Titel/Title: Knowledge-Enabled Parameterization of Whole-Body Control Strategies for Compliant Service Robots

Autor*innen/Author(s): Daniel Leidner, Alexander Dietrich, Michael Beetz, Alin Albu-Schäffer

Veröffentlichungsversion/Published version: Postprint

Publikationsform/Type of publication: Artikel/Aufsatz

Empfohlene Zitierung/Recommended citation:

Leidner, D., Dietrich, A., Beetz, M. et al. Knowledge-enabled parameterization of whole-body control strategies for compliant service robots. Auton Robot 40, 519–536 (2016). https://doi.org/10.1007/s10514-015-9523-3

Verfügbar unter/Available at:
(wenn vorhanden, bitte den DOI angeben/please provide the DOI if available)

10.1007/s10514-015-9523-3

This is a draft to appear in Autonomous Robots
DOI 10.1007/s10514-015-9523-3

Knowledge-Enabled Parameterization of Whole-Body Control Strategies for Compliant Service Robots

Daniel Leidner · Alexander Dietrich · Michael Beetz · Alin Albu-Schäffer

Received: 8 October 2014 / Accepted: 16 November 2015

Abstract Compliant manipulation is one of the grand challenges for autonomous robots. Many household chores in human environments, such as cleaning the floor or wiping windows, rely on this principle. At the same time these tasks often require whole-body motions to cover a larger workspace. The performance of the actual task itself is thereby dependent on a large number of parameters that have to be taken into account. To tackle this issue we propose to utilize low-level compliant whole-body control strategies parameterized by high-level hybrid reasoning mechanisms. We categorize compliant wiping actions in order to determine relevant control parameters. According to these parameters we set up process models for each identified wiping action and implement generalized control strategies based on human task knowledge. We evaluate our approach experimentally on three whole-body manipulation tasks, namely scrubbing a mug with a sponge, skimming a window with a window wiper and bi-manually collecting the shards of a broken mug with a broom.

Daniel Leidner, Alexander Dietrich, Alin Albu-Schäffer
German Aerospace Center (DLR),
Institute of Robotics and Mechatronics
Muenchner Strasse 20, 82234 Wessling, Germany
Tel.: +49-8153-283849, Fax: +49-8153-281134
E-mail: daniel.leidner@dlr.de

Michael Beetz
University of Bremen,
Institute for Artificial Intelligence
Am Fallturm 1, 28359 Bremen, Germany
Tel.: +49-421-21864001, Fax: +49-421-21864047

Alin Albu-Schäffer
Technische Universität München (TUM),
Sensor-Based Robotic Sys. and Intelligent Assistance Sys.
Boltzmannstrasse 3, 85748 Garching, Germany
Tel.: +49-8153-283689, Fax: +49-8153-281134

Keywords Whole-Body Control · AI Reasoning Methods · Task Knowledge · Humanoid Robots · Mobile Manipulation

1 Introduction

Cleaning tasks are the most frequent household chores in human environments according to the analysis of Cakmak and Takayama (2013). Window wiping (Fig. 1) or cleaning the floor, for example, rely on contacts and applied forces with a certain tool to achieve a desired goal state. At the same time the workspace to be covered is spacious. If a humanoid robot is employed to carry out such tasks it has to execute compliant whole-body motions. That requires compliant whole-body control strategies capable of handling various constrained contacts where the correct parameterization is crucial for the performance. Detailed knowledge about the task and the involved objects is mandatory. The robot has to reason symbolically and geometrically about the parameterization in advance w. r. t. the given problem and the current state of the environment. A possible solution for this issue is hybrid reasoning based on prior task knowledge (Wolfe et al 2010; Dornhege et al 2012; Gravot et al 2005).

The combination of hybrid reasoning, which fuses symbolic with geometric planning, and control theory for whole-body manipulation, is a rare research topic so far. However, there exists some work to incorporate the different research fields. Several approaches utilize symbolic and geometric reasoning for mobile manipulation. Wolfe et al (2010) integrate external geometric solvers into a symbolic *Hierarchical Task Network (HTN)* planner. Dornhege et al (2012) compute geometric effects during symbolic planning by calling semantic attach-

ment modules for navigation and manipulation (Dornhege and Hertle 2013). The aSyMov planner by Gravot et al (2005) is designed to deal with multi-robot planning problems including geometric constraints for pick-and-place tasks. Kaelbling and Lozano-Pérez (2013) focus their work on symbolic and geometric planning under uncertainty for mobile manipulation. However, neither of these works consider the combined motion of the mobile base and the manipulator as a whole-body control problem. The works of Yamamoto and Yun (1992), Tan et al (2003), Moro et al (2013), and us (Dietrich et al 2011a; 2012b) show how mobile manipulation is successfully tackled at the control level. The *Operational Space Formulation* (Khatib 1987) is probably the most popular method to implement force control in a reduced space (Sentis and Khatib (2005); Sadeghian et al (2014)), e. g. in the Cartesian coordinates of the end-effector. However, parameterizing controllers for different tasks without knowledge about the involved objects is difficult. Nevertheless, there has already been some advanced research in this context such as by Tenorth et al (2012), where the authors rely on a web-database for actions, objects, and environments to parameterize tasks, or Mosenlechner and Beetz (2011) that infers geometric parameters such as where to place objects based on physics simulation. Bartels et al (2013) ground symbolic actions by the use of a *constraint-based movement description language* based on geometric features, such as points, lines, and planes to be interpreted by the control level. Also to mention are the results by Kallmann and Thalmann (1999), who store articulation trajectories to guide object manipulation, and the work of Levison (1996), which classifies objects by functionality and augments their symbolic domain with hierarchical properties. Based on a similar concept, we introduced a hybrid reasoning framework in our previous work (Leidner et al 2012) to define manipulation tasks in Cartesian space with the aid of object-knowledge. In this work, we show how this framework can be utilized to define compliant tool usage tasks such as *cleaning*.

Recently, compliant cleaning tasks have become subject of intensive research, especially *wiping of surfaces*. Kunze et al (2011) simulate physical effects to infer first-order symbolic representations. In particular they conducted simplified physical simulations where liquids are *absorbed* with a sponge. Do et al (2014) predict action parameters based on object properties by learning from experience in a *table wiping* task. Hess et al (2012) describe a generic approach to autonomously compute time- and effort-optimal cleaning trajectories. However, applied force and stiffness are not parameterized due to the lack of a compliant robot. Vanthienen et al (2013) describe a table wiping task with the *iTaSC* framework for a backdrivable, force-sensorless robot. Okada et al (2005; 2006) show an inverse-kinematics-based programming approach to define whole-body cleaning motions (*sweeping*, *vacuuming*, and *dish washing*) for a humanoid robot exploiting object knowledge. Their method is similar to the geometric reasoning approach we utilize. However, in addition to Okada et al. we emphasize to have a closer look at the cleaning task itself, namely the tool-surface contact together with the applied force and stiffness. In fact, these aspects are crucial for the cleaning result, but can only be taken into account by a *compliant* robot when the reasoning level and the control level act jointly.

Fig. 1 The humanoid robot Rollin' Justin wiping a window as a demonstration for a compliant whole-body task.

The contribution of this paper is a generic approach to parameterize whole-body controllers for compliant manipulation tasks with different contact behavior. In particular, a classification of *wiping tasks* is conducted in order to identify the required task and control parameters. This parameterization is utilized in our whole-body control framework based on hierarchical null space projections. The control layer is integrated into a hybrid reasoning mechanism which grounds symbolic transitions to concrete geometric actions. Within this step action templates transfer human task knowledge by individually parameterizing wiping tasks w. r. t. our classification. This is achieved by applying prior object knowledge including the Cartesian task trajectory, Cartesian contact force, and Cartesian contact stiffness, among others. We evaluate our approach in three whole-body manipulation experiments: cleaning a mug with a sponge, wiping a window with a window wiper and bi-manually sweeping the floor. The humanoid robot Rollin' Justin is employed for the experiments, see Fig. 1. Part of this work was presented in Leidner et al (2014).

The paper is organized as follows: A detailed problem identification is conducted in Sec. 2 to classify the tasks we are interested in. We present the control stra-

tegies to accomplish these tasks in Sec. 3. The reasoning about the task parameterization and the integration into our hybrid reasoning framework is presented in Sec. 4. We evaluate our methods in three elaborate experiments in Sec. 5.

2 Parameter Identification

A large number of tasks in domestic and industrial environments rely on forces applied by tools of different kind. One of the most common compliant activities is cleaning. The parameterization of these tasks have to be reasoned carefully in order to deal with contacts of different nature in a generalized way. So does vacuuming not rely on a specific direction of motion, while sweeping with a broom requires a dedicated trajectory design to collect the particles in one spot. Furthermore, a wide variety of tasks that are not related to cleaning are based on the same principles. One example is the painting of a wall. Besides the tool-surface contact these tasks share one more common component, namely the *medium* (i. e. particles or liquid *between* the tool and the surface, e. g. dust or paint). In summary, we formulate this kind of tasks as *guiding a tool along a target surface while maintaining contact to indirectly manipulate some sort of medium.* Although the purpose of the actions is different, they share many properties. These similarities make it possible to utilize one control design for all problems, but each task requires a different parameterization. In order to identify the required control parameters, we are going to classify the respective tasks first.

Several taxonomies for classifying object manipulation have been developed in the past. Most of them categorize by grasp type. The taxonomies of Kapandji and Honoré (1970) and Cutkosky (1989) are probably the most well-known. Based on these and many other taxonomies, a comprehensive grasp taxonomy was developed by Feix et al (2009). However, they do not specify manipulation actions in greater detail. The hand-centric taxonomy of Bullock and Dollar (2011); Bullock et al (2013) classifies human manipulation behaviors according to the relative movement of the hand to the grasped object during task execution. They classify *wiping a surface* as "*In Contact - Prehensile - Motion - Not Within Hand - No Motion At Contact*". Based on this definition the relative position of a grasped object does not change w. r. t. the manipulator during the object manipulation. The taxonomy of Bloomfield et al (2003) classifies haptic actions by the applied forces and torques. They classify *filing* and *sanding* as action with major amounts of force, where the applied force direction is not aligned with the direction of motion.

However, neither of these taxonomies make goal-oriented assumptions about the nature of motion. Contact properties between a tool and a surface are not considered. Furthermore, the medium to be manipulated is never integrated in the classification. For this reason, we define a taxonomy on our own for the sub-category from now on referred to as *wiping tasks*, derived from the term *wiping a surface*.

We assume that the relative position of the hand and the tool does not change as defined by Bullock et al (2013). The grasp type is secondary. Accordingly, the manipulator is neglected in our classification and can be either robotic or human. The classification does not distinguish between one or more manipulators. Forces and torques are important control parameters but differ significantly for individual wiping tasks based on the environmental conditions (e. g. the amount of dirt on the floor) and are therefore not applicable as categorization metric. For this reason, a sub-categorization of wiping tasks based on geometric features does not seem appropriate. Therefore, we do not directly categorize wiping tasks w. r. t. geometric features, but rather classify them based on symbolic effects to implicitly group actions with similar geometric structures. This is done by investigating wiping tasks based on the symbolic aspects of the *tool - surface - medium* tuples, where the medium is considered to be the main reason for any wiping task. Based on this consideration we are able to identify nine action types related to a specific wiping task. The actions are grouped in *applying*, *removing*, and *modifying* the medium. Additionally, a tool-centric view, a surface-centric view and a medium-centric view can be applied to categorize the actions. The resulting matrix structure is illustrated in Fig. 2 and explained in detail from the top left to the bottom right:

Absorb: A medium is absorbed upon close vicinity to the tool. This may be caused due to electrostatic force as known from dusting or an air draft from a vacuum. The tool is usually applied with a planar contact where the medium defines the region of interest (ROI).

