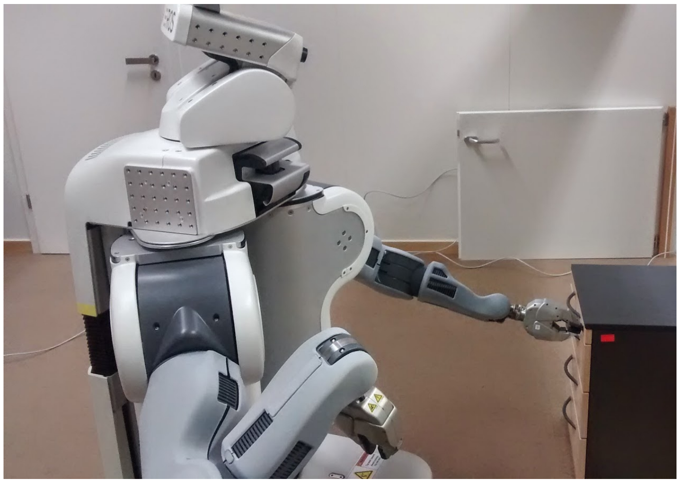


Fig. 1. Real world experiment: our PR2 is exploring the joint dependency structure of a cabinet. After only two observations, the robot learned that in order to open the drawer, it must first unlock it by turning the key (compare to Fig. 6).

The main contributions of this paper are:

- We define a probabilistic model of joint dependency structures which is the basis for active learning (Sec. III).
- The model captures the fact that discontinuities in the dynamics define crucial states of the joint. These states allow the segmentation of the continuous joint state space into discrete segments enabling efficient inference and active learning strategies for exploration.
- Given the probabilistic model, we develop an efficient active learning exploration strategy: *maximum cross-entropy (MaxCE)* is a method which efficiently reduces the entropy of the belief in the long term and avoids choosing sample points which would confirm the wrong hypothesis (Sec. III-C).
- We demonstrate that our PR2, which uses the probabilistic model and the MaxCE strategy, can learn the joint dependency structure of a cabinet (Sec. IV-D).

II. Related Work

Autonomous exploration of real-world environments recently gained the attention of many researchers. In our previous work [11] we introduced the *physical exploration challenge* and demonstrated how different exploration strategies

All authors are with the Machine Learning and Robotics lab at University of Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany. johannes.kulick@ipvs.uni-stuttgart.de stefan.otte@ipvs.uni-stuttgart.de marc.toussaint@ipvs.uni-stuttgart.de

can be applied to infer the type and parameters of degrees of freedom of the environment. Kaelbling and Lozano-Pérez [5] use probabilistic modeling of such environments to plan actions. This laid the foundation for Barragan et al.'s work [1] where the agent identifies different mechanical joint types from interaction. However, these joints only can only have a pre-defined dependency structure and thus new structures cannot be explored.

To model the joints and their parameters we must perceive the objects and joints. In the field of interactive perception Van Hoof et al. [17] generate actions based on a probabilistic segmentation of cluttered scenes. They update their model after performing the action and observing the corresponding reaction of the objects. The work of Martin and Brock [9] and Katz et al. [6] is in a similar spirit. They model the movement of joints in a hierarchical manner and identify the joint type, state, and parameters based on tracking of features in RGB-D data. However, the robot's actions are scripted. They also do not consider dependencies between joints. Sturm et al.'s work [15] also does not consider joint dependencies. They identify joints from object trajectories and generate actions to actuate those objects [16].

Höfer et al. [4] consider joint dependencies as relations in a relational reinforcement learning scenario. The actions in this work are symbolic and do not consider the actual position of a joint. The dependencies learned are also in relation to the actions performed and not the current sub-symbolic state of the environment.

To explore the joint dependency structure, strategies are needed. Active learning is a field of research which deals with the generation of efficient exploration strategies. A good overview of the field is given by Settles [14]. For probabilistic models, such as our joint dependency model, strategies derived from information theory are particularly useful. Bayesian experimental design [2], in particular, is a promising strategy. In [8] we showed that this strategy might get stuck in situations where the prior might be misleading. Thus, this paper uses a slightly different strategy.

To infer where force feedback might be given, we use Bayesian change point detection as described by Fearnhead [3].

