2008 IEEE/RSJ International Conference on Intelligent Robots and Systems
Acropolis Convention Center
Nice, France, Sept, 22-26, 2008

Positioning Mobile Manipulators to Perform Constrained Linear Trajectories

Franziska Zacharias, Christoph Borst, Michael Beetz and Gerd Hirzinger

***Abstract*— For mobile manipulators envisioned in home environments a kitchen scenario provides a challenging testbed for numerous skills. Diverse manipulation actions are required, e.g. simple pick and place for moving objects and constrained motions for opening doors and drawers. The robot kinematics and link limits however are restrictive. Therefore especially a constrained trajectory will not be executable from arbitrary placements of the mobile manipulator.**

A two stage approach is presented to position a mobile manipulator to execute constrained linear trajectories as needed for opening drawers. In a first stage, a representation of a robot arm's reachable workspace is computed. Pattern recognition techniques are used to find regions in the workspace representation where these trajectories are possible. A set of trajectories results. In the second stage mobile manipulator placements are computed and the corresponding trajectories are checked for collisions. Compared to a brute force search through the workspace, the success rate of finding a mobile manipulator placement can be increased from 2% to 70%.

I. INTRODUCTION

As more advanced robot arms and mobile manipulators are developed and the first humanoid robots are commercially available more emphasis is placed on high-level tasks, task specification and task execution. For service robots in home environments e.g. the kitchen is a challenging test bed [1], [2] requiring a multitude of skills, e.g. environment modeling, manipulation or planning. The robot is envisioned as the human's assistant and is expected to perform tasks such as setting the table, cleaning the kitchen or filling the dishwasher. Hereby different kinds of manipulation actions have to be executed. Simple pick and place actions are used to move objects. Constrained motion of the end-effector and its attached tool center point (TCP) is needed for opening cupboards and drawers. The most difficult manipulation action is constrained dual arm manipulation as needed for transporting a pot of water.

The kinematics and link limits restrict the manipulation capabilities of a mobile manipulator. Therefore, we expect that especially a constrained trajectory will only be executable from well chosen placements of the mobile manipulator.

It is an advantage if a mobile manipulator has a model of the reachable workspace of its arms and knows which regions are reachable from which directions. Given information about the location of an object, its reachability can be checked and a placement of the mobile manipulator can be computed from which the object is reachable. Especially constrained motion is often needed in the kitchen. As a first step, we examine the subset of constrained linear trajectories and position the mobile manipulator to execute them. This type of trajectories is needed for e.g. opening drawers, extracting objects from a cluttered scene, but also in assembly tasks. Due to inaccuracies of the robot base movement and resulting forces on the robot arm, trajectories are chosen that allow for a fixed robot base during trajectory execution.

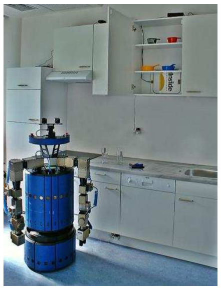


Fig. 1. A mobile manipulator in a kitchen.

Zacharias et al. [3] have presented a model of the reachable workspace for the DLR light weight robot arm. This approach is used for the demonstrator of the IAS Group, a mobile platform with two 6 degree of freedom (DOF) PowerCube arms [4] (fig.1). The reachability representation of the right arm of the manipulator will be analyzed using pattern recognition techniques. Regions will be extracted where a given constrained trajectory is executable. The extracted information will then be used to position the mobile manipulator so that it is able to execute the task.

II. RELATED WORK

Most approaches to constrained trajectory planning for mobile manipulators to date combine the positioning of the robot with the search for feasible trajectories for the robot arm in the configuration space (C-space). Optimization and path planning techniques are used.

Optimization techniques are applied to the whole kinematic chain. When using multi-criteria optimization for positioning a mobile manipulator to reach a point, choosing criteria weights, competing criteria and the resulting local minima pose a great challenge [5]. Multi-criteria optimization can get stuck in local minima due to unfavorable start positions. These local minima may not constitute a valid solution for the task, thus causing the method to fail.

Existing path planning techniques, e.g. the probabilistic

F. Zacharias, Ch. Borst and G. Hirzinger are affiliated with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany, franziska.zacharias@dlr.de

M. Beetz is affliated with the IAS Group, Technische Universität München, München, Germany, beetz@cs.tum.de

978-1-4244-2058-2/08/$25.00 ©2008 IEEE.

roadmap approach (PRM) [6] or the rapidly-exploring random tree (RRT) approach [7] can be adapted to search for constrained trajectories. Some examples will be outlined.

