Grant Agreement no.	**621447**
Project acronym	**R5-COP**
Project full title	**Reconfigurable ROS-based Resilient Reasoning Robotic Cooperating Systems**
Dissemination level	CO / PU
Date of Delivery	dd/mm/yy
Deliverable Number	Dab.cd
AL / Task related	(fill me)
Author	(PARTNERSHORTNAME) Person
Contributors	(PARTNERSHORTNAME) Person
Keywords	(fill me)
Abstract	(fill me)

Document History

Document History			
Ver.	**Date**	**Changes**	**Author**
0.0			

Note: Filename should be
"R5-COP_D##_#.pdf", e.g. „R5-COP_D91.1_v0.1_TUBS.pdf"

Fields are defined as follows

1. **Deliverable number *.***
2. **Revision number (should not be required with LaTeX ... use versioning!)**
   - **draft version** **v**
   - **approved** **a**
   - **version sequence (two digits)** ***.***
3. **Company identification (Partner acronym)** *****

Table of Contents

List of Figures

List of Tables

List of Acronyms

ACRONYM	definition text

1 Introduction

1.1 Summary (abstract)

Whole-body control is a promising control method for domestic service robots. However, since whole-body control is a potential field-based method, it only operates at a local scope and it might therefore be unable to find a global solution in a complex environment with local minima. Therefore, to integrate a whole-body controller with a task executer, a planner is required that provides connectivity information at a global scope while constraining only the DoFs that are relevant for the (sub-task) at hand. The contribution of this chapter is to apply a topological planner for manipulation, where the topological graph is an abstract representation of possible robot configurations that describes possible sequences of motions. All nodes in the topological graph are grounded to specific end-effector attractors. Each attractor can constrain up to six DoFs and possesses a number of properties, such as the target pose, target offset, stiffness and goal tolerances. The feasibility of the motion is subsequently validated by forward integration of the closed-loop whole-body controller.

1.1.1 Results

1.1.1.1 Reports

Title / URL	Author	Year
Reactive collision avoidance for domestic service robots	Metsemakers, P.M.G.	2013
Global path planning for the reactive whole-body controller	Derksen, T.J.J.	2014
Context-aware design and motion planning for autonomous service robots	Lunenburg, J.J.M.	2015

1.1.1.2 Software

Package / URL	Description
tue-robotics/whole_body_controller	Whole-Body Controller ROS Package
tue-robotics/whole_body_planner	Whole-Body Cartesian planning utility ROS Package

1.1.1.3 Videos

Video / URL	Description
youtube/reactive_collision_avoidance	Reactive collision avoidance with the AMIGO robot
youtube/whole_body_planning	Whole-Body motion planning with the AMIGO robot

1.2 Purpose of this document

What should we say here?

1.3 Partners involved

Partners and Contribution	
Short Name	**Contribution**
TU/e	Implementation of a Whole Body Controller

Do we have other partners?

2 An introduction to Whole-body control

Over the past years, increasing research attention has been devoted to domestic service robots. These robots usually consist of a mobile base with one or two robotic arms. These arms have six or seven Degrees-of-Freedom (DoFs) and are commonly mounted on a torso with additional DoFs. A fundamental competence of these robots is the ability to safely manipulate objects in a domestic environment.

Many of the control methods for this purpose are based on the operational space formulation (Khatib, 1987). This formulation eventually resulted in whole-body control, of which implementations can be found in, e.g., Nagasaka et al (2010); Dietrich et al (2012) and Gienger et al (2005); Sentis and Khatib (2005), where the latter references concern biped humanoid robots instead of wheeled mobile robots. There are various motives that justify the use of whole-body controllers in a domestic environment:

- Robustness against dynamic environments due to the presence of humans. This requires reactive motions, which is a key feature of whole-body controllers. A task of a whole-body controller is typically defined as an impedance in Cartesian space, i.e., as a virtual spring between the current end-effector pose and the desired end-effector pose (also called 'attractor'). By defining this as a compliant spring rather than a stiff trajectory, tasks involving interaction with the environment are more robust against inaccuracies in the environment model and this compliance enhances safety in case of undesired interaction with the environment.
- the kinematic redundancy of domestic service robots. Both the path and goal constraints of a task are typically defined in Cartesian space, and due to this redundancy the specification of joint positions and trajectories is no longer a practical means of defining a task (Brock and Khatib, 2002). This is especially true if a robot has multiple manipulators. In whole-body control, the redundancy is utilized for online optimization of secondary motion objectives such as keeping a desired posture and avoiding joint limits.
- Since the end-effectors are controlled in Cartesian space, a point-to-point motion typically results in an end-effector trajectory that is straight in Cartesian space, which looks much more human-like than a trajectory that is straight in joint space.

However, since whole-body control is a potential field based method, it only operates at a local scope. As a result, it might be unable to find a global solution in a complex environment with local minima. This illustrates the need for a global planning method on top of the whole-body controller, as is recognized by, e.g., Yang and Brock (2006); Toussaint et al (2007); Behnisch et al (2010, 2011); Dietrich et al (2012).

