Robotic Agents Representing, Reasoning, and Executing Wiping Tasks for Daily Household Chores

Daniel Leidner[1], Wissam Bejjani[1], Alin Albu-Schäffer[1], and Michael Beetz[2]
[1] German Aerospace Center (DLR), Institute of Robotics and Mechatronics, Weßling, Germany
{daniel.leidner, wissam.bejjani, alin.albu-schaeffer}@dlr.de
[2] University of Bremen, Institute for Artificial Intelligence, Bremen, Germany
beetz@cs.uni-bremen.de

ABSTRACT

Universal robotic agents are envisaged to perform a wide range of manipulation tasks in everyday environments. A common action observed in many household chores is wiping, such as the absorption of spilled water with a sponge, skimming breadcrumbs off the dining table, or collecting shards of a broken mug using a broom. To cope with this versatility, the agents have to represent the tasks on a high level of abstraction. In this work, we propose to represent the medium in wiping tasks (e. g. water, breadcrumbs, or shards) as generic particle distribution. This representation enables us to represent wiping tasks as the desired state change of the particles, which allows the agent to reason about the effects of wiping motions in a qualitative manner. Based on this, we develop three prototypical wiping actions for the generic tasks of absorbing, collecting and skimming. The Cartesian wiping motions are resolved to joint motions exploiting the free degree of freedom of the involved tool. Furthermore, the workspace of the robotic manipulators is used to reason about the reachability of wiping motions. We evaluate our methods in simulated scenarios, as well as in a real experiment with the robotic agent Rollin' Justin.

Keywords

AI Reasoning Methods, Task and Effect Representation, Robotic Manipulation Planning, Service Robotics

1. INTRODUCTION

Research in controlling robotic agents has for a long time been dominated by the study of robot actions that are inspired by the seminal blocks world problem that was introduced with the Shakey system [24]. Even today most research efforts that realize activity on autonomous robots do so in demonstrating scenarios that mostly include only very simple fetch-and-place tasks. The fetch-and-place actions considered in those research efforts so far are very simple in nature. They can be considered as actions that have discrete instantaneous effects. For the execution of the task the only constraint to be satisfied is to find collision free paths to reach, pick up, and transfer the object to the destination. Consequently, robotic agents that reason about these actions can abstract away from how actions are executed and model them using relatively simple action models consisting of preconditions and effects [10].

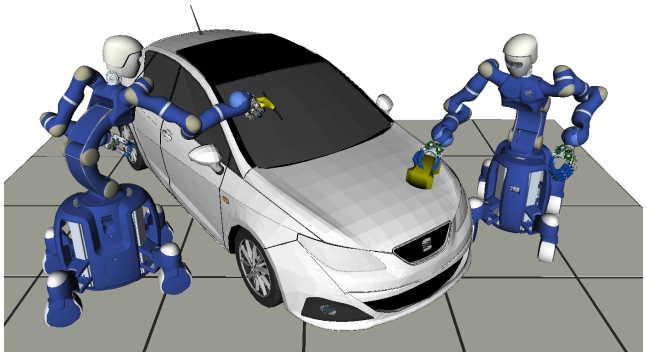

Figure 1: Illustration of the robotic agent Rollin' Justin skimming the windshield of a car and scrubbing the hood with a sponge.

In contrast, the research roadmaps for autonomous robotic agents [4, 23] envisage robotic agents manufacture products in close physical cooperation with their human co-workers or in the more distant future as helpers that enable old and disabled citizens to live their lives more independently. Such robotic agents would have to be capable of taking over a substantial part of the human daily household chores.

Cakmak et al. [3] collected and classified the manipulation actions that characterize the human household chores based on their semantic similarity and identified that almost half of them are based on wiping of surfaces located in rooms, on furniture, or other objects. Most of these tasks are related to cleaning, such as dusting furniture with a feather duster, sweeping breadcrumbs from the kitchen countertop, or collecting shards of a broken mug with a broom.

Compared to fetch-and-place tasks, wiping tasks demand enormously rich and complex actions. Depending on **how** the wiping is done in terms of force applications and motions the wiping can remove sticky dirt, dry a wet surface, or paint an area. This means that the same action can produce very different effects depending on how it is executed. In order to perform the respective tasks successfully the robotic agent must carefully select the motion and force parameters in a continuous parameter space. Moreover, the successful execution of wiping tasks requires a substantial amount of geometric reasoning as the purpose of wiping might be the collection, the spreading, or the elimination of particles.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

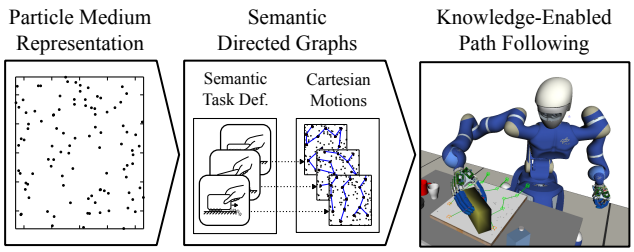

Figure 2: Overview of the proposed framework.

In our earlier work, [20] we proposed to classify wiping tasks based on the set of parameters describing the semantic relation between tools and the environment. In particular, we classify wiping tasks based on the relation between the tool, the surface, and the medium to be manipulated. The tool orientation, and the direction of motion are thereby crucial aspects for the task performance. For example, cleaning the body of a car can be solved by repeatedly moving a sponge in random orientations and directions along the car surface, while skimming snow from a windshield is only successfully if each wiping motion is directed towards the edge of the window, having the tool properly aligned as it is illustrated inf Fig. 1.