Given a start configuration, a goal frame and task constraints, Yao et al. [8] search a collision-free path for a fixed base robot arm by alternating task space and configuration space exploration. Valid path segments in cartesian space are tracked in C-space using trajectory tracking methods from control theory, e.g. Jacobian-based methods. If tracking is not possible due to the kinematics and link limits, the path segment is discarded and another segment is tested.

Oriolo et al. extend their previous approach to constrained trajectory planning [9] to include mobile manipulators [10]. They labeled the n DOF of the mobile manipulator as redundant ($n-6$ DOF) or non-redundant (6 DOF). Random samples are drawn from the C-space of the redundant DOF, i.e. from the 3 DOF C-space of the mobile base. The RRT path planning approach is adapted and combined with inverse kinematics algorithms. The resulting path is a combination of motion of the redundant and the non-redundant joints, i.e. the mobile base and the robot arm. Since no knowledge is used about e.g. which regions are reachable from what direction, many platform positions are examined from which executing the trajectory is impossible.

Yang et al. [11] adapt PRMs to plan task-consistent collision-free motion for mobile manipulators with a holonomic base in dynamic environments. Task-level controllers are used as the local navigation function to connect nodes in the map. This approach is able to plan the mobile manipulator motion for e.g. opening a drawer but needs the base mobility.

Stilman [12] plans task-constrained, collision-free motion for a mobile manipulator. He proposes a task formalism to indicate constrained motion directions in cartesian space. The task frame Jacobian and its pseudo-inverse are used in the extension of an RRT path planner to find joint space displacements that resolve task space errors and then use these to generate samples for a RRT path planner in C-space. A Cartesian start and end-frame of the trajectory is known as well as an initial grasping configuration of the robot arm.

To summarize, two directions are pursued. Either a generalized inverse kinematics is applied to the whole kinematic chain. It is assumed that the task can be completed if the redundancy is adequately exploited. Or a start configuration of the mobile manipulator is given from which the task is feasible and path planning techniques are adapted. In both cases the redundancy provided by the mobile base is used.

With respect to choosing a fixed mobile base position to execute constrained trajectories, few approaches are know to the authors. Neo et al. present an interactive operation system for commanding an HRP-2 humanoid robot to open a fridge. To choose a standing position for HRP-2 from which it is able to execute the task, a heuristic approach is used.

Yoshida et al. [13] reach for target end-effector positions with an HRP-2 humanoid robot. A generalized inverse kinematics is combined with an heuristic approach to support polygon reshaping and stepping.

III. PROBLEM ANALYSIS

Considering the task of opening a drawer using a mobile manipulator the following issues should be observed. The movement accuracy of a mobile base is going to be around ± 2 cm or lower. When opening a drawer the robot ideally maintains a fixed, rigid connection with the drawer handle. Moving the base in the process of opening, results in considerable forces on the arm due to the movement inaccuracies of the base. These have to be compensated. Neo et al. [14] reported considerable forces due to movement inaccuracies of the arm during opening a fridge with an HRP-2 humanoid robot. The orientation of the TCP and the standing position of the humanoid were kept stationary.

Constrained motions involved in opening a door using all available DOF has been demonstrated using the DLR Robutler, a light weight robot arm equipped with a mobile base [15]. Impedance control was used to compensate forces along the constrained trajectory e.g. due to inaccuracies. However, force-torque sensors are needed in each link to apply impedance control. Most industrial robots and many robot arms used in research are not equipped with force torque sensors.

Another option to compensate the afore mentioned forces would be to attach a passive compliance element at the end-effector. This however makes grasping and object handling more difficult and requires closed loop control (admittance control) of the arm to correct the position errors. Therefore, we propose to check first whether the constrained trajectory can be executed while keeping the mobile base stationary.

Linear constrained trajectories are often encountered in manipulation tasks. They are needed to extract an object from a set of objects. Or they may be part of assembly tasks.

Opening a drawer involves a constrained trajectory where the orientation of the gripper and its attached TCP frame remains fixed and motion is only executed in one direction (Fig. 2). It will serve as the example of choice to illustrate the approach presented in this paper.