III. Probabilistic Modeling of Joint Dependency Structures

A. Notions of Joint Dependency Structures

In this paper we investigate joint dependency structures. These are structures where the state of one joint depends on the state of another joint. To study these structures we focus on worlds consisting of rigid bodies only. Rigid bodies may be connected through joints and form kinematic graphs. Between any two rigid bodies in the world there can be a joint constraining the movement of the two bodies. Additionally, each joint can be *locked* or *unlocked*. We call this the *locking state* of the joint. If a joint is *locked* no movement is possible (e.g., if the door handle is not turned the door cannot be opened), if a joint is *unlocked* movement within the normal constraints of the joint (like axis limits or friction) is possible.

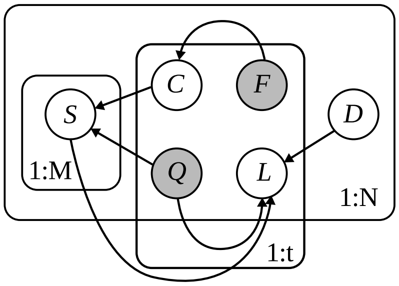


Fig. 2. The probabilistic graphical model for joint dependency structures. Tab. I explains the random variables. Arrows leaving a plate and entering it again denote dependencies from all source RVs to all target RVs.

Symbol	**Description**	**Domain**
N	Number of joints	$\mathbb{N}$
M^j	Maximum joint angle of joint j	$\mathbb{R}$
t, s, u, v	Index for time	$\mathbb{N}$
j	Index for joints	$\{1, \ldots, N\}$
D^j	RV, dependency of joint j	$\{1, \ldots, N+1\}$
L_t^j	RV, locking state of joint j	$\{0, 1\}$
Q_t^j	RV, joint state/position of joint j at time t	$\mathbb{R}$
F_t^j	RV, force/torque measurements of joint j at time t	$\mathbb{R}$
C_t^j	RV, change points of joint j at time t	$\{0, 1\}$
S_p^j	RV, segment borders of joint j at position p	$\{0, 1\}$

TABLE I

Summary of used symbols.

The locking state of one joint can change depending on the state of other joints. We call the joint whose position determines the locking state of another joint the *master* and the joint which locking state depends on the master's position the *slave*. We can divide the master's joint configuration into segments which put the slave into the *locked* or *unlocked* locking state.

With arbitrary possible dependencies between any two joints in the world, the search space for dependencies grows exponentially and can not be searched efficiently even for a small number of joints. This is especially problematic if a real robot is to uncover the joint dependency structure of real world mechanisms. We overcome this problem by using the sensor clues the mechanisms offer. During the manipulation of a joint we measure its force/torque (F/T) feedback. Change points in the F/T measurements should indicate the borders between the locking state segments, e.g., in one segment the master locks the slave, in other segments the master does not lock the slave.

B. Modeling the Joint Dependency Structure

The robot has to infer the joint dependency structure from its actions and the F/T measurements. For this we model the joint dependency structure as a graphical model depicted in Fig. 2. We introduce random variables $D^{1:N}$,

$L_{1:t}^{1:N}$, $Q_{1:t}^{1:N}$, $F_{1:t}^{1:N}$, $C_{1:t}^{1:N}$, and $S_M^{1:N}$, summarized in Tab. I, which we explain in detail below. N is the number of joints to be modeled and t is the time index for the time dependent random variables. M^j is the maximum reachable joint position of joint j and we assume without loss of generality that the minimum joint position is 0.

D^j is a discrete random variable with the domain $\{1, \ldots, N+1\}$. $N-1$ states indicate which other joints joint j depends on. The $(N+1)$-th state indicates if joint j is independent of all other joints. This choice of D^j limits our model to one-to-one joint dependencies. It is easy to extend this to more complex dependencies by extending D^j. That, however, enlarges the space of possible dependency structures significantly.

L_t^j is the locking state of joint j at time step t. It is a binary variable stating whether joint j is *locked* or *unlocked*.

Q_t^j is the joint state of joint j at time t, i.e., the angle for rotational joints and the prismatic extension for prismatic joints.

F_t^j are potentially pre-processed force/torque (F/T) sensor measurements observed at time t for joint j. The measurements can be mapped to a particular joint since the agent knows which joint he is actuating. In this paper only F/T measurements are used but different modalities, such as sound, could be incorporated in a similar fashion.

