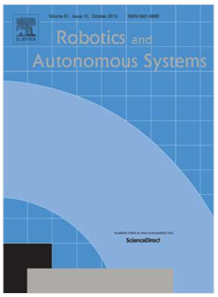
Accepted Manuscript



Cognition-enabled robotic wiping: Representation, planning, execution, and interpretation

Daniel Leidner, Georg Bartels, Wissam Bejjani, Alin Albu-Schäffer, Michael Beetz

PII: S0921-8890(18)30331-2
DOI: https://doi.org/10.1016/j.robot.2018.11.018
Reference: ROBOT 3123

To appear in: *Robotics and Autonomous Systems*

Please cite this article as: D. Leidner, G. Bartels, W. Bejjani et al., Cognition-enabled robotic wiping: Representation, planning, execution, and interpretation, *Robotics and Autonomous Systems* (2018), https://doi.org/10.1016/j.robot.2018.11.018

Cognition-Enabled Robotic Wiping: Representation, Planning, Execution, and Interpretation

Daniel Leidner[a,*], Georg Bartels[b], Wissam Bejjani[c], Alin Albu-Schäffer[a], Michael Beetz[b]

[a]*Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany*
[b]*Institute for Artificial Intelligence, University of Bremen, 28359 Bremen, Germany*
[c]*SensibleRobots research group in the School of Computing at the University of Leeds, Leeds LS2 9JT, England*
[d]*Chair for Sensor-Based Robotic Systems and Intelligent Assistance Systems, Technische Universität München, 85748 Garching, Germany*

Abstract

Advanced cognitive capabilities enable humans to solve even complex tasks by representing and processing internal models of manipulation actions and their effects. Consequently, humans are able to plan the effect of their motions before execution and validate the performance afterwards. In this work, we derive an analog approach for robotic wiping actions which are fundamental for some of the most frequent household chores, including vacuuming the floor, sweeping dust, and cleaning windows. We describe wiping actions and their effects based on a qualitative particle distribution model. This representation enables a robot to plan goal-oriented wiping motions for the prototypical wiping actions of absorbing, collecting and skimming. The particle representation is utilized to simulate the task outcome before execution and infer the real performance afterwards based on haptic perception. This way, the robot is able to estimate the task performance and schedule additional motions if necessary. We evaluate our methods in simulated scenarios, as well as in real experiments with the humanoid service robot Rollin' Justin.

Keywords: AI Reasoning Methods, Action and Effect Representation, Compliant Manipulation, Service Robotics.

1. Introduction

Advanced cognitive capabilities will enable future service robots to master everyday household chores. These robots need to represent, plan, execute, and interpret the required actions and their effects to the environment. Research on neuro-biology, as discussed by Kawato [1], suggests that humans maintain detailed internal models for manipulation tasks which can be accessed and improved to achieve high dexterity for almost every activity of daily living including the most frequent household chores of cooking, organizing and cleaning. For example, in order to clean a desk with a dust-cloth, a human would intuitively wipe along the entire target surface with the tool in order to cover all dusty areas. Fundamental to the skillful task execution is a suitable

*Corresponding author: daniel.leidner@dlr.de

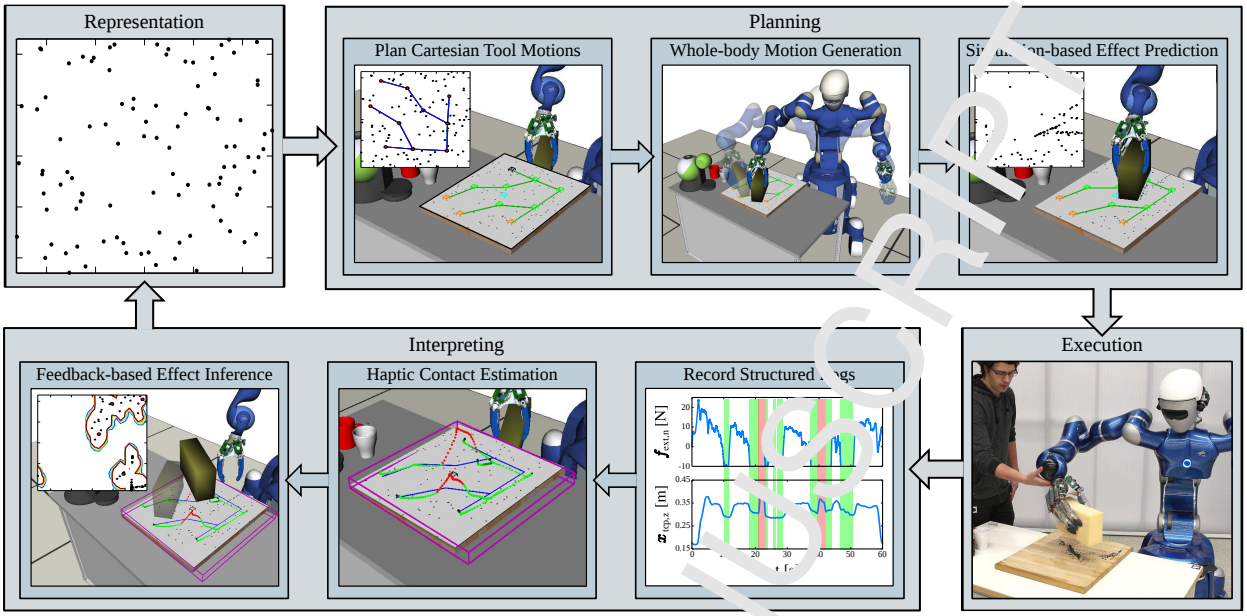


Figure 1: Overview of the proposed framework. A particle distribution model serves as the basis for the proposed motion planning algorithm as well as for the haptic effect inference method. The methods complement each other such that the output of the effect inference can be utilized to plan recovery motions in case of detecting performance errors caused by external disturbances, e. g. collisions with humans as visualized.

representation of wiping motions and their effects, i. e. the knowledge that dust is absorbed by the cleaning cloth and the assumption that dust particles are often equally distributed on planar surfaces. Humans are able to infer that the desired effect (i. e. having the desk cleaned from dust) must be accomplished after the tool has been in physical contact with the whole region of interest. However, a human may schedule additional motions in case of poor outcome. Different internal models may apply to different wiping situations. For example, if breadcrumbs are recognized on the desk the human would try to skim the particles deliberately off the table and into its hand to dispose them afterwards.

Haptic perception is often an important and reliable source of feedback during the execution of the described wiping motions. Even though visual perception is commonly considered the prime sense in human manipulation, visual feedback can be unreliable for many wiping tasks including vacuuming the floor, dusting surfaces, and window wiping, as small dirt particles, dust, and streaks of water are often hard to perceive visually. Especially in the absence of vision it has been observed that the sense of touch is essential for human task reasoning and effect inference [2]. Flanagan *et al.* [3] highlight that haptic feedback does not only provide humans with the information that contact occurred with the environment, but moreover provides the basis for effect inference, task performance ratings, and even the detection of performance errors. This is done based on comparison of the expected contact force w. r. t. the internal task model with the actual sensed force. In case of irregular contact (i. e. introduced by friction or uneven areas) humans may decide to replan additional wiping motions to improve the cleaning result accordingly. In cognitive science, this behavior is associated with the *cognitive control loop* [4], which builds on the hypothesis that humans primarily take conscious action for error correction and novel tasks. This work aims to

develop similar cognitive reasoning capabilities for robots to qualitatively reason about the effects of their motions and solve even complex wiping tasks despite poor visual feedback.

In this article, we combine artificial intelligence (AI) reasoning methods and compliant robotic manipulation to solve wiping tasks of different kind with the humanoid service robot Rollin' Justin [5]. Our contributions extend on our research on manipulation planning for wiping tasks presented in [6] and the interpretation of wiping motions presented in [7], which in combination aim on *representing*, *planing*, *executing*, and *interpreting* robotic wiping actions as it is visualized in Fig. 1.

The contributions of this work include (i) a qualitative particle distribution model to represent the effects of wiping actions, (ii) an approach to generate whole-body wiping motions based on these representations utilizing effect-oriented policies, and (iii) an approach to assess the quality of wiping motions by estimating contact situations during task execution using haptic perception. This enables (iv) the inference of performance errors and the subsequent generation of recovery wiping motions as well as (v) the semantic interpretation of contact situations by means of annotated structured logs.

The remainder of this article is structured as follows. The state-of-the-art is covered in Sec. 2. Sec. 3 sescribes representations and related planning methods to generate semantically meaningful whole-body wiping motions. We describe our effect inference strategy to estimate the task performance of real world wiping motions based on haptic feedback in Sec. 4. We conclude with Sec. 5.

2. Related Work

Cakmak *et al.* [8] classify manipulation actions that characterize human household chores w. r. t. semantic similarities. They found out that almost half of all household chores are related to wiping of surfaces in rooms, furniture, or other objects. In fact, wiping actions are substantial for cleaning tasks, such as dusting furniture with a feather duster, sweeping breadcrumbs from the kitchen countertop, or collecting shards of a broken mug with a broom. Accordingly, cleaning related wiping tasks have been investigated to some extent lately.

The task of wiping a surface is often considered as a *coverage path planning* problem [9], where a robot has to find a path (i. e. for a cleaning device) connecting all nodes of a graph in a time- or effort-optimal way. Hess *et al.* conducted research on robotic cleaning in a series of papers [10, 11]. They investigate the path coverage problem for robotic manipulators. In [10] they describe an approach to autonomously compute cleaning trajectories for redundant robotic manipulators guiding a sponge on 3d surfaces. They utilize a variation of the *Traveling Salesman Problem (TSP)* and resolve the joint motions of the robotic manipulator by means of null-space optimization. In [11] they learn the effect of a vacuum cleaner moving along a planar surface by utilizing visual feedback based on color segmentation. The robot can enhance the task execution for future trials as it generates motions that cover only the dirty areas. These works implicitly assume that dirt is absorbed upon contact. Martinez *et al.* [12] investigate planning for robotic cleaning by wiping with a sponge under the assumption that the particles are pushed upon contact. Do *et al.* [13] predict appropriate action parameters by learning from experience during wiping tasks. Okada *et al.* [14, 15] apply an inverse-kinematics-based programming approach to compute

whole-body motions for the tasks of sweeping the floor, vacuuming the floor, and washing the dishes with a humanoid robot. Lana *et al.* [16] represent robotic manipulation tasks in an algebraic form, which incorporates poses, velocities and forces in a simulated window cleaning task. Vanthienen *et al.* [17] describe table wiping tasks as a set of constraints with the *iTaSC* framework. Ortenzi *et al.* [18] propose to exploit the environment contact constraints of wiping tasks in the operational space, to decouple the motion of the robot from the applied force. Schindlbeck and Haddadin [19] utilize task-energy tanks to react safely upon contact loss. Most recently, [20] proposed an approach to imitate wiping motions from human demonstration. The approach combines impedance control for the execution of wiping motions with learning by demonstration methods. Additionally, the authors apply the Path Integral (PI^2) algorithm to update the imitated force policy. As a result, the deployed light-weight robot is able to successfully execute a wood grinding task repeatedly with different wooden planks. The research by Gams *et al.* applies the concept of movement primitives learned from visual feedback [21]. The trajectories are modified using regression methods, where the feedback is provided through force signals. Eventually, the robot learns new trajectories that are able to maintain the desired contact with the environment. Recently, Gams *et al.* proposed an adaption of this approach by adding a feed forward term that encodes a complete period of motion [22]. The method is able to circumvent perturbations and obstacles and is transferable between different robots. [23] consider the coverage problem arising for a mobile robot such as a autonomous lawn mower. They propose to sub-divide the search space into a grid and apply variations of the *Spanning Tree Covering (STC)* algorithm to cover the area.

The approaches listed so far mainly focus on the physical part of wiping actions, while mainly ignoring the semantic meaning of the motion. In contrast, Kunze *et al.* [24] reason about the semantic effect of the tool interacting with the medium based on a simplified process model. The authors simulate the effect of a sponge contacting liquids, namely the absorption of the liquid. This way, a qualitative effect inference can be conducted based on the absorbed and leftover water particles. Winkler *et al.* [25] maintain expectation about the outcome of planned manipulations in pick-and-place scenarios. Based on observation of relevant task parameters (e. g. gripper forces during object transitions) a robot can learn when an action was successfully executed or failed. Pastor *et al.* [26] propose to learn motor skills in form of Dynamic Movement Primitives. Additionally they predict the task outcome based on low-level sensor streams e. g. fingertip force information. The proposed approach enables the robot to predict failure situations online.

A related research topic is the representation and planning of contact situations. Del Prete *et al.* [27] investigate contact localization on tactile skins to improve the accuracy of force control strategies for robotic manipulators. This approach does not require a model of the external force. Most recently, Denei *et al.* [28] proposed the concept of *tactile maps* for artificial tactile skins. This representation is integrated into a robot control framework which allows a robot to plan motions w. r. t. measured contact. In our previous work we have proposed to represent and classify contact events in force signals by means of multidimensional time-series shapelets [29].

Furthermore, we have investigated robotic wiping actions in one of our earlier articles [30]. We developed a hybrid reasoning framework to plan and parameterize compliant wiping motions executed by whole-body impedance controller. The robot was able to schedule wiping actions for a given high level goal and successfully execute cleaning motions for the tasks of cleaning a window, scrubbing a mug, and collecting shards with a broom. However, the robot had no internal

model of the desired effects and was unable to plan effect-oriented tool motions nor reason about its performance.

Robotic wiping gained some interest in the past, however, there has not yet been comprehensive efforts to cover the cognitive complexity of the problem so far. Consequently, robots are not yet able to mirror the humans cognitive capabilities to reason about wiping actions in detail. In this work, we propose to tackle this issue by equally representing wiping actions and their effects, plan goal-oriented motions accordingly, and estimate real world effects based on haptic feedback.