To open a drawer the gripper is assumed to have a rigid connection with the drawer handle and moves in a plane perpendicular to the handle along a linear trajectory (Fig. 2). The gripper can be oriented in numerous ways w.r.t. the handle of the drawer. The orientation of the gripper differs from the direction in which the drawer can be extracted. Executing the constrained trajectory will not be possible from every starting point in the robot arm workspace due to the robot kinematics and joint limits. The number of valid trajectories distributed across the workspace is nevertheless infinite. If a representation of these regions were available, a mobile base could be positioned so that the arm can operate in one of these regions and is able to perform the required linear motion.

A two-stage approach is presented that determines regions in the robot arm workspace where linear constrained trajectories are possible, as needed for the drawer example. It uses a model of a robot arm's reachable workspace which is briefly introduced in section IV. In the first stage, the model is

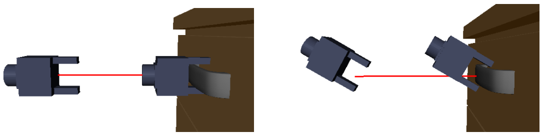
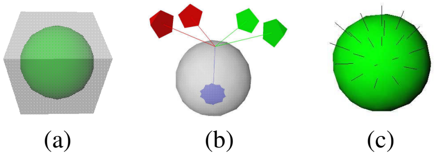


Fig. 2. Different gripper orientations w.r.t. the handle of the drawer. Trajectories lie in a plane perpendicular to the drawer handle.



(a) (b) (c)

Fig. 3. Shows a sphere inscribed into the cube (a), exemplary frames for a point on the sphere (b), valid inverse kinematics solutions on a sphere (c).

analyzed. Regions are extracted where constrained linear cartesian space trajectories are possible. These are used to infer placements for the mobile manipulator (sec. V). In the second stage, the placements are checked for collision-free execution of the trajectory. Results are reported in section VI.

IV. REACHABILITY SPHERE MAPS

Humans use internal models to solve reaching tasks [16], to infer which regions are reachable from what directions. Possessing a similar abstraction of a robot arm's capabilities in its workspace is important for task planning. Zacharias et al. [3] introduced a model that can be used for this purpose. We will briefly summarize its main points.

The theoretically possible robot arm workspace is enveloped by a cube and subdivided into equally-sized smaller cubes Into each cube a sphere is inscribed and on this sphere n points are uniformly distributed (Fig. 3 (a), (c)). Frames are generated for each point and serve as the target TCP for the inverse kinematics of the robot arm (Fig. 3 (b)). The point on the sphere determines the z-axis of the TCP frame. The orientation of the x- and y- axis is subsampled. The result of the inverse kinematics is registered in the data structure that is visualized by the sphere. Reachable points on the sphere are visualized as black lines. The spheres visualize the reachability for a region and are therefore called reachability spheres. The reachability sphere map is the aggregation of all spheres. It can be used to visualize and inspect the reachability across the workspace and to approximate the shape of the robot arm workspace. For visualization, spheres are colored w.r.t. the reachability index D [3] which is the percentage of the points on the sphere that are reachable. For the arm used in this work, the reachability index is relatively low throughout the workspace compared to the 7 DOF DLR light weight robot arm. A maximum value of D=22 is reached. This means that at maximum 22 percent of the points on the sphere are reachable. This underlines the need to search the reachable workspace for regions where the constrained trajectories are possible. Fig. 4 shows examples of reachability spheres for the used robot arm.

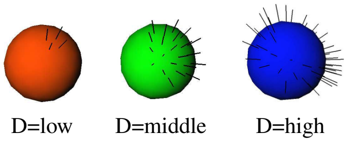


Fig. 4. Spheres with different reachability indices D.

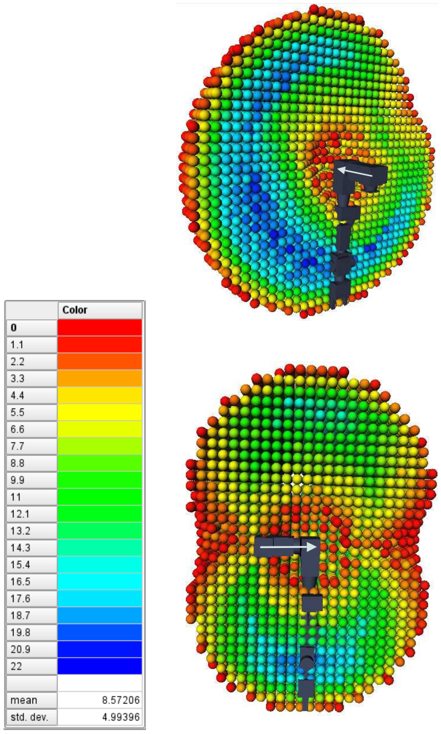


Fig. 5. A model of a PowerCube arm and the reachability index D across the workspace.