An important issue for this global planning method is the question which DoFs to constrain. Already in Nakanishi et al (2007), it is argued that it is desirable to control only the minimum number of DoFs of a robot such that the remaining DoFs can be utilized to optimize secondary motion objectives in the null space of the primary task. The minimum number of DoFs required to complete a task may even vary during a motion. An example is grasping an object from a table. Initially, the end-effector moves to the correct height while being close to the robot to avoid the table (the height and distance to the robot body are constrained by the task), then moves to a pre-grasp pose and a grasp pose (up to six DoFs need to be constrained). After closing the gripper, the object is lifted (the height and possibly, to keep an object such as a coffee cup level, roll and pitch are constrained) and the arm is retracted (the end-effector needs to be close to the robot and roll and pitch are possibly constrained).

To integrate a whole-body controller in a task executer, a planner is required that provides connectivity information in a global scope. Furthermore, it constrains only the DoFs that are relevant for the (sub-) task at hand to maximize the number of redundant DoFs that can subsequently be used to optimize secondary motion objectives.

3 Related work and contribution

In literature, examples can be found of motion planners that i) plan in joint space, ii) plan an end-effector path in Cartesian space, typically represented as a sequence of attractors or iii) call motion primitives from a task executer.

A recent example of a joint space planner can be found in Şucan and Kavraki (2012). However, this planner does not take path constraints into account. To include path constraints when planning in the joint space, a constraint manifold is computed and a path is searched on that manifold (Şucan and Chitta, 2012). Although good results can be obtained with these planners, there is no robustness against dynamic environments: the increasing number of degrees of freedom increases computational complexity for motion planning in joint space. For a 7-DoF arm, planning times are in the order of magnitude of 0.5 s and these planning times will increase with increasing number of DoFs. This limits the ability to replan (Brock and Khatib, 2002). Furthermore, planning in joint space may result in motions that are far from human-like, which is undesired for anthropomorphic domestic robots.

For whole-body controllers, the planners do not result in an explicit trajectory but rather a sequence of attractors in Cartesian space. These attractors can constrain up to six DoFs of the end-effector, i.e., both the position and the orientation.

In Brock and Khatib (2002), the concept of elastic strips is introduced as a framework to robustly execute a global motion. However, it is not discussed how this global motion is computed or how to recover from an invalidation of the global motion. To include this global connectivity information, elastic roadmaps are introduced in Yang and Brock (2006, 2010). Here, a sampling-based planning approach is used to create a roadmap in workspace. The elasticity implies that individual attractors are allowed to move in response to obstacles while constantly updating the connections between them. In these references, there is a distinction between end-effector placement and position-constrained end-effector motion and it is recognized that, depending on the task at hand, the end-effector orientation may also be constrained throughout the motion. Nevertheless, integration of this method with a task planner and executer and the selection of the DoFs to constrain are not discussed.

Hybrid planning methods, i.e., methods that plan in Cartesian space but verify the motions in joint space, can be found in, e.g., Ojdanic and Graser (2007); Behnisch et al (2010) and Behnisch et al (2011). In Ojdanic and Graser (2007), a 3D cell decomposition is searched for a global path. For every visited cell, an inverse kinematics (IK) solution is computed. If no collision free IK solution exists for a certain cell, this will get infinite costs. The orientation of the end-effector during the motion is neither released nor explicitly specified but the end-effector is rotated gradually from its start towards its goal orientation.

In Behnisch et al (2010, 2011), an Expansive Space Tree (Hsu et al, 2002) is used as a global search component. The connectivity between attractors is determined by integrating the closed-loop system forward in time using the Jacobian pseudo-inverse. Here, tasks are defined by either the position (3D) or by both the position and the attitude (5D) of the end-effector.

In Toussaint et al (2007), a motion is represented by a sequence of task space attractors. The exact position of these attractors is optimized with respect to smoothness of the motion, collision distance measures and joint limit avoidance. In Gienger et al (2008), it is also recognized that a task can be described by individual positions or orientations. Nevertheless, these references do not formally describe how to choose these DoFs. Furthermore, the constrained DoFs are also constant over the computed trajectories.

A completely different approach, not relying on whole-body planning and control, is applying parameterized motion primitives which are called from a task executer. An example can be found in, e.g., Stückler et al (2012). In this reference, it is also argued that a direct reach toward the object is often collision-free, making a time-consuming joint-space planner superfluous. This corresponds to our experience in the RoboCup@Home competition (Wisspeintner et al, 2009): in AMIGO's (Lunenburg et al, 2014) grasping and placing motions, the yaw of the gripper is usually left unconstrained because many of the objects that are manipulated fit in the robot's grippers in any orientation. By moving the end-effector up close to the body before a grasping or placing motion and by retracting to a height above the surface where the object is located, collision-free grasping and placing motions can be performed, even without an explicit collision-avoidance algorithm. In the approach AMIGO used during previous RoboCup competitions, these motions are called separately by a pre-programmed state machine. Appropriate values for the DoFs that are irrelevant for the task at hand are selected ad-hoc rather than optimized in realtime. Although this approach works in practice, it limits the re-usability for other tasks.

