

Capturing Robot Workspace Structure: Representing Robot Capabilities

Franziska Zacharias, Christoph Borst and Gerd Hirzinger

Abstract—Humans have at some point learned an abstraction of the capabilities of their arms. By just looking at the scene they can decide which places or objects they can easily reach and which are difficult to approach. Possessing a similar abstraction of a robot arm's capabilities in its workspace is important for grasp planners, path planners and task planners. In this paper, we show that robot arm capabilities manifest themselves as directional structures specific to workspace regions. We introduce a representation scheme that enables to visualize and inspect the directional structures. The directional structures are then captured in the form of a map, which we name the *capability map*.

Using this capability map, a manipulator is able to deduce places that are easy to reach. Furthermore, a manipulator can either transport an object to a place where versatile manipulation is possible or a mobile manipulator or humanoid torso can position itself to enable optimal manipulation of an object.

I. INTRODUCTION

In infancy, humans learn to control their limbs. After this learning process, they can reach for and grasp objects without consciously thinking. By just looking at the scene they know which places or objects they can easily reach and which are difficult to approach. The survey by Kawato [1] already verified the existence of internal models for human motor control. But humans also seem to have an internal map/model of their arm's capabilities and are able to use this representation to accomplish various tasks. If the arm was represented as a redundant serial link manipulator, these models could be seen to project the configuration space of the serial chain into its cartesian workspace forming a kind of capability map. If we had a similar map of the capabilities for the redundant arms of our humanoid robot Justin [2] (fig. 1), we would be able to visualize directional preferences and existing structure in the robot arm workspace imposed by the robot design. The information can be used to direct grasp planners to search for valid grasps in regions where the hand tool center point (TCP) can easily be placed by the robot arm.

In [3] we already emphasized the importance of being able to easily find robot target configurations that solve the subtask and facilitate the subsequent process of finding paths in a task planning process. The simple approach presented, worked on a planar projection of the scene. In this paper we derive a compact representation of kinematic reachability and directional structure information for the whole Cartesian



Fig. 1. The DLR humanoid two-arm system JUSTIN.

workspace of a robot arm. This representation, named capability map, is a good basis for finding easy to reach robot arm target configurations in 3D.

To arrive at a map of robot capabilities, we will first determine if existing criteria applied in the robot design process can be used (section II). The applicability of the popular manipulability concept proposed by Yoshikawa [4] will be discussed. Then, we will derive requirements to be met by a map of robot capabilities. In section III, we develop a representation that fulfills the mentioned criteria. We will evaluate the derived representation in section IV.

II. CRITERIA USED IN ROBOT DESIGN OPTIMIZATION

In the design stage, a robot manipulator is optimized with respect to kinematic and dynamic criteria. As we intend to use the results of the presented work to support static planning methods, we only focus on the kinematic aspects here. The kinematic design can furthermore be divided into task-oriented robot design (e.g.[5]) and the design of robots to accommodate a large variety of tasks.

In general, the robot kinematics can be optimized to maximize its workspace or to maximize various dexterity indices with respect to specific positions or with respect to the entire workspace. Park et al. [6] introduce general performance criteria for workspace volume and dexterity using differential geometry. Global indices are obtained through integration of local criteria. Sturges et al. [7] define a dexterity measure that relates the difficulty of an assembly task to the capabilities of a planar robot arm. Some directional information is included in this task-dependent difficulty measure, especially concerning the accuracy of TCP movements at specific positions. However, these indices are hard to extend to redundant spatial manipulators for service tasks.

All authors are affiliated with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Germany, franziska.zacharias@dlr.de

A popular means used in robot design is the analysis of the Jacobian matrix of a manipulator. Several indices to describe the dexterity of the manipulator have been derived thereof. Klein et al. [8] examined the relationship of the determinant, the condition number and the smallest singular value. With the goal to obtain a global isotropy design parameter, Stocco et al. [9] optimized the ratio of the maximum and the minimum singular value of the Jacobian in the entire workspace to obtain a global version of the condition number. We are interested in capturing directional preferences rather than in finding configurations where directionally uniform movement is possible. Therefore especially the manipulability ellipsoid [4] and derived measures proposed by Yoshikawa will receive a closer inspection.