The distribution of the reachability index throughout the workspace is shown in Fig. 5. The reachability sphere map is cut along two different planes. For reference the direction of the z-axis of the first link is shown as a gray arrow. In the blue region the reachability index is highest. The effect of the asymmetric limit of the first link (tab. I) can be seen in the asymmetry of the blue region.

V. POSITIONING A MOBILE MANIPULATOR

A. *Reachability across the task space*

The problem analysis in section III showed that for the linear cartesian movement involved in opening a drawer all movement of the TCP happens in a plane which will be named *task plane* from now on. The reachability sphere map can be intersected with this plane. This results in a description of the reachability just for this task plane (Fig. 6). The spheres are colored w.r.t. their reachability index. It can be seen that the reachability varies across the task plane.

The points on the sphere are uniformly distributed using a deterministic algorithm [17]. Thus for each sphere in the workspace the points are placed exactly the same. Direction

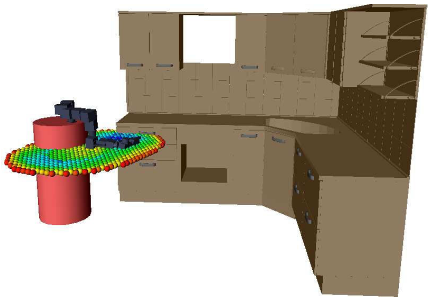


Fig. 6. Intersection of the reachability sphere map with the task plane.

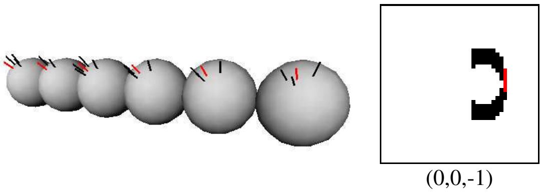


Fig. 7. Neighboring spheres with the line corresponding to the vector (0,0,-1) colored red and their position in the binary map.

vectors belonging to the same point on a sphere at two different locations in the workspace have the same coordinates. To illustrate this, Fig. 7 shows several neighboring spheres. The line corresponding to vector (0,0,-1) is colored red.

The reachability of a point p on a sphere, i.e. the direction it represents can be analyzed across the task plane. For each point p, this results in a two dimensional binary map where a pixel corresponds to a sphere on the planar cut through the reachability sphere map (compare fig. 7). A pixel is colored black if the point is marked valid for the respective sphere, i.e. it is reachable. Otherwise it is white. Fig. 8 shows two of these binary maps and lists the coordinates of the corresponding sphere points. The binary maps are an alternative visualization of the reachability data represented by the cut through the reachability sphere map (Fig. 6).

B. Extracting feasible regions using pattern recognition

In this section we extract regions from the reachability representation where the required constrained trajectory is possible. The reachability sphere map is anchored at the robot arm base and moves with the movable base. Our goal is to infer a placement of the mobile base using this representation. Fig. 9 (left) shows two placements of the mobile manipulator w.r.t. the drawer. For each placement of the mobile manipulator, the trajectory is positioned and oriented differently in the robot arm workspace and also in the binary map. This is indicated by the bold white line in the left hand images. In Fig. 9 (right) the location of the trajectory in the binary map is shown by the red pixels.

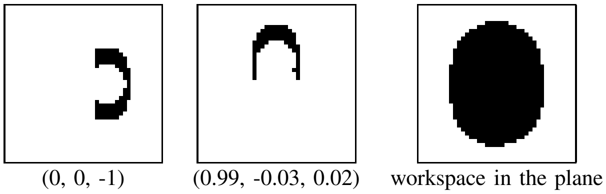


Fig. 8. (Left, middle) Binary maps for two different points on the sphere. (right) The union of the binary maps of all points. It shows the extension of the task plane. A black image border was added for better comparability.