In the references mentioned in this section, either i) the number of DoFs that need to be constrained is constant, e.g., three, five or six DoFs are constrained over the entire motion ii) the integration with task planner or executer is not discussed or iii) motions are pre-programmed in a state-machine.

To obtain a planner that can be used for multiple tasks while constraining the minimum number of DoFs during the motion, one can find inspiration in navigation of mobile robots. In navigation systems for mobile robots, an additional hierarchical layer on top of the metric planners can be added in the form of a topological map, which is an abstract representation that describes relationships among features of the environment, without any absolute reference system (Zavlangas and Tzafestas, 2002). A topological map is usually represented in graph form, where the nodes represent sectors and the edges represent gateways. These sectors and gateways commonly have a semantic label associated with them, such as a room, a hallway or a door.

The contribution of this chapter is to apply a topological planner for manipulation. The nodes of the topological graph represent the pose of the attractors and also the stiffness and the tolerances. Unlike the aforementioned planners, different nodes can constraint different DoFs. Additionally, the nodes have a semantic label, e.g., pre-grasp, grasp and retract, associated with them. This approach:

- minimizes the number of constrained DoFs during motions, resulting in motions that are closer to optimality and more robust to external disturbances since the redundant DoFs can be used to optimize secondary motion objectives. Furthermore, the motions are more human-like compared to sampling-based joint space planners because no random sampling is involved.
- formalizes the various motion stages, offloading the calls to the various motion primitives from the task executer and thereby enhancing re-usability.

This chapter is organized as follows: in Section 4 whole-body control will be briefly introduced.

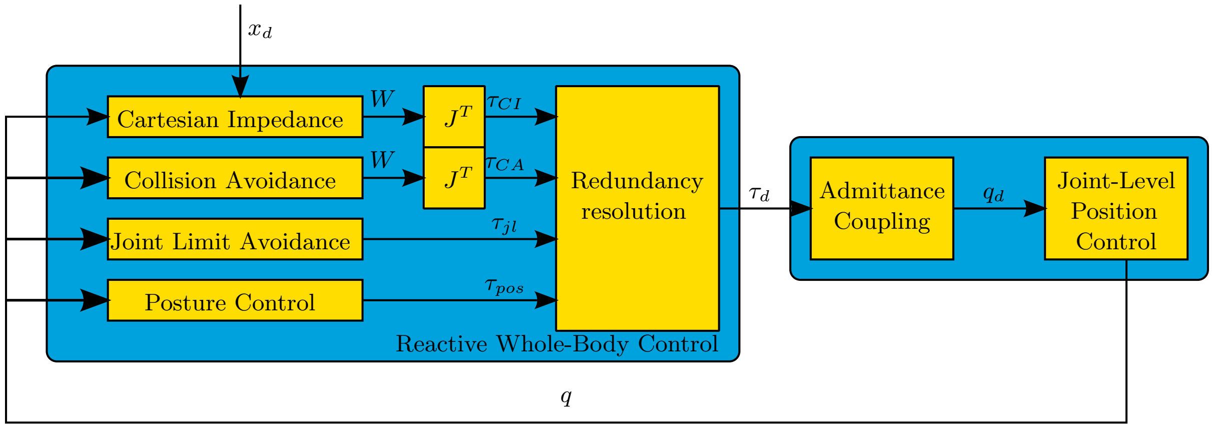


Figure 1: The whole-body controller.

The topological motion planner will be introduced in Section 5, including verification of the computed attractors. Since the implementation of the planner and controller have not been completed, only a preliminary simulation result is shown in Section 6. This chapter ends with the discussion in Section 7.

4 Whole-body control

The whole-body controller that is used in this chapter is similar to the approach in Dietrich et al (2012), see Figure 1. In this scheme, a Cartesian impedance is present for task execution. As mentioned in the introduction, it basically spans a virtual spring and damper in up to six DoFs between the desired and the actual end-effector position, resulting in a wrench $\mathbf{W}$ on that specific link. Using the transpose of the relevant Jacobian matrices $J^T(\mathbf{q})$, joint torques τ are computed using

$$\tau = J^T(\mathbf{q})\,\mathbf{W}, \tag{1}$$

where $\mathbf{q}$ are the n joint positions and τ the corresponding desired joint torques. Since AMIGO currently does not have torque controlled joints, an admittance coupling is used to computed desired joint velocities and joint positions:

$$M_{\mathbf{a}}\ddot{\mathbf{q}} + D_{\mathbf{a}}\dot{\mathbf{q}} = \tau. \tag{2}$$

Here, $M_{\mathbf{a}}$ and $D_{\mathbf{a}}$ are a virtual diagonal mass and damping matrix. The parameters are chosen such that the closed-loop position controlled joints are able to track the desired references q_d.