Skim: The goal position of the medium is not of interest and is therefore illustrated as skimmed from the surface. Scratching ice from a car window is considered as skimming. The direction of motion is defined by the individual geometric topology.

Collect: Collecting can be related to skimming. However, the medium has to be collected afterwards, e. g. to remove it accumulated. Exemplary actions are collecting leaves with a rake or sweeping up shards. The tool alignment w. r. t. the surface is a crucial aspect.

Emit: The emitting action is the counterpart of the absorbing action. The medium is initially located on or

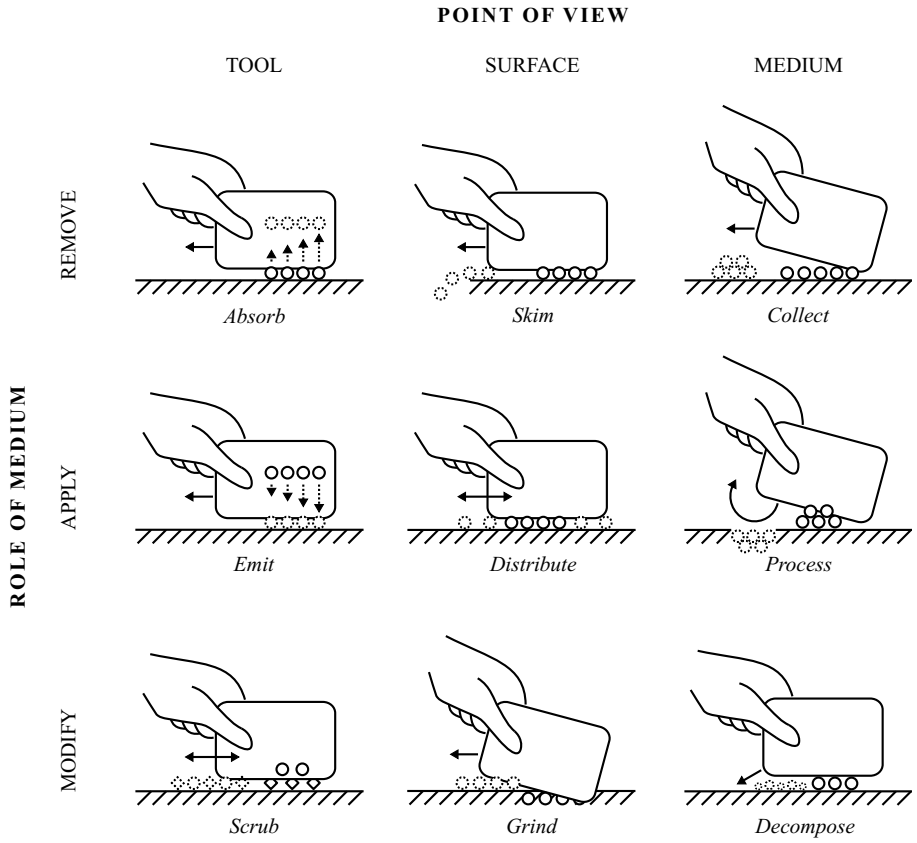


Fig. 2 Matrix classification of wiping tasks according to the action w. r. t. the tool, surface, and medium versus the role of the medium. The tool (grasped by a human hand) is abstracted as a rectangle capable of all illustrated actions. The motion of the tool is indicated as a solid arrow. The surface is always shown on the bottom of the corresponding cell. It might be flat as illustrated, curved, or of any other shape. The medium is shown in the initial state (solid circles/squares) and in the goal state after the action is performed (dashed circles/squares) where dashed arrows may indicate the transition of the medium.

in the tool and is applied to the surface as it is done for painting a wall. Typically, the whole surface is involved.

Distribute: Distributing a medium is related to emitting a medium. However, the medium is already located on the surface. Applying shoe polish is such a task. The task trajectory is important to equally distribute the medium on the surface.

Process: Processing is a medium-centric action. The medium is on purpose used to alter the surface. One example for this is the use of cement to fill holes in a wall. It is also possible that the surface is only used to directly manipulate the medium as done when rolling cookie dough, for example.

Scrub: Scrubbing merges an auxiliary medium with an unwanted medium (e. g. detergent and dirt) by exerting force under repetitive motions to remove the unwanted medium from the surface. Many cleaning tasks can be categorized as scrubbing.

Grind: Grinding is often used in manufacturing such as planing wood. The medium is separated from the surface and is a waste product of no greater value. The tool alignment is crucial for the result.

Decompose: Decomposing splits the medium into smaller particles. The contact between the medium and the tool is planar. Pestle with a mortar is one example.

Our classification emphasizes the versatility of wiping actions. For each action, numerous tools might be suitable to achieve the desired goal. It is also possible that one particular task is actually realized as a combination of several wiping actions such as mopping, which is a combination of *distributing*, *scrubbing*, and *absorbing*. Furthermore, we investigated that based on the topology (e. g. varying medium size and surface friction) a different parameter range might be suitable. We identified three types of parameters: *symbolic*, *subsymbolic/geometric*, and *control parameters*.

With reference to the symbolic classification of wiping tasks, the identified symbolic parameters reflect the identified action verb catalog, namely *absorbed*, *skimmed*, *collected*, *emitted*, *distributed*, *processed*, *scrubbed*, *ground*, and *decomposed*. According to our definition, these symbolic parameters are equal to *predicates* in the *Planning Domain Definition Language (PDDL)* (Ghallab et al 1998). They are utilized to describe symbolic precondi-

tions and effects for symbolic planning. Based on the desired state of the medium, a symbolic planner can compute the required actions to solve a task w. r. t. the target surface and the provided tools in the environment.

Sub-symbolic parameters[1] have to be considered during geometric reasoning, such as the *shape* of the object and the *ROI* on the target surface in order to plan feasible *Cartesian* and *joint trajectories* (including time information) not colliding with the environment. There are several sub-symbolic parameters which have direct influence on the control level. The most important ones are the *mass/inertia* of the tool, the *center of mass (CoM)*, the *tool center point (TCP)*, and the relative *alignment* of the tool w. r. t. the target surface.

The identified wiping actions rely all on the same control parameters, namely *Cartesian motion*, *Cartesian stiffness*, and *Cartesian force*. The Cartesian motion defines the trajectory for the tool, and therefore the motion of the robotic manipulators. The robot has to reason about whether it is necessary to execute the motion with the whole body using the locomotion system or only a subset of joints. The stiffness parameters are based on the tool alignment and the direction of motion. Depending on the task, different rotational and translational stiffnesses have to be applied. The force parameter may depend on the material of tool and surface. Also different environmental conditions may require a different force level.

Summarized one can say that properly executed wiping tasks depend on the coupling between symbolic, geometric and control parameters. Therefore, we believe that it is mandatory to combine low-level control strategies and high-level reasoning for a correct parameterization. In conclusion, we propose to set up one particular *process model* for each identified wiping action. Based on our classification each process model has reference to one of the identified symbolic parameters and the corresponding symbolic effects to the environment. Consequently, each symbolic parameter is grounded to one geometric procedure with an individual geometric parameterization considering the involved objects. Accordingly, the control parameters are chosen to fit the overall *control strategy*.

In this paper we explain this parameterization bottom up to emphasize the significance of exploiting contacts and physical interaction on the control level. The whole-body control framework and the related low-level parameterization is explained in Sec. 3. The integration of the symbolic and geometric reasoning procedure along with the high-level parameterization is described in Sec. 4, where we illustrate the generalized utilization of our control framework w. r. t. different domains. In Sec. 5 the approach is evaluated on a humanoid robot.

[1] commonly summarized as geometric parameters in the automated planning community

3 Whole-Body Control

In this section, the whole-body control framework is explained in detail. That includes the structure of the closed loop (Sec. 3.1), the control tasks in use (Sec. 3.2), and the redundancy resolution to realize the commanded control task hierarchy (Sec. 3.3). An example of a controller parameterization is given in Sec. 3.4.

3.1 Structure of the Controller

The generic whole-body control scheme is illustrated in Fig. 3. One can divide the structure into three basic parts. The first subsystem is the robot itself providing measurements. In case of the humanoid mobile robot Rollin' Justin (Borst et al 2009), these are for example the joint configuration $\boldsymbol{q}$, the joint velocities $\dot{\boldsymbol{q}}$, or the joint torques $\boldsymbol{\tau}$. Additionally other sensor information is provided, e. g. vision information from various cameras. The second subsystem is the whole-body controller, which gets the measurements of the robot. From a stack of various control tasks, a selection is made and the order of priority is given to the controller. Based on that, null space projection techniques are used to realize this choice. The reasoning instance constitutes the third subsystem which parameterizes the controller for arbitrary manipulation tasks. The reasoning includes symbolic and geometric planning. This part will be detailed in Sec. 4.

3.2 Control Tasks

A large variety of control tasks are available for the higher-level reasoning instance in order to properly plan the overall application. In the following, the most commonly used methods on Rollin' Justin are briefly reviewed and associated with the wiping tasks explained in Sec. 2. More details on the methods can be found in Dietrich et al (2012b). The parameterization of the control level is realized by the reasoning instance which will be explained in Sec. 4.2. However, it is mentionable that although the parameterization of control tasks may depend on several factors it is not mandatory to define all parameters for every single action. General-purpose default values apply if a parameter is skipped during task reasoning.

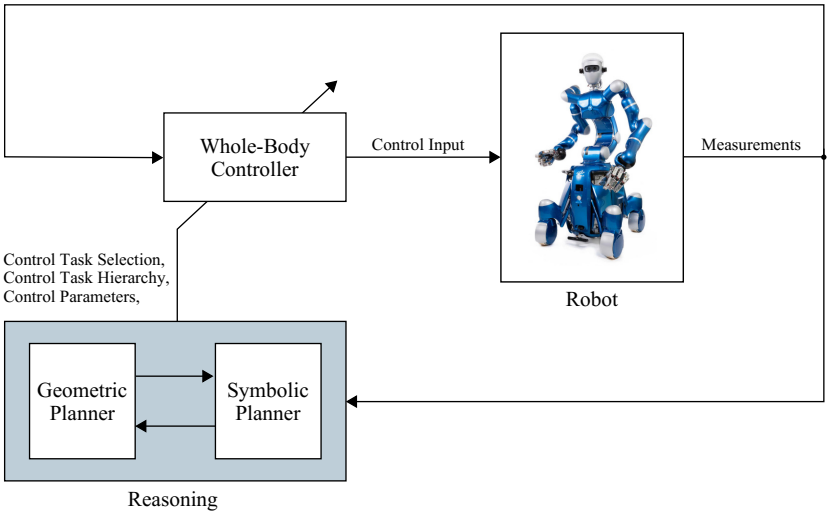


Fig. 3 The whole-body controller contains a stack of tasks arranged in a hierarchical order. It is parameterized by a reasoning instance which consists of a symbolic part and a geometric part.