3. Planning Wiping Actions in the Effect-Space

The causal relation between the motion parameterization and the effects of wiping tasks is particularly rich. That is, similar motions that show comparable trajectories can produce very different effects depending on *how* they are executed in terms of contact force and stiffness settings. In addition, the effects of wiping actions are hardly representable by means of simple numeric properties e. g. homogeneous transformations. Depending on how much force is applied, a wiping motions may dry a wet surface or remove sticky dirt. Moreover, the successful execution of wiping tasks requires geometric reasoning as the purpose of wiping might be the collection, the distribution, or the absorption of particles. In order to perform the commanded tasks successfully the robot has to carefully select the task parameters in a continuous parameter space. We propose a qualitative representation for wiping actions, capable of describing both the wiping motions itself and the resulting effects.

3.1. Particle Distribution Model

According to our earlier work in the classification of compliant manipulation tasks [31], we represent robotic manipulation actions w. r. t. the semantic contact situation between the manipulated objects and the environment. This high-level of abstraction allows us to define semantic actions in accordance with the descriptive manner of the action definitions for automated planning [32]. To further derive geometric models, wiping actions are geometrically represented by the relation between the tool, the medium to be manipulated (e. g. particles or liquids), and the target surface. The medium in wiping tasks (as defined in [31]) is representative for arbitrary liquids or particles with different properties. In order to incorporate different types of media, we propose to project a particle distribution onto the planar target surface

$$\begin{aligned} \boldsymbol{P} = \big\{ &(x_1, y_1),\, (x_2, y_2),\, \ldots,\, (x_N, y_N) \,|\, x_i, y_i \in \mathbb{R} \wedge \\ &x_{\min} \leq x_i \leq x_{\max} \wedge\; y_{\min} \leq y_i \leq y_{\max} \big\}, \end{aligned} \tag{1}$$

where N particles (x_i, y_i) are distributed on the target surface $(x_{\min},\, x_{\max},\, y_{\min},\, y_{\max})$. An exemplary particle distribution is shown in Fig. 2. The scenario shows a kitchen environment where the humanoid robot Rollin' Justin is commanded to collect bread crumbs distributed on a chopping board. In some scenarios, the medium can be perceived visually if the particles are big enough or the liquid is not transparent. However, especially water and other transparent liquids or dust and other small particles are very hard to perceive in camera images. Consequently, the real distribution of the medium cannot be modeled by the robot. In this case, we propose to assume a uniform distribution as it is shown for the initial particle distribution in Fig. 2.

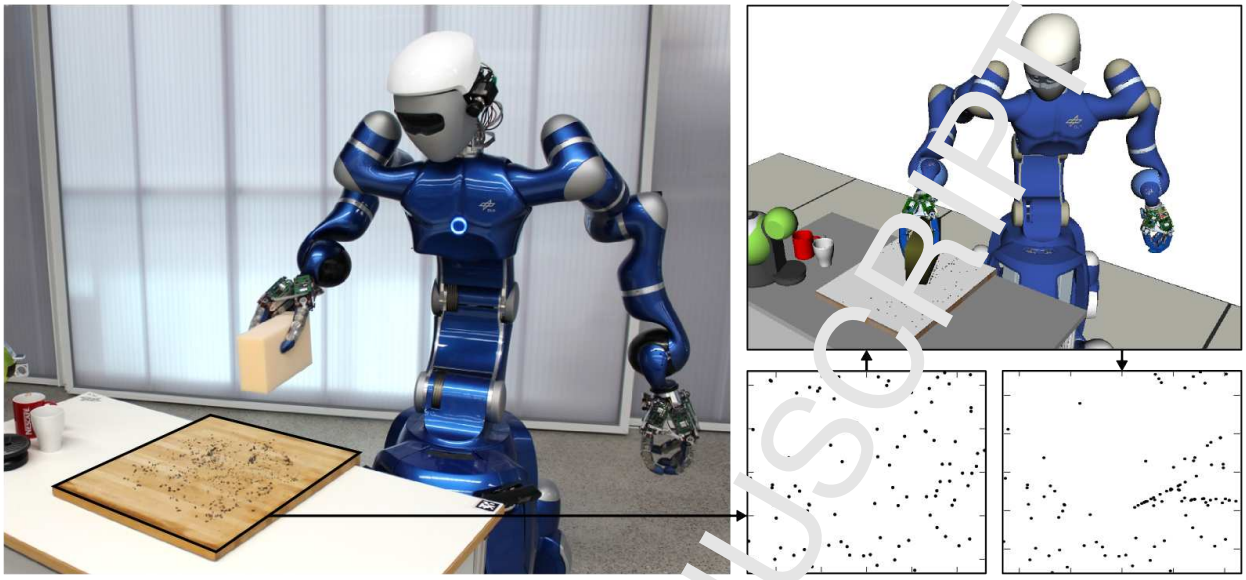


Figure 2: Change estimation for solid particles in contact with the sponge during wiping. As the CAD model of the sponge touches the particles, the particles are moved along the direction of motion.

The main purpose of the proposed particle representation is to have a naive predictive model of the effects of wiping actions. The particle model builds the basis to *qualitatively reason* about the effects of wiping motions. The term qualitative reasoning is coined by Forbus [33]. It refers to simulations that use simplified models. In our case, the simulated particles provide only a rough estimate of the real process. The particles do not incorporate contact with each other and they do not adhere to physical standard models. As a result, the model is considered qualitative (in contrast to quantitative) as the calculated performance metrics provide only rough approximations. Nevertheless the qualitative model can be used to plan and interpret wiping motions. The applied tool-particle interaction model considers the exact CAD data of the tool and the position of each particle. Depending on the type of the tool and the properties of the medium (i. e. solid particles or liquids), the contact results in different effects. For example, if a sponge is simulated to wipe a liquid, the resulting effect is the absorption of the liquid, which is implemented as a delete operation of the respective particles. In case of a solid medium, the contact with the sponge pushes the particles in parallel to the direction of the tool motion. An exemplary simulation of solid particles pushed in contact with the sponge is visualized in Fig. 2.

3.2. Cartesian Wiping Motions

The particle distribution model is not only utilized to simulate the wiping effect, it is also used to plan the wiping motions. This requires the coverage of the entire particle distribution model by means of a waypoint graph in Cartesian space. The surface coverage, i. e. the node distribution for this graph, is restricted by the current state of the geometric environment G_s, i. e. the volumetric model for geometric planning of wiping motions. A collision avoidance strategy based on a *collision sphere model* is implemented to explore the target surface as shown in Fig. 3. The collision sphere is utilized to validate hypothetical node positions during the graph development

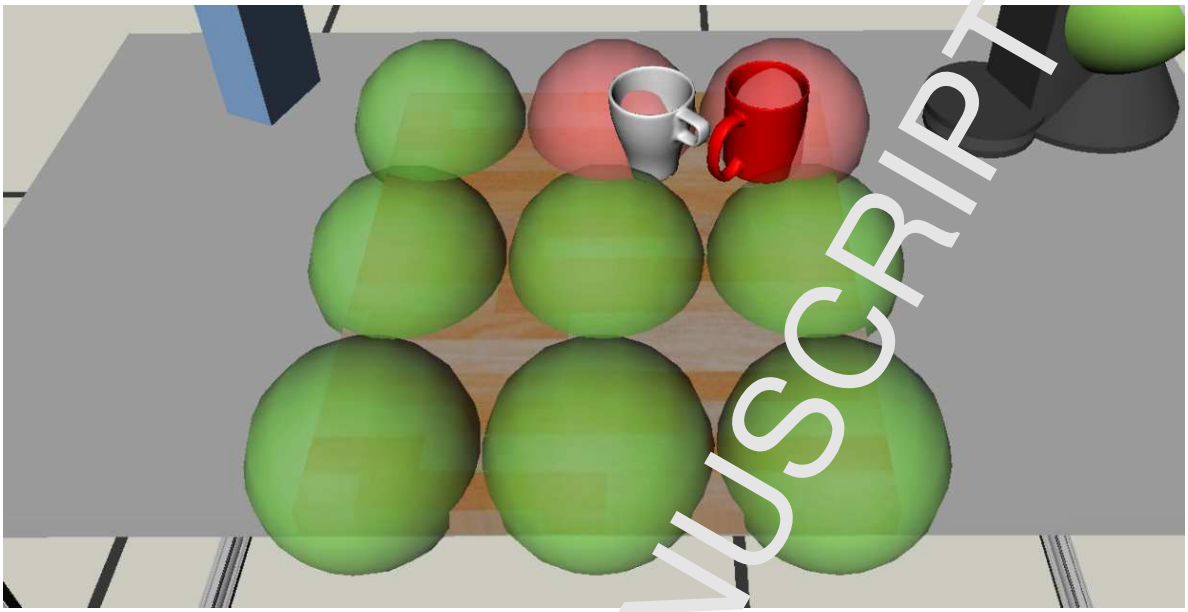


Figure 3: Exemplary collision sphere model distribution.

procedure. A node position is valid if the sphere is not in collision with the environment (except for collisions with the robot, the tool, and the target surface). The collision sphere diameter is defined as $d_s = ||\boldsymbol{D}_{aoe}||$, where $\boldsymbol{D}_{aoe}$ is the area of effect of the tool end-effector (e. g. the nozzle of a vacuum cleaner).

Based on the collision sphere mechanism we have investigated three different coverage strategies. A discretized grid (GRID), Rapidly Exploring Random Trees (RRT) [34], and a Kernel Density Estimation (KDE). The methods are compared in Fig. 4, where red dots mark the resulting graph nodes.

Discretized Grid: The first baseline coverage strategy constitutes a simple grid heuristic. It is adapted from several state-of-the-art approaches that typically rely on direct visual feedback [11, 12, 13] or simply try to cover the entire region by means of minimum path length [23]. This method does not incorporate reasoning on the particle model and is therefore considered a non-cognitive approach. The radius $r_s = d_s/2$, of the collision sphere is used to calculate the grid resolution within the bounds of the target area. The grid-based strategy is uninformed and may be applied if no prior knowledge on the particle distribution is available.

Rapidly Exploring Random Trees: RRTs [34] is a well established method in research on path planning and exploration and is therefore used as second baseline algorithm. The algorithm samples a random configuration $\boldsymbol{q}_{rand}$ in the free space C, calculates the nearest neighbor $\boldsymbol{q}_{near}$, and extends the tree starting from this configuration towards $\boldsymbol{q}_{new}$, which incorporates the maximal expansion length $\boldsymbol{q}_{delta}$. For our approach $\boldsymbol{q} \in \mathbb{R}^2$ and $\boldsymbol{q}_{delta} = r_s$. The algorithm is biased to explore uncovered regions and it is therefore predestined for region coverage. The RRT may be augmented to reject nodes that are too far away from the particles to include prior knowledge about the particle distribution, yet the baseline implementation does not include this feature.

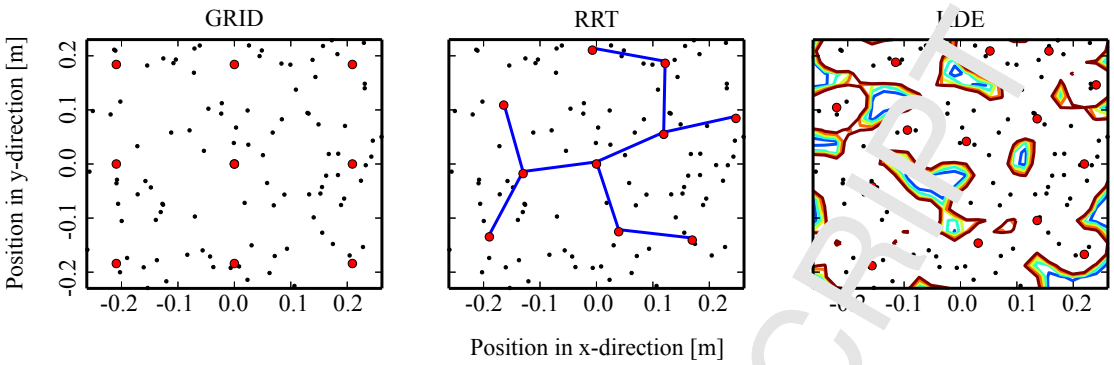


Figure 4: The coverage strategies utilized to explore the target surface. The visualized area corresponds to the obstacle free chopping board surface.

Kernel Density Estimation: The third strategy is a Gaussian KDE to estimate particle probability regions within the particle distribution $\boldsymbol{P}$:

$$\boldsymbol{K}(\boldsymbol{x}) = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} e^{-||\boldsymbol{P}_i - \boldsymbol{x}||^2/h^2} \tag{2}$$

where h is the bandwidth of the kernels. The multivariate KDE is visualized as a contour plot on the right in Fig. 4. This continuous representation is used to select significant peaks with a clearance of d_s, which places the nodes naturally at the position with the highest effect. This is considered a cognitiive approach as it is aware of the particle distribution and thus most beneficial if prior knowledge about the distribution is available, e. g. perceived by a vision system.

The node distribution builds the basis to generate goal orientated wiping motions. To achieve this, we introduce *Semantic Directed Graphs (SDGs)*

$$\mathrm{SDG} = \{\boldsymbol{P}, S_g, G_s\}, \tag{3}$$

which are specified by the particle set $\boldsymbol{P}$, the semantic goal state S_g that is represented as discrete PDDL state [35], e. g. (`collected breadcrumbs chopping_board`), and the geometric environment state G_s that is provided by means of a static scene description of the environment and the related CAD data as it is shown on the top, right in Fig. 2. Based on this information, SDGs project a graph structure on a planar surface, where

- each node n_i represents a waypoint for the Cartesian motion of the *Tool Center Point (TCP)*,
- the edge (n_i, n_{i-1}) connecting two nodes represents the interpolated tool motion in contact.

SDGs can implement wiping actions of different kind. The desired goal state is thereby represented by means of the change to the particle distribution $\boldsymbol{P}_{t_0} \rightarrow \boldsymbol{P}_g$. In this work, we investigate three different actions, namely absorb actions, collect actions, and skim actions. Each action is related to one particular goal state.