C_t^j is a binary variable that states whether at time t a change point in the F/T measurements was detected. Again, this can be mapped to a joint since the agent knows which joint it is actuating.

S_q^j is a binary variable stating whether there is a segment border at position q in joint state space of joint j. Since the joint space is a continuous space, S^j is strictly speaking a random field over the joint space. However, we simplified this by finely discretizing the joint space in our implementation such that S^j becomes a set of random variables.

The agent observes the joint state $Q_t^{1:N}$ at a given time t and the F/T measurements F_t^j. To query the locking state L_t^j an oracle can be asked. Alternatively an action could be performed (e.g., pushing, pulling or rotating the joint) to compute the locking state from the observations.

C. MaxCE Exploration Strategy

We formalize the goal to uncover the joint dependency structure as to minimize the uncertainty over D^j. This can be achieved by minimizing the entropy over D^j. We would have to solve a POMDP to optimally minimize the entropy of a random variable. As that includes computing the expectation over all possible futures, it is computationally very hard. To avoid solving this NP-complete problem, one-step lookahead objective functions are mostly used. Typically, the expected entropy over possible outcomes of one step actions is used. However, as we have shown in [8], minimizing the expected one-step entropy can be misleading whe the aim is active learning about latent model parameters, especially when the prior distribution is biased to a wrong state. This is the case in this work because we assume that most joints are not dependent on others (see Sec. IV for more details).

Thus, we leverage the *maximum cross-entropy (MaxCE)* criterion to minimize the uncertainty over D^j. This criterion has the benefit that it does not try to strengthen a wrongly biased belief but it values change in the posterior independent of the actual direction of change. It does this by measuring the change in the model distribution – in our case this is the distribution over possible joint dependency structures.

For this purpose we compute the expected one-step cross-entropy between the current joint dependency structure distribution $P_{D_t^j}$ and the expected joint dependency structure distribution one time step ahead $P_{D_{t+1}^j}$ (also called the augmented posterior), and maximize this expectation to get the optimal next sample position $Q_{t+1}^{1:N^*}$:

$$(Q_{t+1}^{1:N^*}, j) = \underset{(Q_{t+1}^{1:N}, j)}{\operatorname{argmax}} \sum_{L_{t+1}^j} P\left(L_{t+1}^j | Q_{t+1}^{1:N}, S^{1:N}\right) \cdot H\left[P_{D_t^j}; P_{D_{t+1}^j}\right] \tag{1}$$

with

$$P_{D_t^j} = P(D^j | L_{1:t}^j, Q_{1:t}^{1:N}, S^{1:N}) \tag{2}$$

$$P_{D_{t+1}^j} = P(D^j | L_{1:t+1}^j, Q_{1:t+1}^{1:N}, S^{1:N}). \tag{3}$$

$H[\cdot;\cdot]$ is the cross-entropy between two probability distributions, defined as

$$H[P(X); Q(X)] = -\sum_X P(X) \log\left(Q(X)\right). \tag{4}$$

D. Marginal Data Likelihood

The posterior of the joint dependency structure is dependent on the marginal data likelihood of the queries acquired $P(L_{t+1}^j | D^j, Q_{t+1}^{1:N}, S^{1:N})$. Since the locking state is a discrete variable, the usual choice for its probability is a Dirichlet prior. The marginal likelihood is then:

$$P(L_{t+1}^j | D^j, Q_{t+1}^{1:N}, S^{1:N}) = \frac{\Gamma(\sum_k \alpha_k)}{\Gamma(O + \sum_k \alpha_k)} \prod_{k=1}^{K} \frac{\Gamma(c_k + \alpha_k)}{\Gamma(\alpha_k)} \tag{5}$$

Where O is the number of observations, K is the cardinality of L_{t+1}^j, α_k are the Dirichlet hyper-parameters, c_k are the observation counts (i.e., counts for each locking state), and Γ is the multinomial gamma function.

Normally the observation counts are just counts of how often each locking state occurred in the queries acquired so far. We have, however, to account for the fact that the locking state changes over time. We thus weight each query by the probability that it was acquired in the same segment, i.e., for a query acquired at time step i the probability that there is no segment border between Q_i^k and Q_{t+1}^k. How we can compute this will be subject of Sec. III-E.