Cartesian impedance: Primarily applied in the context of the task execution, the Cartesian impedance (Hogan 1985; Ott et al 2008) at the end-effectors constitutes one of the basic functionalities on Rollin' Justin. The Cartesian impedance is modeled as a mass-spring-damper system in the Cartesian directions of the TCP to create a compliant behavior for the robotic end-effector. The planning layer defines a *virtual, spatial equilibrium* and the TCP will follow it depending on the parameterization of the *stiffness*, the *damping*, and the maximum permissible *Cartesian forces/torques*. The control torque is

$$\boldsymbol{\tau}_{\mathrm{car}} = -\left(\frac{\partial V_{\mathrm{car}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{car}})}{\partial \boldsymbol{q}}\right)^{T} - \boldsymbol{D}_{\mathrm{car}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{car}})\dot{\boldsymbol{q}} \ , \qquad (1)$$

where V_{car} denotes the spatial spring potential and $\boldsymbol{\mathcal{P}}_{\mathrm{car}}$ describes the parameterization, e. g. the potential stiffness, the trajectories, and maximum Cartesian forces. Damping is injected by the positive definite damping matrix $\boldsymbol{D}_{\mathrm{car}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{car}})$. Further information on the implementation of the Cartesian impedance control can be found in Ott et al (2008).

Cartesian impedance is probably the most crucial control behavior to solve wiping tasks. Since we treat all wiping actions according to Sec. 2 as robot-independent problems, all assumptions on force and stiffness are made w. r. t. the involved objects and therefore in Cartesian object space (Wimböck 2013) and later transformed into robot coordinates. By carefully parameterizing the Cartesian impedance controller, contacts of different nature can be managed.

Joint impedance: In contrast to the Cartesian impedance, a joint impedance does not relate a task space (e. g. the Cartesian space) to the joint space, but it is defined in the joint space. The control torque is

$$\boldsymbol{\tau}_{\mathrm{jnt}} = -\left(\frac{\partial V_{\mathrm{jnt}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{jnt}})}{\partial \boldsymbol{q}}\right)^{T} - \boldsymbol{D}_{\mathrm{jnt}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{jnt}})\dot{\boldsymbol{q}} \ , \qquad (2)$$

where V_{jnt} denotes the spring potential, the parameterization is denoted by $\boldsymbol{\mathcal{P}}_{\mathrm{jnt}}$, and $\boldsymbol{D}_{\mathrm{jnt}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{jnt}})$ is the damping matrix related to the joint impedance.

A joint impedance can be applied to the complete robot or any subsystem such as an arm or the torso. This is particularly useful for bi-manual wiping tasks (e. g. sweeping, see Sec. 5.3), since the arms can be set to be less stiff compared to the torso. This ensures a steady torso posture while external influences can be compensated at the same time.

Self-collision avoidance: In manipulators with many degrees of freedom (DOF), self-collision avoidance is relevant due to the large number of possible collisions between body parts. We have developed a reactive technique (Dietrich et al 2011b) which applies artificial, repulsive forces between potentially colliding links. In each collision direction, a desired mass-spring-damper relation is commanded which allows to realize damping ratios. The control torque is

$$\boldsymbol{\tau}_{\mathrm{sca}} = -\left(\frac{\partial V_{\mathrm{sca}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{sca}})}{\partial \boldsymbol{q}}\right)^{T} - \boldsymbol{D}_{\mathrm{sca}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{sca}})\dot{\boldsymbol{q}} \ , \qquad (3)$$

where V_{sca} denotes the repulsive potential which is parameterized by $\boldsymbol{\mathcal{P}}_{\mathrm{sca}}$. The parameters contain, among others, the minimum distance between links where the brakes are engaged (emergency stop), the potential stiffness, and the distance at which the potential starts to generate repulsive forces. The damping is injected via $\boldsymbol{D}_{\mathrm{sca}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{sca}})$.

Description	Value	Parameters
Cartesian impedance	$\boldsymbol{\mathcal{P}}_{\mathrm{car}}$	stiffness (3 transl., 3 rot.), trajectories, damping, max. Cart. wrench (3 transl., 3 rot.)
Joint impedance	$\boldsymbol{\mathcal{P}}_{\mathrm{jnt}}$	stiffness (n joints), trajectories, joint damping
Self-collision avoidance	$\boldsymbol{\mathcal{P}}_{\mathrm{sca}}$	stiffness (in collision direction), thresholds, emergency stop conditions, damping ratios
Avoidance of end stops	$\boldsymbol{\mathcal{P}}_{\mathrm{mes}}$	stiffness (n joints), joint damping
Singularity avoidance	$\boldsymbol{\mathcal{P}}_{\mathrm{sav}}$	stiffness (in singularity space), thresholds, damping (in singularity space)
Gravity compensation	$\boldsymbol{\mathcal{P}}_{\mathrm{g}}$	loads (mass, inertia, center of mass)

Table 1 Parameters of the control tasks to be determined by the reasoning instance. Note that all parameters are instantiated with general-purpose values. The overall parameter vector is defined as $\boldsymbol{\mathcal{P}} = (\boldsymbol{\mathcal{P}}_{\mathrm{car}}, \boldsymbol{\mathcal{P}}_{\mathrm{jnt}}, \boldsymbol{\mathcal{P}}_{\mathrm{sca}}, \boldsymbol{\mathcal{P}}_{\mathrm{mes}}, \boldsymbol{\mathcal{P}}_{\mathrm{sav}}, \boldsymbol{\mathcal{P}}_{\mathrm{g}})$.

The self-collision avoidance or parts of it can optionally be disabled for tasks where multiple manipulators work close to each other such as washing the dishes (see Sec. 5.1).

Avoidance of mechanical end stops: A standard control task in robotics is the avoidance of mechanical end stops. Artificial repulsive potentials around the end stops are designed. This control subtask is active close to the end stops only, and it is inactive in the remaining workspace. Similar to the preceding tasks, the control torque is

$$\boldsymbol{\tau}_{\mathrm{mes}} = -\left(\frac{\partial V_{\mathrm{mes}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{mes}})}{\partial \boldsymbol{q}}\right)^T - \boldsymbol{D}_{\mathrm{mes}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{mes}})\dot{\boldsymbol{q}} \ , \quad (4)$$

where V_{mes} denotes the repulsive potential and $\boldsymbol{\mathcal{P}}_{\mathrm{mes}}$ determines the parameterization, e. g. the stiffness and and thresholds. Damping is injected via $\boldsymbol{D}_{\mathrm{mes}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{mes}})$.

Singularity avoidance: In order to be able to properly react to unexpected events or disturbances, it is important to maintain a high manipulability throughout the whole time. Various methods are known in the literature but the representations based on the kinematic and the dynamic manipulability measure (Yoshikawa 1990) are probably the most popular ones. The control torque can be described as

$$\boldsymbol{\tau}_{\mathrm{sav}} = -\left(\frac{\partial V_{\mathrm{sav}}(m_{\mathrm{sav}}(\boldsymbol{q}), \boldsymbol{\mathcal{P}}_{\mathrm{sav}})}{\partial \boldsymbol{q}}\right)^T - \boldsymbol{D}_{\mathrm{sav}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{sav}})\dot{\boldsymbol{q}} \ , \quad (5)$$

where V_{sav} is the potential to repel from singular configurations, $m_{\mathrm{sav}}(\boldsymbol{q})$ is the manipulability measure, and $\boldsymbol{\mathcal{P}}_{\mathrm{sav}}$ parameterizes the control task by defining the stiffness and the manipulability thresholds (Ott 2008), for example. Damping is injected via $\boldsymbol{D}_{\mathrm{sav}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{sav}})$.

Singularities as well as mechanical end stops and self-collisions can additionally be avoided by providing roughly approximated reference trajectories for use in a joint impedance controller (2). This is done by calculating inverse kinematics solutions for discretized steps along the Cartesian task trajectory during the geometric reasoning step. Nevertheless, it is mandatory to parameterize the control tasks w. r. t. the environmental circumstances, in order to react on unforeseen external disturbances such as humans in the workspace of the robot.

Gravity compensation: Gravitational effects can be compensated by simulating a gravity model of the system online (Ott et al 2004). The respective control torque can be written as

$$\boldsymbol{\tau}_{\mathrm{g}} = \left(\frac{\partial V_{\mathrm{g}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{g}})}{\partial \boldsymbol{q}}\right)^T \ , \quad (6)$$

where V_{g} denotes the gravity potential parameterized by $\boldsymbol{\mathcal{P}}_{\mathrm{g}}$, which contains information about *mass*, *inertia*, and *center of mass* of the links of the robot, handled objects, and additional loads such as backpacks.

Platform control: Nonholonomic, mobile platforms are usually controlled via kinematic controllers to realize desired trajectories by implicitly complying with the rolling constraints. In case of Rollin' Justin, such a control law has been implemented by Giordano et al (2009), which expects a trajectory in the plane of motion, i. e. in one rotational and two translational directions.

The utilization of the mobile base is crucial for wide-area tasks (e. g. cleaning the floor or wiping large windows). Whether to move only a subset of joints, arms and torso, or to include the mobile base for solving a given task has to be defined during the reasoning process. The geometric topology (e. g. tool dimension and ROI) is thereby considered.

An overview of the control parameters that can be set and the respective representation is given in Table 1.

3.3 Redundancy Resolution

The reasoning instance does not only choose the control tasks necessary for the considered application but it also determines the task hierarchy and the order of priorities, respectively, cf. Fig. 3. In the whole-body control framework implemented on Rollin' Justin, a strict task hierarchy with r priority levels is realized via null space projections (Nakamura et al 1987; Siciliano and

Slotine 1991), which is a standard tool in redundancy resolutions. One can write the overall control torque as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_1 + \sum_{i=2}^{r} \boldsymbol{N}_i(\boldsymbol{q})\boldsymbol{\tau}_i \ , \tag{7}$$

where the indices describe the priority levels, and $j < k$ means that j has higher priority than k. Each lower level control action $\boldsymbol{\tau}_i$ for $1 < i \leq r$ is projected in the null space of all higher priority control tasks by the respective null space projector $\boldsymbol{N}_i(\boldsymbol{q})$.

These null space projectors are not unique and their properties depend on the specification of the task hierarchy. One can distinguish between *statically consistent* (Albu-Schäffer et al 2003) and *dynamically consistent* (Khatib 1987) null space projections, for example. Another criterion is whether the projections are of *successive* or *augmented* type (Antonelli 2009). An overview is given in (Dietrich et al 2015). Applying a control task hierarchy with several priority levels also requires to deal with arising singularities and the resulting discontinuities in the control law due to task conflicts or activation/deactivation processes of unilateral constraints. That also has an influence on the shape and properties of the projectors when applying techniques such as Dietrich et al (2012a). Moreover, there are also differences in the kind of control itself. One may require a high tracking performance in specific scenarios (*Operational Space Formulation* (Khatib 1987)) or may give more weight to improved contact/interaction behavior with guaranteed stability of the complete dynamic system (*Compliance Control* (Ott 2008; Dietrich et al 2013)).

The tasks and the hierarchy are assigned by the reasoning instance depending on the application such that e. g. $\boldsymbol{\tau}_1 = \boldsymbol{\tau}_{\mathrm{car}}$ and $\boldsymbol{\tau}_2 = \boldsymbol{\tau}_{\mathrm{jnt}}$, and so on. Three examples for the assignment are given in Sec. 5.