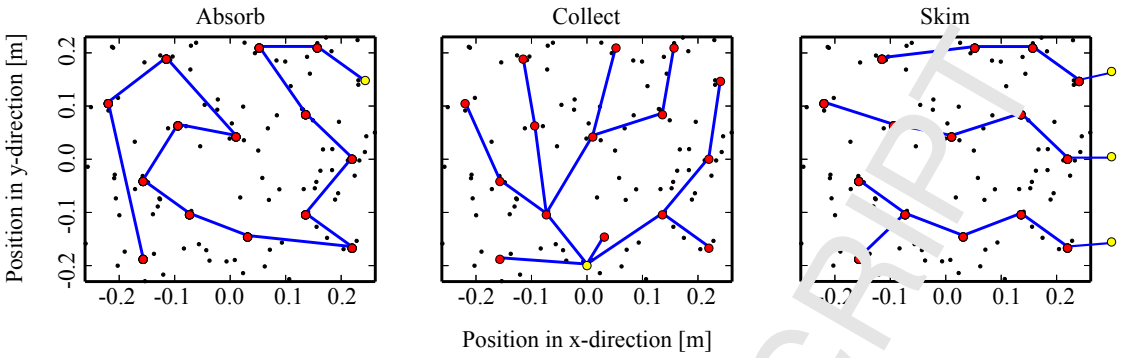


Figure 5: Top view of the developed graph structures for the three prototypical removal actions absorb, collect, and skim. The goal nodes are marked in yellow. Please note that the goal node for the collect action is for now manually defined by the user after the initial region coverage. Accordingly, it may happen that some branches appear redundant (such as the short path segment next to the collect goal).

Absorbing: The absorb action is the first investigated action. It occurs e. g. in vacuuming, dusting, or soaking up water with a sponge. The interaction characteristics between the tool and the medium (i. e. the absorption) is independent of the tool motion, as long as contact is made with the entire region of interest to remove all particles, such that

$$\boldsymbol{P}_{\text{g,absorb}} = \emptyset. \tag{4}$$

Collecting: The second removal action is to collect the medium, where the particles are pushed upon contact. The goal state S_g is geometrically represented as dedicated goal node n_{goal} on the target surface, such that

$$\begin{aligned}\boldsymbol{P}_{\text{g,collect}} = \big\{(x_1, y_1),\, (x_2, y_2),\, \ldots,\, (x_{\text{N}}, y_{\text{N}}) \,|\, x_i, y_i \in \mathbb{R} \wedge \\ ||(n_{\text{goal},x}, n_{\text{goal},y}) - (x_i, y_i)|| \leq r_s\big\}.\end{aligned} \tag{5}$$

Skimming: Skimming is related to collecting the medium. Upon contact with the tool the particles are pushed along the direction of motion of the tool. The semantic goal S_g for this action can be described as a geometric state where all particles are located outside the boundaries of the target surface, such that

$$\begin{aligned}\boldsymbol{P}_{\text{g,skim}} = \big\{(x_1, y_1),\, (x_2, y_2),\, \ldots,\, (x_{\text{N}}, y_{\text{N}}) \,|\, x_i, y_i \in \mathbb{R} \wedge \\ (x_{\text{min}} > x_i \vee x_i > x_{\text{max}} \vee y_{\text{min}} > y_i \vee y_i > y_{\text{max}})\big\}.\end{aligned} \tag{6}$$

Depending on the desired goal state, SDGs implement context-aware motion generate policies. These policies have to correlate semantically to the goal state of the particle distribution $\boldsymbol{P}_{\text{g}}$. For example, the desired effect of the absorb action is to remove all particles by getting into contact with each particle to trigger a delete operation. Collecting and skimming, however, require directed tool motions to have the particles pushed towards a certain goal area (collect), or pushed over the edge of the surface (skim), respectively. Suitable off-the-shelf policies for these issues can be found in graph theory:

A suitable policy for the absorb action is the *Traveling Sales Person (TSP)* algorithm. Hess *et al.* [10] showed that this is a performant approach to solve unconstrained wiping tasks in a generalized way. The outcome is a natural curved motion covering all nodes of the graph.

The collect action requires to direct the graph to a certain goal region, i. e. a single root node. In graph theory this problem is described by a *Minimum Spanning Tree (MST)* [36]. The distance between the nodes serves thereby as cost function with a maximum connection length $l_{\max} = d_s$.

Skimming is related to collecting and therefore implemented as multiple MSTs. Multiple nodes outside the boundary of the surface are user-defined as goal. Depending on the object that provides the target surface, only a subset of boundaries may be valid. For the chopping board, only the boundary facing the robot is defined as goal region (right in Fig. 5). As a result of this policy, multiple trees expand towards the closest goal node.

In addition, it is furthermore possible to describe other wiping tasks in a similar way. To outline an example, the distribution of particles can be described by a uniform distribution of all particles. This is similar to tasks where particles are emitted on the target surface (e. g. painting a wall). Different tasks may include polishing a car, where repetitive motions are executed. However, the full coverage of all identified wiping actions would exceed the scope of this work. To get an intuition on the extension towards further tasks, please refer to [31].

3.3. Whole-Body Joint Motions

The development of Semantic Directed Graphs (SDGs) in Cartesian space is only a first estimate for the feasibility of the planned wiping motions. The collision sphere model is utilized to check for collision free translational motion of the TCP along the developed graph structure. However, the orientation of the tool is not considered until now. Similarly, the joint state of the robot has to be integrated into the reasoning process to verify the overall feasibility. It is most desirable to move the tool perpendicular to the planar motion, such that as many particles as possible are affected. For some cases it is, however, required to rotate the tool to circumvent collisions or local minima in the joint space.

The graph structure of an SDG represents Cartesian tool motions that serve as the basis for whole-body robot motions. The underlying problem to resolve a Cartesian path into joint motions is formulated as *path following problem*. For each Cartesian pose $\boldsymbol{x}$ on a Cartesian path $\boldsymbol{X}$, the robot has to find a joint velocity $\dot{\boldsymbol{q}}$

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{\dagger}\dot{\boldsymbol{x}} + (I - \boldsymbol{J}^{\dagger}\boldsymbol{J})\dot{\boldsymbol{q}}_0 \tag{7}$$

where $\boldsymbol{J}^{\dagger}$ is the generalized inverse of the Jacobian matrix [37]. The joint velocity $\dot{\boldsymbol{q}}$ and the joint acceleration $\ddot{\boldsymbol{q}}$ must not exceed the limits of the robotic manipulators. Moreover, the resulting joint path $\boldsymbol{Q}$ must not collide with any obstacle nor the robot itself.

Path following algorithms try to follow a given Cartesian path as exactly as possible, where all six dimensions of the task are tracked (i. e. three translational dimensions and three rotational dimensions). However, this poses an unnecessary restriction for wiping tasks. For instance, a sponge may be rotated along the normal of the target surface to yield better reachability, yet resulting in a decreased wiping effect. In any case the overall Cartesian motions are oriented towards the next node in order to avoid the degeneration ot the wiping effect. Another example is the cleaning motion of a window wiper which has to be moved orthogonally to the wiper blade in order to achieve the desired effect. However, rotating the wiper along the main axis of the blade does not impair the cleaning result. To this end, we propose a path following method that is aware of the free tool DOF available in the Cartesian space as it is outlined in Algorithm 1.

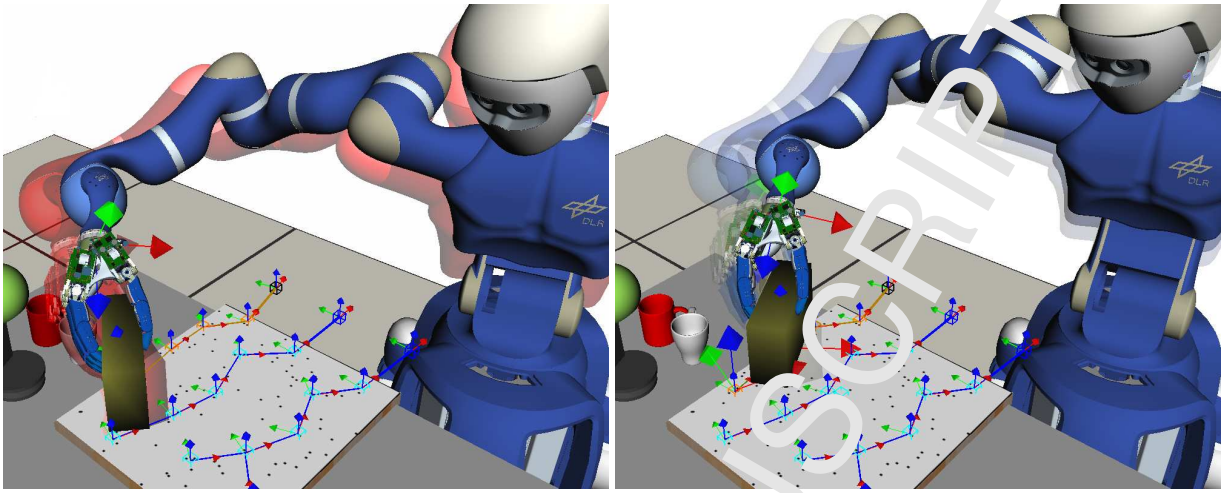


Figure 6: Initially, each node n_i is oriented towards the next node n_{i+1} in the graph. The tool rotation is interpolated along the Cartesian path, i. e. the edge (n_i, n_{i+1}) between this nodes. If a pose along this path is in collision (left), or unreachable, e. g. due to joint limitations (right), the free DOF of the tool are exploited.

Algorithm 1: PathFollowing($\boldsymbol{q}_{n_i}$, n_{i+1}, δ)

Input: The initial joint configuration $\boldsymbol{q}_{n_i}$, the goal node n_{i+1}, and the step-size δ
Output: A continuous joint path $\boldsymbol{Q}$

$\boldsymbol{x}_{\text{start}} \leftarrow$ CalculateToolPose($\boldsymbol{q}_{n_i}$, $\boldsymbol{x}_{\text{grasp}}$)
foreach $\boldsymbol{x}_{goal}$ *in IterateFreeDOF*(n_{i+1}) **do**
  $\boldsymbol{Q} \leftarrow$ List()
  $\boldsymbol{X} \leftarrow$ Interpolate($\boldsymbol{x}_{\text{start}}$, $\boldsymbol{x}_{\text{goal}}$, δ)
  foreach $\boldsymbol{x}_i$ *in* $\boldsymbol{X}$ **do**
    $\boldsymbol{x}_{\text{eef},i} \leftarrow \boldsymbol{x}_i \cdot \boldsymbol{x}_{\text{grasp}}$
    $\boldsymbol{q}_i \leftarrow$ FindIK($\boldsymbol{x}_{\text{eef},i}$)
    if *IsValid*($\boldsymbol{q}_i$) **then**
      $\boldsymbol{Q}[i] \leftarrow \boldsymbol{q}_i$
    else
      break
  if *Length*($\boldsymbol{Q}$) = *Length*($\boldsymbol{X}$) **then**
    return $\boldsymbol{Q}$

The tool poses and the respective TCP poses of the end-effector are computed based on the pose of the nodes in the SDG. Each node n_i is oriented towards the next node n_{i+1} in the branch. The resulting translations and orientations for the nodes are utilized to calculate the initial hypothesis for the start pose $\boldsymbol{x}_{\text{start}}$ and the goal pose $\boldsymbol{x}_{\text{goal}}$ of the TCP as formulated in (7). The edge (n_i, n_{i+1}) in between the nodes is interpolated to resolve the constraints of the path following task. Ideally, all interpolated poses $\boldsymbol{x}_i$ are reachable and collision free so that the robot can manipulate the tool

accordingly. However, in cluttered environments it is unlikely that the edge in between two nodes is collision-free while the robot circumvents local minima (see Fig. 6). In either case we backtrack to the initial pose $\boldsymbol{x}_{\text{start}}$, select an alternative goal pose $\boldsymbol{x}_{\text{goal}}$ w. r. t. the free DOF of the tool, and repeat the path following task. The sequence in which the SDG branches are processed depends on the wiping action. The absorb motions consists of a single path and is therefore unambiguous. The branches developed for the collect and skim actions are resolved iteratively, starting from the leaf node closest to the initial tool pose, back-propagating to the root node of the branch.

In order to transform the computed Cartesian motions into whole-body joint motions, we apply a local path following method as it is described by Konietschke and Hirzinger [38]. If the local method is unable to find a feasible path, a global method as proposed by Huaman and Stilman [39] is applied. This approach constitutes an exhaustive search. Accordingly, it searches the entire joint space without violating joint limits, velocity limits, and acceleration limits. The highly redundant robot can this way resolve local minima on its own. It is therefore complete within the bounds of the discrete search space defined by the SDG. Given enough time the method is guaranteed to find a solution if a collision-free path exists. If no feasible inverse kinematics solution is found at all, the respective node is abandoned and the algorithm moves on with the remaining nodes of the branch.

As it is quite common for wiping tasks in everyday environments to cover large areas, we propose to augment the graph nodes n_i with reachability information for the robotic manipulator in the so called *extended Semantic Directed Graphs (eSDG)* representation. This is done by means of *Capability Maps* [40], which represent the reachability of a robotic manipulator. Capability Maps can be utilized to rate the position of the robot base w. r. t. optimal reachability for a certain task e. g. cleaning a surface. This information is used to reposition the base of the robot during the manipulation planing procedure using an A* algorithm [41]. The motion generation procedure is not detailed at this point in order to focus on the contributions of this work, i. e. the semantic action representation and the related effect-space planning and inference methods. Please refer to our previous publications for a detailed analysis on the particular topic of whole-body motion generation and mobile manipulation [42, 6].

3.4. Simulation-based Effect Prediction and Evaluation

The particle distribution model introduced in Sec. 3.1 is utilized to predict the effect of the planned wiping motions in simulation as visualized in Fig. 2. As the task performance may depend on the initial node distribution on the target surface, this sub-section evaluates the different node distribution strategies w. r. t. the investigated wiping actions. The evaluation is conducted in three different scenarios. Scenario I constitutes the chopping board scenario illustrated in Fig. 2. For scenario II, we assume the same environment without the chopping board, where the particles are distributed on the entire table surface. Scenario III is a car cleaning scenario, where the target area is approximated by a plane aligned with the windshield.