In this paper, we investigate the reasoning and action execution problems that enable robotic agents to successfully carry out abstractly specified wiping tasks. We propose to represent wiping tasks on a high level of abstraction and develop generalized action execution mechanisms that are able to ground these high-level task descriptions to low-level robotic control programs. Our method constitutes an effect-oriented reasoning approach to manipulate the medium in wiping tasks, i. e. liquids or particles, in a generalized manner. Our contribution include (i) a qualitative representation of the medium in wiping tasks based on a particle distribution; (ii) an effect-oriented approach to reason about the desired effects of wiping tasks, i. e. the state change of the medium, by means of Semantic Directed Graphs (SDG); (iii) a knowledge-enabled path following approach to resolve Cartesian tool motions into joint motions of the agent; and (iv) an extended approach that integrates reasoning about the reachability of the agent to solve wiping tasks in wider areas by utilizing capability maps. In conclusion, we enable robotic agents to reason about the effect of wiping tasks on a high level of abstraction and enable the agents to execute the matching goal oriented joint motions for the tasks of absorbing, collecting and skimming. The methods are evaluated in three simulated scenarios, where the target surface is defined by (a) a chopping board, (b) the surface of a table, and (c) the windshield of a car. Additionally, we show experiment (a) in a real world setting with the humanoid robot Rollin' Justin successfully collecting breadcrumbs on a chopping board with a sponge.

An overview of the proposed framework is provided in Fig. 2. A particle based medium representation serves as the basis of our reasoning methods (Sec. 3). It is used to describe the desired state change, i. e. the semantic goal states and derive Cartesian tool motions for the prototypical wiping tasks investigated (Sec. 4). This Cartesian motions are resolved to whole-body joint motions (including the mobile base of the agent) by exploiting the free Degrees of Freedom (DOF) of the involved tool (Sec. 5).

2. RELATED WORK

Automated manipulation planning with a universal robotic agent in domestic environments has ever since been a major goal in robotics research, as well as in research on autonomous agents. Recently, this field has regained momentum thanks to affordable and safe light weight robots such as the LBR III [14]. These compliant manipulators enable a robotic agent to get in deliberate soft contact with their environment, which is a key element for many manipulation tasks, including wiping of surfaces.

Especially cleaning related wiping tasks have been investigated in detail lately. Urbanek *et al.* [30] demonstrated how machine learning can be used to teach a robot different movement primitives by demonstration in the context of wiping a table. Do *et al.* [7] solve the inverse problem of predicting appropriate action parameters by learning from experience during wiping tasks. Vanthienen *et al.* [31] describe table wiping tasks as a set of constraints with the *iTaSC* framework. Okada *et al.* [26, 25] 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.* [28] represent robotic manipulation tasks in an algebraic form, which incorporates poses, velocities and forces in a simulated window cleaning task. In robotics, wiping tasks are often investigated from a control theoretic point of view. Ortenzi *et al.* [27] propose to exploit the environment contact constraints of wiping tasks in the operational space, to decouple the motion of the robotic agent from the applied force. Schindlbeck and Haddadin [29] utilize task-energy tanks to react safely upon contact loss.

The task of wiping a surface is often considered as a *coverage path planning* problem [18], where an agent has to find a path (i. e. for a cleaning device) connecting all nodes of a graph in a time- or effort-optimal way. Gabriely and Rimon [9] consider the coverage problem arising for a mobile robot such as a autonomous lawn mower. They propose to subdivide the search space into a grid and apply variations of the *Spanning Tree Covering (STC)* algorithm to cover the area. Hess *et al.* [13] 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 nulls-space optimization in a discretized Jacobian null-space along the Cartesian path (see [15]). While the work of Hess *et al.* [13] is most related to our work, it does not integrate different semantic goals and is therefore only applicable to undirected tasks, such as vacuuming or dusting.

The approaches listed so far mainly focus on the physical part of the problem, while mainly ignoring the semantic meaning of the motion. In contrast, Kunze *et al.* [17] 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. In our earlier work [22], we utilized a knowledge-based approach to formulate a window cleaning task on a high level of abstraction. The approach combines the desired semantic effect of the task with a concrete geometric process model, which integrates the parameterization of the low-level control program of the agent. However, the Cartesian tool motions are pre-defined properties of the manipulated objects (e. g. the window pane). Consequently, the

agent cannot adapt to new situations in case of unforeseen obstacles. In this work we extend this approach by reasoning autonomously about the appropriate tool motions w. r. t. the desired semantic effect and the environmental state.

3. WIPING EFFECT REPRESENTATION

This work is based on our classification of compliant manipulation tasks [20]. The classification is based on the semantic contact situation between manipulated objects and the environment. Wiping tasks are thereby represented based on the geometric relation of the tool, the surface and the medium to be manipulated. This taxonomy will be used in this work to develop an abstract representation of wiping motions to be executed by a robotic agent. In particular, we will investigate the removal actions *absorbing*, *collecting*, and *skimming* the medium as illustrated in Fig. 3.

The medium in wiping tasks, as defined in [20], is representative for arbitrary liquids or particles with different properties. To reason about wiping motions in a generalized form, a medium representation has to incorporate the properties of arbitrary media. For example, the medium in absorption tasks may be a variation of dust or dirt or a liquid. An example for the medium in skimming tasks is water or detergent on a window, or snow on a windshield. Shards of a broken mug, leaves, or rubble are exemplary for collecting tasks. To incorporate these different types of media, we propose a qualitative model based on a particle distribution estimation on a planar target surface

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

where N particles (x_i, y_i) are estimated within the boundaries of the target surface (x_{min}, x_{max}, y_{min}, y_{max}). An example for such a distribution is provided in Fig. 4 which resembles a kitchen scenario with bread crumbs on a chopping board. In most real world scenario, the dirt distribution can be provided by a visual perception system if the particles are big enough to be perceived. However, especially dust and dirt is very hard to perceive and the real distribution on the surface can hardly be modeled. Instead, we assume a unified distribution in this case.