A. The manipulability measure

The manipulability ellipsoid in the m -dimensional Euclidean space introduced by Yoshikawa [4] is intended to quantify the ease of arbitrarily changing the position and orientation of the end-effector. It is derived by analyzing the Jacobian matrix of a manipulator (equation 1). The Jacobian matrix at a robot arm configuration \mathbf{q} relates the joint velocities $\dot{\mathbf{q}}$ with the total end-effector velocity in the Cartesian space (the angular velocity ω_E and the translational velocity $\dot{\mathbf{p}}_E$).

$$\begin{pmatrix} \dot{\mathbf{p}}_E \\ \omega_E \end{pmatrix} = J(\mathbf{q})\dot{\mathbf{q}} \quad (1)$$

$$J = U\Sigma V^T \quad (2)$$

The principal axes and singular values σ_i of the Jacobian are obtained by singular-value decomposition of the Jacobian matrix \mathbf{J} (equation 2). Here \mathbf{U} and \mathbf{V}^T are orthogonal matrices, and Σ is a diagonal matrix containing the singular values of \mathbf{J} . The principal axes and singular values define the orientation and the shape of the so-called manipulability ellipsoid. The size of the ellipsoid and its major and minor axes are assumed to represent an ability of manipulation at a certain configuration. A singular value is interpreted as the radius of the ellipsoid in the direction of the corresponding principal axis. The ratio of the minimum and maximum singular value of the ellipsoid can be used to describe the directional uniformity of the ellipsoid and thus the directional uniformity of possible movements at the considered configuration. The volume of the ellipsoid is known as the *manipulability measure* and can be interpreted as a distance of the manipulator from a singular configuration.

Abdel-Malek et al. [10] augment the Jacobian matrix with joint limit criteria and redefine the manipulability measure. The resulting measure is then used to evaluate a robot placement with respect to certain target points.

The manipulability measure is also used in fields other than robot design, two examples shall be mentioned. Guilamo et al. [11] use the manipulability measure to optimize simple trajectories. In a two stage inverse kinematics computation for a redundant robot arm, Guilamo et al. [12] choose configurations according to their manipulability measure.

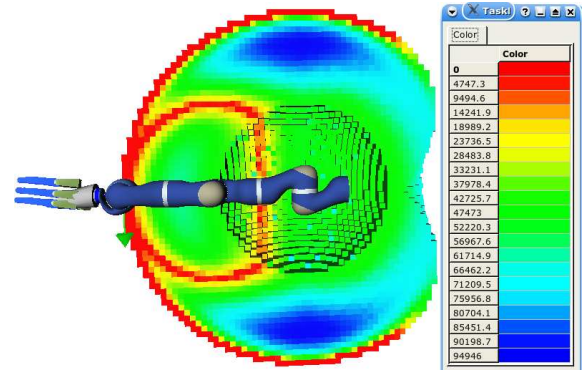


Fig. 2. The singularities of the right arm of the DLR humanoid robot *Justin* are visualized using the manipulability measure. Link lengths were given in cm. The TCP is located in the wrist. The workspace is cut in half for visualization of the 3D structure.

B. Evaluation of the fitness of robot design parameters to our problem

Criteria used for manipulator design often aim either at reaching an isotropic performance of the manipulator in its workspace or maximizing some global performance index derived from local ones. These criteria are not aimed at capturing directional structure in the workspace, rather perhaps at smoothing the structure across the workspace.

The manipulability measure itself is directionless making it impossible to discern directional preferences. However, it approaches zero the closer a robot arm configuration is to a singularity. We have used the manipulability measure to visualize (figure¹ 2) the structure of the singularities for the right arm of the DLR humanoid robot *Justin* [2]. The robot workspace is discretized using small equally sized cubes. A robot configuration is obtained by randomly sampling the configuration space. Via the direct kinematics the Cartesian position of the TCP is computed and mapped to a cube. For this configuration the manipulability measure is computed. Figure 2 shows an approximation of the distribution of the minimum value of the manipulability measure across the workspace. In areas which are red or very close to red, the manipulability measure is very low. These areas either contain singularities or are close to singularities.

When working with the manipulability measure, the choice of the TCP is important. For our robot, we placed the TCP in the wrist. As a result only the translational part of our Jacobian is rank deficient at a singular configuration within link limits. If the TCP was chosen differently the singularities would be distributed between the translational and rotational part of the Jacobian. The visualization of the manipulability measure for a projection of the Jacobian onto e.g. the translational subspace can therefore be vastly different for two different TCPs. For the manipulability measure in figure 2 only the translational part of the Jacobian was used. The exact location of all singularities can be computed by decomposing the Jacobian using the approach

¹The color table used is not optimal for printing the paper in black and white. Please refer to the color pictures in the pdf.