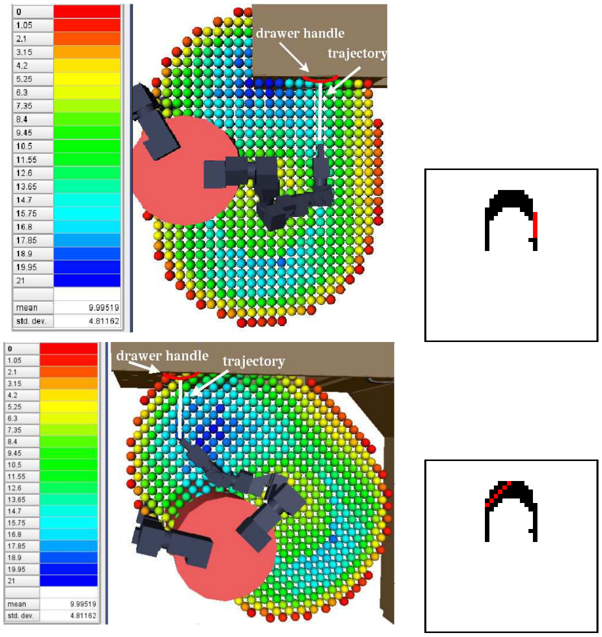


Fig. 9. (left) Trajectory in the robot arm workspace, (right) location of the trajectory in the binary map.

The aim of this work is not to obtain a continuous representation of the regions in the robot arm workspace that contain task-consistent trajectories. Rather we aim at a discrete representation of valid solutions. The discretized snapshots that the binary maps represent, can be used for this purpose. A suitable solution can be found in a connected region in the binary map with a dimension corresponding to the length of the linear trajectory in the task plane.

Morphological image processing techniques can be used to search for these regions. The morphological operation *erosion* [18] moves a pattern across the image and marks where it is a subset of the data w.r.t. the anchor point of the pattern. Positions where the pattern is found are directly obvious from the resulting image. In Fig. 10, a binary map is eroded with a pattern of 6 pixels in a vertical line. The resulting binary map (Fig. 10 (c)) contains only the possible starting points of the pattern w.r.t. the original binary map. These starting points are marked by black pixels.

However, if the pattern occurs in the image in a different orientation, it will not be found. The standard solution to this is to rotate the pattern using a fixed stepsize. Fig. 11 illustrates this procedure by rotating and subsampling a trajectory template. Each rotated trajectory is mapped onto a grid where a grid cell covers the same space as a reachability

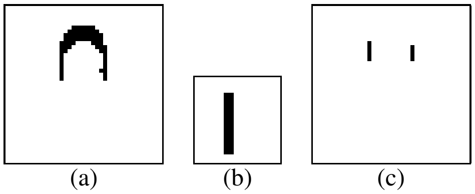


Fig. 10. (a) binary map, (b) search pattern (enlarged), (c) binary map after erosion with the pattern

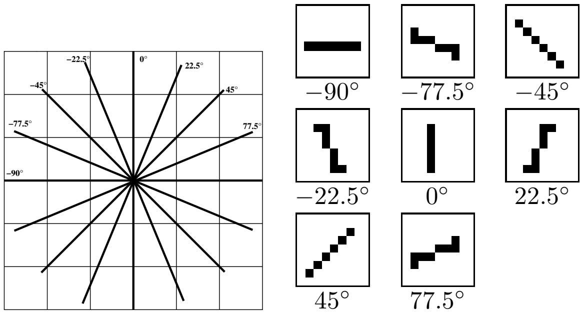


Fig. 11. Patterns resulting from the trajectory template being rotated from -90 to +90 degrees with an angle stepsize of 22.5 degrees.

i	$a(i-1)$	$\alpha(i-1)$	$d(i)$	$\theta(i)$	limits
1	0	0	0	0	[-60,170]
2	0	-90	0.26 m	-90	[-170,170]
3	0	-90	0	180	[-90,90]
4	0	-90	0.31 m	180	[-170,170]
5	0	-90	0	180	[-120,120]
6	0	-90	0	180	[-170,170]
TCP	0	0	0.265 m	0	

TABLE I

DH PARAMETERS AND LINK LIMITS FOR THE 6 DOF POWERCUBE ARM.

sphere. The pattern is saved as a binary image.

C. Validation of solutions

The pattern recognition process provides a number of starting points for trajectories. These need to be checked for actual reachability since the reachability sphere map implies an interpolation assumption. It is assumed that at each point of a subcube the same orientations are reachable as on the sphere located at its center, i.e. that the reachability structure does not change for small increments in space. This assumption will hold in general. However at the inner and outer border of the reachable workspace, this assumption may be violated.