The sponge introduced with the exemplary wiping task of collecting breadcrumbs is used in all experiments in order to obtain comparable results w. r. t. the task performance. A uniform particle distribution is assumed to be able to compare the uninformed methods (GRID, RRT) with the KDE method that exploits the particle model. All three coverage strategies (i. e. GRID, RRT, KDE) are paired with the three removal actions (i. e. absorbing, collecting, skimming) and

evaluated w. r. t. the traveled Cartesian distance in contact with the surface, the computation time, the execution time, and the task performance. The calculation of the computation time involves the node distribution, the base optimization, the path following algorithm, and all collision checks, where OpenRAVE [43] is utilized. The execution time is based on the assumption of a maximum velocity of $1\,rad/s$ joint speed, and a maximum joint acceleration of $2\,rad/s^2$. The maximum base velocity is limited to $0.5\,m/s$, and a maximum acceleration of $1\,m/s^2$. The performance measurement is conducted according to the constraint definitions (4), (5), and (6). Accordingly, the metric for the absorb action can be described as the number of deleted particles, the collect action is evaluated based on the number of particles within the radius r_s around the goal node n_{goal}, and the metric for the skim action is the number of particles pushed outside the boundaries of the target surface. All results are based on the average of five trials with different initial particle distributions. The results are visualized in Fig. 7 and listed in Table 1.

The different coverage strategies vary strongly in performance for the individual settings. Therefore, it might be useful for the robot to reason about the most effective coverage strategy given a concrete problem definition. In case of lesser obstructed environments (as for example posed by the chopping board scenario and the windshield scenario) the KDE mostly outperforms the other methods w. r. t. the defined performance metrics. However, the KDE coverage strategy seems to be biased by obstacles as it is less efficient for the table scenario. In conclusion, a robot may have to test the available coverage strategies for a given scenario in order to figure out and execute the most effective one. While this observation may be considered as a drawback for one-time tasks, it is especially interesting for recurring tasks, e. g. industrial manufacturing tasks, such as polishing the surface of a car. These tasks can be autonomously optimized w. r. t. the execution

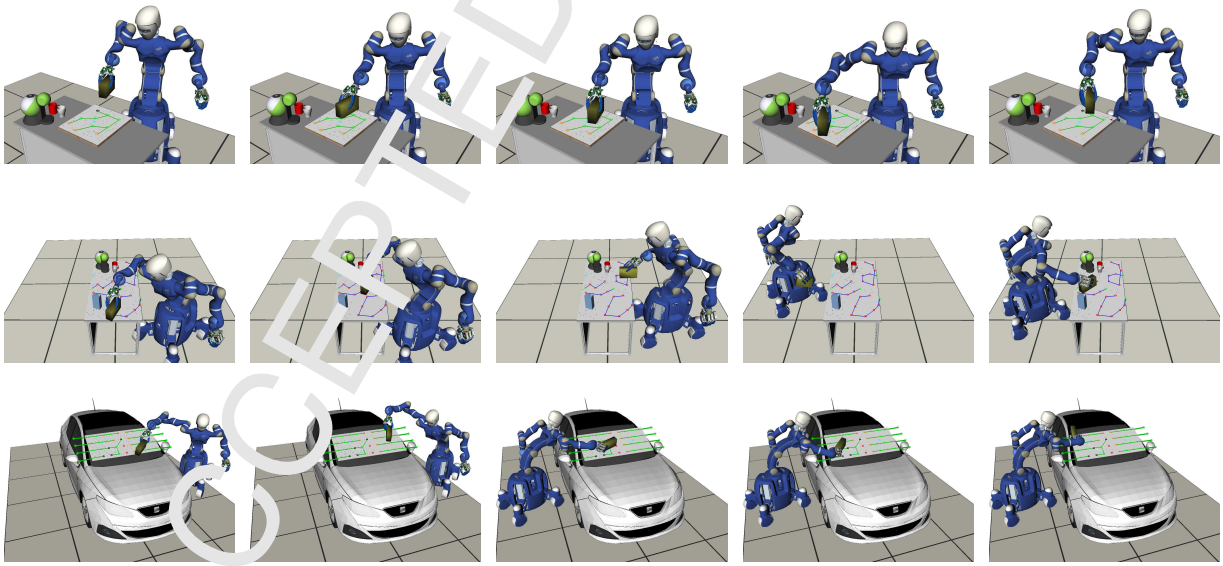


Figure 7: Simulation-based effect prediction for the collect action in the Chopping Board scenario (top), the absorb action in the Table scenario (center), and the skim action in the Windshield scenario (bottom). The particles contact behavior is visualized over time (best viewed in a digital copy of this work).

Table 1: Evaluation of the three scenarios Chopping Board, Table, and Windshield.

		Chopping Board			Table			Windshield		
		GRID	RRT	KDE	GRID	RRT	KDE	GRID	RRT	KDE
Absorb	Cart. dist. [m]	1.52	1.41	1.90	4.34	4.10	3.24	4.44	4.84	4.69
	comp. time [s]	45.96	57.12	62.00	144.47	312.45	169.92	314.04	423.21	417.66
	exec. time [s]	18.01	25.92	27.71	41.30	66.96	52.43	77.59	111.90	66.11
	performance [%]	84.25	85.00	95.75	77.00	79.50	70.50	90.13	90.50	91.00
Collect	Cart. dist. [m]	1.54	1.45	1.90	3.68	3.60	2.92	4.97	4.76	4.22
	comp. time [s]	69.89	68.48	72.24	110.92	161.60	270.56	885.45	262.55	428.05
	exec. time [s]	16.17	32.42	43.28	65.71	83.36	62.14	112.47	116.26	97.56
	performance [%]	86.50	86.75	90.50	70.25	69.25	41.75	48.50	39.25	62.75
Skim	Cart. dist. [m]	1.81	1.62	2.01	5.48	4.63	3.75	5.58	5.65	5.89
	comp. time [s]	107.17	44.25	90.57	211.14	260.34	255.58	325.97	212.34	199.03
	exec. time [s]	32.11	26.51	45.81	113.37	122.50	102.48	108.18	103.29	96.03
	performance [%]	88.00	80.25	97.00	71.75	66.50	93.00	94.50	88.50	88.50

time or the task performance by iterating over the available coverage strategies.

One may argue that a greedy algorithm that ignores the particle distribution would most likely outperform all of the other coverage methods by means of planning time. While this may be true, we argue that the effort measured by the Cartesian distance as well as the execution time would increase due to unnecessary repetitions as covered regions (presumably close to the target region) may be revisited unknowingly.

In addition, it is often necessary to follow a particular procedure to achieve the desired effects and avoid unwanted ones (e. g. painting a wall in a grid, moving from top to bottom, in order to avoid stroke artifacts). Therefore, we argue that it may be beneficial for future robots to provide a semantically meaningful portfolio of coverage algorithms that can be selected to the needs of the task. Yet, we recommend to favor the KDE strategy as it is the only one that is able to take the actual dirt distribution into account. While the KDE method is able to reduce planning effort on known particle distributions, the other two methods generate unnecessary motions if the particle distribution is not covering the entire surface. This is especially valuable if the dirt distribution can be estimated in advance (e. g. by means of visual perception in case of visible media) or if the robot has to recover from failure situations as it is discussed in Sec. 4.3.

4. Inferring the Effects of Wiping Actions

Up to this point of our work we have utilized the proposed particle distribution model to plan wiping motions and predict the wiping effect based on simulated motions. This section shall investigate real world wiping actions and the estimation of the real task performance. This is usually done by means of visual feedback as it is described by several research groups [11, 12, 13]. In contrast, we propose a vision independent feedback method as visual data is often unreliable for wiping actions for two main reasons. First, the medium in wiping tasks may be invisible due to lighting conditions or the properties of the medium such as small dust particles, or transparent liquids. Second, the simulated experiments indicate that the robot itself is often occluding the manipulated areas which makes visual perception of the effect only available after the task execution.

4.1. Compliant Execution of Wiping Motions

The planned wiping motions are executed by the real robot with the compliant whole-body impedance control framework introduced by Dietrich *et al.* [44]. The controller parameters are provided by an object-centric knowledge-base. That is, the applied Cartesian force, the Cartesian stiffness, and the whole-body control task hierarchy are parameterized according to the requirements of the tool, the surface, and the medium to be manipulated. We do not apply a force-control strategy as the robot is intended to work in close collaboration with humans. A hybrid force-position controller would not allow to safely interact with the robot during manipulation. Instead we apply a soft impedance controller and exploit the compliant behavior to command the wiping motions *into* the target surface. As a result, our control strategy allows for safe physical human-robot interaction while it is still able to get into desired contact with the environment. A force controller would rather counteract the human during interventions which is not desirable for a service robot. For a detailed discussion on this topic, please refer to [30].

In a nutshell, we apply a whole-body impedance controller to establish compliant contact, where the controller force $\boldsymbol{f}_\text{c}$ is saturated to the needs of the task and transformed into the joint space by the Jacobian transposed as it is illustrated in (8), where V is a virtual elastic potential and $\boldsymbol{D}$ the damping matrix.

$$\boldsymbol{\tau} = -\boldsymbol{J}^T \underbrace{\boldsymbol{\mathcal{S}} \left(\left(\frac{\partial V}{\partial \boldsymbol{x}} \right)^T + \boldsymbol{D}\dot{\boldsymbol{x}} \right)}_{\boldsymbol{f}_\text{c}}, \tag{8}$$

$\boldsymbol{\mathcal{S}}$ is the saturation function for the two force terms $\boldsymbol{f}_\text{spring} = (\partial V/\partial \boldsymbol{x})^T$ plus $\boldsymbol{f}_\text{damper} = \boldsymbol{D}\dot{\boldsymbol{x}}$. The external contact force $\boldsymbol{f}_\text{ext}$ counteracts the controller force $\boldsymbol{f}_\text{c}$ under the assumptions of a static equilibrium with negligible model uncertainty. Accordingly, the saturation function $\boldsymbol{\mathcal{S}}$ clamps the controller force to limit the applied contact force.

4.2. Feedback-based Contact Estimation

Our inference method is based on the force sensing capabilities of the compliant manipulators of the robot. In particular, we execute the planned wiping motions by means of a compliant whole-body impedance controller and record telemetry data of the robot. This includes end-effector forces calculated from joint torque sensor measurements, as well as measured Cartesian end-effector positions. With this information we are able to reproduce the executed motions in simulation and estimate the contact between the tool and the particle distribution w. r. t. real world sensor readings and compute the real task performance respectively.

At each timestamp i, we record the actual measured end-effector position $\boldsymbol{x}_\text{act}$ and the controller forces at the end-effector $\boldsymbol{f}_\text{c}$. An example path of the computed TCP position is visualized as black dotted line in Fig. 8. The transformation of the TCP is thereby defined as $\boldsymbol{H}_{\text{tcp},i} = \boldsymbol{H}_{\text{act},i} \cdot \boldsymbol{H}_\text{grasp}^{-1}$, where the homogeneous transformation matrices $\boldsymbol{H}_*$ correspond to the task space coordinates $\boldsymbol{x}_*$. The grasp transformation matrix $\boldsymbol{H}_\text{grasp}$ is assumed to be constant during the task execution as the soft material of the sponge aligns firmly with the curvature of the hand. The blue lines on the target surface indicate the desired wiping motions of the sponge TCP

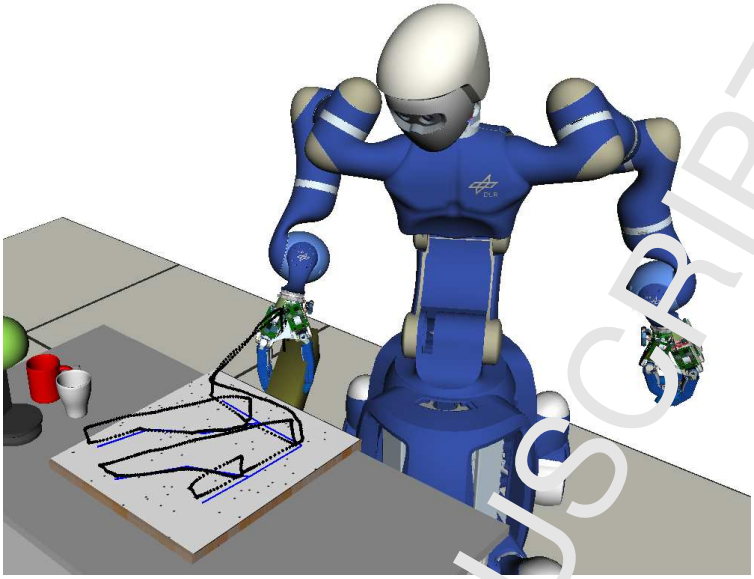


Figure 8: The recorded Cartesian wiping motion of the robotic manipulator, i. e. the TCP of the sponge it is holding respectively (black dotted line) and the desired wiping motion in contact (blue lines).

in contact with the chopping board. The overall motion of the planned collect action is illustrated in Fig. 8. By coincidence, an arrow-shaped path is formed pointing towards the goal node n_{goal}.

Each TCP position is related to the corresponding contact force measurement provided by the structured log. This allows us to infer the segments of motion that are most likely in contact with the environment. In fact, we are only interested in the force normal to the target surface $\boldsymbol{f}_{\text{ext,n}}$. We analyze the normalized contact force

$$\boldsymbol{f}'_{\text{n},i} = \frac{\min(\boldsymbol{f}_{\text{ext,n,max}},\ \boldsymbol{f}_{\text{ext,n},i})}{\boldsymbol{f}_{\text{ext,n,min}}}, \tag{9}$$

for each force sample $\boldsymbol{f}_{\text{ext,n},i}$. In the example at hand $\boldsymbol{f}_{\text{ext,n,max}} = 0\,\text{N}$ and $\boldsymbol{f}_{\text{ext,n,min}} = -10\,\text{N}$.