The particle distribution is utilized to simulate the desired semantic *state change* $S_{t_0} \rightarrow S_g$ provided as desired effect in PDDL syntax [10], e. g. *(`collected` breadcrumbs chopping_board)*. The respective estimated change of the particle distribution $\boldsymbol{P}_{t_0} \rightarrow \boldsymbol{P}_g$ is visualized in Fig. 4. The contact model considers the exact CAD data of the tool and the position of each particle. Depending on the tool type and the properties of the medium (i. e. liquid or solid particles), the particles react with different effects upon contact. 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. In case of solid particles, the contact with the sponge moves the particles in parallel to the direction of motion. We use this behavior for the collect and the skim action. The state change of the particle distribution enable the agent to qualitatively reason about the effect of its motions [8], and subsequently the task performance used to evaluate our methods in Sec. 6.

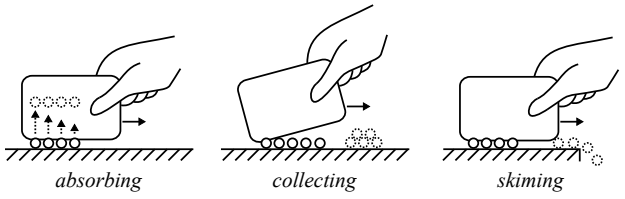

Figure 3: The three prototypical removal actions identified in the classification of wiping tasks. The initial state of the medium is pictured as solid circles, the desired state as dashed circles.

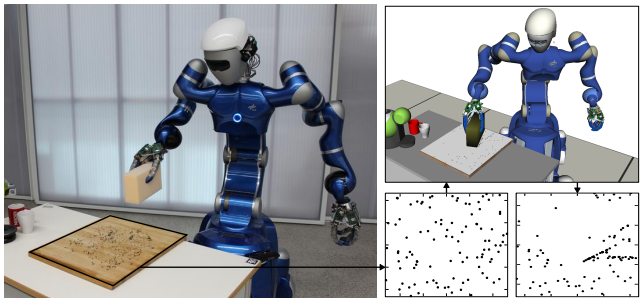

Figure 4: Change estimation of the particle distribution in contact with the sponge during wiping.

4. REASONING WIPING MOTIONS WITH SEMANTIC DIRECTED GRAPHS

Wiping tasks can be considered as a coverage path planning problem. The removal actions to be investigated in this paper have to cover the entire particle distribution and simultaneously satisfy the semantic goal state. To achieve this, we utilize *Semantic Directed Graphs (SDGs)*

$$\boldsymbol{SDG} = f(\boldsymbol{P}, S_g, G_s), \tag{2}$$

which incorporate the coverage problem for the particle distribution $\boldsymbol{P}$, the semantic goal state S_g of the respective action, and the geometric state G_s of the environment. SDGs define a graph structure projected on the planar target surface to be wiped, where

- each node n_i represents a waypoint of the Cartesian tool motion w. r. t. the *Tool Center Point (TCP)*,
- the edge (n_i, n_{i+1}) in between two nodes represents the interpolated tool motion in contact with the surface.

The surface coverage, i. e. the node distribution, is thereby dependent on the current state of the geometric environment G_s, which assembles the volumetric model for geometric planning of wiping motions. Accordingly, a collision avoidance strategy based on a *collision sphere model* is utilized to explore the target surface as visualized in Fig. 5. This sphere is of variable diameter $d_s = ||\boldsymbol{D}_{aoe}||$, where $\boldsymbol{D}_{aoe}$ is the dimension of the area of effect of the tool. We do not consider the full extend of the tool, since this might be much bigger than the actual required space. For example. vacuuming under a bed requires to move the nozzle of the tool to the desired goal region. Although the vacuum cleaner does not fit under the bed, the nozzle is clear to reach the dust.

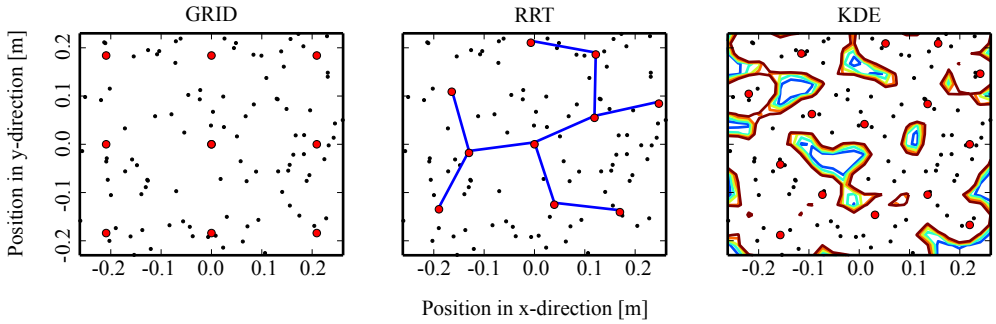

Figure 6: The coverage strategies utilized to explore the target surface.

We have investigated three different coverage strategies to distribute the initial set of graph nodes, namely a discretized grid (GRID), Rapidly Exploring Random Trees (RRT) [19], and a Kernel Density Estimation (KDE), as illustrated in Fig. 6, where red dots mark the nodes of the graph. The coverage strategies are compared below.