The (self-) collision avoidance algorithm works in a similar fashion: it puts a repulsive force on a robot link if the distance between this two links or between a link and the environment decreases below a certain safety threshold.

As secondary objectives, joint limit avoidance and posture control are present. The joint limit avoidance algorithm puts a torque on the joints if q_i gets too close to its limits:

$$\begin{cases} \tau_{i,jl} = k_{i,jl}\frac{q_{\min,\text{thresh},i}-q_i}{(q_{\max}-q_{\min})^2} & \text{for } q_i < q_{\min,\text{thresh},i} \\ \tau_{i,jl} = k_{i,jl}\frac{q_{\max,\text{thresh},i}-q_i}{(q_{\max}-q_{\min})^2} & \text{for } q_i > q_{\max,\text{thresh},i} \end{cases} \tag{3}$$

where q_i is the joint position of joint i, $k_{i,jl}$ is a gain, $q_{\min}$ and $q_{\max}$ are the lower and upper joint limits and $q_{\min,\text{thresh},i}$ and $q_{\max,\text{thresh},i}$ are the thresholds below/above which a repulsive torque is applied.

The posture controller works similarly to keep the robot configuration as close as possible to its desired configuration:

$$\tau_{i,pos} = k_{i,pos} \frac{q_{0,i} - q_i}{(q_{\max} - q_{\min})^2} \tag{4}$$

where $k_{i,pos}$ is a gain and $q_{0,i}$ is the desired joint position of joint i.

In principle, more motion objectives could be integrated that, e.g., minimize torques or keep a certain object in view of the camera. As some objectives such as collision avoidance are more important than others such as posture control, a hierarchy is constructed. Hereto, the torques of less important objectives are projected into the nullspace of more important objectives. This is called the redundancy resolution.

5 Topological arm navigation

The attractors that are fed to the Cartesian impedance in Section 4 are computed by a topogical planner, which is discussed in this section. First, the topological graph is introduced, followed by the specific properties that must be defined for each node, i.e., grounding the node. Thereafter, it is discussed how the computed motion is validated in joint space.

5.1 Topological graph

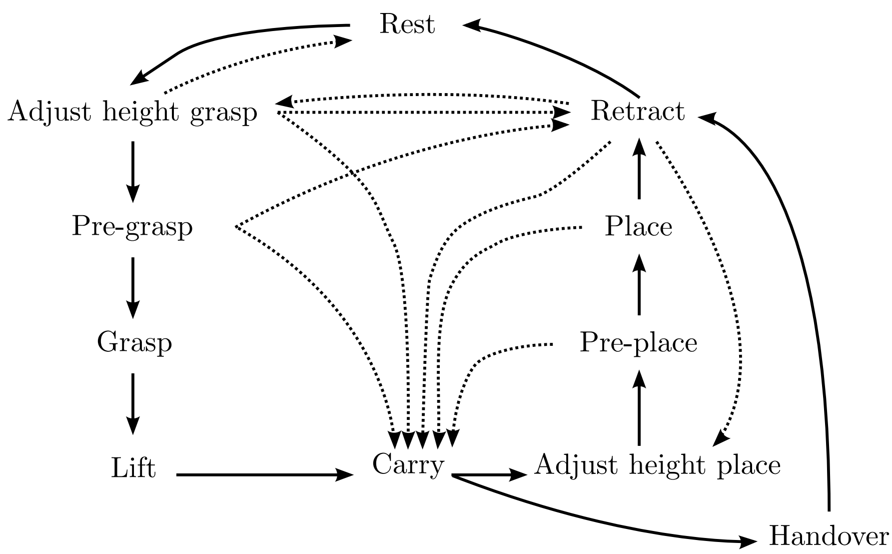


Figure 2: Graph of the various manipulation (intermediate) goals used in RoboCup 2014. The solid arrows denote motions that were performed in practice, while the dotted arrows represent feasible motions that are not common in practice.

According to Zavlangas and Tzafestas (2002), a topological map is an abstract representation that describes relationships among features of the environment, without any absolute reference system. A topological map is typically represented in graph form. In our implementation of a topological planner for manipulation, the topological graph is an abstract representation of possible robot configurations that describes possible sequences of motions.

In Figure 3, an overview of the various manipulator (intermediate) goals that the AMIGO robot Lunenburg et al (2014) performed at the 2014 RoboCup@Home competition Wisspeintner et al (2009) is presented. During the competition, these goals were issues separately by a task executive, which implies that all transitions were pre-programmed. By appropriately selecting these intermediate points collision-free trajectories were obtained, even without an explicit collision-avoidance algorithm.