Only the TCP positions that show a high confidence for contact with the chopping board are considered in the first place. That is, we investigate only measurements that show a normalized force value of [0.9, 1.0], such that

$$\boldsymbol{x}'_{\text{tcp},i} = \left\{\boldsymbol{x}_{\text{tcp},i} \,|\, 0.9 \leq \boldsymbol{f}'_{\text{n},i} \leq 1.0\right\}, \tag{10}$$

i. e. the measurements that show only 10% deviation from the desired contact force as it is visualized in Fig. 9. The corresponding motion is visualized by green dots in the simulation presented on the left of Fig. 10. Most of the measurements close to the target surface match this constraint and resemble the desired wiping motion. This is already a quite accurate estimate of the contact motion, yet, some segments were omitted due to lower contact forces introduced by friction and stick-slip effects (t=25s). However, these segments cannot be ignored as they participate to the overall wiping effect.

We incorporate these left out yet still contact-rich segments, by applying the *Random Sample Consensus (RanSaC)* algorithm [45]. The RanSaC algorithm is often used in computer vision to

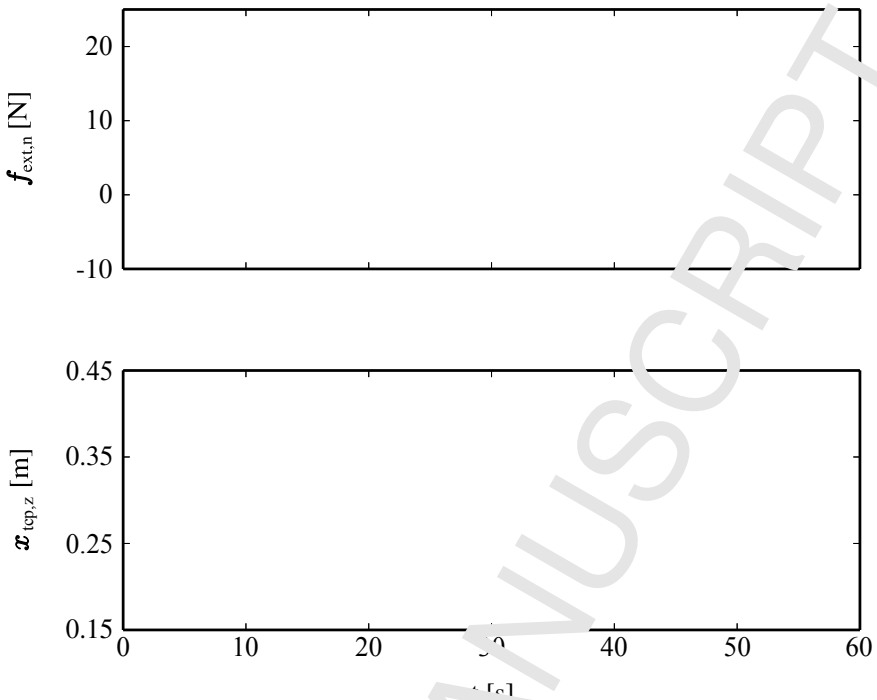


Figure 9: Plot of the external force normal to the target surface $f_{ext,n}$ and the corresponding TCP position in z-direction $x_{tcp,z}$. The measurements with high contact confidence $x_{tcp,i}$ are emphasized by the green bars.

fit a plane onto surface elements perceived in visual data, such as point clouds computed based on depth camera images. We propose to apply the RanSaC algorithm on the Cartesian TCP positions with high contact confidence $x'_{tcp,i}$ to fit a plane onto the target surface, such that

$$c_{plane,i} = \text{RanSaC}\left(\sum_{i=0}^{N} x'_{tcp,i},\ \epsilon\right), \tag{11}$$

where $x_{plane,i}$ constitute the measurements within the estimated target plane visualized as purple box in Fig. 10. All measurements $x_{plane,i}$ that fit within the inlier threshold ϵ along the estimated surface orientation are considered in contact with the surface. The corresponding measurements are visualized as green dots of different brightness. High normalized contact forces $f'_{n,i}$ are represented by bright green colors. Darker green colors (eventually fading to black) represent lower normalized contact forces $f_{n,i}$. The RanSaC based approach allows us to incorporate all measurements that are potentially in contact with the target surface for contact modeling, and not solely the data points with high contact confidence. As motivated in the introduction, this way of haptic perception allows the robot to estimate the target surface despite poor lighting conditions as they may occur on transparent glass panes or reflecting solar panels, for example.

The contact model for the simulation-based effect prediction in Sec. 3.4 only considers the volumetric model of the tool in relation to the position of the particles. There are no forces involved to simulate the wiping effect. However, in real world applications lower contact forces may result in poor contact situations. Therefore, we propose to incorporate the normalized contact force $f'_{n,i}$

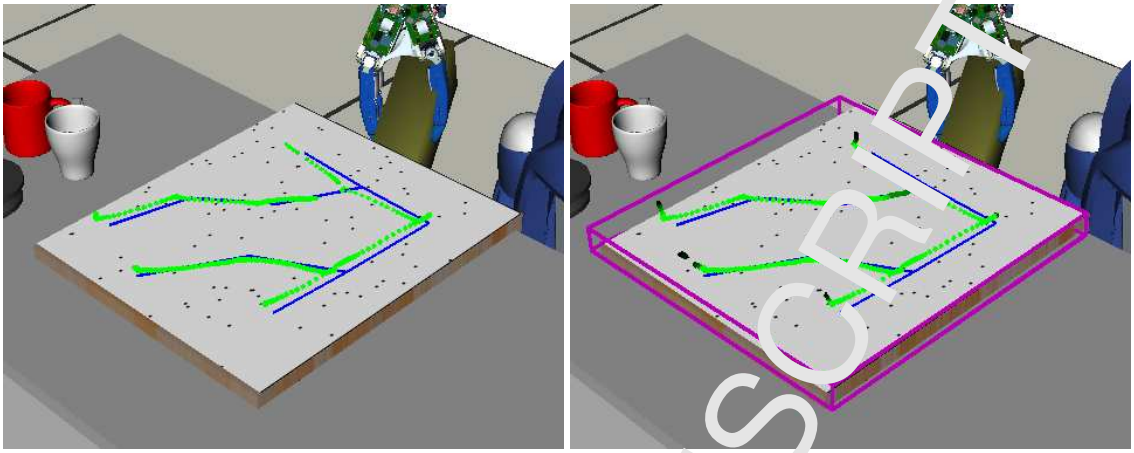


Figure 10: Left: closeup view of all measurements with high contact confidence $\boldsymbol{x}'_{\mathrm{tcp},i}$ (green dots), which are the basis for the target surface estimation. Right: the estimated target surface visualized as purple box. All positions of the TCP within this box are considered in contact with the target surface and colored in green. The brighter the color, the higher the normalized contact force $\boldsymbol{f}'_{\mathrm{n},i}$.

to model the effect of real world wiping actions. The enhanced contact model is based on

- the position of the TCP of the tool $\boldsymbol{x}$
- the particle distribution $\boldsymbol{P}$ w. r. t. the tool CAD data,
- and the applied controller force $\boldsymbol{f}_c$, respectively the counteracting contact force $\boldsymbol{f}_{\mathrm{ext}}$.

By re-executing the recorded motions in simulation we are able to infer the real world effect of the previously executed wiping motions under consideration of the logged contact forces. We

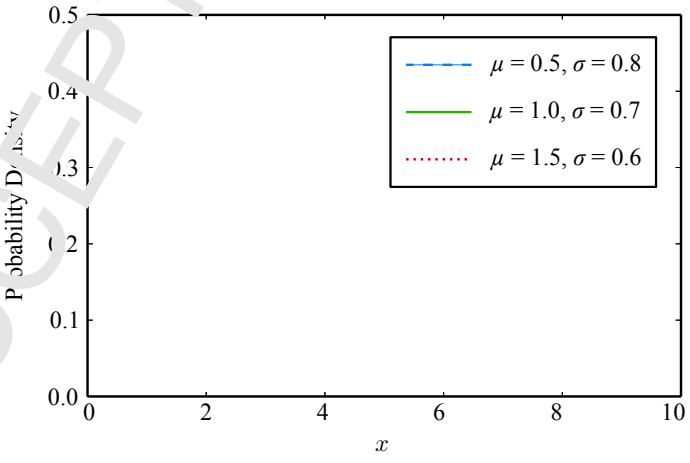


Figure 11: Exemplary plots of the probability density function for different contact situations.

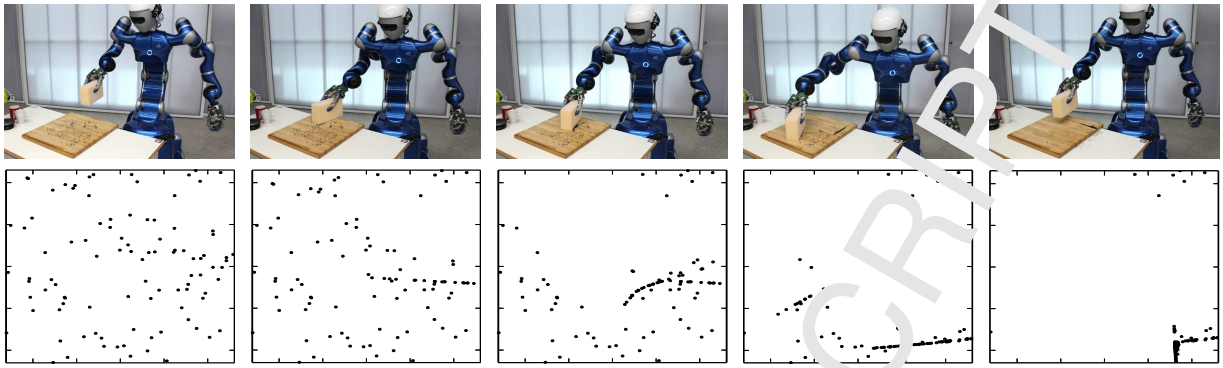


Figure 12: The real collect action executed by Rollin' Justin (top) compared to the estimated outcome (bottom). The particle distribution in the bottom row is rotated clockwise, where the upper left corner corresponds to the left corner of the chopping board in the top row. Black chippings of glass are placed to visualize the task performance and assess the effect inference. They are not visually perceived. Instead a uniform distribution is assumed.

argue that contact forces close to the desired force have higher probability to produce the desired wiping effect to the medium. However, the actual effect depends on the properties of the tool-medium-surface tuple as described in our previous investigations on compliant manipulation tasks [31]. Among others, the exact tool geometry, the tool elasticity, the surface friction, the surface inclination, the medium friction, as well as the medium size and geometry influence the outcome of wiping actions. Most of these parameters are not assessable. Therefore, we utilize the *log-likelihood function* $\log(\mathcal{L}(x))$ to model the effect on a qualitative basis. At every timestamp, the log-likelihood is computed for each particle in contact with the tool.

- If $\boldsymbol{f}'_{\mathrm{n},i} \geq ||\log(\mathcal{L}(x_i))||$, the contact behavior is simulated as described in Sec 3.1 (i. e. push the affected particles in case of collect and skim actions and delete the particles for the absorb action).
- If $\boldsymbol{f}'_{\mathrm{n},i} < ||\log(\mathcal{L}(x_i))||$ the simulation step is skipped without applying the effect and the algorithm proceeds with the next measurement.

The *probability density function* $p(x)$ for the log-likelihood function is visualized in Fig. 11. The visualized plots shall illustrate the possibility to simulate different contact situations. It is defined as

$$p(x) = \frac{1}{x\,\sigma\,\sqrt{2\,\pi}}\,e^{-\frac{(\log(x)\,-\,\mu)^2}{2\,\sigma^2}}, \tag{12}$$

where μ is the mean and σ is the standard deviation of the distribution. These variables can be altered to simulate contact models for different tools-medium-surface combinations with varying properties. This approach allows us to avoid a fixed force threshold by exploiting the variance of the likelihood function. While a fixed force threshold may be sufficient to distinguish contact from no contact, it often results in false positives in borderline situations. Utilizing a steeply parameterized log-likelihood function (e. g. to simulate a window wiper skimming water from a

window, blue in Fig. 11), even path segments with a lower force measurement have a chance to produce the desired outcome. Vice versa, even high contact forces may result in no effect for more flat log-likelihood functions (e. g. to simulate bristles collecting fine sand, red in Fig. 11).

A collect action with a sponge is shown in Fig. 12. The resulting effect estimation over time is visualized below. Please note that the initial particle distribution is not visually perceived (Which is of course always an option for visible media). Similarly to the earlier experiments conducted in Sec. 3.4, we assume a unified particle distribution to emulate the absence of visual feedback. The minor contact loss observed in Fig. 9 and Fig. 10 does not affect the overall particle distribution estimation. Accordingly, all particles are collected on the lower right corner of the target surface.

The second scenario showcases a bi-manual example action where a broom is used to collect particles on the floor (left in Fig. 13). This experiment is executed twice with particles of different kind. The robot motion as well as the controller parameterization is identical in both trials. First, the chippings (2 mm - 6 mm) introduced in the previous example are distributed on a sheet of paper on the floor. As the broom swipes over the surface most of the particles are effected. Few chippings remain as the bristles of the broom are of irregular nature. Second, small grains of sand (0.2 mm - 2 mm) are distributed. These particles are too small to be collected by the broom as the bristles bend. This effect can be modeled with the particle representation by utilizing a flatter log-likelihood function.

The third scenario investigates the effect of different tools to a certain medium. In particular, the robot is commanded to skim detergent from a solar panel utilizing the rubber blade of a window wiper in comparison to the bristles of a brush (right in Fig. 13). Since, the brush is not designed to manipulate liquids, the window wiper outperforms the brush in this task. The effect can only be assessed by means of macro recordings under improved lighting conditions as the solar panel reflects the light such that one cannot clearly distinguish between wet surfaces and reflections. In fact, the detergent is not perceivable in the images recorded with the cameras of the robot, which would prevent us from applying a vision-based estimation in this case. In the before-after image, one can see that the window wiper removes most of the liquid from the solar panel surface, whereas a layer of detergent remains utilizing the brush. Similarly to the other experiments, this circumstance can be modeled by adapting the parameters of the log-likelihood function. While the particle distribution is not designed to model the influence of the brush on the detergent structure (e. g. the bubble size), the simulated contact with the brush reflects the overall observation of the real world effect. The distribution shows local particle variance while the global distribution is mostly unaltered. Even though the detergent is invisible to the robot due to the low resolution cameras, the particle model allows to estimate the poor performance qualitatively. In conclusion, the applied computational model presents a suitable estimation of the real world effect.