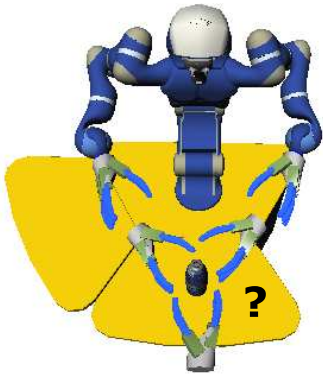


Fig. 3. Illustrates the choices to be made by the humanoid robot Justin concerning arm usage and approach direction.

presented by Konietschke et al. [13]. Compared to this, the manipulability measure can be interpreted as a kind of distance from singular configurations (compare figure 2). It can be used to detect and avoid these.

The manipulability ellipsoid and its axes could be taken to describe directional preferences. However, since the Jacobian (equation 1) relates link velocities to cartesian translational and rotational velocities, the various manipulability criteria for the 6D cartesian space are unit dependent. Furthermore the principal axes of the 6D manipulability ellipsoid, being derived from the Jacobian, mix rotational and translational components as well. Therefore we see problems concerning their interpretability.

Since manipulability values are derived from the Jacobian at a configuration c they are purely local measurements valid only for a small ϵ -neighborhood. Furthermore, they do not take link limits into account. Thus, attested good movability at a configuration may not be possible in the desired direction due to link limits. Abdel-Malek et al. [10] augmented the Jacobian with joint limit criteria. By introducing these additional criteria, the manipulability measure is even harder to interpret due to the additional unit dependency.

Due to the above mentioned problems we refrained from using the manipulability ellipsoid or measures derived thereof to represent the robot arm capabilities. Instead, in the next sections we reformulate the problem and develop a representation based on different ideas.

C. Reformulation of the problem

We already have a completed robot. The question is no longer how to design it, but rather what follows from the design. A robot arm's workspace is not uniform with respect to reachability. Instead, there are regions that can only be reached from specific directions. This directional information needs to be captured.

The DLR robot Justin is a humanoid upper body with 42 degrees of freedom (DOF). It has two redundant arms with 7 DOF each. Using Justin, we want to grasp and manipulate objects using both arms. To decide when to use which arm, we need to be able to evaluate which arm can e.g. best grasp certain objects in the task space (figure 3). Moreover, due to

the redundancy of our robot arms we have to choose among an infinite number of alternative configurations that can be used to approach and grasp an object. Considering a mobile manipulator the question arises how best to position the mobile platform to have optimal manipulation capabilities with respect to the operating area, e.g. a table.

In general, we need a representation of manipulator capabilities that can be used to characterize which places are easily reached. If only a specific direction is of interest, this direction should be applicable to the map, resulting in a filtered representation that masks all information but that lying in the requested direction. Structure inherent to the robot arm's capabilities inside its workspace should be easy to recognize. Using this representation the manipulator should be able to choose good approach directions for objects.

Representing the reachable workspace has already received attention from other research groups. A monte carlo approach to represent the reachable workspace by randomized sampling was introduced by Guan et al. [14]. However, they only provide true/false information concerning the reachability of regions. No directional structure can be discerned from their representation.

III. THE CAPABILITY MAP APPROACH

The ability of humans to manipulate objects depends on the position of their arm in the workspace. Two-handed manipulation is limited to a region where the workspaces of both arms overlap. The best performance is achieved in an even smaller subspace. The same is true for the humanoid robot Justin, whose design is oriented at the human model. In general, every robot arm is designed differently, and therefore has different capabilities. We show that these capabilities result in directional structures specific to workspace regions, and that these structures can be captured and represented in the form of a directional map. As a first step, we introduce a visualization scheme to detect the structure. In a second step, the data of the map is reduced while preserving the stored information to use it for grasp and task planning. We illustrate our approach using the right arm of the DLR humanoid robot Justin.

In a nutshell, the workspace structure is extracted through discretization, randomized sampling, analysis and optimization processes. The robot arm reachability in a certain region of the workspace is examined using inverse kinematics. We will show that indeed structure of the workspace does become obvious. And we will then capture this structure using shape primitives.