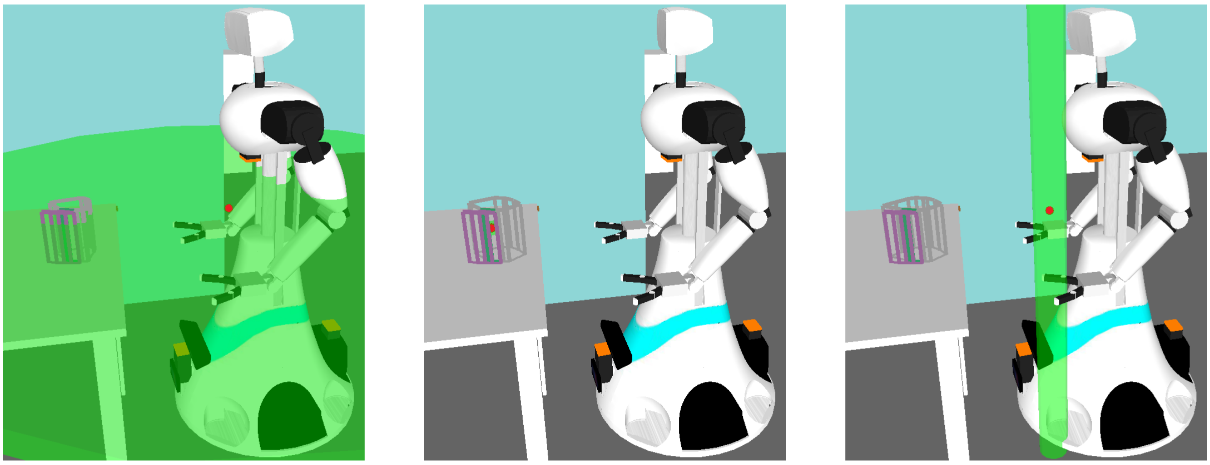
Table 6: Arm motions during RoboCup 2014

Motion	**Constrained DoFs**	**Comments**
Reset	-	Arm is at rest
Adjust height grasp	$x,\ y,\ z$	
Pre-grasp	$x,\ y,\ z,\ r_x,\ r_y,\ r_z$	Less with axissymmetric objects
Grasp	$x,\ y,\ z,\ r_x,\ r_y,\ r_z$	Less with axissymmetric objects
Lift	z	r_x and r_y are constrained in case an object needs to be carried level
Carry	$x,\ y$	r_x and r_y are constrained in case an object needs to be carried level
Adjust height place	$x,\ y$	r_x and r_y are constrained in case an object needs to be carried level
Pre-place	$x,\ y,\ z$	r_x and r_y are constrained in case an object needs to be carried level
Place	$x,\ y,\ z$	r_x and r_y are constrained in case an object needs to be carried level
Retract	$x,\ y$	r_x and r_y are constrained in case an object needs to be carried level
Handover	$x,\ y,\ z$	r_x and r_y are constrained in case an object needs to be carried level

In this chapter, we propose to formalize these goals in a topological graph. A task executer can then, e.g., ask the motion planner for a 'grasp' motion, resulting in three subsequent attractors (adjusting the height, pre-grasping and grasping). At this point, the graph only contains nodes for pick-and-place actions but more nodes can be added as the robot is supposed to perform more complicated manipulation tasks. A rule of thumb when adding nodes is that a collision-free trajectory is often possible if the robot always retracts its end-effector to close to its body before proceeding to the next task. The definition of the attractors, i.e., grounding the nodes, is discussed in the next section.

5.2 Attractors

All nodes in the graph in Figure 3 have to be grounded to specific end-effector attractors. Each attractor can constrain up to six DoFs and possesses a number of properties:



(a) Adjust height grasp: since mainly the height of the end-effector is important, the constraint volume is a cylinder with a large diameter and small height.

(b) Grasp: the attractor and constraint volume are inside the wire frame.

(c) Retracting: since the exact height is irrelevant, the constraint volume is a narrow but tall cylinder.

Figure 3: Visualization of three attractors. The red spheres indicate the exact location of the attractor. The green volume indicates the constraint, i.e., once the end-effector is inside the volume, it can proceed to the next attractor.

- Target pose: As mentioned in De Schutter et al (2007), at least two object frames and two feature frames are required to define a constraint. The target pose represents the first object frame and the first feature frame. Up to three position and three orientation DoFs can be specified.
- Target offset: The target offset represents the second object frame and feature frame.
- Stiffness: Defines the stiffness of the virtual spring between the current end-effector pose and the desired end-effector pose. If the stiffness is zero, this DoF is unconstrained.
- Goal tolerance: The position tolerance is currently defined by a box, sphere or cylinder. If the end-effector position is within the position tolerance, the position constraint is met. If a certain DoF is free, the tolerance of this DoF should be chosen such that it is always met. The orientation tolerance is defined by roll, pitch and yaw tolerance. If the end-effector orientation is within these tolerances, the orientation constraint is met. Again, if a certain DoF is free, the tolerance of this DoF should be chosen such that it is always met. In case of orientations, 2π is always sufficient. In the future, it should be possible to define these tolerances in a more flexible way that is not limited to these simple shapes. Furthermore, force tolerances would also be a valuable addition.