3.4 Example of a Potential Definition - Cartesian Impedance Control

In order to demonstrate how the control tasks in Sec. 3.2 are parameterized in practice, $\boldsymbol{\mathcal{P}}_{\mathrm{car}}$ is detailed here. The spring potential of the Cartesian impedance (1) is defined by

- Stiffness (3 translations, 3 rotations): The positive definite, symmetric matrices $\boldsymbol{K}_{\mathrm{t}}$, $\boldsymbol{K}_{\mathrm{r}} \in \mathbb{R}^{3\times3}$ define the translational and rotational stiffness at the TCP, respectively.
- Trajectories: The homogeneous transformation matrix $\boldsymbol{H}_{\mathrm{cmd}}(t) = (\boldsymbol{R}_{\mathrm{cmd}}(t),\, \boldsymbol{p}_{\mathrm{cmd}}(t)) \in \mathbb{R}^{3\times4}$ describes the commanded/desired orientation of the TCP via the rotation matrix $\boldsymbol{R}_{\mathrm{cmd}}(t) \in \mathbb{R}^{3\times3}$ w. r. t. time t and the commanded/desired position of the TCP through $\boldsymbol{p}_{\mathrm{cmd}}(t) \in \mathbb{R}^3$.

The damping and the minimum/maximum Cartesian wrench specification are also contained in $\boldsymbol{\mathcal{P}}_{\mathrm{car}}$, yet they are not part of the potential definition and taken into account separately. Based on (1), the potential of the Cartesian impedance can be written as

$$V_{\mathrm{car}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{car}}) = \frac{1}{2}\tilde{\boldsymbol{p}}^T \boldsymbol{K}_{\mathrm{t}} \tilde{\boldsymbol{p}} + 2\boldsymbol{\epsilon}^T \boldsymbol{K}_{\mathrm{r}} \boldsymbol{\epsilon} \ , \tag{8}$$

where $\tilde{\boldsymbol{p}} = \boldsymbol{p}_{\mathrm{act}}(\boldsymbol{q}) - \boldsymbol{p}_{\mathrm{cmd}}(t)$ denotes the error between the current position $\boldsymbol{p}_{\mathrm{act}}(\boldsymbol{q})$ of the TCP and the commanded one. Note that analogous to $\boldsymbol{H}_{\mathrm{cmd}}(t)$, the forward kinematics delivers the homogeneous transformation matrix $\boldsymbol{H}_{\mathrm{act}}(\boldsymbol{q}) = (\boldsymbol{R}_{\mathrm{act}}(\boldsymbol{q}),\, \boldsymbol{p}_{\mathrm{act}}(\boldsymbol{q}))$, which describes the actual TCP orientation and position through $\boldsymbol{R}_{\mathrm{act}}(\boldsymbol{q}) \in \mathbb{R}^{3\times3}$ and $\boldsymbol{p}_{\mathrm{act}}(\boldsymbol{q}) \in \mathbb{R}^3$, respectively. The rotation between $\boldsymbol{H}_{\mathrm{act}}(\boldsymbol{q})$ and $\boldsymbol{H}_{\mathrm{cmd}}(t)$ can be uniquely described by unit quaternions. The vector part of the quaternions is $\boldsymbol{\epsilon} \in \mathbb{R}^3$. The gradient of the potential can be derived as follows:

$$\frac{\partial V_{\mathrm{car}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{car}})}{\partial \boldsymbol{q}} = \frac{\partial V_{\mathrm{car}}}{\partial \tilde{\boldsymbol{p}}} \frac{\partial \tilde{\boldsymbol{p}}}{\partial \boldsymbol{q}} + \frac{\partial V_{\mathrm{car}}}{\partial \boldsymbol{\epsilon}} \frac{\partial \boldsymbol{\epsilon}}{\partial \boldsymbol{\omega}} \frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{q}} \tag{9}$$

$$= \tilde{\boldsymbol{p}}^T \boldsymbol{K}_{\mathrm{t}} \boldsymbol{J}_{\boldsymbol{p},\boldsymbol{q}} + 4\boldsymbol{\epsilon}^T \boldsymbol{K}_{\mathrm{r}} \boldsymbol{J}_{\boldsymbol{\epsilon},\boldsymbol{\omega}} \boldsymbol{J}_{\boldsymbol{\omega},\boldsymbol{q}} \tag{10}$$

The dependencies in the notations are omitted for the sake of simplicity. The variable $\boldsymbol{\omega}$ describes the rotation of the TCP. The Jacobian matrix $\boldsymbol{J}_{\boldsymbol{\epsilon},\boldsymbol{\omega}}$ in (10) only depends on the unit quaternions, while $\boldsymbol{J}_{\boldsymbol{p},\boldsymbol{q}}$ and $\boldsymbol{J}_{\boldsymbol{\omega},\boldsymbol{q}}$ constitute the body Jacobian matrix. The overall control torque can be written as

$$\boldsymbol{\tau}_{\mathrm{car}} = -\left(\frac{\partial V_{\mathrm{car}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{car}})}{\partial \boldsymbol{q}}\right)^T - \boldsymbol{D}_{\mathrm{car}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{car}})\dot{\boldsymbol{q}} \tag{11}$$

$$= -\begin{pmatrix} \boldsymbol{J}_{\boldsymbol{p},\boldsymbol{q}} \\ \boldsymbol{J}_{\boldsymbol{\omega},\boldsymbol{q}} \end{pmatrix}^T \begin{pmatrix} \boldsymbol{K}_{\mathrm{t}}\tilde{\boldsymbol{p}} \\ 4\boldsymbol{J}_{\boldsymbol{\epsilon},\boldsymbol{\omega}}^T \boldsymbol{K}_{\mathrm{r}}\boldsymbol{\epsilon} \end{pmatrix} - \boldsymbol{D}_{\mathrm{car}}(\boldsymbol{q}, \boldsymbol{\mathcal{P}}_{\mathrm{car}})\dot{\boldsymbol{q}} \tag{12}$$

In accordance with the parameters given in Table 1, the reasoning level can now take the commanded stiffness ($\boldsymbol{K}_{\mathrm{t}}$, $\boldsymbol{K}_{\mathrm{r}}$) and the trajectories (used in $\tilde{\boldsymbol{p}}$ and $\boldsymbol{\epsilon}$) into account by providing them to the controller (12). Similarly, the reasoning level can limit the Cartesian wrench by providing minimum/maximum values for the Cartesian forces $\boldsymbol{K}_{\mathrm{t}}\tilde{\boldsymbol{p}}$ and torques $4\boldsymbol{J}_{\boldsymbol{\epsilon},\boldsymbol{\omega}}^T \boldsymbol{K}_{\mathrm{r}}\boldsymbol{\epsilon}$. In that case, these terms have to be saturated in (12). The damping in (11) will not be particularized here, but it can be realized by classical approaches such as the Double Diagonalization Design (Albu-Schäffer et al 2003). For a more detailed discussion on the Cartesian stiffness implementation, please refer to Ott (2008).

4 Task Reasoning and Parameterization

Based on the classification conducted in Sec. 2 we identified task parameters for the symbolic level, the geometric level, and the control level of *wiping tasks*. To incorporate the entire parameter set on the three levels of abstraction, we utilize a *hybrid reasoning* framework for combined symbolic and geometric reasoning. Furthermore, the task requirements may vary on the environmental conditions which demands detailed knowledge about the task and the involved objects. Therefore, we proposed to categorize objects in a hierarchical structure w. r. t. their functionality and additionally stored process models to define arbitrary manipulation instructions in our earlier work (Leidner et al 2012). The architecture is outlined in the following subsection in the context of the compliant manipulation task of *scrubbing a mug with a sponge*.

4.1 Hybrid Reasoning

An *object storage* provides prior knowledge for all available objects. The objects are hierarchically arranged in the object-oriented paradigm and categorized by functionality. Objects of the same class share the same process models to handle them and can therefore be manipulated in the same way but under consideration of their specific properties such as size and shape. The *world representation* holds the current state for the environment of the robot. Objects as described in the object storage are instantiated here with specific symbolic and geometric properties.

The process models within the object context are described in so-called *action templates*. Fig. 4 shows the action template for the *scrub action* of the abstract object class *_sponge* (Abstract object classes are marked with a leading underscore). Action templates consist of two segments. The first segment provides symbolic action definitions for symbolic planning in the PDDL language. The second segment specifies geometric instructions as executable code to implement the symbolic effects by the use of modular geometric reasoning such as navigation, motion planning or dynamics computations. Action templates constitute the main element for our approach on object-centered hybrid reasoning in a two-step approach:

First, all symbolic action definitions are gathered from the action templates of the object types currently in the world state. Additionally the matching symbolic properties are collected from the knowledge base. Hence, the symbolic domain is only filled with information of the current state. With this information, a symbolic planner is able to generate a symbolic transition leading to the desired goal state. For example, in order to scrub a mug with a sponge, both objects have to be picked up first, and a detergent has to be absorbed by the sponge. Only if these *preconditions* apply the scrubbing action can be scheduled. As *effect* the dirt is *scrubbed* away from the mug surface.

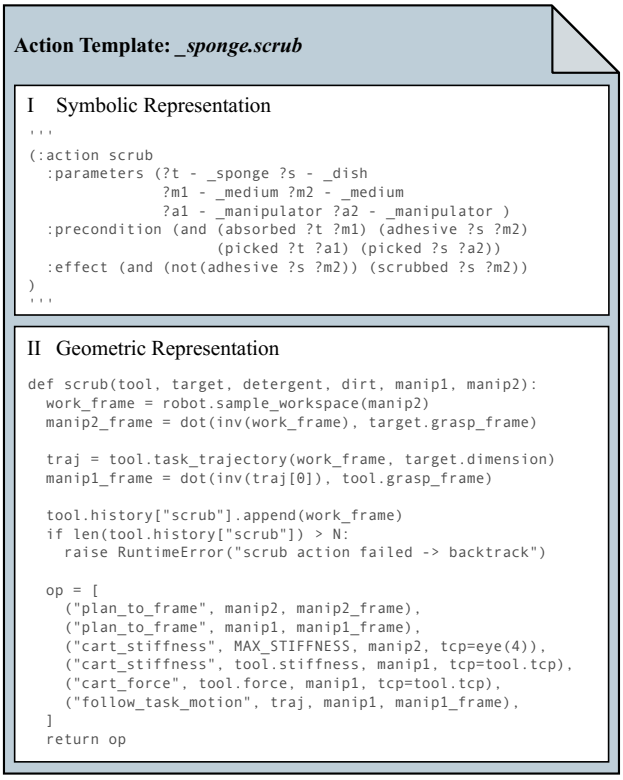


Fig. 4 Action templates are interpreted in a two-step approach. First, the symbolic action template header is parsed to solve a given task with symbolic reasoning. Second, the resulting symbolic transition is grounded based on the geometric process model, given by the second part of the action template. The controller parameterization is inherent in this step. Note that the code is visualized in two segments for better understanding, but constitutes one file.