E. Change Points

As described in the last section we require the probability of a state change between two points in joint space. While we could use typical distance functions like radial basis functions, we leverage the knowledge that state changes can often be inferred from feedback. To model this we use the F/T sensor measurements to generate a probability distribution over state changes.

We assume that the state changes are independent from each other. The probability of no state change between two positions Q_u^j and Q_v^j at time u and v of joint j is then

$$\prod_{q=Q_u^j}^{Q_v^j} \left(1 - P\left(S_q^j | Q_{1:t}^j, C_{1:t}^j, F_{1:t}^j\right)\right). \quad (6)$$

As depicted in the graphical model, S_q^j is dependent on $Q_{1:t}^j$ and $C_{1:t}^j$. We model this dependency as the probability of a segment boarder being the mean probability of a change point at a particular joint position. Thus we can infer S_q^j by

$$P\left(S_q^j | Q_{1:t}^j, C_{1:t}^j, F_{1:t}^j\right) = \frac{\sum_{s=1}^{t} \delta(Q_s^j, q) P(C_s^j | F_{1:t}^j)}{\sum_{s=1}^{t} \delta(Q_s^j, q)} \quad (7)$$

with $\delta(\cdot, \cdot)$ the Kroenecker delta

$$\delta(a, b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ else} \end{cases} . \quad (8)$$

To infer the change point probability from the sensor measurements we leverage Bayesian change point detection. For the sake of brevity it is sufficient to know that it computes the posterior $P(C_s^j | F_{1:t}^j)$ – the probability of a change point at a time s given the sensor input. Although the sensor input can in principle be arbitrary, we can compute the change point probabilities analytically if we assume piecewise constant data. We will leverage this knowledge later when dealing with the F/T data of the robot. For a detailed description of the change point detection refer to [13].

IV. Experiments

A. General Setup

We test our dependency model in a physical simulation and on a real robot. Here we describe the parts of the experiments that are shared. The details specific to one experiment are described in the corresponding sections. The goal of the robot in both experiments is to uncover the joint dependency structure D of complex mechanisms.

1) Assumptions: We assume the agent initially has akinematic model of the environment, including the existance of joints and their parameters, but not their dependencies.

For simplicity we also assume the F/T profile of each joint is known, i.e., the F/T recording of each joint of the entire joint space. This can and eventually should be recorded during the exploration.

2) Actions: The result of Eq. 1 is the next query point, i.e., the desired target configurations of all joints $Q_{t+1}^{1:N*}$ and the joint j of which the locking state is supposed to be checked. Given the model of the mechanism, a controller can bring the mechanism to the desired configuration if possible. The locking state of joint j is supplied via the simulation or via a human oracle respectively.

3) Prior: For the probabilistic graphical model we have to choose priors. We want to incorporate the knowledge that most joints in the world are independent of other joints, e.g., most drawers and cabinet doors can be opened directly without unlocking them and they do not lock each other. Therefore, for the dependency prior $P(D)$ we set the probability of a joint being independent to 0.7. We also think that proximity is a good indicator of the dependency of joints. Joints that are close to each other are more likely to depend on each other than joints that are far apart. For the dependency prior $P(D)$ we set the probability of being dependent on another joint proportional to $\frac{1}{d(i,j)}$, with $d(i, j)$ being the Euclidean distance between joint i and j. The probability of a joint locking itself is set to zero. More formally:

$$P(D^j = i) = \begin{cases} 0 & \text{if } j = i \quad \text{(self-dependent)} \\ .7 & \text{if } j = N + 1 \quad \text{(independent)} \\ \frac{1}{d(i,j)c_N} & \text{else} \quad (j \text{ depends on } i) \end{cases} \quad (9)$$

with c_N being a normalization constant.

4) Data Pre-processing: The second graph in Fig. 3 shows the F/T sensor reading applied to a joint when controlling it, which is a non-linear time series. To efficiently compute the change point probabilities we, assume a piecewise constant model plus Gaussian noise. We pre-process the F/T as follows, to account for that assumption.