In Table 6 the DoFs that are constrained in case of the motions in Figure 2 are summarized. Furthermore, Figure 3 shows three examples where specific parameters for the atrractors and tolerances are defined for the AMIGO robot. Here, the attractors in Figure 3a and Figure 3c are defined with respect to the robot and the attractor in Figure 3b is inside the object to grasp.

5.3 Forward integration

By updating the graph based on object to grasp and searching the graph, a sequence of attractors with accompanying stiffness is obtained. Next, it must be validated whether these attractors result in a feasible motion. This is done by forward integration of the closed-loop whole-body controller, similar to Behnisch et al (2010).

The forward integration simply means the sequence of attractors is fed to a simulator of the system in Figure 1. As soon as the end-effector meets the specified goal tolerances, the next attractor is fed to the system. If the goal tolerance of one of the attractors is not met after $ni_{\max}$ iterations, this motion is invalidated. Since the admittance coupling is designed such that the closed-loop position controlled joints are able to track the desired trajectories, $q = q_d$ to speed up the simulation.

If the entire sequence of attractors is feasible it is fed to the real robot in exactly the same way. In case a part of the trajectory is not feasible, a possible solution is to repair this interval using a geometric motion planning method such as a probabilistic roadmap (Amato et al, 1998; Laumond and Nissoux, 2000; Kurniawati and Hsu, 2004; Yeh et al, 2012) or a rapidly exploring random tree (Hsu et al, 1997; Kuffner and LaValle, 2000). This is left as future work.

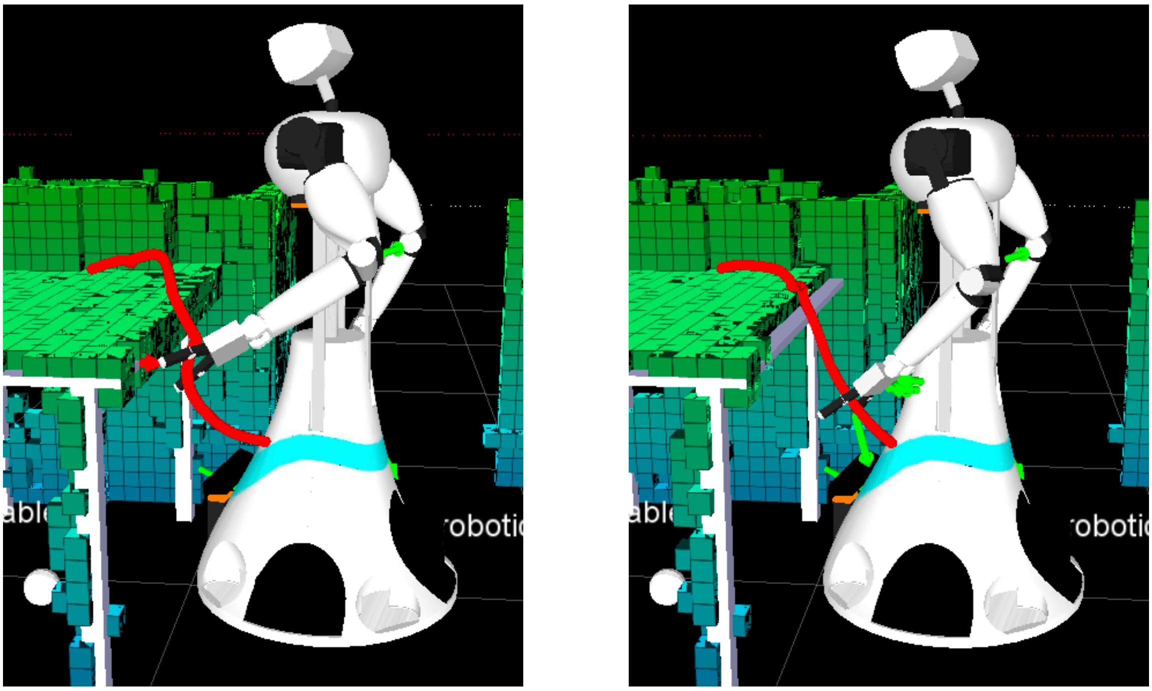
6 Simulation results

Although the implementation has not yet been finalized, a simulation has been performed to illustrate the approach. AMIGO is the domestic service robot of the Eindhoven University of Technology. Its base platform has four omniwheels and is hence fully holonomic. It is equipped with two 7-DoF Philips Experimental Robotic Arms. These have the dimensions of the arms of a large person and are placed on an upper body that can move up and down with a telescopic spine. In its lower position, AMIGO can grasp objects from the floor while in its upper position it has the size of a child and can therefore operate most features in a domestic environment. The kinematic structure of AMIGO is redundant: the two arms and torso have a $2 \times 7 + 1 = 15$ DoFs.