Discretized Grid: The first coverage strategy constitutes a simple grid heuristic within the bounds of the target area. The radius, $r_s = d_s/2$, of the collision sphere is used to calculate the grid resolution, given the dimension of the target surface. This coverage strategy is uninformed and can also be applied if no prior knowledge on the particle distribution is available.

Rapidly Exploring Random Trees: RRTs [19] are a well established generic method in research on path planning and exploration. In a nutshell, 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 $\boldsymbol{q}_{delta}$ is defined by the radius r_s. As the algorithm is biased to explore uncovered regions, it is predestined to generate the initial nodes for wiping tasks. However, RRTs do not necessarily yield goal oriented paths w. r. t. the desired semantic goal, especially considering the skim action. Therefore, we do not use the tree developed structure to derive wiping motions, but only the nodes that were created during the expansion.

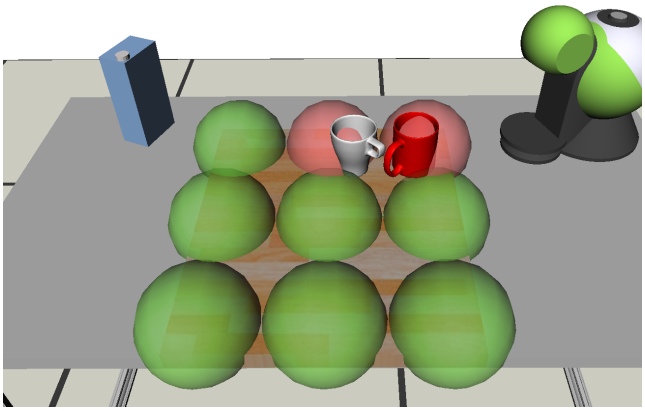

Figure 5: An exemplary geometric state G_s. The collision sphere is utilized to verify the SDG nodes in a grid distribution, where green is valid and red in collision.

Kernel Density Estimation: As a third strategy, we apply a Gaussian KDE to estimate the regions with high particle probability within the particle distribution $\boldsymbol{P}$:

$$\boldsymbol{K}(\boldsymbol{x}) = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} e^{-||\boldsymbol{P}_i - \boldsymbol{x}||^2/h^2} \tag{3}$$

where h is the bandwidth of the kernels. The multivariate KDE is visualized as a contour plot on the left in Fig. 6. The resulting continuous representation is used to select the M most significant peaks. This approach is most valuable if prior knowledge about the distribution is available, e. g. perceived by a vision system. The nodes are naturally placed at the position with the highest effect.

The calculated nodes n_i serve as the starting point to grow SDGs with different semantic goals S_g. As the coverage strategy may influence the task performance, we will evaluate the execution time and task performance of the resulting wiping motions in Sec. 6. In the following we will use the distribution generated by the KDE to outline the graph generation for the three removal actions.

Every wiping action can be described as the desired state change of the medium in an abstract form. We represent these state changes as *constraints* for the set describing the goal particle distribution $\boldsymbol{P}_g$, as listed in (4), (5), and (6). For example, the goal state of absorbing is to remove all particles from the surface, by simply getting in contact with each particle to trigger a delete operation. Collecting and skimming, however, require directed tool motions to have the medium moved towards a certain goal area (collect), or moved from the edge of the surface (skim), respectively. The geometric process models of these actions have to correspond to this state change, respectively the desired effect to the particle distribution $\boldsymbol{P}$ as illustrated in Fig. 7. Based on our previous investigations [20], these process models are designed in accordance with the tool, the surface, and the medium to be manipulated as hand crafted Cartesian task motions are hardly applicable in most of the cases. For example, absorbing dust from a sideboard using a feather duster may be represented as a set of Cartesian straight line motions along the sideboard surface, however, if it is obstructed with obstacles, e. g. a television, flowers, and books, these simple representation is not anymore applicable. Having generated, instead of a hand crafted task space paths one can overcome this issue. To this end, we implement three prototypical strategies to ground the semantic actions corresponding to the three investigated semantic goal states S_g, i. e. (`absorbed` *?m - medium ?s - surface*), (`collected` *?m - medium ?s - surface*), (`skimmed` *?m - medium ?s - surface*). Accordingly, SDGs embody one of the following actions.

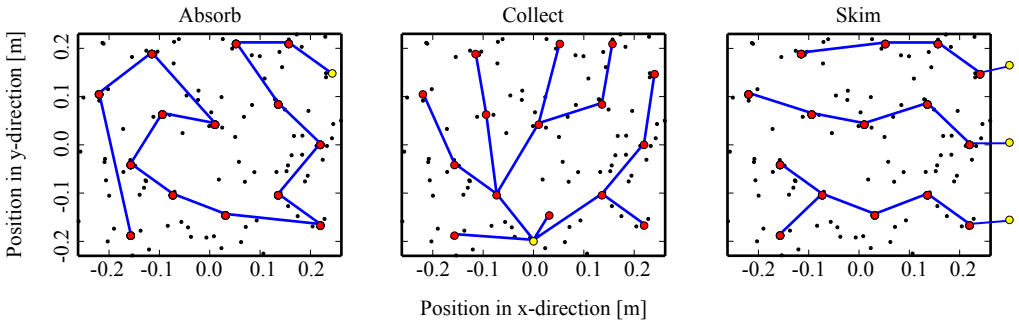

Figure 7: SDGs for the three prototypical removal actions absorb, collect, and skim.