In the second reasoning step, the main part of the action templates is revisited to resolve the geometric grounding. Modules to simulate the geometric execution can be integrated according to different requirements. If one geometric reasoning step succeeds, the next action is simulated until the symbolic transition is processed. Should one step of the geometric reasoning fail, the action template is reviewed for geometric alternatives. Should the predefined set of alternatives turn out to be insufficient, geometric backtracking is initiated to find a prior action with remaining alternatives to start over. If the given symbolic transition is not feasible at all, it is removed from the symbolic domain and the symbolic planner is called once again in order to find a symbolic alternative. Fig. 5 illustrates

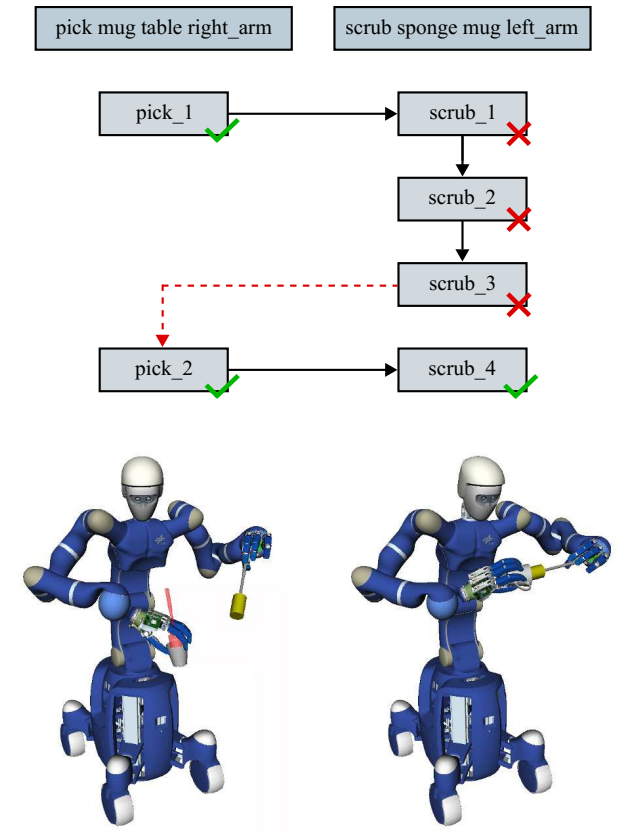


Fig. 5 A simple example on the geometric reasoning procedure is outlined in a block diagram where the symbolic transition is shown at the top (darker blocks) while different geometric alternatives are depicted below (lighter blocks). A top grasp is not suitable to scrub the mug with the sponge since all alternatives to reach the mug with the sponge (red crosses) are colliding with the right hand of the robot (lower left). After geometric backtracking to the previous pick action (red, dashed arrow), a bottom grasp is chosen to pick the mug. The inner curvature of the mug is no longer blocked (lower right).

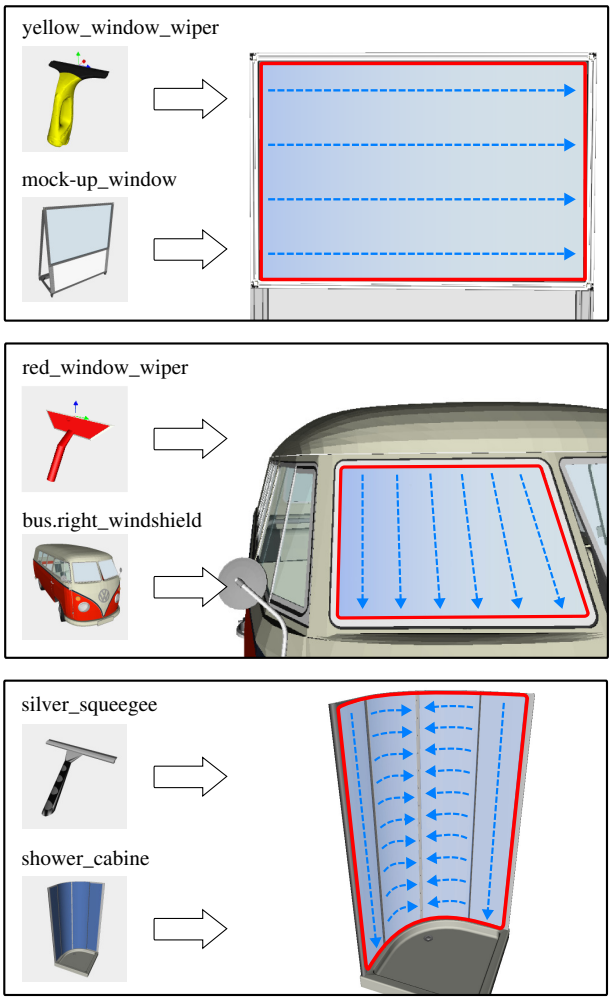


Fig. 6 Exemplary parameterization of the geometric process models of the *skim* action template with different wipers and windows. The ROI is highlighted in red. The generated workspace trajectories are drawn in blue for the respective tools.

the selection of geometric alternatives as well as the geometric backtracking mechanism in the conceptual example, *scrubbing* a mug with a sponge. Alternatives for the pick action consist of different grasps while the scrub action samples different orientations to reach the mug with the sponge. The backtracking mechanism resembles an exhaustive search algorithm. The reasoning procedure is therefore complete within the bounds of the discrete search space defined by the available alternatives for all scheduled action templates. The method is guaranteed to find a solution if one set of alternatives describes a feasible geometric action sequence. Consequently one can state that the more alternatives are available for the individual action templates, the higher the probability to find a feasible solution satisfying the desired symbolic goal state.

Action templates try to define process models at the highest possible level of abstraction to provide generic control strategies. Individual variations are defined at concrete object level, i. e. scrubbing a plate with a squared sponge applies the same action template with different parameters (e. g. different trajectory, force, and stiffness). If necessary, it is anyhow possible to specialize the process model for specialized object types. This concept is illustrated in Fig. 6 at the example of another compliant manipulation task, namely *skimming* a window with a window wiper. Depending on the window pane a different *trajectory design* might be required. This concept is applied to windows directly, or objects that contain windows such as the bus in the mid-row, or the shower on the bottom. The trajectory design is not limited to rectangular shapes but can also be applied to curved surfaces and other geometries. The strategy may even depend on particular regions of the object

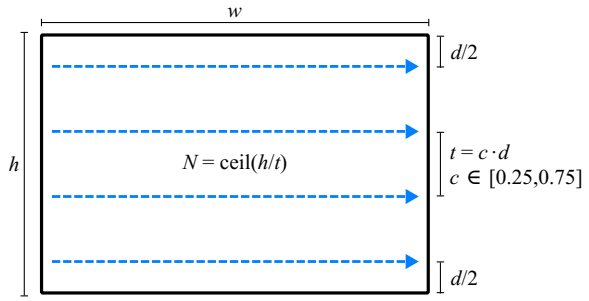


Fig. 7 Exemplary illustration of action template parameters for the the mock-up window, where h and w are height and width of the rectangular window pane (black) and d is the width of the wiper. These parameters are automatically extracted from the knowledge base and used to calculate the N Cartesian end-effector motions (blue) and the gap t in between. To guarantee full coverage of the surface, c is set to 0.75 by default which may be adapted after calculating N. The principle trajectory design is currently (but derivable) hand-crafted for the respective window classes (see Fig. 6).

(e. g. from left to right on curved areas and from top to bottom on flat surfaces). For now, the trajectory design is hand-crafted and stored as part of the window object. It is determined through parameters provided by distinct tools to calculate concrete Cartesian workspace motions for the execution. The parameter substitution is illustrated at the example of the mock-up window in Fig. 7. Object properties and the resulting task parameters can be mined from the web Tenorth et al (2011), demonstrated by a human Kronander and Billard (2014), learned by the robot Do et al (2014) or empirically estimated based on the experience from preceding research (e. g. Albu-Schäffer et al (2008)).

The single reasoning modules (i. e. symbolic planners and geometric planners) are individually provided by the robot. This way heterogeneous robotic systems can utilize the reasoning frameworks w. r. t. their special needs. For example, an unmanned aerial vehicle has to use a different geometric planner for motion generation than a ground-based humanoid. In the case of Rollin' Justin we use almost only out-of-the-shelf software. In particular, we use the *Fast Downward* planner by Helmert (2006) for symbolic planning and OpenRAVE by Diankov (2010) for geometric planning.

4.2 Task Parameterization

The reasoning procedure is responsible for the controller parameterization. Symbolically, the effects of a wiping action match one of the actions classified in Sec. 2. The symbolic planner selects the correct action based on the current symbolic world state, i.e. the available tools and the given problem to solve. Since the classification is based on the tool-surface-medium tuple, the appropriate control strategy is predetermined this way.

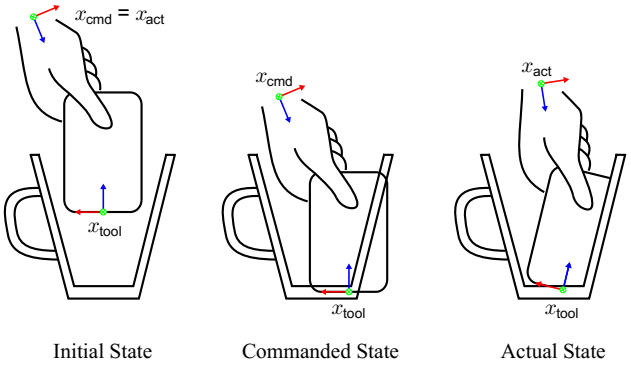


Fig. 8 The commanded state for the tool $\boldsymbol{x}_{\text{tool}}$, resulting from the virtual, spatial equilibrium $\boldsymbol{x}_{\text{cmd}}$ of the right end-effector, is geometrically unfeasible. In contact, the parameterization of the robot forces the correct alignment of the tool (including deformations) to solve the wiping task (see $\boldsymbol{x}_{\text{act}}$).

On the geometric level the task is described according to geometric relations between the involved objects. However, the compliant behavior of the real robot is difficult to model. Not speaking of possible elastic contact behavior between tool and surface which can hardly be predicted. To solve this issue human task knowledge is exploited by integrating the control level into the reasoning step. This is done by defining the Cartesian trajectory for the tool TCP $\boldsymbol{x}_{\text{tool}}$ w. r. t. the *virtual, spatial equilibrium point* $\boldsymbol{x}_{\text{cmd}}$ for the end-effector. The trajectory is computed w. r. t. the involved objects and their properties (e. g. the mug radius defines the circular trajectory for the sponge in our conceptual example). In the example at hand, only translational motions are commanded. The rotations of the tool are introduced by defining the *Cartesian object stiffness* and the *Cartesian contact force* at control level. Deviations on contact are neglected during planning time. The resulting Cartesian control torque $\boldsymbol{\tau}_{\text{car}}$ according to (12) enables a compliant robot to adapt itself to the curvature of the target surface resulting in a deliberate deviation of the end-effector from the commanded Cartesian trajectory:

$$\boldsymbol{x}_{\text{act}} = \boldsymbol{x}_{\text{cmd}} + \boldsymbol{x}_{\text{dev}}\left(\boldsymbol{q}, \boldsymbol{\mathcal{P}}, \boldsymbol{\tau}_{\text{ext}}\right) \ . \tag{13}$$

Herein $\boldsymbol{x}_{\text{act}}(\boldsymbol{q})$ describes the forward kinematics based on the link side measurements (see Fig. 8). It contains both the translation $\boldsymbol{p}_{\text{act}}(\boldsymbol{q})$ of the TCP and its orientation, depending on the chosen representation (e. g. via Euler angles). The term $\boldsymbol{x}_{\text{act}}(\boldsymbol{q})$ can be represented as the combination of the commanded TCP position/orientation $\boldsymbol{x}_{\text{cmd}}$ and the deviation $\boldsymbol{x}_{\text{dev}}$ between this command and the actual TCP position/orientation. The deviation is subject to the parameterization $\boldsymbol{\mathcal{P}}$ of the whole-body controller and the external forces/torques $\boldsymbol{\tau}_{\text{ext}}$.

The control level alone cannot guarantee a successful task execution by only following the Cartesian virtual equilibrium, since the applied local control methods are vulnerable to get stuck in local minima. Furthermore, collisions with the environment cannot be prevented that way. Therefore, a roughly approximated reference trajectory is provided during the reasoning step by computing discrete inverse kinematics solutions along the task trajectory (similar to Okada et al (2005)). During run-time, the reference trajectory is interpolated to provide an input for the joint impedance controller based on (2). As described in Sec. 3.3, the joint impedance control action is projected into the null space of the Cartesian impedance resulting in an overall whole-body motion of the robot, maintaining contact and free of local minima. Additionally control torques for self-collision avoidance $\boldsymbol{\tau}_{\text{sca}}$ (3), avoidance of mechanical end stops $\boldsymbol{\tau}_{\text{mes}}$ (4), and singularity avoidance $\boldsymbol{\tau}_{\text{sav}}$ (5) are computed with lower control task priority to react on unforeseen events.