In this simulation, AMIGO is performing a typical grasping move. As a comparison, we have also performed this simulation with waypoints constructed in a more common probabilistic roadmap. Here, a visibility-based sampling method (Laumond and Nissoux, 2000) has been used in 6-DoF. The resulting end-effector trajectory can be seen Figure 4a, while the trajectory of the topological planner is displayed in Figure 4b. A striking difference in these trajectories is that the maximum height of the random trajectory equals 1.13 m, while the actual target is more than 20 cm lower at 0.90 m. As a result, the motion using the PRM planner looks very unnatural and not human-like. Furthermore, the PRM planner does not enforce the end-effector to approach the object to grasp from the correct direction. Finally, the topological planner will always result in a similar trajectory, while a PRM-based planner will result in a different trajectory every time a new roadmap is constructed.

7 Conclusions & future work

This chapter demonstrates the use of a topological graph for motion planning for manipulation. Although the implementation of the entire pipeline has not yet been finished, preliminary



(a) Using a visibility-based probabilistic roadmap.

(b) Using the topological planner.

Figure 4: End-effector trajectories when using a visibility-based probabilistic roadmap and using the topological planner.

simulations show promising results. A first prerequisite to see the benefits of this planner in practice is to have a decent implementation of the whole-body controller. This allows to make a proper comparison between a geometric planner and a topological planner to assess whether the decreased number of constrained DoFs indeed leads to improved performance. The graph in Figure 2 and Table 6 only contain nodes for simple pick-and-place operations. To use this approach in practice, more nodes should be added to the graph so that more tasks are covered.

As discussed in Section 5.2, the nodes are grounded using pose constraints. Extending the definition of the attractors, i.e., the grounding of the nodes, offers numerous possibilities to improve performance. One of the possible extensions is to include force constraints. Consider, for example, the task of placing an object. The ‘place’ move, can then be implemented with the attractor position at the table surface and a force constraint in vertical direction. As a result, the robot will move the object down until it ‘feels’ that it has reached the table.

8 Appendix A. How to integrate?

This appendix shows where the whole-body controller and planner are positioned within the TU/e Robotics architecture, depicted in Figure 5. The whole body controller is positioned in the 'Robot skills Implementation' block. The input of the whole body controller is the world model that describes the environment around the robot; as an output, the whole-body controller sends joint references to the low level control or the simulator. The whole-body

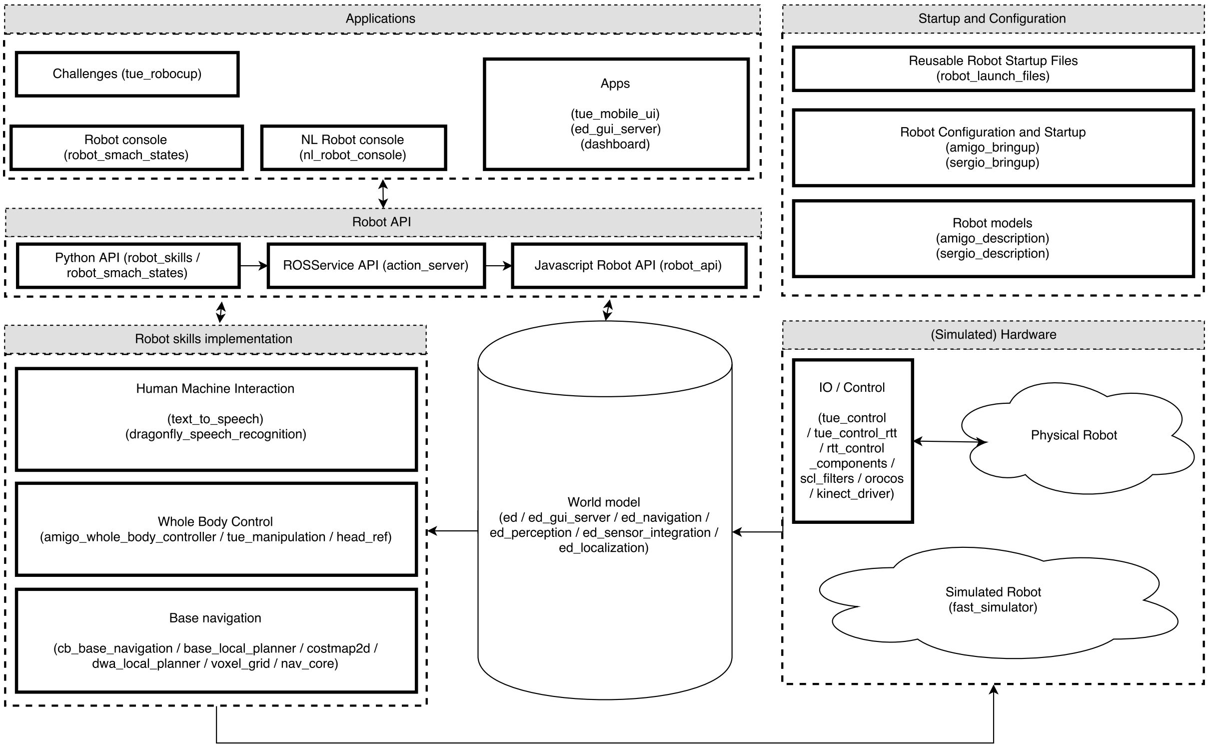


Figure 5: TU/e Robotics Architecture

controller exposes an API using a ROS actionlib to the Robot API layer so that its functionality can be used as a skill. The actionlib interface definition can be found here; the arguments basically consist of a link name, position constraints and orientation constraints.