A. Discretization

The theoretically possible workspace of the robot arm can be encapsulated by a cube with a sidelength of two arm lengths centered at the robot arm base (figure 4). The maximum workspace of the arm is thereby overestimated. The envelopping cube is then subdivided into equally sized smaller cubes. Using this discretization, we make visual-

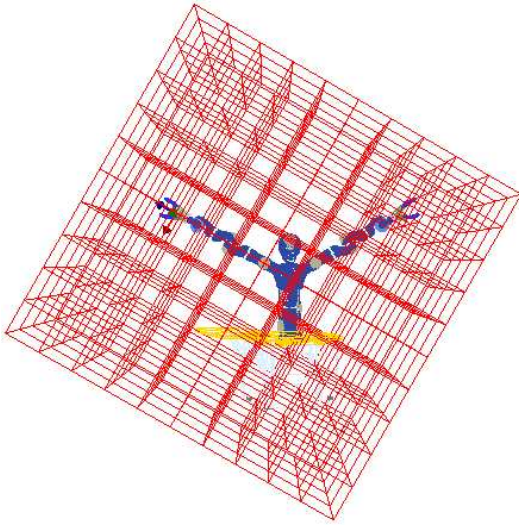


Fig. 4. The maximum workspace of the right arm is overestimated by the enveloping cube which is divided into subcubes of 300 mm sidelength.

ization possible and can analyse specific regions of the workspace in task planning processes.

B. Randomized sampling

The *configuration space* is randomly sampled according to a uniform distribution. For each configuration the position of the TCP is computed via the direct kinematics. In Figure 4 the TCP position is indicated by the coordinate frame in the hand. The TCP position is then mapped to the subcube that contains this position.

It could be considered to use the number of randomly sampled configurations mapped to a subcube as a measure of reachability for a region. The hope would be that the number of sampled configurations assigned to a specific subcube correlates with that region being easily reachable especially w.r.t. exploiting redundancy and versatile manipulation. However this is a false conclusion. When a robot is in a singular configuration, large steps in the configuration space for the links causing the singularity result in small motions in the cartesian workspace. Thus especially in regions containing singularities a large amount of sampled configurations are mapped to subcubes. Therefore the number of configurations mapped to a cube cannot be used to distinguish regions where versatile manipulation is possible. Furthermore it does not allow to discern directional structure. As a consequence, in the next step we use inverse kinematics to examine the workspace.

C. Using the inverse kinematics to examine the workspace

Into each cube a sphere with a diameter equal to the width of the cube is inscribed (figure 5(a)). Using the spiral point algorithm proposed by Saff et al. [15] we generate N equally distributed points on the sphere. For each point thus obtained, we generate a frame. In figure 5(b) the frame is shown with the x-axis (red) and the y-axis (green) tangential to the sphere and the z-axis (blue) pointing towards its center. The

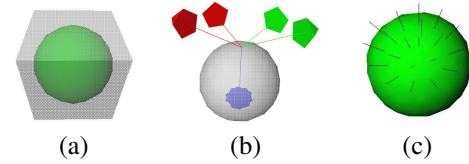


Fig. 5. Shows a sphere inscribed into the cube (a), exemplary frames for a point on the sphere (b), valid inverse kinematics solutions on a sphere (c).

frame is then turned around its z-axis according to a fixed stepsize. Each resulting frame is considered to constitute a TCP frame to be reached by the arm in question and an inverse kinematics solution is computed. If for one of the rotated frames at a specific point p on the sphere an inverse kinematics solution is available, that point p is marked in the underlying datastructure. It is important to mention that thus the z-orientation of the TCP w.r.t. the sphere center is discarded.

The inverse kinematics for our 7 DOF redundant robot arm is computed by combining an analytical solution as proposed by Craig [16] with optimization procedures to optimize the redundant degree of freedom [17]. The randomly sampled configuration is taken as the initial configuration supplied to the inverse kinematics. Since an inverse kinematics for redundant robots does not have a single unique solution and involves some iterative optimization, a starting solution, that is already near the desired solution is beneficial for the computation.

D. Reachability spheres to characterize the workspace

In the visualization, for each valid inverse kinematics solution on a sphere, a line is drawn originating in the sphere center (figure 5(c)). The spheres visualize the reachability for a region. We therefore call them reachability spheres. We assign a measure called the *reachability index* D (equation 3) to each sphere characterizing the reachability of the region enclosed by the sphere. In equation 3, N is the total number of points on a sphere and R is the number of valid inverse kinematics solutions recorded. The resulting value informs about the percentage of points on the sphere, having an inverse kinematics solution.

$$D = \frac{R}{N} \cdot 100 \text{ with } R \leq N \quad (3)$$

Using the reachability index, already some structure inherent to the robot workspace can be visualized. To achieve this, the spheres are colored with respect to their reachability index D . Figure 6 presents the change of the reachability index across the robot arm workspace. As expected, as we move into the interior of the workspace the index gets better reaching its optimum in the blue region. The nearer we then come to the robot arm base the more the index decreases. The reachability index D for our robot arm ranges from 0 to 76. For better visibility, in some figures the full workspace is cut in half along the arm as shown in figure 7. Figure 8 (top) shows all spheres with an index D in the lowest 10 percent of the reachability index ($D \in [0, 8]$) across the workspace.