Absorb: The first action to be investigated is the absorb action, as it occurs e. g. in vacuuming, dusting, or soaking up water with a sponge. The effect to the particles (i. e. the absorption) is independent of the direction of motion, as long as the entire region is covered by the tool at least once to remove all particles, such that

$$\boldsymbol{P}_g = \emptyset. \tag{4}$$

A possible solution to this issue is the *Traveling Sales Person (TSP)* algorithm. Hess *et al.* [13] showed that this is a performant approach to solve unconstrained wiping tasks in a generalized way. We apply a variation of the TSP to absorbing, as seen in Fig. 7. The outcome is a natural curved motion covering all nodes of the graph.

Collect: Collecting the medium constitutes the second removal action. The particles are modeled to be pushed upon contact. The desired goal state S_g can be geometrically represented as a single goal region on the target surface. This region is defined by an additional node n_g added to the SDG (yellow dot). The goal is currently given to the robotic agent by the operator but may be inferred from the current world state in future. All branches of the graph have to be directed towards this single goal node, such that

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

To do so, we implement a *Minimum Spanning Tree (MST)* [11] for a single root node. The distance between nodes serves as cost function with a maximum connection length $l_{max} = d_s$.

Skim: The third SDG strategy is designed to solve arbitrary skimming tasks. In principle, skimming can be related to collecting. The particles are pushed away upon contact with the tool. The semantic goal S_g of the action corresponds to a geometric state where all particles are pushed off the target surface, such that

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

We implement this action as a collecting action with multiple goals, i. e. multiple MSTs. These goals are represented by multiple virtual goal nodes equally distributed along the edges of the target surface. Depending on the target object, only a subset of the available edges may be used. In the scenario at hand, only the edge parallel to the table edge is a valid goal region (right in Fig. 7). As a result, multiple trees expand towards the upper edge of the surface. Each tree forms thereby a MST w. r. t. the closest goal node.

5. RESOLVING WHOLE-BODY WIPING MOTIONS FOR A ROBOTIC AGENT

So far, we have discussed the problem of wiping only from a Cartesian point of view. In particular, SDGs only consider translational paths of a spherical body along the target surface w. r. t. a desired semantic goal. This is only a first estimate for the feasibility of the action. The joint state of the robotic agent, as well as the orientation of the tool are yet to be integrated into the reasoning process. The highly redundant robotic agent is able to resolve the Cartesian tool motions in manifold ways, where some configurations result in local minima, while others allow to follow the desired path. A similar problem is observed for the tool orientation. In some cases it is better to rotate the tool to obtain a better reachability in favor of a decreasing task performance, i. e. a poorer cleaning result. These issues are discussed in details in the following sub-sections.

5.1 Generate Joint Motions

The branches of the SDG represent Cartesian tool motions that serve as the basis for the joint motions of the agent. 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 agent has to find a joint configuration $\boldsymbol{q}$, respectively $\dot{\boldsymbol{q}}$

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

where $\boldsymbol{J}^\dagger$ is the generalized inverse of the Jacobian matrix (e. g. the Moore-Penrose pseudoinverse [2]). The first term on the right minimizes the norm of the joint distance and the second term exploits the redundancy of the robotic agent to satisfy secondary criteria, such as collision and singularity avoidance. For instance, the joint velocity $\dot{\boldsymbol{q}}$ and the joint acceleration $\ddot{\boldsymbol{q}}$ must not exceed the limits of the robotic manipulators and the resulting joint path $\boldsymbol{Q}$ must not collide with obstacles in the environment nor the agent itself. This problem is especially challenging for highly redundant robotic manipulators with a high number of DOF, such as the robotic agent Rollin' Justin [1].

A possible solution to this problem has been proposed by Konietschke and Hirzinger [16]. They utilize an inverse kinematics solver based on non-linear optimization. The algorithm enables Rollin' Justin to track arbitrary Cartesian trajectories on-line. The algorithm does not consider collisions with the environment, nor with the agent itself. Moreover, since it is based on local optimization techniques, it might diverge towards local minima, i. e. singularities in the joint space. A deterministic path following approach

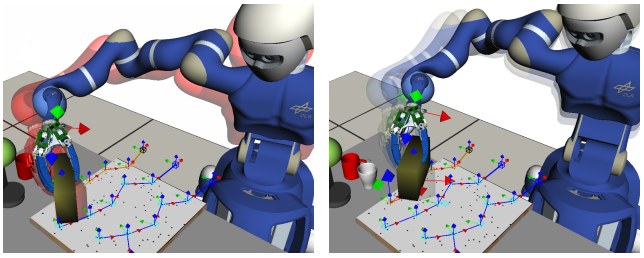

Figure 8: Initially, each node is directed towards the next node in the graph. If this pose is in collision (left), or unreachable, e. g. due to joint limitations (right), the free DOF of the tool are exploited.

was proposed by Huaman and Stilman [15]. They propose a resolution complete solution that exploits the redundancy in discretized Jacobian null-space of robotic manipulators. The algorithm implements a breadth-first backwards search procedure when encountering local minimas or obstacles in the environment. Compared to the former method, the drawback of this solution is the high computation time. Both methods try to exactly follow the Cartesian path, where all six dimension (i. e. three translational dimension and three rotational dimension) of the task are considered. This is not necessarily mandatory for wiping tasks. For example, a sponge can be rotated w. r. t. the normal of the target surface, without decreasing the area of effect. On the other hand, a window wiper has to be moved orthogonal to the wiper blade in order to achieve the desired effect, but it is allowed be rotated along the main axis of the blade, up to a certain degree. To this end, we propose a path following method that is aware of the free tool DOF available in the Cartesian space. Furthermore, we propose to exploit local path following methods whenever possible, and utilize global search-based methods only if no other alternative is feasible to decrease the overall planning time.