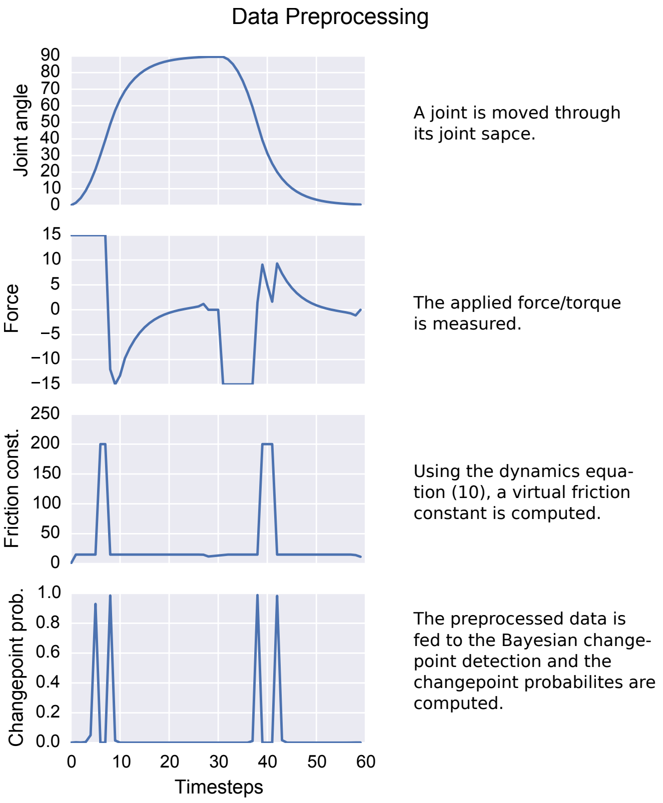
The force we measure is the force needed to induce a certain velocity change on the joint, which can be calculated using physical laws as

$$\Delta v = \sqrt{v^2 - c_f v \Delta t}, \quad (10)$$

with v the velocity, Δt the change in time and c_f a virtual friction constant incorporating the mass of the object, the contact force and actual friction coefficient. From the actual F/T measurements we can thus compute the virtual friction constant which changes if there is feedback in the joint mechanism. As can be seen in the third graph of Fig. 3, the virtual friction constant is piecewise constant up to sensor noise. We can feed it to the Bayesian change point detection and compute change point probabilities (fourth graph of Fig. 3).

We map the change point probabilities, which are in time space (bottom plot in Fig. 3), into joint space and get the probabilities over segment borders (top plot in Fig. 4). With this data we can then start the actual exploration of the mechanism.

5) The Exploration Sequence: The data pre-processing yields the segment border probabilities as depicted in the first graph of Fig. 4. From the segment border probabilities and all observation data we can compute the next query point/action (see Sec. III and Sec. IV-A.2). Then, the robot performs the given action, i.e., it brings all joints into the goal configuration $Q_{t+1}^{1:N*}$ and checks the locking state of joint j. The grapical model is updated with the resulting



A joint is moved through its joint sapce.

The applied force/torque is measured.

Using the dynamics equation (10), a virtual friction constant is computed.

The preprocessed data is fed to the Bayesian changepoint detection and the changepoint probabilites are computed.

Fig. 3. Data gathered from exploring a (simulated) cupboard, which has a key that locks the door between 20 and 50 degrees. All graphs show data from the key. The pre-processing of the F/T data shown here is done once once before the exploration.

observation and the process is repeated. Fig. 4 shows some exploration steps of this procedure.