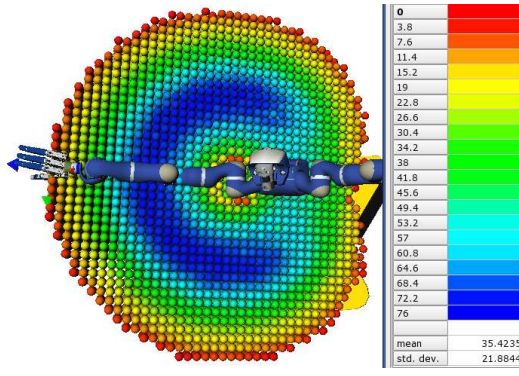


Fig. 6. Shows the reachability spheres across the workspace. The workspace representation was cut as shown in fig. 7 for better visibility of the structure.

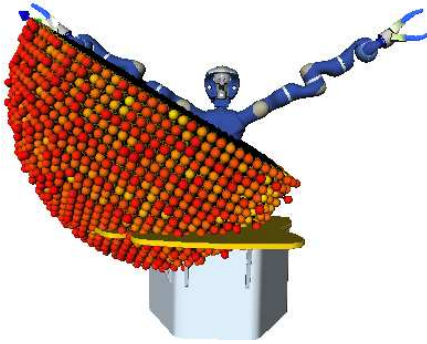


Fig. 7. Visualizes the cut across the workspace of the robot arm.

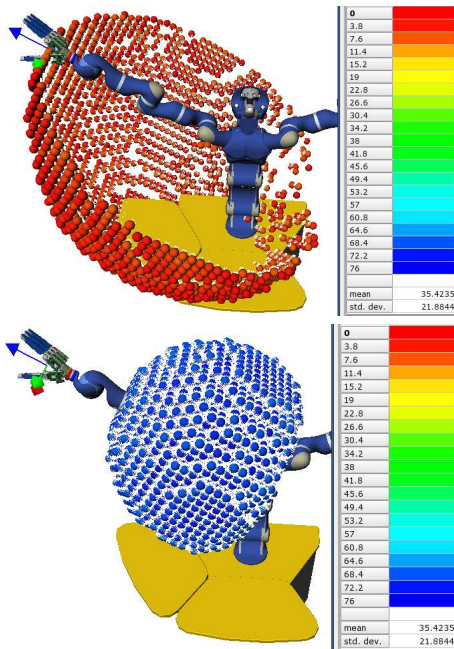


Fig. 8. Shows the spheres with an index D in the lowest 10% of the reachability index (top) and in the upper 10% of the reachability index (bottom).

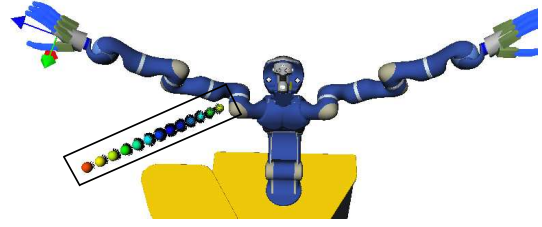


Fig. 9. Shows spheres from the border of the workspace moving inward.

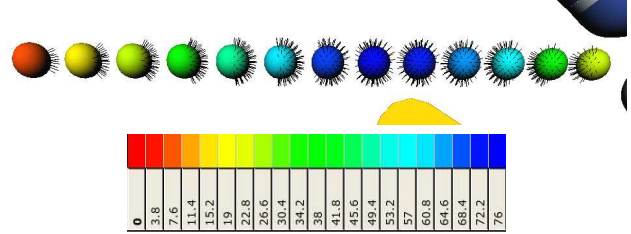


Fig. 10. A close up of the spheres from fig. 9 with corresponding color table for the reachability index.

As expected the spheres with the lowest index are on the border of the workspace. Figure 8 (bottom) shows spheres with an index D in the top 10 percent of the reachability index ($D \in [68, 76]$) across the workspace. It can be seen that spheres with a good index D lie on somewhat more than a half a sphere shell around the robot arm base (also compare figure 6) with a diameter of approximately half the robot arm length. We would get a complete sphere shell when disregarding the link limits. For figure 6 to 8, 10^6 random samples were drawn. The spheres have a radius of 25 mm and 200 points are distributed on a sphere. The stepsize for turning the frame around its z-axis is 30 degrees.