The sequence in which the SDG branches are processed depends on the wiping action. The absorb motion is unambiguous since it has only one branch to follow from start to end. For each branch, in the collect and skim actions, the Cartesian space is resolved iteratively starting from a leaf node and expanding backward to the branch's root node. Initially, the orientation of the nodes n_i is directed towards the next node n_{i+1} in the branch. The resulting pose, i. e. the translation and rotation of the node, is used as initial hypothesis for the goal pose of the tool (see Fig. 8). The edge in between the nodes is interpolated to resolve the path following task outlined in (7). We apply the local path following method described in [16], with additional checks for collision between the agent and the environment. Ideally, all poses are reachable and collision free so that the agent can manipulate the tool accordingly. The default orientation $\boldsymbol{x}_{goal,0}$ is most effective w. r. t. the manipulation of the medium, i. e. the particle distribution. However, in a cluttered environment it is likely that the edge between two nodes cannot be tracked without collision, or the robotic agent falling into a local minima. In this case we backtrack to the initial pose $\boldsymbol{x}_{start}$, select an alternative goal pose $\boldsymbol{x}_{goal}$ w. r. t. the free DOF of the tool, and repeat the path following task. The procedure is outlined in Algorithm 1.

The procedure is repeated until all alternative combinations in the discretized search space of the free DOF are evaluated. If none is successful, the path that had the largest part of it solved by the local method is selected to be solved using the global method described in [15], which is guaranteed to find a path if one exists (given enough time). If no feasible inverse kinematics solution is found at all, the agent abandons the node and moves on to the remaining nodes of the branch. Another solution to this issue would be to alter the Cartesian graph at this point until it can be resolved by the agent, e. g. by displacing the goal node or changing the curvature of the graph edge.

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 continous joint path $\boldsymbol{Q}$

```
x_start ← CalculateToolPose(q_{n_i}, x_grasp^{-1})
foreach x_goal in IterateFreeDOF(n_{i+1}) do
    Q ← List()
    X ← Interpolate(x_start, x_goal, δ)
    foreach x_i in X do
        x_eef,i ← x_i · x_grasp
        q_i ← FindIK(x_eef,i)
        if IsValid(q_i) then
            Q[i] ← q_i
        else
            break
    if Length(Q) = Length(X) then
        return Q
```

5.2 Extended Semantic Directed Graphs

Up to this point, we have considered wiping tasks as globally realizable in terms of reachability, albeit it is quite common for wiping tasks to cover larger regions, which may be unreachable for a static manipulator. For example, if the agent is commanded to clean a whole table surface, the larger task space requires to move the mobile base. A mobile robotic agent is able to reposition itself to overcome this issue and execute tasks in extended areas. We incorporate this by utilizing *extended Semantic Directed Graphs (eSDG)*, which augment the graph nodes n_i with reachability information to reposition the base of the robotic agent.

One approach to position a robotic manipulator optimally w. r. t. the reachability of a certain task is the use of *capability maps* [32]. We have already utilized capability maps to initially optimize the base position of a robotic agent in front of a windowpane, in order to clean it [22]. While this provides a solution for the initial positioning problem, the base motions during the task execution are handcrafted. We incorporate the reachability information for single graph nodes in eSDGs by utilizing capability maps. This way, we enable the agent to optimize its base positions for sub-graphs of eSDGs, respectively the underlying particle distribution. Capability maps represent the workspace of robotic manipulators in a discretized grid. For each voxel in the grid, the reachability index $r = \frac{R}{N}$ is computed, where N is the maximum number of hypothetically reachable positions, and R the actual reachable positions [32].

The sub-graphs of eSDGs, here called cluster, are computed according to Algorithm 2. Initially, all nodes are associated to one cluster $\boldsymbol{C}$. The agent computes the base

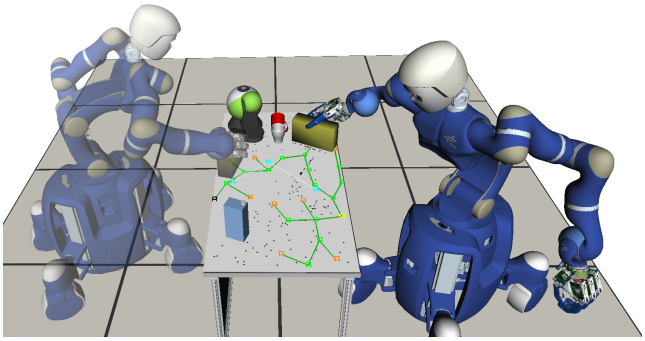

Figure 9: The eSDG collect graph consists of three clusters. As the goal is located on the right (yellow node), the clusters on the left are resolved first.

position by means of the capability maps in order to reach all nodes in the cluster. For every node n_i, the reachability index is obtained for all the available alternative poses of the end-effector $\boldsymbol{x}_{eef,i} = \boldsymbol{x}_{n_i} \cdot \boldsymbol{x}_{grasp}$. If the mean reachability for these poses is higher than $0.5\, r_{\max}$, the node is considered as reachable [22]. If all nodes are reachable, the cluster and the corresponding base position is stored. If a sub-set of nodes is unreachable the cluster is split into two new clusters. This is done using the K-Means algorithm with $K = 2$ and the two centers initialized as the centers of the reachable, and unreachable nodes, respectively. This approach yields large connected areas instead of smaller scattered regions. The algorithm is recursively executed.