D. Computing the robot base position

The current robot position in the world coordinate system is assumed to be known. The new mobile manipulator position can be computed using eq. 1. The z-axis of the world system is assumed to point upwards. The transformation from the old to the new robot base position involves only a rotation around the z-axis of the world system and a translation in the xy plane.

$$T_{base} \cdot T_{base2TCP} = T_{goal} \qquad (1)$$

Eroding a binary map with a pattern provides a number of starting points for the constrained trajectory in the robot coordinate system. One of these is selected. The orientation of the TCP frame attached to the gripper is obtained as follows. The direction vector connected with the binary map provides the z-axis. The orientation of the x-axis corresponds to the cross-product of the task plane normal and the z-axis. The y-axis follows naturally. Thus the transformation $T_{base2TCP}$ from the robot coordinate system to the TCP coordinate system can be computed. A grasping point on the drawer handle is given in world coordinates. The pattern provides the orientation of the trajectory in the robot base coordinate system. Using this information and the orientation of the TCP frame in robot coordinates, the TCP frame T_{goal} in world coordinates can be computed. Eq. 1 can now be solved for the new base frame T_{Base}.

VI. ANALYSIS OF THE RESULTS

A PowerCube arm is attached as the right arm to a mobile platform as shown in Fig. 1. The DH-parameters according to [19] for the arm kinematics and the link limits are given in Tab. I. Analytic inverse kinematic solutions were computed as detailed in [19]. We assume that a grasp point on the drawer handle is given in the world coordinate system. This point is used to determine the task plane.

A. Brute force search vs. using the reachability model

Without a representation of reachability, a brute force approach could be thought of. To find linear constrained trajectories for opening drawers, a position on the task plane could be randomly chosen. From this starting position, a discrete number of directions can be searched for reachable trajectories of the correct length.
In this section the likelihood to find trajectories using the brute force search will be examined. It will be compared to the likelihood achieved by making use of the reachability model. Results are reported for linear trajectories of 0.3 m and 0.4 m length.

1) Results using brute force search: The TCP start frame of the trajectory and the trajectory direction are randomly sampled. The region of the task plane shown in fig. 8 (right) is taken as an approximation of the reachable space. A position is sampled from this region according to a uniform distribution. The frame orientation is obtained by randomly choosing one of 200 points distributed uniformly on a sphere. The corresponding direction vector is used as the z-axis of the TCP start frame. An angle is randomly sampled to determine the orientation of the x- and y-axis of the TCP start frame. One of the patterns from fig. 11 is randomly chosen to determine the trajectory direction in the robot arm workspace. The generated frame only contributes to the computed success rate if the last TCP frame along the trajectory also lies within the reachable region of the task plane (black region in fig. 8 (right)). The trajectory is checked

	brute force search			reachability model		
length t	samples	valid	%	predicted	valid	%
0.3 m	$1{\cdot}10^7$	195875	1.96	13869	10160	73
0.4 m	$1{\cdot}10^7$	96169	0.96	7320	4337	59.2

TABLE II

RESULTS OF THE BRUTE FORCE SEARCH ARE COMPARED WITH THE REACHABILITY MODEL-BASED APPROACH.

for reachability using inverse kinematics. To obtain a relative success measure, the number of successes is weighted with the total number of samples. For linear trajectories t of 0.3 m length in the given task plane a success rate of only 2% can be expected (tab. II).

2) Results using the reachability model: A reachability sphere map was computed for the robot arm. The sphere diameter was empirically set to 0.05 m. The 200 points distributed uniformly on the sphere result in 200 binary maps. Eigth patterns are computed for trajectories of length 0.3 and 0.4 m as illustrated in Fig. 11. Using the approach described in the last sections, each binary map is eroded with each of the eight patterns. The method finds 13869 possible trajectories. Because of the interpolation assumption mentioned in section V-C, each of these cartesian trajectories is subsampled and checked for reachability using an inverse kinematics. Failures and successes are counted for each workspace region, i.e. pixel in the picture. From these two counts the relative error is computed (eq. 2)

$$\text{interpolation error} = \frac{\#\text{failure}}{\#\text{failure} + \#\text{success}} \qquad (2)$$

This relative error is depicted in Fig. 12 (left). Regions with low error values are colored blue while those with high errors are red. At the inner and outer border of the workspace the interpolation assumption is often violated. Cases accumulate where a position is reachable on the sphere, but not reachable in the immediate vicinity. The last three columns of tab. II show that 73% of the predicted solutions are actually reachable. Compared to the brute force search success rate of 2% this is a significant performance gain.