In conclusion, the control strategy for a particular process model is represented by one particular action template. The process model involves the symbolical meaning and the geometric execution of the control strategy. The required sub-symbolic parameters such as task frames or regions of interest are provided by the involved objects. Additionally the actual controller parameterization is inherent. Preprocessed by a human programmer, this task-related knowledge is used to encode the desired compliant control behavior for the wiping action according to our classification. Such a behavior can hardly be generalized by using local control strategies only, neither, is no geometric planner known which could possibly simulate such a behavior sufficiently *stable* in a generalized way. Therefore we believe that the control level and the reasoning level have to act jointly.

This way of programming and parameterizing manipulation actions is beneficial since the programmer is forced to define the process model based on predicted physical behavior and human task knowledge, which is otherwise hardly transferred to the control level of a robot. In the following section we demonstrate this approach for whole-body manipulation tasks.

5 Evaluation

To evaluate our approach we conduct three elaborate experiments on the mobile humanoid robot Rollin' Justin (Borst et al 2009). The experiments are designed as whole-body manipulation tasks inspired by natural everyday household chores in human environments. The first experiment addresses the mug scrubbing task already introduced in the previous section. The second experiment is a window wiping task in which the robot has to clean a large window with a one-handed window wiper. The third experiment is a sweeping task where shards of broken dishes have to be collected by bimanually handling a broom. Each experiment demonstrates one particular wiping action as classified in Sec. 2. Due to the different classification and the resulting varying contact behavior the tasks have to be parameterized differently. In this paper we focus on the task parameterization and not on the reasoning aspect which is therefore less detailed. This section describes directions and rotations in robot coordinates (positive x-axis forward, positive y-axis left, positive z-axis up), while the Cartesian force and stiffness parameters are defined in the local TCP coordinate systems of the involved tools as illustrated in Table 2. The parameters are chosen based on previous experiments on wiping tasks related to learning by demonstration (Urbanek et al 2004).

5.1 Scrubbing a Mug with a Sponge

The first experiment is based on the conceptual example described in Sec. 4, scrubbing with a sponge. The goal of the wiping task is to clean the inner curvature of the mug. We classify this action as *scrubbing*. The tool is the sponge and the target surface is the inner region of the mug. The surface area can therefore actively be positioned by the robot. There are two media involved. The first medium is the dried up liquid remaining in the mug. It is not of value for the robot. The second medium is the detergent in the sponge. The symbolic representation, introduced in Fig. 4, is defined in the action template as follows:

```
_sponge.scrub:
  :parameters (?t - _sponge ?s - _dish
               ?m1 - _medium ?m2 - _medium
               ?a1 - _manipulator ?a2 - _manipulator)
  :precondition (and (absorbed ?t ?m1) (adhesive ?s ?m2)
                     (picked ?t ?a1) (picked ?s ?a2))
  :effect (and (not(adhesive ?s ?m2)) (scrubbed ?s ?m2))
```

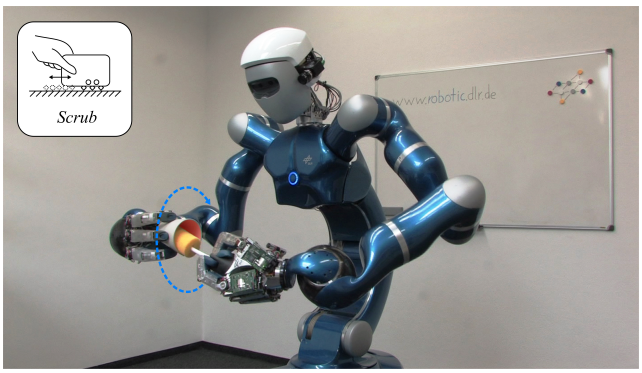
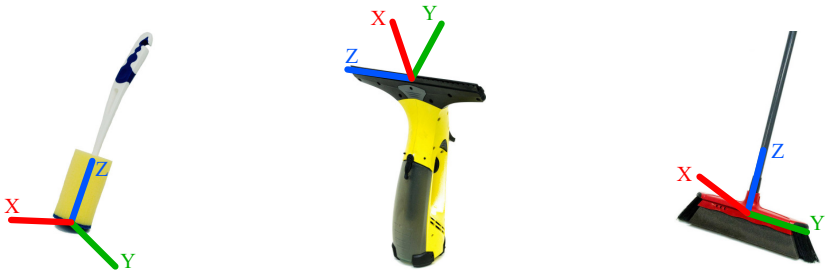


Fig. 9 The humanoid robot Rollin' Justin scrubbing a mug with a sponge.



Task Relevant Actuators	both arms, torso	one arm, torso, base	both arms, torso, base
Control Task Hierarchy	$\tau_{\text{car}}, \tau_{\text{jnt}}, \tau_{\text{mes}}, \tau_{\text{sav}}$	$\tau_{\text{car}}, \tau_{\text{sca}}, \tau_{\text{sav}}, \tau_{\text{jnt}}, \tau_{\text{mes}}$	$\tau_{\text{car}}, \tau_{\text{sca}}, \tau_{\text{jnt}}, \tau_{\text{mes}}, \tau_{\text{sav}}$
Translational Stiffness x, y, z [N/m]	400, 400, 800	100, 500, 1000	1000, 500, 300
Rotational Stiffness θ_x, θ_y, θ_z [Nm/rad]	30, 30, 60	500, 10, 10	200, 10, 500
Cartesian Force Limits x, y, z [N]	± 20, ± 20, ± 20	$-\infty/+10$, $\pm\infty$, $\pm\infty$	$\pm\infty$, $\pm\infty$, $-10/+\infty$

Table 2 Table of tool-specific parameters. Cartesian force and stiffness parameters are given w. r. t. the visualized frames.

Symbolically, the mug has to be cleaned which requires to combine detergent (previously *absorbed* by the sponge) with the coffee leftovers in the coffee mug by *scrubbing*. The preconditions require to have the mug and the sponge grasped at the same time, with two distinct manipulators. As effect, the dirt is not *adhesive* to the mug anymore but *scrubbed* and can be *skimmed* out of the mug or washed out. See Fig. 4 for the explicit symbolic representation.

In terms of reasoning, this task is especially challenging since the mug needs to be picked up upside-down to clean the inner curvature of it, which is not feasible by only executing a single *pick* action. The symbolic representation of the action template (see Fig. 4) requires the mug and the sponge to be picked as precondition. Only after backtracking on the geometric level *and* on the symbolic level (by cutting the symbolic domain, see Sec. 4.1) the symbolic planner finds the correct solution by scheduling a *handover* action:

```
_object.pick mug left_arm table,
_object.handover mug left_arm right_arm,
_object.pick sponge left_arm table,
_sponge.scrub sponge left_arm mug
```

The geometric representation of the *_sponge.scrub* action (see Fig. 4) defines a circle approximated by twice the diameter of the mug for the TCP of the sponge in order to execute forces on the medium plotted as blue dashed line in Fig. 9. The translational and rotational stiffness for the left hand (holding the sponge) are set to be low along/about the lateral axis of the sponge as described in Table 2. A Cartesian force is commanded along the circular task trajectory. The right hand (holding the mug) is commanded to be stiff for all rotations

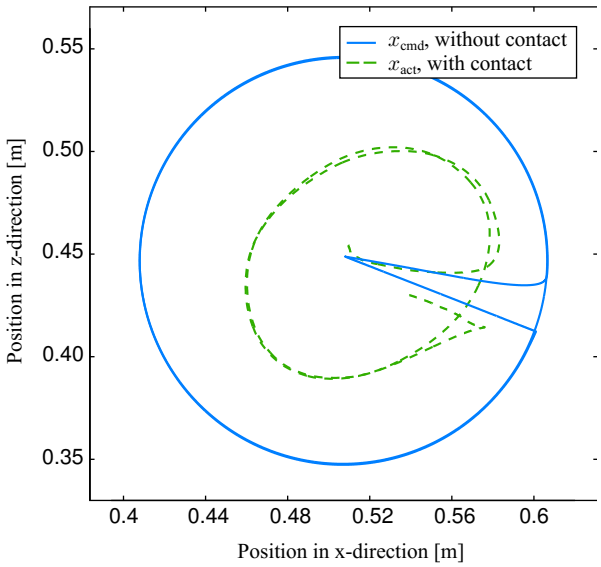


Fig. 10 The commanded position, compared with the measured position of the left hand during scrubbing.

and translations. With this parameterization we deliberately *allow* the resulting motion to significantly deviate from the commanded trajectory. This is exactly what is required to solve the given wiping task. Fig. 10 shows the resulting measured trajectory of the manipulator. During contact, the force limitations according to Table 2 apply.

Regarding whole-body control, the torso and the base are not mandatory to solve the task of cleaning dishes. However, by treating the action equally to a whole-body task, the torso and the base can be used to compensate for external disturbances and avoid obstacles while simultaneously remaining over the sink with

potentially wet dishes. Self-collision avoidance is disabled during this task to enable both manipulators to approach each other.

5.2 Skimming a Window with a Window Wiper

Window cleaning is a diverse task. It can be solved with several tools, one-handed or bi-manually, and it is applicable to many window types such as squared, round, or curved ones as outlined in Sec. 4.2. Based on the size of the window the whole body might be required to reach the entire window pane. An individual parameterization is mandatory w. r. t. individual settings.

With a width of 1.5 m and a height of 1.0 m, the window pane in our example is too large to follow the Cartesian task motion by only using the arm of the robot. Even with the aid of the torso it is not possible to maintain the contact with the window along the complete trajectory. Especially in the corners the reachability decreases until the task gets unfeasible, not to mention the reduced capability to compensate for possible disturbances during task execution. Consequently, the mobile base is mandatory to accomplish the task. The additional DOF of the mobile base can be considered by the applied discrete inverse kinematics solver. As described in Sec. 3.2, a Cartesian trajectory in the plane of motion can be directly commanded to the platform controller. This way, the base follows the lateral motion of the end-effector, respectively the window wiper, along the window pane to extend the workspace of the robot. Nevertheless it is beneficial to set the control task priority of $\boldsymbol{\tau}_{\text{sav}}$ higher during the task of window wiping to prevent singularities, especially in the corners of the window (see Table 2). The robot is initially positioned in front of the window w. r. t. the metric we defined in our prior work (Leidner and Borst 2013; Leidner et al 2014), where we used *capability maps* (Zacharias et al 2007) to compute the optimal torso configuration and base position based on the ROI of the target object.