Although the contact behavior poses an abstraction of the actual process (i. e. the contact between the tool-medium-surface tuple as well as the motion of the particles are strongly simplified), the resulting patterns in the particle model match the real world observations in general which allows for a qualitative assessment. Given the right parameterization of the log-likelihood function, the applied effect model roughly matches the real world outcome for the showcased scenarios.

While the results of the effect inference are quite satisfactory, the parameterization of the effect model introduces some limitations. For now, the tool dependent parameters for the log-likelihood function (i. e. μ and σ), as well as the maximum allowed contact force $\boldsymbol{f}_{\text{ext,n,max}}$ are

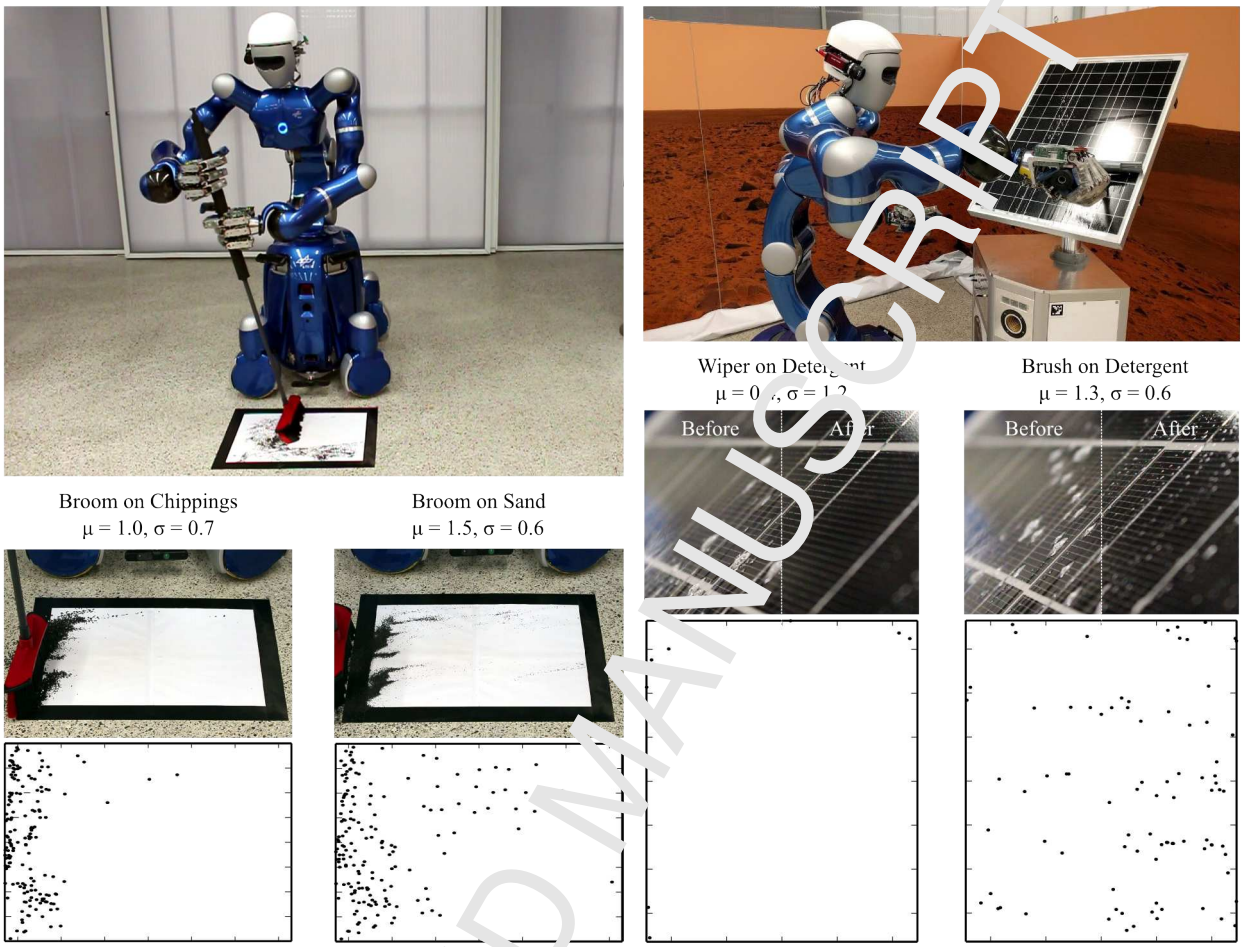
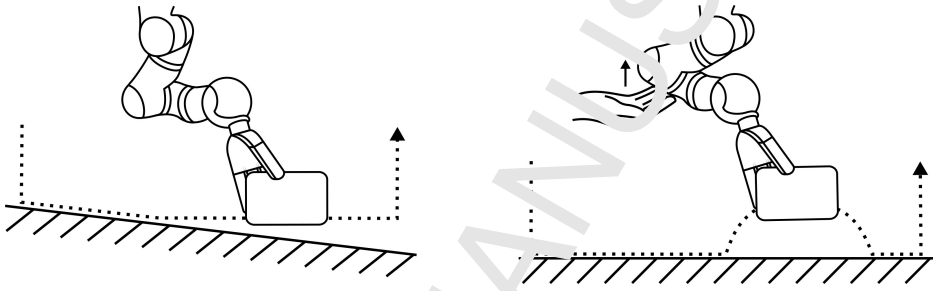


Figure 13: Left: a broom is used to collect particles. The broom is in general less accurate than the sponge utilized in the previous example. Some of the chippings are not effected by the broom. The task performance decreases with the size of the particles as it is observed for the fine grained sand. Right: detergent is applied to a solar panel. The robot executes a skim action with a window wiper and a brush in order to remove the liquid. The model parameters μ and σ are designed to match the effect of the two tools on the medium.

defined empirically. However, just recently it was shown that the task parameters in the context of wiping motions can be learned by the robot. In particular, Hazara and Kyrki [20] proved that it is possible to learn a wood planing task by human demonstration and even improve the resulting motions by means of reinforcement learning strategies. The authors utilize the PI^2 algorithm to enhance the imitated force profile. Additionally, Do *et al.* [13] show that it is possible to learn tool parameters including the tool stiffness over time based on the visual observation of the effect. Utilizing visual perception, machine learning could be applied to the problem at hand in order to learn the parameters of the log-likelihood function and thus autonomously improve the effect prediction.

4.3. Failure Detection and Recovery

As already emphasized, we execute wiping motions by means of a whole-body impedance controller [44]. The applied contact force is thereby determined by the controller force which is transformed into the commanded torque (8). The controller force is thereby saturated w. r. t. the requirements of the wiping action. This approach is proven to be suitable for wiping actions [30] and allow for safe human-robot interaction likewise [46]. However, as the controller is not designed to adapt the Cartesian position w. r. t. the force measurements (unlike a hybrid position-force controller as described in [28]), it is prone to errors arising from imprecise localization (see Fig. 14 left) and external perturbation (see Fig. 14 right). Nevertheless, the proposed inference method is able to detect these execution errors and adapt accordingly.



Failure Scenario 1:
Imprecise Localization

Failure Scenario 2:
External Perturbation

Figure 14: The two investigated failure scenarios. Left: contact loss due to imprecise localization (left). Right: contact loss due to external human perturbation.

The first failure scenario is the loss of contact due to imprecise localization. In particular, we tilt the table in front of the robot to emulate a rotational localization error. Fig. 15 shows the task execution in five snapshots. The contact loss is captured in the fourth frame. The commanded wiping motions are the same as in the successful task execution. A plot of the contact force and the TCP position as well as the corresponding TCP motion and the plane estimation are provided in Fig. 16. The plot already hints at the fact that only few confident contact situations occurred during the task execution. This gets more obvious in the visualized motion in the simulation environment. While the left segments of the arrow-shaped path roughly match the perceived table height, the path segments on the right are rendered too high and show low contact forces (dark green and black dots). The purple box is visualizing that the plane estimate is shifted to the right accordingly (Please note that the box constitutes only a visual element that does not limit the target plane extension). However, a simple comparison of the volumetric model of the sponge and the particle distribution would not be sufficient to estimate the contact situation as the sponge is still very close to the target surface. By referencing the measured tool motion to the force at the end-effector we are able to detect the contact loss and model the effect as visualized in the lower row of Fig. 15. The final estimation of the particle model is shown to be very similar to the real breadcrumb distribution on the chopping board (Please refer to a digital copy of this article for

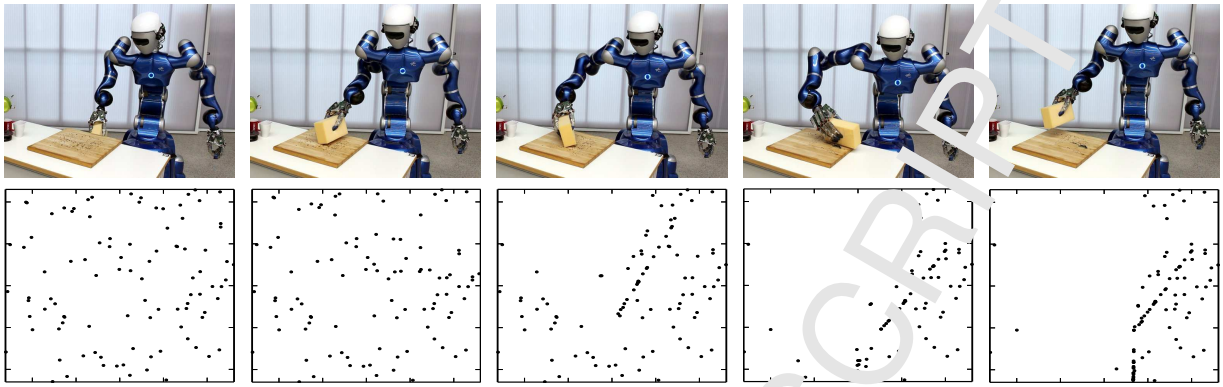


Figure 15: Failure Scenario I: The tilted table simulates an incorrect localization leading to partially poor contact situations shown in the top row. The estimated particle distribution is shown in the lower row.

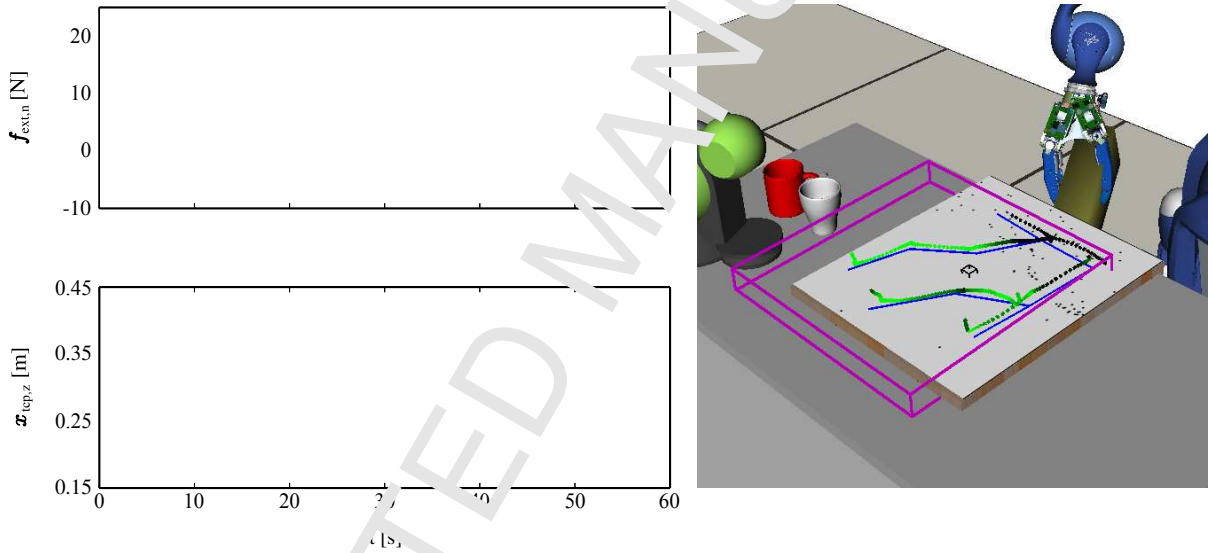


Figure 16: Left: a plot of the contact force $\boldsymbol{f}_{\text{ext,n}}$ and the TCP position in z-direction $\boldsymbol{x}_{\text{tcp,z}}$ in a failure situation. The contact loss is evident in the reduced number of confident contact situations $\boldsymbol{x}'_{\text{tcp},i}$ (green bars). Right: the recorded Cartesian wiping motion in a failure situation arising from a tilted table. The green and black path visualizes the actual measured TCP motion. The commanded path is visualized as blue lines.

optimal picture quality).

The second failure scenario showcases a deliberate human intervention. The interaction is visualized in frame two and four in Fig. 17. The robot is pushed up, such that the sponge looses contact with the chopping board. As the maximum controller force is saturated to satisfy the contact behavior, the robot does not counteract the human and the controller force measurement is not different from the nominal case. However, obviously the position of the end-effector does not match the desired position. Since we apply a RanSaC based approach to estimate the target plane, our approach is able to detect these outliers and ignore them during the effect estimation

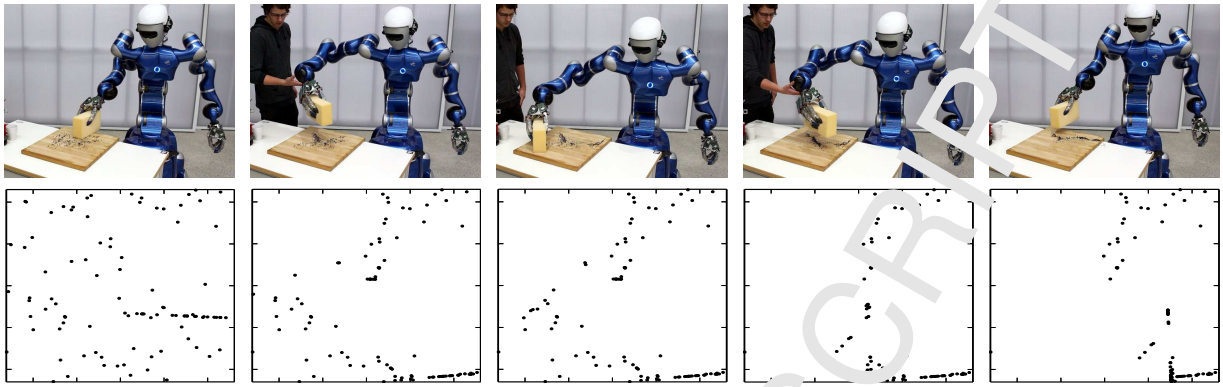


Figure 17: Failure Scenario II: The robot is pushed twice at the end-effector during task execution as it is shown in the top row. The estimated particle distribution visualized in the bottom row reflects this disturbance.