E. Analyzing the structures in the workspace

The reachability index again is a directionless measure. While it does help to recognize structure it does not fulfill our requirements defined in section II-C. The reachability spheres are able to represent directional preferences but testing a specific direction against all valid candidates on every sphere is computationally quite expensive. Furthermore, the spheres also cover all those points where no valid inverse kinematics solutions were found. In this section we have a closer look at the geometric structure present in the workspace and represented by reachability spheres.

Figure 9 shows reachability spheres on a line across the workspace marked by a box. For better recognizability, figure 10 shows a zoomed view. As expected, it can be observed that as we move into the workspace (starting with the red sphere) the number of points with valid inverse kinematics solutions increases. Having a closer look at each sphere and the distribution of the inverse kinematics solutions, we observe a cone like structure for the red, yellow and light green spheres. As we move further inwards, the cones open out and the structure changes. For the dark blue spheres, a ring structure can be observed. Although not present in figure

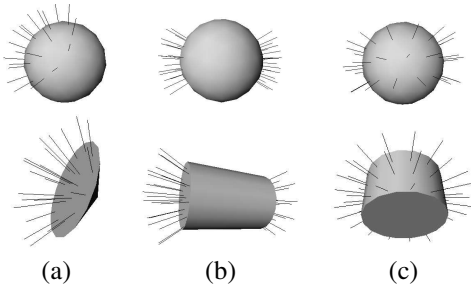


Fig. 11. Cones (a) and two cylinder types to capture structures. (b) Cylinder type t1 (c) Cylinder type t2

10, double cone structures have also been found.

Capturing and approximating the structures using shape primitives such as cones and cylinders would result in immense data reduction. Testing directions against these shape primitives is a lot faster than testing against single inverse kinematics solutions. To determine whether a direction vector lies in a cone and thus whether inverse kinematics solutions are available reduces to one computation of the angle between the shape's axis and the direction.

The attached video shows the spheres from figure 10 and fitted shape primitives. The spheres and shape primitives are rotated to enable a thorough inspection.

F. Capturing the structure to construct a map

In this section, we describe how and what shape primitives were fitted to the data. We introduce a measure based on the relative error of an approximation to evaluate the appropriateness of a shape primitive to the structure captured by a reachability sphere.

As a consequence of the observations from the last section, we propose the shape primitives presented in figure 11. Cones can be used to approximate cone-like structures (figure 11 (a)). A cylinder of type t1 (figure 11 (b)) can capture double cone structures and a cylinder of type t2 (figure 11 (c)) can capture ring structures. The process of fitting the shape primitives to the data involves optimizing the main axis of the shape primitive and its opening angle to best approximate the data. The axis is optimized using principal component analysis. To optimize the opening angle we evaluate the appropriateness of the shape primitive to represent the data for different opening angles.

The appropriateness of a shape primitive is determined by how many of the valid inverse kinematics solutions the shape captures (reachable points), but also by how many points of the sphere are included, where no inverse kinematics solution could be computed (unreachable points). We combine these observations in the computation of the shape fit error (SFE). The SFE is computed for a shape primitive with axis \vec{a} and opening angle α (compare figure 12). It is derived from the relative error made by the shape fitting process to capture all inverse kinematics solutions available for a sphere. The ideal shape covers all R inverse kinematics solutions available for a sphere. Compared to an ideal approximation, a suboptimal shape fails to cover r reachable points and wrongly covers u

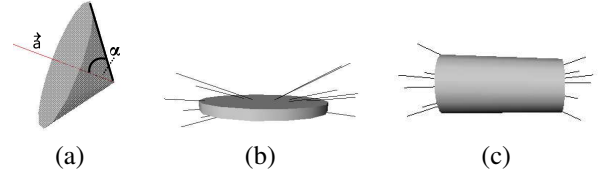


Fig. 12. (a) A cone with axis \vec{a} and opening angle α . (b) The Cylinder with a SFE=100 is not fitted well to the data. (c) The cylinder is optimally fitted to the data used in (b) and has a SFE=0.

unreachable points. It has an approximation value of $R - u - r$. Thus we define the relative error of an approximation by subtracting the ideal approximation value R from the value for a suboptimal shape and divide the result by the ideal value R (equation 4 with $r, u \geq 0$). The shape fit error is then defined in equation 5 using this relative error. The SFE is limited to $[0, 100]$. If a shape has a relative error greater than 1, it is no better approximation than a shape with a relative error of 1. Both are unacceptable. Therefore both receive the maximum SFE of 100.