The task is parameterized according to prior task knowledge deposited w. r. t. the involved objects. From a symbolical planning point of view the detergent (applied by a human co-operator) has to be removed from the surface area by skimming it away. As a result, the medium, i. e. the detergent will no longer be applied to the surface. The symbolic representation is defined as follows:

```
_wiper.skim:
  :parameters (?t - _wiper ?s - _window
               ?m - _detergent ?a1 - _manipulator)
  :precondition (and (picked ?t ?a1)
                     (applied ?s ?m))
  :effect (and (skimmed ?s ?m)
               (not (applied ?s ?m)))
```

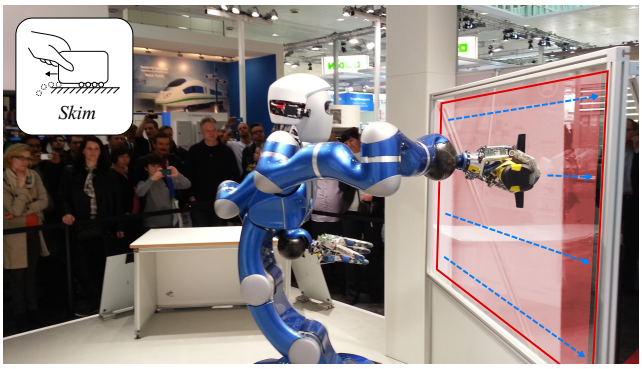


Fig. 11 Rollin' Justin wiping a window. Based on the ROI (red) and the wiper width the Cartesian task trajectory (blue, dashed arrows) is computed. The experiment was successfully demonstrated to the public at Hanover Fair 2013.

As precondition, the window wiper has to be grasped by the robot, while a detergent is applied to the surface. The resulting symbolic transition consists of a *pick* and a *skim* action:

```
_object.pick wiper table right_arm,
_wiper.skim window right_arm
```

The skim *action template* also describes the geometric wiping task based on our classification in Sec. 2. The ROI is defined in the data storage of the mock-up window. Together with the dimension of the wiper blade the Cartesian task trajectory can be computed as illustrated in Fig. 11. The TCP of the wiper is defined in the center of the blade and is used to describe the tool alignment (see Table 2). The Cartesian stiffness refines the alignment by introducing freedom about the y-axis and the z-axis of the object TCP. This way a compliant behavior of the blade can be achieved leading to a more robust task execution. As already described in Fig. 8 the wiper aligns near optimal with the window plane. Local errors and external disturbances can be compensated this way while maintaining the wiping. The Cartesian force is predefined by the window wiper and may vary from tool to tool. In this case we exert a maximum Cartesian force of $\boldsymbol{f}_x^{\max} = 10\,\text{N}$. The object stiffness parameterization for the window wiper is listed in Table 2.

One of the main difficulties in wiping windows is the perception part. The transparent window pane offers almost no visual features so that the window pane can only be estimated by localizing the window frame. However, even with an imperfect localization the robot is able to execute the commanded window wiping task. This is achieved due to the compliant behavior of the robot and the individual controller parameterization which can only be provided if the reasoning level and the control level act jointly. The comparison of the commanded and the measured trajectory in Fig. 12 illus-

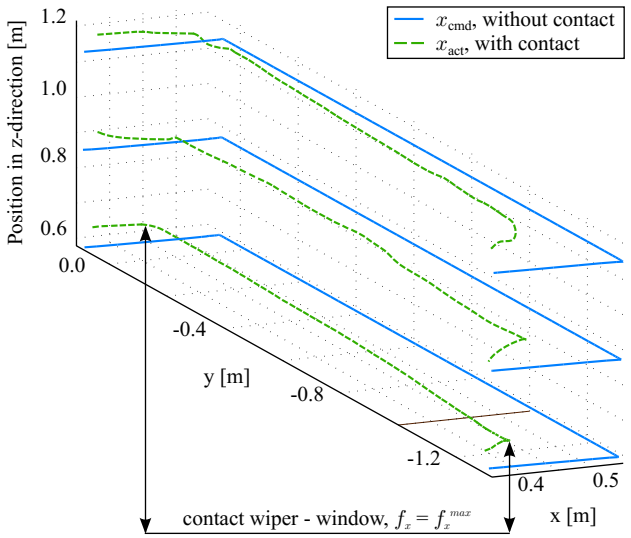


Fig. 12 This plot compares the commanded virtual equilibrium point with the measured position of the right hand during the window wiping task. The three axes correspond to the three dimensions of the window (similar to robot coordinates), where z is the height and y is the width. The deviation along the x-axis corresponds to the contact of the wiper with the window pane. Note that the transit paths, connecting two contact situations, are removed for clarity.

trates the resulting behavior in the translational deviation of the right arm guiding the wiper along the window pane. Even collisions between the wiper and the window frame (see Fig. 12, first row, upper left) can be handled this way. In the future we plan to integrate the localization uncertainties directly into the controller parameterization to further enhance the task performance.

5.3 Collecting Shards with a Broom

The third task is a *collecting* task where a broom is used to collect shards of a broken mug. This requires both arms to handle the broom while the mobile base and the torso support the motion to create an overall wiping trajectory along a larger area. Both arms, the torso, and the mobile base act jointly. Similar to the window cleaning task, the broom needs to be picked before it can be used. Since the broom has to be handled bi-manually, the mass parameters have to be divided among the manipulators. The pick action template is therefore specialized by the *_broom* object class. The symbolic representation of the *collect* action template is defined as follows:

```
_broom.collect:
  :parameters (?t - _broom ?s - _floor ?m - _dish
               ?a1 - _manipulator ?a2 - _manipulator)
  :precondition (and (picked ?t ?a1) (picked ?t ?a2)
                     (broken ?m))
  :effect (and (collected ?s ?m))
```

This example illustrates the use of a medium other than small particles or liquids. As defined in the symbolic precondition section, the mug can only be collected with the broom if it is marked as broken. In this form the mug constitutes the medium for the wiping task of collecting shards. In case of the desired symbolic goal state *collected floor mug*, the symbolic planner yields the following symbolic transition:

```
_broom.pick broom table right_arm left_arm,
_broom.collect broom mug right_arm left_arm
```

The shards define the ROI in this particular collecting task. A *skimming* task to remove dust from the floor would, however, rely on the whole accessible floor of a room as target surface and a completely different parameterization (e. g. higher contact force, lower stiffness). The experiment is shown in Fig. 13.

The Cartesian task trajectory for the broom is stored in the knowledge base and defines a simple ellipse intersecting the floor plus a linear lateral offset. The elliptic motion is thought to be executed by the arms and the lateral motion by the mobile base. This way the contact during the trajectory execution is slightly overlapping. The trajectory can be imagined as a loop (see Fig. 14). However, with this trajectory alone the robot is not able to collect the shards with the broom. Only after setting the correct Cartesian stiffness the task can be solved. The desired motion of the tool involves a tilted TCP, wiping along the floor (see Table 2). This requires a low rotational stiffness along the brush attachment (y-axis) and a low translational stiffness along the shaft (z-axis). The force against the ground (z-axis) is limited to $\boldsymbol{f}_z^{\min} = -10\,\mathrm{N}$ so that the virtual equilibrium point for the broom is not pushed too far. The desired

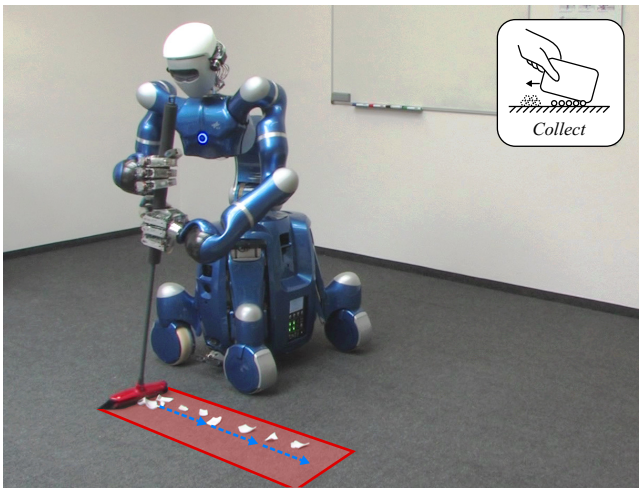


Fig. 13 Rollin' Justin collecting shards of a broken mug. The ROI (red) is defined by the broom width and the shards. Along this region, a cyclic trajectory has to be executed by the broom (blue, dashed arrow).

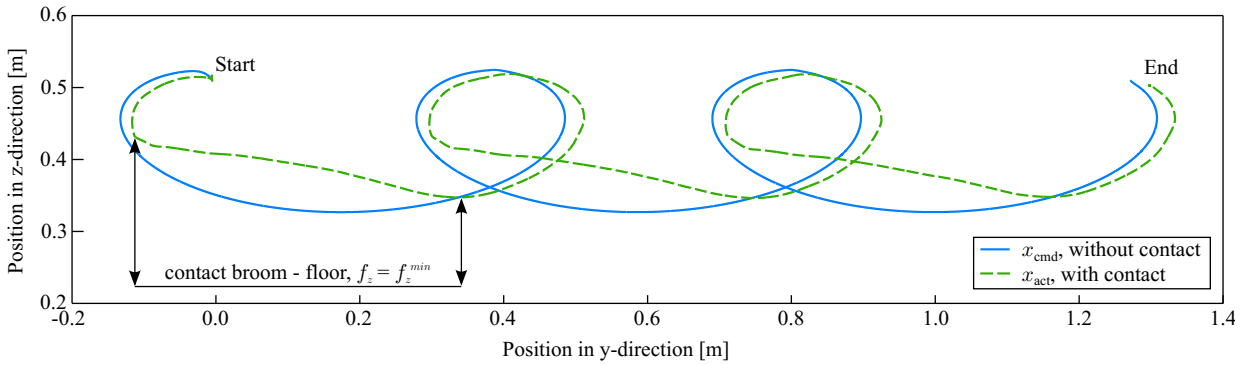


Fig. 14 Frontal view of the commanded and measured TCP of the right manipulator. During the contact phase the hands are both following a tilt motion introduced by the sweeping broom which reflects the low rotational stiffness around the brush.

Cartesian object stiffness for the broom is listed in Table 2. The coupling of stiffness and damping for bimanual tool handling was already examined on Rollin' Justin (Wimböck 2013; Florek-Jasinska et al 2014). Yet we approximate the desired stiffness of the individual manipulators for this experiment, since the strategy is not yet integrated in this framework. Additionally to the Cartesian stiffness we define the *joint stiffness* for the torso to be twice the default value, so that the main compliance is anticipated by the arms.

This experiment shows how whole-body manipulation of closed kinematic chains can be integrated into a hybrid reasoning framework without actually handling the closed kinematic chain. All relevant assumptions have been made in the object space and are therefore robot-independent (except for the torso parameterization, which is exemplary, not mandatory). The resulting behavior solves the commanded sweeping task. Eventually all shards are collected in one area. The actual measured trajectory of the right manipulator in Fig. 14 shows that it is not straightforward to define the correct task motion by programming it in detail. We believe that the *local* adaption on control level paired with *globally* applied human task knowledge is a well-suited method to solve this issue.

The experiments show that the concept of action templates is suitable to describe the process models of wiping tasks w. r. t. whole-body control behavior of a humanoid robot. For all tasks the same controller was utilized. It is shown that various components can be addressed to solve even complex whole-body manipulation tasks rather than only pick-and-place tasks. All information needed to describe the tasks was stored within the context of the involved objects. The task reasoning and controller parameterization was autonomously performed by the robot with the use of the provided action templates. All tasks are shown in the video attachment.

6 Discussion

The main advantage of the approach presented in this paper is the relative ease with which a manipulation task can be defined in order to act correctly on the control level. A programmer has to develop a single action template only, utilizing prior task knowledge including the task trajectory, Cartesian object stiffness, and Cartesian force for the controller parameterization. However, it is not straightforward to select the correct parameters. Unfortunately, there exist no detailed studies on exerted force and stiffness during manipulation tasks for robots, neither for humans. This lack can be mainly traced back to impracticable measuring procedures during tool usage. Nevertheless, Cartesian tool trajectories can be tracked. Doing so it would be possible to automatically learn the correct force, torque, and stiffness parameterization with suitable machine learning strategies, providing the desired Cartesian trajectory by human demonstration.