The clusters are considered during the reasoning process of the wiping action. Each cluster, resembles the strategy of the main graph, i. e. absorbing, collecting and skimming. The clusters for the absorb action can be executed independently. Collecting requires to align the clusters towards the goal node of the eSDG. Each sub-graph implements the collect strategy with an intermediate goal. For skimming, the cluster on the edge of the target surface may also be considered independently. If the cluster is not connected to the edge of the target surface, the skimming paths are directed to the closest cluster with a connection to the surface edge. We utilize an A* algorithm [12] to navigate between the base positions of the clusters. As an example, the eSDG collect strategy is visualized in Fig. 9.

6. EVALUATION AND CONCLUSION

We evaluate the proposed methods in three different scenarios. Scenario I constitutes the chopping board scenario illustrated in Fig. 4. This scenario is executed by the humanoid robotic agent Rollin' Justin as illustrated in Fig. 10. The proposed reasoning methods command a whole-body impedance control framework [6] which enables the agent to compliantly interact with the environment. Besides the pure motion aspect discussed in this work, we have conducted detailed research efforts on the knowledge-based parameterization of compliant contact behavior [22], [21]. Among others, the desired Cartesian force, the Cartesian stiffness and the damping are parameterized according to the requirements of the tool, the surface and the manipulated medium.

For scenario II, we assume the same environment without the chopping board, where the particles are distributed on the entire table surface as shown in Fig. 9. Scenario III is a variation of the car cleaning example visualized in Fig. 1,

Algorithm 2: ExtendSDG($\boldsymbol{C}$, $\boldsymbol{eSDG}$)

Input: The initial cluster $\boldsymbol{C}$, and the $\boldsymbol{eSDG}$ structure
Output: The $\boldsymbol{eSDG}$ structure, including a list of clusters $\boldsymbol{C}$, and their base positions $\boldsymbol{x}_{base}$

```
x_base ← OptimizeBasePose(C)
R, U ← List()
foreach n_i in C do
    r_i ← CalculateReachability(n_i, x_grasp)
    if r_i >= 0.5 r_max^1 then
        R[i] ← n_i
    else
        U[i] ← n_i
if U = ∅ then
    eSDG.append([C, x_base])
else
    foreach C_i in KMeans(C, Centers(R, U)) do
        eSDG ← ExtendSDG(C_i, eSDG)
return eSDG
```

where we approximate a planar target surface aligned with the windshield. In all scenarios the agent manipulates the sponge illustrated in Fig. 4 to obtain comparable results. We utilize the eSDG approach to incorporate the optimization of the base position for all scenarios. We evaluate the three coverage strategies (i. e. GRID, RRT, KDE), paired with the three removal actions (i. e. absorb, collect, skim), w. r. t. the traveled Cartesian distance during contact, the computation time, the execution time, and the task performance. The calculation time includes the node distribution, the cluster computation, and the path following. For all steps the collision check time using OpenRAVE [5] is included. 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 performance measurement is conducted according to the constraint definitions (4), (5), and (6). The metric for the absorb action is 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, and the metric for the skim action is the number of particles pushed outside the boundaries of the surface area. All numbers are averaged over five trials with different particle distributions. The results are listed in Table 1. A video of the task executions is available at `https://youtu.be/ZA2U9upkrjo`.

In the lesser obstructed chopping board scenario, as well as in the windshield scenario, the KDE coverage strategy mainly outperforms the other two methods w. r. t. the task performance. In general, this indicates that the KDE approach is favorable for tasks with a detailed model of the medium. However, the KDE strategy is noticeably biased by the obstacles in the table scenario, which results in many collisions and unreachable, i. e. unconnectable nodes, which is also reflected by the poorly developed path lengths. The GRID heuristic and the RRT coverage strategy perform similarly good for most of the cases, where the GRID strategy generates reproducible motions independent of the particle estimation or probabilistic effects.

In general, the proposed particle distribution is a suitable representation to estimate the outcome of wiping motions in compliant contact. The task execution illustrated in Fig. 10 shows that the outcome in a real environment matches the

Table 1: Results for 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	52.485	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	4.95	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	122.76	102.48	108.18	103.29	96.03
	performance [%]	88.00	80.25	97.00	71.75	66.50	53.00	94.50	88.50	88.50

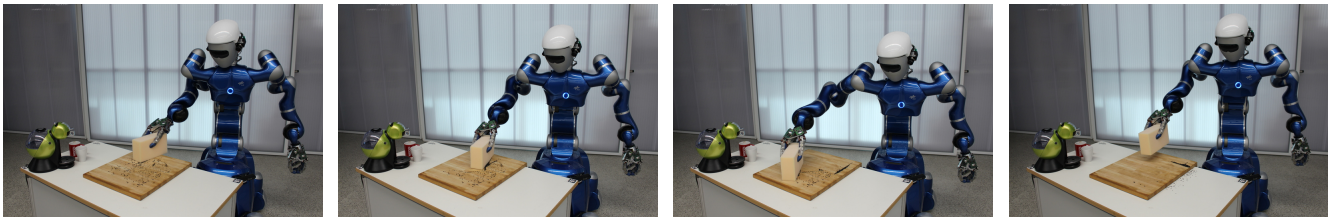

Figure 10: The robotic agent Rollin' Justin accomplishing the collect task in the chopping board scenario. The outcome is similar to the prediction obtained with the particle distribution representation.

expectations of the simulated behavior of the particles for small grain dirt particles. It is therefore applicable to infer the effect of wiping motions generated with the different coverage strategies and measure the task performance w. r. t. the constraint definition for the three removal actions. In conclusion, our approach enables a robotic agent to reason about the task performance of wiping actions as it is aware of the desired state change of the particle distribution. This allows to decide for the most effective strategy to clean a surface given a certain problem instance.