Figure 12(b), (c) show a cylinder of type t1 with two different axes and opening angles fit to the same data set. While the first version collects all lines, it also covers many unreachable points and receives the highest SFE of 100 as a result. The data is optimally represented by the second cylinder (fig. 12(c)) receiving the optimum SFE=0.

$$\left| \frac{(R - u - r) - R}{R} \right| = \left| \frac{-u - r}{R} \right| = \frac{u + r}{R} \quad (4)$$

$$SFE(\vec{a}, \alpha) = \begin{cases} \frac{u+r}{R} \cdot 100 & \text{if } u + r \leq R \\ 100 & \text{if } u + r > R \end{cases} \quad (5)$$

The results of replacing the spheres by the best fitting cone or the best fitting cylinder of type t2 (compare fig. 11) are shown in figure 13 and figure 14. The lines symbolizing inverse kinematics solutions have been removed for better visibility. As we expected and see in figure 13, in the outer zones of the workspace cones are good approximations and have a low SFE shown by the blue coloring. Only for the inner workspace region do cones have high SFEs. Here, cylinders are a solution. Cylinders receive low SFEs in the inner (blue) workspace regions (figure 14). For the remaining workspace, the cylinder hypothesis is inadequate and the SFE is maximum. Thus cones and cylinders representations complement each other. These results suggest that it would be best to take the shapes with the lowest SFE from all three shape fitting processes to obtain a mixed map that is better able to represent all structures found in the workspace.

IV. EVALUATION OF THE ROBOT CAPABILITY MAPS

In this section we will evaluate the quality of the derived capability maps. We computed the cone, cylinder and mixed map for the right arm of the DLR humanoid two-arm system Justin. 10^6 configuration space samples were drawn. And 200 points were distributed on a sphere with radius=25 mm. The mean and the standard deviation of the SFE across the robot arm workspace will serve as a performance indicator.

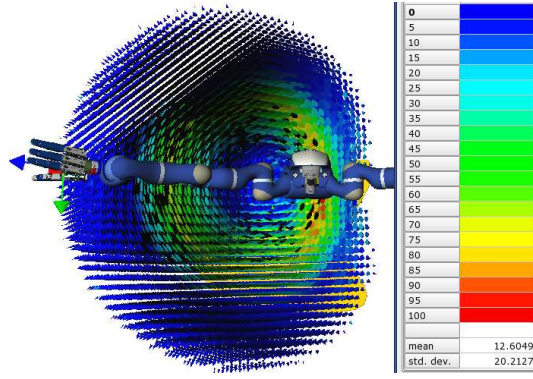


Fig. 13. The cone best fitting the inverse kinematics data replaces the reachability sphere.

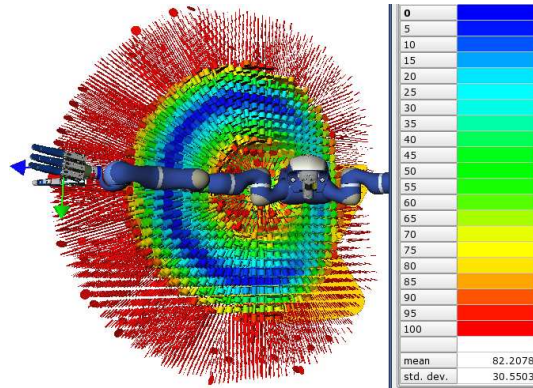


Fig. 14. The cylinder (type t2) best fitting the inverse kinematics data replaces the reachability sphere.

SFE	spheres	cones	cylinder t1	cylinder t2	mixed	optimal
mean	87.8	12.6	85.9	82.2	7.2	0
std. dev.	21.3	20.2	23	30.6	10.7	0

TABLE I

PERFORMANCE FIGURES FOR DIFFERENT REPRESENTATIONS (SPHERE
RADIUS=25 MM, 10^6 SAMPLES, Z-STEP SIZE=30 DEG)

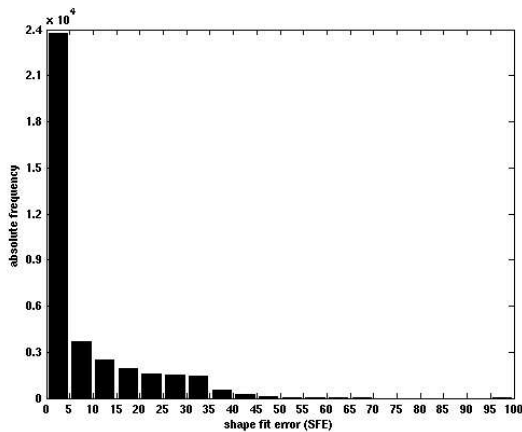


Fig. 15. Absolute frequency of the SFE visualized as a histogram.