The output of the whole body controller are references for the different joints that are taken into account by the controller.

References

Amato N, Bayazit O, Dale L, Jones C, Vallejo D (1998) OBPRM: an obstacle-based PRM for 3D workspaces. Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective pp 155–168

Behnisch M, Haschke R, Gienger M (2010) Task space motion planning using reactive control. In: 23rd IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, Taipei, pp 5934–5940

Behnisch M, Haschke R, Ritter H, Gienger M (2011) Deformable trees - exploiting local obstacle avoidance. In: 2011 IEEE-RAS International Conference on Humanoid Robots, pp 658–663

Brock O, Khatib O (2002) Elastic strips: A framework for motion generation in human environments. International Journal of Robotics Research 21(12):1031–1052

Şucan I, Chitta S (2012) Motion planning with constraints using configuration space approximations. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1904–1910

Şucan I, Kavraki L (2012) A sampling-based tree planner for systems with complex dynamics. IEEE Transactions on Robotics 28(1):116–131

De Schutter J, De Laet T, Rutgeerts J, Decré W, Smits R, Aertbeliën E, Claes K, Bruyninckx H (2007) Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. International Journal of Robotics Research 26(5):433–455

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

Gienger M, Janssen H, Goerick C (2005) Task-oriented whole body motion for humanoid robots. In: 2005 5th IEEE-RAS International Conference on Humanoid Robots, pp 238–244

Gienger M, Toussaint M, Goerick C (2008) Task maps in humanoid robot manipulation. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 2758–2764

Hsu D, Latombe JC, Motwani R (1997) Path planning in expansive configuration spaces. In: Proceedings of the 1997 IEEE International Conference on Robotics and Automation, vol 3, pp 2719–2726 vol.3

Hsu D, Kindel R, Latombe JC, Rock S (2002) Randomized kinodynamic motion planning with moving obstacles. International Journal of Robotics Research 21(3):233–255

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

Kuffner J JJ, LaValle S (2000) Rrt-connect: An efficient approach to single-query path planning. In: Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA), vol 2, pp 995–1001 vol.2

Kurniawati H, Hsu D (2004) Workspace importance sampling for probabilistic roadmap planning. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1618–1623

Laumond JP, Nissoux C (2000) Visibility-based probabilistic roadmaps for motion planning. Advanced Robotics 14(6):477–493

Lunenburg J, Coenen S, Derksen T, van den Dries S, Elfring J, van de Molengraft M (2014) Tech United Eindhoven @Home team description 2014

Nagasaka K, Kawanami Y, Shimizu S, Kito T, Tsuboi T, Miyamoto A, Fukushima T, Shimomura H (2010) Whole-body cooperative force control for a two-armed and two-wheeled mobile robot using generalized inverse dynamics and idealized joint units. In: 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, pp 3377–3383

Nakanishi J, Mistry M, Peters J, Schaal S (2007) Towards compliant humanoids-an experimental assessment of suitable task space position/orientation controllers. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 2520–2527

Ojdanic D, Graser A (2007) A fast motion planning for a 7dof rehabilitation robot. In: 2007 IEEE 10th International Conference on Rehabilitation Robotics, pp 171–178

Sentis L, Khatib O (2005) Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. International Journal of Humanoid Robotics 02(04):505–518

Stückler J, Holz D, Behnke S (2012) Robocup@home: Demonstrating everyday manipulation skills in robocup@home. IEEE Robotics & Automation Magazine 19(2):34–42

Toussaint M, Gienger M, Goerick C (2007) Optimization of sequential attractor-based movement for compact behaviour generation. In: 2007 7th IEEE-RAS International Conference on Humanoid Robots, pp 122–129

Wisspeintner T, Van Der Zant T, Iocchi L, Schiffer S (2009) RoboCup@Home scientific competition and benchmarking for domestic service robots. Interaction Studies 10(3):392–426

Yang Y, Brock O (2006) Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation in dynamic environments. In: Proceedings of Robotics: Science and Systems

Yang Y, Brock O (2010) Elastic roadmaps-motion generation for autonomous mobile manipulation. Autonomous Robots 28(1):113–130

Yeh HY, Thomas S, Amato NM (2012) UOPRM: A uniformly distributed obstacle-based PRM. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems

Zavlangas P, Tzafestas S (2002) Integration of topological and metric maps for indoor mobile robot path planning and navigation. In: Proceedings of the Second Hellenic Conference on AI: Methods and Applications of Artificial Intelligence, Springer-Verlag, pp 121–130