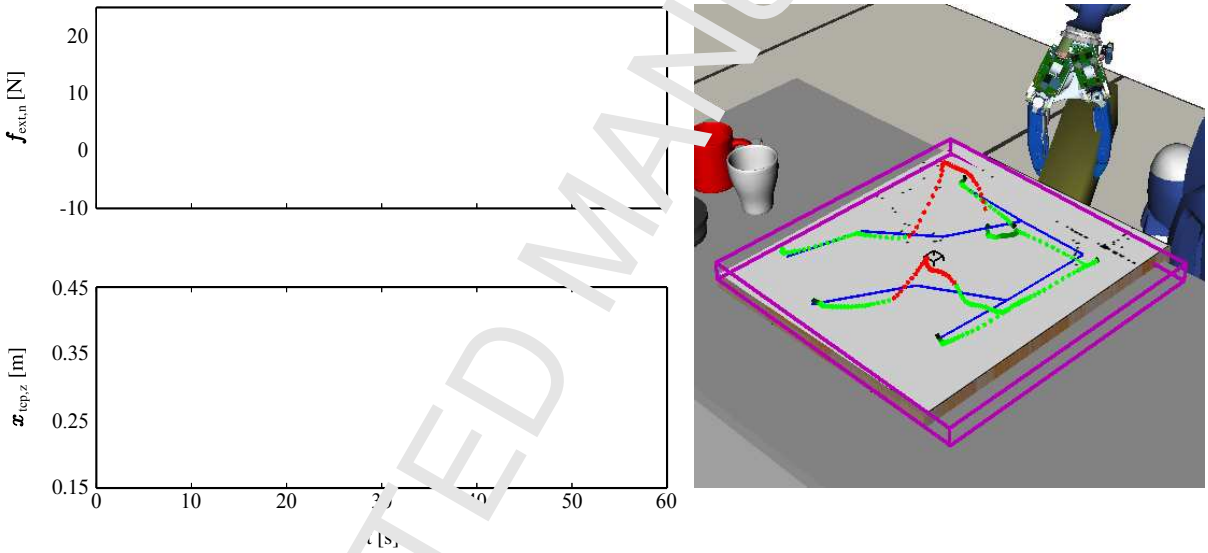


Figure 18: Left: the plot for the contact force $\boldsymbol{f}_{\text{ext,n}}$ and the TCP position in z-direction $\boldsymbol{x}_{\text{tcp,z}}$ in the second failure situation. The red bars indicate false positive contacts introduced due to perturbation. Right: the Cartesian wiping motion for the second failure situation where the robot was pushed twice at the end-effector. The real world measurements of the TCP position are here visualized as green and red path.

procedure. Similarly to the first failure scenario, the robot is eventually able to correctly infer where the wiping motions have been effective, and where the desired effect was not carried out. The false positive measurements are visualized as red bars and the matching red path segments in Fig. 18. In the plot one can see that the contact force stays almost constant during the perturbation while the position is significantly changed. The interaction right before the two main intersections of the wiping motions is a significant impairment for the overall task performance. Some particles are estimated to remain on the chopping board similarly as observed for the real execution.

As argued in the introduction of this article, humans are capable of detecting execution errors

based on haptic perception, update their internal task models accordingly, and use the new information to recover from the failure situation. Our representation of wiping tasks and their effects enables a robot to close this cognitive loop in a similar way for wiping tasks. In particular, utilizing the output of the effect inference method to replan additional wiping motions allows the robot to recover from possible failures introduced due to poor contact situations. Consequently, the robot is able to plan the recovery motion directly in the effect-space. As the effect inference is mostly matching the task performance of the real world execution, there is no visual feedback required. This makes the proposed approach also applicable to wiping tasks involving transparent liquids or small dirt particles, e. g. water or dust.

The recovery procedure for the second failure scenario is outlined in Fig. 19. The initial particle distribution to plan the recovery motion is based on the final estimation of the particle distribution after the robot was pushed twice at the right manipulator. To plan the recovery motion most effectively, we utilize the KDE strategy to distribute the nodes for the SDG w. r. t. the regions with high particle density (red dots in the top left) and replan the collect action accordingly. Similarly to the previous executions, the robot generates whole-body joint motions, executes them, and records structured logs during run-time. This enables the robot to re-evaluate the task performance w. r. t.

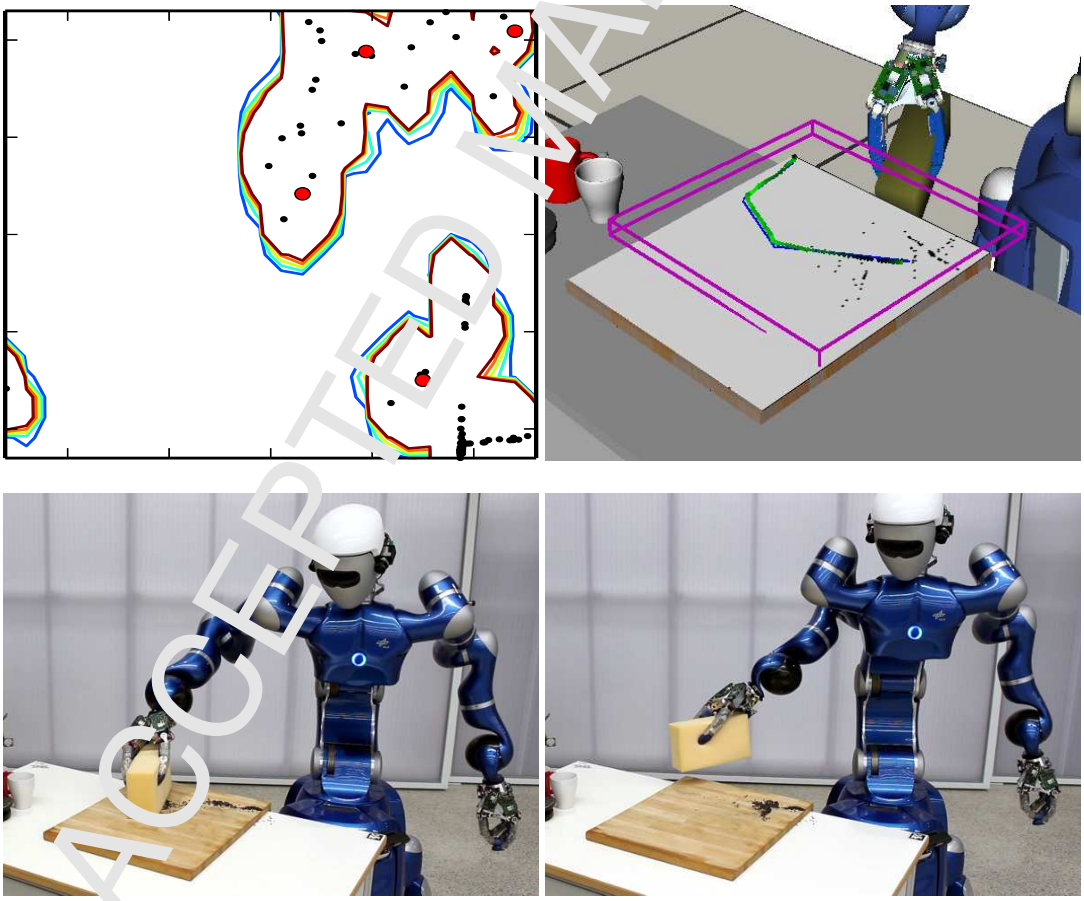


Figure 19: Top left: KDE for the remaining particle distribution after the second failure scenario. Top right: the recorded recovery motion and the final particle distribution. Bottom: The eventually successful real world execution.

the new measurements and infer the improvement qualitatively. In the example at hand, the robot increased the performance for the correctly accumulated particles from 80 % to 97 % w. r. t. the performance metric for collect actions (5). Similarly a recovery plan could be applied to the first failure scenario (after the robot is commanded to relocalize itself) or any other scenario with uncertain contact situations to improve the outcome.

Furthermore, the inference strategy can also be applied in a greedy fashion. That is, the robot would only plan one particular path segment to clean the surface and immediately afterwards interpret the outcome and schedule additional wiping motions according to the updated particle distribution model. This would enable the robot to react on-line on external disturbances.

4.4. Semantic Analysis with openEASE

A cognitive system should not only be able to reason about actions and effects on different levels of abstraction, but also provide the means to communicate its decisions and observations to the operator providing semantically enriched task information. Especially for recurring tasks such as household chores that include wiping actions this is extremely valuable as large amounts of data is generated continuously over long time periods. An operator can neither monitor the process end-to-end nor search through endless data streams to investigate performance errors. Instead, the increasing complexity requires to query the system based on specific interest, such as quality control or safety aspects. Possible queries could include requests like *"visualize all motions with an estimated task performance of 80 % or lower."* or *"highlight all motion segments where a human collided with the robot"*. The *openEASE framework* [47] is equipped with the necessary reasoning mechanisms, visualization techniques, and tool-chains to query episodic memories of robotic manipulation.

In this section, we utilize the openEASE framework to close the cognitive control loop [4] for wiping actions as it was stated in the introduction. The robot telemetry data (i.e. motions, torques, and forces) is therefore augmented with the semantic information about desired and undesired contacts, effects on the particle model and the resulting task performance. These episodic memories are imported into the openEASE framework to make them accessible on a semantic level.

The semantic augmentation of logged data streams allows to query big data based on narratives that are grounded in an ontology. In the example at hand, this allows researchers to query multiple episodic memories of compliant manipulation tasks at once, by formulating relatively simple queries based on symbols related to physical interaction, such as contact or collision. Ideally, these narratives are generated during runtime by the control program of the robot [48]. This is possible if all states and transitions are fully observable as it is common for state machines and symbolic planners. In the example at hand, this would mean that the procedure (e. g. a wiping action) logs its current state based on planned events, such as the expected contact with the table surface. However, as argued in the introduction, this is often impossible due to difficult lighting conditions, imprecise localization, or external disturbances that are unpredictable. As a result, the logged semantics do not correspond to the pattern of the sensor stream. Moreover, many systems lack symbolic task information, such as teleoperated robots for space operations or minimally invasive surgery. To circumvent these issues, we propose to annotate the sensor streams in a post-processing step [49], based on the reasoning mechanisms presented in Sec. 4. To give

Figure 20: The openEASE query visualizing trajectories in collision as yellow trajectory.

an illustration, the openEASE system can be queried to visualize all collision events that occurred during a trial as seen in Listing 1.

Listing 1: Prolog query to request all sponge trajectories for all collision events from openEASE.

```
owl_individual_of(C, semco:'Collision'),    % select all 'Collision' events C
    occurs(C, [S, E]),                      % get start & end times for each C
    M is 0.5 * (S + E),                     % calculate middle time point M
    show(trajectory('sponge')),             % visualize sponge trajectory for
    interval(S, E, dt(0.5))),               % the event, sampled at 2 Hz
    show(justin:'justin_robot1', M).        % visualize robot for time point M
```

The query language in openEASE is based on Prolog [50]. In the example at hand, the episodic memory of a particular trial is queried for all collisions `C`. Given the start time `S` and the end time `E`, one can visualize the trajectory of the sponge at the entire time interval of the collision. Additionally, one can calculate the intermediate time frame `M` to show the configuration of the robot in the middle of the event. The result is visualized in Fig. 20. This relatively short and simple query reveals that the robot was interrupted in the center of the chopping board, which leads to a significantly decreased performance. This information would otherwise be unavailable from feedforward data and hard to identify based on purely numeric telemetry. The integration of the framework into the openEASE system closes the cognitive control loop [4] as it allows the robot to query episodic memories of past executions. In the example at hand, the robot would be potentially able to identify future failure situations on-line by comparing them to the recorded deviations of the end-effector in similar situations, which we plan to investigate in the future. This way, it would also be possible to identify bad localization as it slowly results in lower contact

forces.

5. Conclusion

This work describes an approach to plan wiping motions in the effect-space and infer the effect based on haptic perception, subsequently. (i) A qualitative representation of the medium in wiping tasks was proposed as the basis of the two approaches. (ii) The representation was used to explore the target surface and derive Semantic Directed Graphs (SDGs) which enable a robot to plan Cartesian task motions w. r. t. a desired semantic goal state. (iii) A path following method for the manipulators and the mobile base of the robot was developed to execute the Cartesian wiping motions. (iv) The particle distribution model was utilized to predict and estimate the effect of the robotic cleaning actions w. r. t. haptic feedback information. (v) The approach was extended with a log-likelihood based contact model that allows to simulate different tool-medium-surface combinations. (vi) The proposed method is capable of detecting failure situations occurring from bad localization and human intervention. (vii) We showed that the robot is able to plan additional wiping motions based on the inferred information, in order to successfully accomplish the commanded tasks despite prior failure situations. ($viii$) Eventually, the openEASE framework was integrated to query episodic memories based on the symbolic nature of the task. The approach was tested in simulation as well as in real world scenarios with the humanoid robot Rollin' Justin.