Currently, the Cartesian task trajectories are handcrafted by a skilled programmer. However, the trajectories rely mainly on the physical appearance of the objects involved in the task execution, such as the shape of a window pane. We plan to use this information together with the goals of the wiping tasks defined in Fig. 2 to automatically generate whole-body motions for humanoid robots in realistic scenarios. One challenge w. r. t. this issue is to generate continuous whole-body motions in complex environments based on a desired Cartesian workspace trajectory for a given tool. Additionally, superimposing oscillating/cyclic motions will be mandatory to guarantee appropriate task outcome in case of cleaning tasks, such as scrubbing the floor with a mop.

Integrating the parameterization of whole-body controllers into the task reasoning procedure is only the first step to solve more complex compliant tool usage scenarios. As we emphasized with our classification of

wiping tasks, the effects on the medium are the main aspects of any wiping task. To sufficiently achieve this it is mandatory to verify the outcome of the commanded action during and after task execution. Our next step is therefore the integration of the control strategies into the *feedback loop* of the reasoning level, yet suitable metrics have to be defined. Together with visual feedback we believe that the task performance can be increased further.

7 Summary

In this work we presented a generic combination of high-level hybrid reasoning mechanisms with a low-level whole-body control framework. Within this context a classification of compliant wiping tasks was conducted to extract relevant task parameters. Based on these parameters we provide human task knowledge in the context of the involved objects to describe tasks symbolically and geometrically. The compliant contact behavior of the whole-body controller is realized as a hierarchical stack of control tasks via null space projections. The wiping actions are thereby described by a Cartesian task trajectory for the virtual equilibrium of the end-effectors and the Cartesian object stiffness of the tool. This way the whole-body controller can react on the actual environmental conditions. We evaluated our approach in three experiments on the humanoid robot Rollin' Justin. First, a scrubbing task was executed to clean the inner curvature of a mug with a sponge. Second, a window wiping task was conducted where the robot had to coordinate right arm, torso and base motion to skim along an entire window pane. The third task involved both arms, the torso, and the mobile base in a sweeping task to collect shards of a broken mug. The attached video clearly demonstrated the potential of the proposed concept as well.

8 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

Albu-Schäffer A, Ott C, Frese U, Hirzinger G (2003) Cartesian Impedance Control of Redundant Robots: Recent Results with the DLR-Light-Weight-Arms. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp 3704–3709

Albu-Schäffer A, Eiberger O, Grebenstein M, Haddadin S, Ott C, Wimböck T, Wolf S, Hirzinger G (2008) Soft robotics. Robotics Automation Magazine, IEEE 15(3):20–30

Antonelli G (2009) Stability Analysis for Prioritized Closed-Loop Inverse Kinematic Algorithms for Redundant Robotic Systems. IEEE Transactions on Robotics 25(5):985–994

Bartels G, Kresse I, Beetz M (2013) Constraint-based movement representation grounded in geometric features. In: Proceedings of the IEEE/RAS International Conference on Humanoid Robots (ICHR), pp 547–554

Bloomfield A, Deng Y, Wampler J, Rondot P, Harth D, McManus M, Badler N (2003) A taxonomy and comparison of haptic actions for disassembly tasks. In: Proc. of the Virtual Reality Conference, pp 225–231

Borst C, Wimböck T, Schmidt F, Fuchs M, Brunner B, Zacharias F, Giordano PR, Konietschke R, Sepp W, Fuchs S, et al (2009) Rollin'justin-mobile platform with variable base. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp 1597–1598

Bullock IM, Dollar AM (2011) Classifying human manipulation behavior. In: Proc. of the IEEE International Conference on Rehabilitation Robotics (ICORR), pp 1–6

Bullock IM, Ma RR, Dollar AM (2013) A hand-centric classification of human and robot dexterous manipulation. IEEE Transactions on Haptics 6(2):129–144

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

Cutkosky MR (1989) On grasp choice, grasp models, and the design of hands for manufacturing tasks. IEEE Transactions on Robotics and Automation 5(3):269–279

Diankov R (2010) Automated construction of robotic manipulation programs. PhD thesis, Carnegie Mellon University, Robotics Institute

Dietrich A, Wimböck T, Albu-Schäffer A (2011a) Dynamic Whole-Body Mobile Manipulation with a Torque Controlled Humanoid Robot via Impedance Control Laws. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3199–3206

Dietrich A, Wimböck T, Täubig H, Albu-Schäffer A, Hirzinger G (2011b) Extensions to Reactive Self-Collision Avoidance for Torque and Position Controlled Humanoids. In: Proc. of the IEEE International Conference on Robotics and Automation

(ICRA), pp 3455–3462

Dietrich A, Albu-Schäffer A, Hirzinger G (2012a) On Continuous Null Space Projections for Torque-Based, Hierarchical, Multi-Objective Manipulation. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp 2978–2985

Dietrich A, Wimböck T, Albu-Schäffer A, Hirzinger G (2012b) Reactive Whole-Body Control: Dynamic Mobile Manipulation Using a Large Number of Actuated Degrees of Freedom. IEEE Robotics & Automation Magazine 19(2):20–33

Dietrich A, Ott C, Albu-Schäffer A (2013) Multi-Objective Compliance Control of Redundant Manipulators: Hierarchy, Control, and Stability. In: In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 3043–3050

Dietrich A, Ott C, Albu-Schäffer A (2015) An Overview of Null Space Projections for Redundant, Torque-Controlled Robots. International Journal of Robotics Research Doi:10.1177/0278364914566516

Do M, Schill J, Ernesti J, Asfour T (2014) Learn to wipe: A case study of structural bootstrapping from sensorimotor experience. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp 1858–1864

Dornhege C, Hertle A (2013) Integrated symbolic planning in the tidyup-robot project. In: AAAI Spring Symposium Series

Dornhege C, Eyerich P, Keller T, Trüg S, Brenner M, Nebel B (2012) Semantic attachments for domain-independent planning systems. In: Towards Service Robots for Everyday Environments, Springer, pp 99–115

Feix T, Pawlik R, Schmiedmayer HB, Romero J, Kragic D (2009) A comprehensive grasp taxonomy. In: Robotics, Science and Systems: Workshop on Understanding the Human Hand for Advancing Robotic Manipulation, pp 2–3

Florek-Jasinska M, Wimböck T, Ott C (2014) Humanoid compliant whole arm dexterous manipulation: Control design and experiments. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 1616–1621

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

Giordano PR, Fuchs M, Albu-Schäffer A, Hirzinger G (2009) On the Kinematic Modeling and Control of a Mobile Platform Equipped with Steering Wheels and Movable Legs. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp 4080–4087

Gravot F, Cambon S, Alami R (2005) asymov: a planner that deals with intricate symbolic and geometric problems. In: The Eleventh International Symposium Robotics Research, Springer, pp 100–110

Helmert M (2006) The fast downward planning system. Journal of Artifcial Intelligence Research 26:191–246

Hess JM, Tipaldi GD, Burgard W (2012) 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), pp 1923–1928

Hogan N (1985) Impedance Control: An Approach to Manipulation: Part I - Theory, Part II - Implementation, Part III - Applications. Journal of Dynamic Systems, Measurement, and Control 107:1–24

Kaelbling LP, Lozano-Pérez T (2013) Integrated task and motion planning in belief space. The International Journal of Robotics Research 32:1–60

Kallmann M, Thalmann D (1999) Modeling objects for interaction tasks. In: Computer Animation and Simulation 98, Springer, pp 73–86

Kapandji IA, Honoré LH (1970) The physiology of the joints: annotated diagrams of the mechanics of the human joints, vol 1. E. & S. Livingstone London

Khatib O (1987) A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. IEEE Journal of Robotics and Automation RA-3(1):43–53

Kronander K, Billard A (2014) Learning compliant manipulation through kinesthetic and tactile human-robot interaction. Haptics, IEEE Transactions on 7(3):367–380

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

Leidner D, Borst C (2013) Hybrid reasoning for mobile manipulation based on object knowledge. In: Workshop on AI-based Robotics at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Leidner D, Borst C, Hirzinger G (2012) Things are made for what they are: Solving manipulation tasks by using functional object classes. In: Proc. of the IEEE/RAS International Conference on Humanoid Robots (ICHR), pp 429–435

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

Levison L (1996) Connecting planning and acting via object-specific reasoning. PhD thesis, University of Pennsylvania

Moro FL, Gienger M, Goswami A, Tsagarakis NG, Caldwell DG (2013) An Attractor-based Whole-Body Motion Control (WBMC) System for Humanoid Robots. In: Proc. of the IEEE-RAS International Conference on Humanoid Robots (ICHR), pp 42–49

Mosenlechner L, Beetz M (2011) Parameterizing actions to have the appropriate effects. In: Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp 4141–4147

Nakamura Y, Hanafusa H, Yoshikawa T (1987) Task-Priority Based Redundancy Control of Robot Manipulators. International Journal of Robotics Research 6(2):3–15

Okada K, Ogura T, Haneda A, Fujimoto J, Gravot F, Inaba M (2005) Humanoid motion generation system on hrp2-jsk for daily life environment. In: Proc. of the IEEE International Conference on Mechatronics and Automation (ICMA), pp 1772–1777

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

Ott C (2008) Cartesian Impedance Control of Redundant and Flexible-Joint Robots, Springer Tracts in Advanced Robotics, vol 49. Springer Publishing Company, Berlin Heidelberg

Ott C, Albu-Schäffer A, Kugi A, Stramigioli S, Hirzinger G (2004) A Passivity Based Cartesian Impedance Controller for Flexible Joint Robots - Part I: Torque Feedback and Gravity Compensation. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp 2659–2665

Ott C, Albu-Schäffer A, Kugi A, Hirzinger G (2008) On the Passivity-Based Impedance Control of Flexible Joint Robots. IEEE Transactions on Robotics 24(2):416–429

Sadeghian H, Villani L, Keshmiri M, Siciliano B (2014) Task-Space Control of Robot Manipulators With Null-Space Compliance. IEEE Transactions on Robotics 30(2):493–506

Sentis L, Khatib O (2005) Synthesis of Whole-Body Behaviors through Hierarchical Control of Behavioral Primitives. International Journal of Humanoid Robotics 2(4):505–518

Siciliano B, Slotine JJ (1991) A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems. In: Proc. of the International Conference on Advanced Robotics (ICAR), pp 1211–1216

Tan J, Xi N, Wang Y (2003) Integrated task planning and control for mobile manipulators. The International Journal of Robotics Research 22(5):337–354

Tenorth M, Klank U, Pangercic D, Beetz M (2011) Web-enabled robots. IEEE Robotics & Automation Magazine 18(2):58–68

Tenorth M, Perzylo A, Lafrenz R, Beetz M (2012) The roboearth language: Representing and exchanging knowledge about actions, objects, and environments. In: Proc. of the IEEE International Conference on Robotics and Automation (ICRA), pp 1284–1289

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

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

Wimböck T (2013) Controllers for compliant two-handed dexterous manipulation. PhD thesis, Vienna University of Technology

Wolfe J, Marthi B, Russell SJ (2010) Combined task and motion planning for mobile manipulation. In: Proc. of the International Conference on Automated Planning and Scheduling (ICAPS), pp 254–258

Yamamoto Y, Yun X (1992) Coordinating locomotion and manipulation of a mobile manipulator. In: Proc. of the IEEE Conference on Decision and Control (CDC), pp 2643–2648

Yoshikawa T (1990) Foundations of Robotics: Analysis and Control. The MIT Press, Cambridge

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