Fig. 12 (right) shows how often regions of the workspace were involved in a kinematically reachable solution. As before, each pixel corresponds to a sphere of the task plane. Regions of the task plane that are reachable but are not used in the solutions remain black. Due to the link limits of the first link, most solutions are found in the top part of the task plane (blue area), i.e. in front of the manipulator.

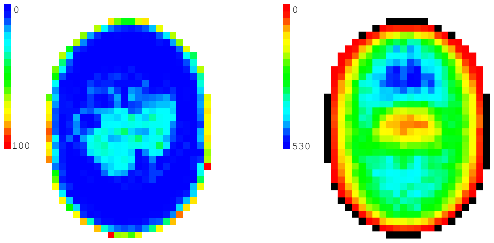


Fig. 12. (Left) Distribution of the interpolation error across the task plane, (Right) Distribution of solutions.

		nr. of solutions			
length t	predicted	real	reconf.	colliding	total
0.3 m	13869	10160	1356	8920	1059
0.4 m	7320	4337	973	3843	400

TABLE III

THE SET OF SOLUTIONS IS ANALYZED W.R.T. DIFFERENT CRITERIA.

B. Collision-free and consistent trajectories

A new mobile manipulator position is computed for each valid solution resulting from the presented approach. Each trajectory is checked for consistency and collisions. In general the assumption will hold that if two adjacent positions are reachable, a consistent trajectory will exist between them. However there will be regions in the workspace e.g. in the vicinity of singularities where reconfigurations of the robot arm will occur. Column *reconf.* of tab. III shows the number of reachable solutions where this is the case. The trajectory is checked for collisions with the mobile manipulator and with the kitchen environment. Results can be seen in the column of tab. III titled *colliding*. The last column of tab. III shows the total number of consistent and collision-free trajectories w.r.t. a given grasp point. A valid mobile manipulator position can be chosen out of 1059 collision free solutions in the case of trajectories of 0.3 m length or out of 400 solutions for trajectories of 0.4 m length. The steps of the presented approach are summarized in fig. 13 and illustrated in the accompanying video. Fig. 14 shows four valid robot placements w.r.t. a grasping point on a drawer handle.

Computation times were determined for the presented approach. However, no optimization was applied. Computing the 200 binary maps took 0.1s. The pattern recognition step took 0.3s. Checking *all* predicted solutions with inverse kinematics needed 6s. The consistency and collision check was performed for *all* reachable trajectories in 17s. Despite these times, the approach can be used online. Depending on preference criteria only a small subset of the solutions needs

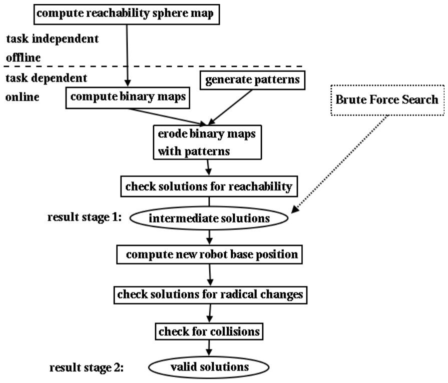


Fig. 13. Sequence of processing steps.

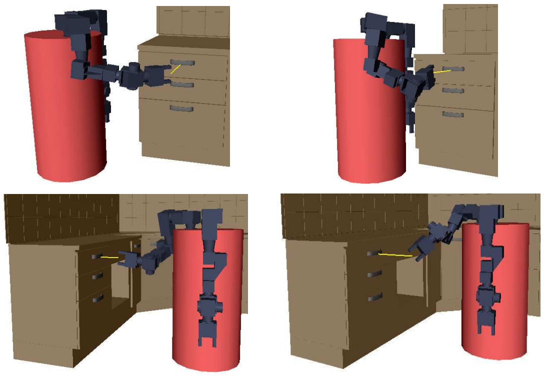


Fig. 14. Valid mobile manipulator placement examples.

to be examined till a valid placement is found.

VII. CONCLUSION AND OUTLOOK

A two-stage approach was presented for positioning a mobile manipulator to execute linear constrained trajectories. The example of opening drawers was used for illustration. It requires a linear constrained trajectory with fixed TCP orientation to be executed in a plane, the so-called task plane. In the first stage, a model of the reachable robot arm workspace is combined with pattern recognition techniques to extract regions from the task plane where the constrained trajectories are executable. In the second stage, valid positions are computed for the mobile manipulator. The trajectories are checked for consistency and collisions. From the remaining solutions, one can be used to position the mobile manipulator. If e.g. minimum torques or stability issues are of importance, the solutions can be ranked accordingly.