The combination of planning methods and evaluation methods based on a shared effect representation is of great potential. In the particular case discussed in this work, it allows us to assess the efficiency of wiping actions executed by the robot and improve them directly in the effect-space. This feature leads to a significant quality boost as the robot becomes aware of its own performance. As a consequence, the robot may put the quality of its own motions into question, especially if it identifies a failure situation, such as the collision with a human. In light of this, we believe that the contributions of this work present useful additions towards the full automation of effect-oriented compliant robotic manipulation. In the future, we are planning to integrate visual feedback whenever applicable. This could not only provide a better estimate on the task performance, but also be utilized to update the parameters of the dirt particle behavior. This is especially crucial for failure recovery since a biased effect simulation may result in poorly parameterized recovery motions as the KDE is influenced by the residual particle distribution.

The proposed planning methods may be well suited for 3D surfaces. A possible solution to this is provided by [10]. They describe an approach to guide a sponge on 3D surfaces with a robotic manipulator by subdividing the surface into smaller patches that are connected by means of the TSP algorithm. Similarly, SDGs could be generated on 3D surfaces. However, the proposed reasoning method to infer the effect of wiping motions is not directly applicable to 3D surfaces. That is, the qualitative particle distribution model used to simulate the dirt behavior is not based on physical properties. For example, it does not consider gravitational effects or adhesive effects. An example of such a case is provided with the liquid simulated on the solar panel - while the liquid flows down the panel, the particles remain stationary. Accordingly, a more accurate particle model has to be developed before the inference methods can be applied to 3D surfaces.

The insights of this paper allow us to draw conclusions about the relevance of the developed methods w. r. t. related manipulation tasks that require both physical compliance and intelligent

decision making. In general it can be stated that it is valuable to plan a task based on the same representations that are utilized to evaluate the task performance. This way it is possible to improve the outcome directly in the effect-space. One particular example is found in cutting vegetables with a knife as it was presented in [51]. The contact force and the cutting motion show a relation that is comparable to that of wiping actions. Therefore, it might be possible to make assertions about the cutting performance based on episodic memories and enhance the task performance by adapting the parameters of the controller accordingly. By further investigating these topics we hope to foster the development towards a generic concept of intelligent compliant behavior in the future.

6. Acknowledgments

This work was supported by the Bavarian Ministry of Economic Affairs and Media, Enegery and Technology within the SMiLE Project (grant LABAY07), and by the German Research Foundation (DFG) within the Collaborative Research Center EASE (SFB 1320).

References

[1] M. Kawato, Internal models for motor control and trajectory planning, Current opinion in neurobiology 9 (6) (1999) 718–727.
[2] J. J. Gibson, Observations on active touch., Psychological review 69 (6) (1962) 477.
[3] J. R. Flanagan, M. C. Bowman, R. S. Johansson, Control strategies in object manipulation tasks, Current opinion in neurobiology 16 (6) (2006) 650–659.
[4] D. A. Norman, T. Shallice, Attention to Action: Willed and Automatic Control of Behavior., Tech. rep., DTIC Document (1980).
[5] C. Borst, T. Wimböck, F. Schmidt, M. Fuchs, B. Brunner, F. Zacharias, P. R. Giordano, R. Konietschke, W. Sepp, S. Fuchs, et al., Rollin'justin-mobile platform with variable base, in: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2009, pp. 1597–1598.
[6] D. Leidner, W. Bejjani, A. Albu-Schäffer, M. Beetz, Robotic agents representing, reasoning, and executing wiping tasks for daily household chores, in: Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2016, pp. 1006–1014.
[7] D. Leidner, M. Beetz, Inferring the effects of wiping motions based on haptic perception, in: Proc. of the IEEE/RAS International Conference on Humanoid Robots (ICHR), 2016, pp. 461–468.
[8] M. Cakmak, L. Takayama, Towards a comprehensive chore list for domestic robots, in: Proc. of the ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2013, pp. 93–94.
[9] J.-C. Latombe, Robot Motion Planning, Springer, 1990.
[10] J. M. Hess, G. D. Tipaldi, W. Burgard, Null space optimization for effective coverage of 3d surfaces using redundant manipulators, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 1923–1928.
[11] J. Hess, J. Sturm, W. Burgard, Learning the state transition model to efficiently clean surfaces with mobile manipulation robots, in: Proc. of the Workshop on Manipulation under Uncertainty at the IEEE International Conference on Robotics and Automation (ICRA), 2011.
[12] D. Martínez, G. Alenya, C. Torras, Planning robot manipulation to clean planar surfaces, Engineering Applications of Artificial Intelligence 39 (2015) 23–32.
[13] M. Do, J. Schill, J. Ernesti, T. Asfour, Learn to wipe: A case study of structural bootstrapping from sensorimotor experience, in: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1858–1864.
[14] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, M. Inaba, Humanoid motion generation system on hrp2-jsk for daily life environment, in: Proc. of the IEEE International Conference on Mechatronics and Automation (ICMA), 2005, pp. 1772–1777.

[15] K. Okada, M. Kojima, Y. Sagawa, T. Ichino, K. Sato, M. Inaba, Vision based behavior verification system of humanoid robot for daily environment tasks, in: Proc. of the IEEE-RAS International Conference on Humanoid Robots (ICHR), 2006, pp. 7–12.

[16] E. Pablo Lana, B. Vilhena Adorno, C. Andrey Maia, A new algebraic approach for the description of robotic manipulation tasks, in: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 3083–3088.

[17] D. Vanthienen, S. Robyns, E. Aertbeliën, J. De Schutter, Force-sensorless robot force control within the instantaneous task specification and estimation (iTaSC) framework, in: Benelux Meeting on Systems and Control, 2013.

[18] V. Ortenzi, M. Adjigble, K. Jeffrey, R. Stolkin, M. Mistry, An experimental study of robot control during environmental contacts based on projected operational space dynamics, in: IEEE-RAS International Conference on Humanoid Robots (ICHR), 2014, pp. 407–412.

[19] C. Schindlbeck, S. Haddadin, Unified passivity-based cartesian force/impedance control for rigid and flexible joint robots via task-energy tanks, in: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 440–447.

[20] M. Hazara, V. Kyrki, Reinforcement learning for improving imitated in-contact skills, in: Proc. of the IEEE-RAS 16th International Conference on Humanoid Robots (ICHR), IEEE, 2016, pp. 194–201.

[21] A. Gams, M. Do, A. Ude, T. Asfour, R. Dillmann, On-line periodic movement and force-profile learning for adaptation to new surfaces, in: Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on, IEEE, 2010, pp. 560–565.

[22] A. Gams, T. Petrič, M. Do, B. Nemec, J. Morimoto, T. Asfour, A. Ude, Adaptation and coaching of periodic motion primitives through physical and visual interaction, Robotics and Autonomous Systems 75 (2016) 340–351.

[23] Y. Gabriely, E. Rimon, Spanning-tree based coverage of continuous areas by a mobile robot, Annals of Mathematics and Artificial Intelligence 31 (1-4) (2001) 77–98.

[24] L. Kunze, M. E. Dolha, E. Guzman, M. Beetz, Simulation-based temporal projection of everyday robot object manipulation, in: Proc. of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2011, pp. 107–114.

[25] J. Winkler, M. Beetz, Robot action plans that learn and maintain expectations, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 5174–5180.

[26] P. Pastor, M. Kalakrishnan, S. Chitta, E. Theodorou, S. Schaal, Skill learning and task outcome prediction for manipulation, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, IEEE, 2011, pp. 3828–3834.

[27] A. D. Prete, F. Nori, G. Metta, L. Natale, Control of contact forces: The role of tactile feedback for contact localization, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 4048–4053.

[28] S. Denei, F. Mastrogiovanni, G. Cannata, Towards the creation of tactile maps for robots and their use in robot contact motion control, Robotics and Autonomous Systems 63 (2015) 293–308.

[29] S. Stelter, G. Bartels, M. Beetz, Multidimensional time-series shapelets reliably detect and classify contact events in force measurements of wiping actions, IEEE Robotics and Automation Letters 3 (1) (2018) 320–327.

[30] D. Leidner, A. Dietrich, M. Beetz, A. Albu-Schäffer, Knowledge-enabled parameterization of whole-body control strategies for force-sensitive tasks, Autonomous Robots (AURO): Special Issue on Whole-Body Control of Contacts and Dynamics for Humanoid Robots 40 (3) (2016) 519–536.

[31] D. Leidner, C. Borst, A. Dietrich, A. Albu-Schäffer, Classifying compliant manipulation tasks for automated planning in robotics, in: in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 1769–1776.

[32] M. Ghallab, D. Nau, P. Traverso, Automated Planning: theory and practice, Morgan Kaufmann, 2004.

[33] K. D. Forbus, Qualitative process theory, Artificial intelligence 24 (1) (1984) 85–168.

[34] S. LaValle, Rapidly-exploring random trees: A new tool for path planning.

[35] M. Ghallab, A. Howe, D. Christianson, D. McDermott, A. Ram, M. Veloso, D. Weld, D. Wilkins, Pddl—the planning domain definition language, AIPS98 planning committee 78 (4) (1998) 1–27.

[36] R. Graham, P. Hell, On the history of the minimum spanning tree problem, Annals of the History of Computing 7 (1) (1985) 43–57.

[37] S. R. Buss, Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods, IEEE Journal of Robotics and Automation 17 (1-19) (2004) 16.

[38] R. Konietschke, G. Hirzinger, Inverse kinematics with closed form solutions for highly redundant robotic systems, in: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2009, pp. 2945–2950.

[39] A. Huaman, M. Stilman, Deterministic motion planning for redundant robots along end-effector paths, in: Proc. of the International Conference on Humanoid Robots (ICHR), 2012, pp. 785–790.

[40] F. Zacharias, C. Borst, G. Hirzinger, Capturing robot workspace structure: representing robot capabilities, in: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2007, pp. 3229–3236.

[41] P. E. Hart, N. J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Transactions on Systems Science and Cybernetics 4 (2) (1968) 100–107.

[42] D. Leidner, A. Dietrich, F. Schmidt, C. Borst, A. Albu-Schäffer, Object-centered hybrid reasoning for whole-body mobile manipulation, in: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1828–1835.

[43] R. Diankov, Automated construction of robotic manipulation programs, Ph.D. thesis, Carnegie Mellon University, Robotics Institute (2010).

[44] A. Dietrich, T. Wimböck, A. Albu-Schäffer, G. Hirzinger, Reactive whole-body control: Dynamic mobile manipulation using a large number of actuated degrees of freedom, IEEE Robotics & Automation Magazine 19 (2) (2012) 20–33.

[45] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[46] A. Albu-Schäffer, S. Haddadin, C. Ott, A. Stemmer, T. Wimböck, G. Hirzinger, The dlr lightweight robot: design and control concepts for robots in human environments, Industrial Robot: an international journal 34 (5) (2007) 376–385.

[47] M. Beetz, M. Tenorth, J. Winkler, Open-ease–a knowledge processing service for robots and robotics/ai researchers, in: IEEE International Conference on Robotics and Automation (ICRA), Seattle, Washington, USA, 2015, pp. 1983–1990.

[48] M. Beetz, D. Beßler, J. Winkler, J. H. Worch, F. Balint-Benczedi, G. Bartels, A. Billard, A. K. Bozcuoglu, Z. Fang, N. Figueroa, A. Haidu, H. Langer, A. Maldonado, A.-L. Pais, M. Tenorth, T. Wiedemeyer, Open robotics research using web-based knowledge services, in: In Proc. of the International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 5380–5387.

[49] A. Haidu, M. Beetz, Action recognition and interpretation from virtual demonstrations, in: Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ international Conference on, IEEE, 2016, pp. 2833–2838.

[50] D. H. Warren, L. M. Pereira, F. Pereira, Prolog-the language and its implementation compared with lisp, ACM SIGART Bulletin 12 (64) (1977) 109–115.

[51] I. Lenz, R. A. Knepper, A. Saxena, Deepmpc: Learning deep latent features for model predictive control., in: Proc. of the Robotics: Science and Systems Conference (RSS), 2015.

AUTHOR BIOGRAPHIES

Daniel Leidner received his diploma degree in communications engineering in 2010, and his M.Sc. degree in information technology in 2011 with distinction from the Mannheim University of Applied Sciences, Mannheim, Germany. In 2017 he received the Ph.D. degree in artificial intelligence from the University of Bremen, Bremen, Germany. He is currently leading the Semantic Planning Group at the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. In his current research interests he combines artificial intelligence and compliant robot manipulation.

Georg Bartels received M.Sc. and Dipl.-Ing. degrees in computer engineering from Shanghai Jiao Tong University, Shanghai, China and Technical University of Berlin, Berlin, Germany as part of a dual degree program in 2012. In the same year, he joined the Institute for Artificial Intelligence, University of Bremen, Germany as a research scientist. His research interests include plan-based control of autonomous robots, knowledge representation and processing for robots, and mobile manipulation.

Wissam Bejjani received his B.E. degree in mechanical engineering from Notre Dame University (NDU), Zouk Mosbeh, Lebanon in 2013 and his M.Sc. degree in automation and robotics from Technical University of Dortmund, Dortmund, Germany as a visiting researcher at the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany in 2015. He is currently affiliated with the SensorimoleRobots research group, in the School of Computing at the University of Leeds, His current research interests include learning from demonstration and physics-based manipulation planning.

Alin Albu-Schäffer received the diploma degree in electrical engineering from the Technical University of Timisoara, Timisoara, Romania, in 1993, and the Ph.D. degree in control systems from the Technical University of Munich, Munich, Germany, in 2002. Since 2012 he has been Head of the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany, which he joined in 1995. His research interests include robot design, modeling, and control, nonlinear control, flexible joint and variable compliance robots, impedance and force control, and physical human-robot interaction.

Michael Beetz received his diploma degree in Computer Science with distinction from the University of Kaiserslautern, Kaiserslautern, Germany. He received his M.Sc., M.Phil., and Ph.D. degrees from Yale University, New Haven, CT, USA in 1993, 1994, and 1996 and his Venia Legendi from the University of Bonn, Bonn, Germany in 2000. Since 2012, he has been Head of the Institute for Artificial Intelligence, University of Bremen, Bremen, Germany. His research interests include integrated cognition-enabled robotic systems, plan-based control of autonomous robots, knowledge representation and processing for robots, integrated robot learning, and cognitive perception.