The task performance for a specific scenario is highly dependent on the selected coverage strategy. While this is a drawback for onetime tasks, this observation can be exploited to improve 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 time by iterating over the available coverage strategies. Moreover, the agent can continuously improve on the task performance by integrating episodic memories of previous executions into the reasoning process. For example, the task performance of previous trials can be used to benchmark future trials with alternative task parameters, e. g. a different tool alignment or a different force profile. The generic approach can be utilized to solve previously unseen wiping tasks, only given the desired semantic goal and the geometric properties of the tool and the environment. To this end, we believe that our approach constitutes a valuable addition for autonomous robotic agents.

Building on our findings, we plan to extend our research on compliant manipulation towards the closure of the semantic feedback loop based on visual and haptic perception and the integration of an appropriate uncertainty model to incorporate bad contact situations. In the case of wiping tasks, this will allow the agent to reason about the real effect of its actions and the resulting task performance, update the qualitative medium representation, and schedule additional wiping motions to improve the cleaning result.

7. SUMMARY

We described a qualitative reasoning approach to solve everyday wiping tasks with a robotic agent on a high level of abstraction. We proposed (*i*) a qualitative representation of the medium in wiping tasks (*ii*) in combination with Semantic Directed Graphs (SDG) which enable the agent to plan Cartesian task motions w. r. t. the desired semantic goal state, i. e. the state change of the particle distribution. Furthermore, we proposed (*iii*) a path following method that is aware of the free DOF of the tool and the redundant joint space of the agent to allow for a robust task execution. Eventually, (*iv*) we integrated reachability information into the reasoning process to extend our approach towards wide area tasks. We have shown that our approach is applicable to various tasks, i. e. absorbing, collecting, and skimming, in three different scenarios in simulation, as well as in a real experiment with the humanoid robotic agent Rollin' Justin. The robotic agent experiment proves that our qualitative representation matches the effects of the real world.

ACKNOWLEDGMENTS

This work was partially funded by the European Community's Seventh Framework Programme under grant agreement no. 608849 EuRoC and partially by the Helmholtz Association Project HVF-0029 RACE-LAB.

REFERENCES

[1] 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)*, pages 1597–1598, 2009.

[2] 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):16, 2004.

[3] M. Cakmak and L. Takayama. Towards a comprehensive chore list for domestic robots. In *Proc. of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 93–94, 2013.

[4] H. I. Christensen et al. A roadmap for u.s. robotics from internet to robotics. Technical report, Robotics Virtual Organization, 2013.

[5] R. Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute, 2010.

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

[7] M. Do, J. Schill, J. Ernesti, and 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)*, pages 1858–1864, 2014.

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

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

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

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

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

[13] J. M. Hess, G. D. Tipaldi, and 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)*, pages 1923–1928, 2012.

[14] G. Hirzinger, N. Sporer, A. Albu-Schäffer, M. Hahnle, R. Krenn, A. Pascucci, and M. Schedl. Dlr's torque-controlled light weight robot iii-are we reaching the technological limits now? In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1710–1716. IEEE, 2002.

[15] A. Huaman and M. Stilman. Deterministic motion planning for redundant robots along end-effector paths. In *Proc. of the International Conference on Humanoid Robots (ICHR)*, pages 785–790, 2012.

[16] R. Konietschke and 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)*, pages 2945–2950. IEEE, 2009.

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

[18] J.-C. Latombe. *Robot Motion Planning*. Springer, 1990.

[19] S. LaValle. Rapidly-exploring random trees: A new tool for path planning. 1998.

[20] D. Leidner, C. Borst, A. Dietrich, and A. Albu-SchÃd'ffer. Classifying compliant manipulation tasks for automated planning in robotics. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1769–1776, 2015.

[21] D. Leidner, A. Dietrich, M. Beetz, and A. Albu-Schäffer. Knowledge-enabled parameterization of whole-body control strategies for compliant service robots. *Autonomous Robots (AURO): Special Issue on Whole-Body Control of Contacts and Dynamics for Humanoid Robots*, pages 1–18, 2015.

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

[23] B. Liepert, S. Stramigioli, R. Bischoff, U. Haass, et al. Multi-annual roadmap for robotics in europe. Technical report, euRobotics Association Internationale Sans But Lucratif (AISBL), 2014.

[24] N. J. Nilsson. Shakey the robot. Technical report, DTIC Document, 1984.

[25] K. Okada, M. Kojima, Y. Sagawa, T. Ichino, K. Sato, and 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)*, pages 7–12, 2006.

[26] K. Okada, T. Ogura, A. Haneda, J. Fujimoto, F. Gravot, and 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)*, pages 1772–1777, 2005.

[27] V. Ortenzi, M. Adjigble, K. Jeffrey, R. Stolkin, and M. Mistry. An experimental study of robot control during environmental contacts based on projected operational space dynamics. In *Proc. of the IEEE-RAS International Conference on Humanoid Robots (ICHR)*, pages 407–412, 2014.

[28] E. Pablo Lana, B. Vilhena Adorno, and 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)*, pages 3083–3088, 2015.

[29] C. Schindlbeck and 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)*, pages 440–447, 2015.

[30] H. Urbanek, A. Albu-Schäffer, and P. van der Smagt. Learning from demonstration: repetitive movements for autonomous service robotics. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3495–3500, 2004.

[31] D. Vanthienen, S. Robyns, E. Aertbeliën, and 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.

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