Since the solutions were extracted from a discretization of the robot arm workspace, it is possible that for a specific discrete solution there is a solution in its immediate vicinity that is judged even better w.r.t. some optimality criterion. Therefore the set of discrete solutions can be used as starting solutions for a subsequent optimization step. The difference to previous optimization approaches is that we already start with a valid solution that is not randomly generated but determined according to a deterministic method.

The arms of the mobile manipulator used in this paper were empirically attached to its mobile base observing criteria such as fitting through doors and reaching a weight distribution that forestalls falling over. For a given type of constrained linear trajectories and a given task plane, the proposed method computes regions in the robot arm workspace where these trajectories are possible without collisions. For different attachments of the arm to the mobile base, this number of regions varies. This can serve as an additional criterion to determine an optimum attachment of the arms to the manipulator. Furthermore, it can be determined how well a given mobile manipulator is suited for specific tasks and a specific environment. In the kitchen scenario it can be checked which drawers can or cannot be opened or in which regions objects may be placed and still be reachable. If the task is not solvable with a fixed robot base, the presented approach could be extended. However the base motion during trajectory execution should be restricted to limit the search space. Furthermore, closed-loop control techniques need to be applied (sec. III). In future work, it needs to be examined how well the method generalizes to constrained non-linear trajectories, e.g. for opening doors.

VIII. ACKNOWLEDGMENTS

The research has been partially funded by the EC Seventh Framework Programme (FP7) under grant agreement no. 216239 as part of the IP DEXMART and by the German Research Foundation (DFG) within the cluster of excellence CoTeSys (Cognition for Technical Systems).

REFERENCES

[1] T. Asfour, K. Regenstein, P. Azad, J. Schröder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "Armar-III: An integrated humanoid platfrom for sensory-motor control," in *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 169–175.

[2] M. Beetz, J. Bandouch, A. Kirsch, A. Maldonado, A. Müller, and R. B. Rusu, "The assistive kitchen - a demonstration scenario for cognitive technical systems," in *Proc. COE Workshop on Human Adaptive Mechatronics (HAM)*, 2007.

[3] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007, pp. 3229–3236.

[4] "Amtec robotics: Reconfigurable light weight robots," http://www.amtec-robotics.com.

[5] F. G. Pin and J.-C. Culioli, "Optimal positioning of combined mobile platform-manipulator systems for material handling tasks," *J. Intelligent and Robotic Systems*, vol. 6, no. 2-3, pp. 165–182, 1992.

[6] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 566–580, Aug. 1996.

[7] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Oct. 1998, TR 98-11, Computer Science Dept., Iowa State University.

[8] Z. Yao and K. Gupta, "Path planning with general end-effector constraints: using task space to guide configuration space search," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005, pp. 1875 – 1880.

[9] G. Oriolo, M. Ottavi, and M. Vendittelli, "Probabilistic motion planning for redundant robots along given end-effector paths," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2002, pp. 1657–1662.

[10] G. Oriolo and C. Mongillo, "Motion planning for mobile manipulators along given end-effector paths," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, Barcelona, Spain, Apr. 2005, pp. 2154 – 2160.

[11] Y. Yang and O. Brock, "Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation," in *Proc. Robotics: Science and Systems (RSS)*, 2006.

[12] M. Stilman, "Task constrained motion planning in robot joint space," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2007, pp. 3074–3081.

[13] E. Yoshida, O. Kanoun, and C. E. J.-P. Laumond, "Task-driven support polygon reshaping for humanoids," in *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 208–213.

[14] E. Neo, T. Sakaguchi, K. Yokoi, Y. Kawai, and K. Maruyama, "A behavior level operation system for humanoid robots," in *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 327 – 332.

[15] C. Ott, B. Baeuml, C. Borst, and G. Hirzinger, "Autonomous opening of a door with a mobile manipulator: A case study," in *Proc. 6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, 2007.

[16] M. Kawato, "Internal models for motor control and trajectory planning," *Curr. Opin. Neurobiol.*, vol. 9, no. 6, pp. 718–727, 1999.

[17] E. Saff and A. Kuijlaars, "Distributing many points on the sphere," *Mathematical Intelligencer*, vol. 19, no. 1, pp. 5–11, 1997.

[18] Open computer vision library (OpenCV). [Online]. Available: http://sourceforge.net/projects/opencvlibrary/

[19] J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.