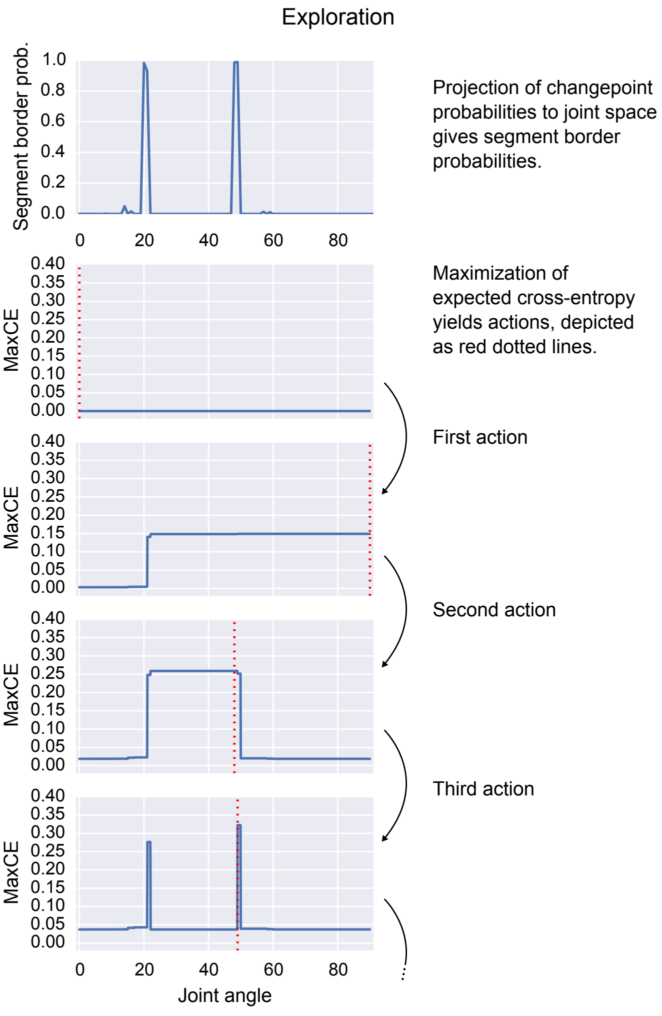
B. Physical Simulation: Setup

We first test our method in a physical simulation. We simulate the dynamics of the joints to determine change points in the movement profile. The agent applies forces to the joints to move them to desired positions using a PD-controller. The agent senses the applied forces and the position of the joint and can use this information to infer the change points and segments.

We test three different strategies to uncover joint dependency structures: **(a)** the *baseline strategy* selects actions randomly, **(b)** the *expected entropy strategy* minimizes the expected entropy of the distribution of D^j, and **(c)** the *MaxCE strategy* maximizes the cross-entropy as described in [8].

Additionally, we test the influence of the change point detection on the overall performance. We test all the strategies with **(a)** the change point detection enabled and **(b)** using a common exponential distance function as likelihood of two experiences belonging to the same joint state instead of using the segments detected by the change point detection.

Each strategy is evaluated in 50 different environments. An environment consists of furniture with different joint dependency structures and different parameters of the locking state. An environment is generated by randomly choosing



Projection of changepoint probabilities to joint space gives segment border probabilities.

Maximization of expected cross-entropy yields actions, depicted as red dotted lines.

First action

Second action

Third action

Fig. 4. The exploration sequence of the same simulated cupboard as depicted in Fig. 3. Again, only data from the key is shown. Each action moves the key to the desired configuration shown in red. After each action the robot also observes the locking state of one joint. In this particular example it explores the key of the cupboard and always asks for the locking state of the door. Note that in the first step no observations are present and thus all actions are equally interesting. Tiebreaking is done randomly. After three actions, the probability of the key being the master of the door is already greater than 70% (not shown).

three different objects from Tab. II and randomly choosing the locking state. One instance of an environment might consist of a cupboard with a key (which is unlocked if the key is, e.g., between 73 and 93 degree), a cupboard with a handle (which opens, e.g., at its upper limit), and a drawer with a key (which is unlocked if the key is, e.g., between 122 and 142 degree). In an exploration sequence the agent can perform 30 actions.

C. Physical Simulation: Results and Discussion

Fig. 5 shows the performances of the different strategies with and without change point detection. The upper plot shows the average correctly classified joint dependencies over time. We define correctly classified as recognizing the correct joint dependency with a probability of ≥ 0.5. Note that due to the prior, three dependencies are always classified

Name	**Description**	**Locking mechanism**
Cupboard with handle	A cupboard with a door and a handle attached to it.	The handle must be at upper or lower limit to unlock the door
Cupboard with lock	A cupboard with a door and a key in a lock	The key must be in a particular position to unlock the door
Drawer with handle	A drawer with a movable handle	The handle must be at upper or lower limit to unlock the drawer
Drawer with lock	A drawer with a key in a lock	The key must be in a particular position to unlock the drawer

TABLE II
FURNITURE USED IN THE SIMULATION.

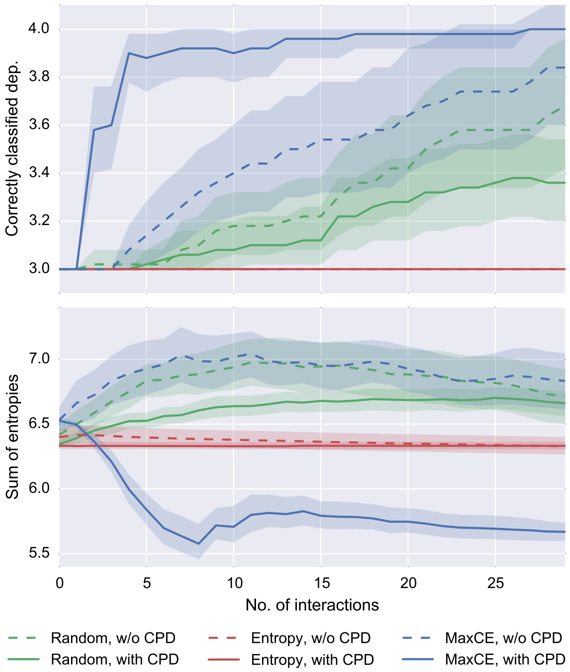


Fig. 5. Results of the simulation experiment. We show the sum of entropies of all D^j random variables. The error bars reflect a 99% confidence interval of the mean estimator.

correctly. Here one can see that the MaxCE method is able to classify significantly more dependencies correctly as well as faster when compared to all other strategies. One can also see that the change point detection increases the performance of the strategies. Note that the entropy method is not able to find out anything apart from the three independent assumptions, which are already correct due to the prior. The entropy is already quite low and cannot easily be reduced. Querying configurations that lower the independence belief would increase the entropy which is undesired although it would improve the belief in the long run.

In the lower graph we show the sum of entropies of all D^j over the actions. Different things can be seen in this plot. First we can see that the MaxCE strategy with change point detection is the only strategy which is able to lower the entropies significantly. All other strategies raise the entropy. This is due to the strong prior that joints are independent of each other. Thus lowering the belief of independence raises the entropy.

Note that the strategies that use the change point detection reduce the uncertainty faster than their counterparts without change point detection. This is due to the fact that an observation yields information for the entire segment (when using change point detection). The random strategy shows this effect very clearly. While the change points have no influence on the action selection process, the inference over the dependency structure can use them to weight the observations. Therefore, the entropy of the random strategy with change point detection is lower than the random strategy without change point detection.

D. Real World Experiment: Setup

We also test our dependency model using a real PR2 and a typical office cabinet. The cabinet has a drawer, which can be locked/unlocked by a key. The key also works as handle to open the drawer once the drawer is unlocked. Again, the robot knows the model of the cabinet (see Sec IV-A.1). The configuration of the furniture is measured through proprioception of the robot. The robot's wrist can measure the joint angle of the key. The extension of the arm corresponds to the joint value/extension of the drawer. We manually position the gripper of the robot around the key. No re-grasping is required during the experiment. The MaxCE strategy is used to explore the cabinet. Fig 1 shows our PR2 exploring the joint dependency structure of a cabinet drawer.

As additional sensory clues the robot uses F/T feedback of the drawer, measured by a sensor located in the wrists of the PR2. The measurements consist of the three dimensional force vector and three dimensional torque vector acting on the wrist of the robot. We sum up these values and preprocess them as stated in Sec. IV-A.4.

E. Real World Experiment: Results and Discussion

Fig. 6 shows the results of the conducted experiment. The robot was able to uncover the dependency of the drawer from the key position. Looking at the exploration sequence in more detail, we see that the robot was not able to increase the probability that the key is an independent joint. This results from the fact that there are no direct observations which imply independence and therefore would increase the probability for the joint being independent. During the experiment we only observed that the key was in the "unlocked" locking state, but this does not increase the probability of the key being an independent joint. Note that being unlocked either means that no master was found, or that the joint

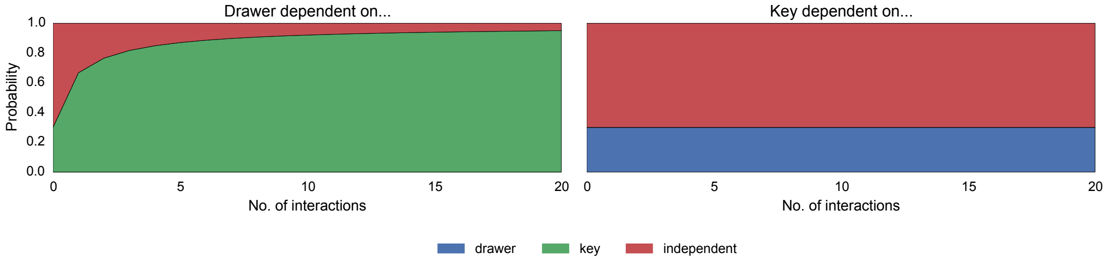


Fig. 6. The results of the real world experiment. We show the probability distribution of the two involved D^j random variables over the number of interactions the robot made. We can see that the dependency of the key as master over the slave drawer is uncovered very early, whereas the probability of the independence of the key is not significantly raised during the interactions. The self-dependency is zero throughout the whole experiment as it has a hard zero prior.

is independent. To guarantee independence we would have to observe the "unlocked" state for the entire configuration space. This could only be accomplished by exhaustively searching the configuration space.

V. Conclusion

We developed an active learning method to explore complex joint dependency structures. The method leveraged the sensory clues from the mechanisms (F/T measurements were used) to segment the joint space into meaningful discrete clusters. The MaxCE strategy proved to be an efficient strategy to explore joint dependency structures whereas the common active learning strategy, expected entropy, did not succeed. We demonstrated our method in simulation and on a real PR2.

One major limitation of our approach is the fact that only one-to-one dependencies can be modeled, since the D^j random variable only gives the dependency to exactly one other joint. While this could easily be extended to multiple dependencies, the exploration strategies would have to cope with $O(N^2)$ different models for two dependencies of each joint up to exponentially many models for arbitrary dependencies. Arbitrary joint dependencies are, however, rare. An according prior could reduce the amount of models drastically.

We are planning to integrate our method into the current implementation of the physical exploration challenge [11] to make the system more general.

Acknowledgments

This work is partially funded by the DFG (German Science Foundation) within Priority Programme 1527, "Autonomous Learning". We would like to thank Robert Lieck for help with the physical model of movement and friction, and the anonymous reviewers for the helpful feedback.

References

[1] P. R. Barragán, L. P. Kaelbling, and T. Lozano-Pérez. Interactive bayesian identification of kinematic mechanisms. In *Proc. of the IEEE Int. Conf. on Robotics & Automation(ICRA 2014)*, 2014.

[2] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[3] P. Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16.2:203–213, 2006.

[4] S. Höfer, T. Lang, and O. Brock. Extracting kinematic background knowledge from interactions using task-sensitive relational learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[5] L. P. Kaelbling and T. Lozano-Pérez. Integrated task and motion planning in belief space. *Int. Journal of Robotics Research*, 32, 2013.

[6] D. Katz, A. Orthey, and O. Brock. Interactive perception of articulated objects. In O. Khatib and G. Sukhatme V. Kumar, editors, *Experimental Robotics*, volume 79 of *Springer Tracts in Advanced Robotics*, pages 301–315. Springer, 2014.

[7] E. Klingbeil, A. Saxena, and A. Ng. Learning to open new doors. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[8] J. Kulick, R. Lieck, and M. Toussaint. Active learning of hyperparameters: An expected cross entropy criterion for active model selection. *ArXiv e-prints*, September 2014.

[9] R. Martin-Martin and O. Brock. Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In *Proc. of the Int. Conf. on Intelligent Robots and SystemsIROS 2014)*, 2014.

[10] K. Nagatani and S. Yuta. An experiment on opening-door-behavior by an autonomous mobile robot with a manipulator. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 2, pages 45–50, 1995.

[11] S. Otte, J. Kulick, M. Toussaint, and O. Brock. Entropy based strategies for physical exploration of the environment's degrees of freedom. In *Proc. of the Int. Conf. on Intelligent Robots and Systems (IROS 2014)*, 2014.

[12] L. Peterson, D. Austin, and D. Kragic. High-level control of a mobile manipulator for door opening. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, volume 3, pages 2333–2338 vol.3, 2000.

[13] R. Prescott Adams and D. J. C. MacKay. Bayesian online changepoint detection. *ArXiv e-prints*, October 2007.

[14] B. Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2010.

[15] J. Sturm, C. Plagemann, and W. Burgard. Body schema learning for robotic manipulators from visual self-perception. *Special Issue of the Journal of Physiology*, 2009.

[16] J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research (JAIR)*, 41(2):477–526, 2011.

[17] H. van Hoof, O Kroemer, and J. Peters. Probabilistic segmentation and targeted exploration of objects in cluttered environments. (5):1198–1209, 2014.