Table I shows the performance figures rounded to the first decimal place for capturing structures with spheres, cones and cylinders. To obtain the mixed map (fifth column), we always choose the shape with the lowest SFE and place it in the map.

Using cones to approximate the structures in the workspace already leads to a low mean SFE. Figure 13 shows that for the majority of the cones, the SFE is low which explains the low mean in table I. Both cylinder maps have a mean SFE significantly higher than the cone map. Compared to the cones, only the inner part of the workspace can be represented well by cylinders, resulting in this high mean SFE. As expected, the mean SFE is lowest for the map that mixes cones and cylinders. The standard deviation has also improved significantly. Figure 15 shows the histogram of the absolute frequency of the SFE. It confirms that the majority of the shapes fit the data well.

We can conclude that the mixed map presents the workspace structure well and therefore the robot capabilities in its workspace. We will now call it *capability map*. It fulfills the criteria developed in section II-C. Combined with the reachability index, it can be used to recognize places in the workspace enabling versatile manipulation of objects. Furthermore, the capability map facilitates the search for good directions to approach objects since it encodes directions in a compact form. If only a specific direction is of interest all shapes failing to satisfy this direction can be masked.

V. THE CAPABILITY MAP IN GRASP AND MOTION PLANNING

In this section, we will sketch two application szenarios for the capability map. The shapes in the capability map are characterized by two indices, the reachability index (eq. 3) and the SFE (eq. 5). Using the reachability index, regions can be identified where versatile manipulation is possible, i.e. the blue regions in figure 6. These regions could then be superimposed over objects to be manipulated e.g. by moving the torso. Figure 16 shows the DLR robot Justin in its home position. The bottle is situated at the border of the robot reachable workspace. By positioning of the torso, the bottle can be overlaid with the region of the capability map where the reachability index is highest ($D \in [68, 76]$) allowing for more alternatives to reach the object and exploitation of redundancy. The map can thus be used to position the torso of the DLR robot Justin or a mobile manipulator to optimally reach objects.

A grasp planer as developed by Borst et al. [18] generates stable grasps for objects disregarding kinematic constraints or obstacles. Figure 3 illustrates a number of grasp directions for one object. Some are merely awkward and others are kinematically unreachable. If additional obstacles were added more grasps would have to be discarded. However, the number of valid grasps is still infinite.

Choosing a single position on the object and requesting a grasp can result in the failure of the grasp planning algorithm. While the position may be kinematically reachable, no stable grasp can be computed. At this point the capability map can

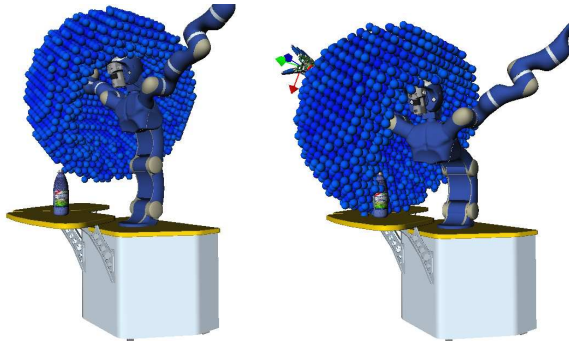


Fig. 16. Positioning the torso using the capability map.

representation type	reachability spheres	homogenous shape map	capability map	total
time	12.3 h	1.8 min	6.4 min	12.4 h

TABLE II

COMPUTATION TIMES FOR BUILDING THE REPRESENTATIONS.

be used to preselect regions for generating grasps that give the grasp planner the freedom it needs while ensuring that the generated grasps can be reached. We will therefore combine the map with our grasp planner [18] to speed up the grasp generation process.

VI. CONCLUSIONS AND FUTURE WORK

Structure is indeed present in a manipulator's workspace and can be captured in the form of a map. We developed the representation called the *capability map* to capture the manipulator capabilities in its workspace. The map is anchored at the robot arm base and moves with the upper body.

The capability map is computed offline for a specific robot arm. Table II shows computation times for building the reachability sphere representation, a homogenous shape map (only one shape type) and the final mixed map, i.e. the capability map. The total computation time is derived by adding column 1 and 3. While we did not optimize the algorithms, it can be seen that computation has to be done offline. The derived capability map can then be consulted online e.g. in grasp placement decision modules.

It is especially useful for two-armed humanoid robots. Humanoid robots, e.g. Justin, could use this representation to decide which arm is best suited for certain tasks. Two-handed manipulation of an object can be done at a location where versatile regrasping is possible. Using the capability map to solve the problem of finding an approach and grasp direction that facilitates the work of path planners [3] is possible. Furthermore, the maps could perhaps be used to develop cartesian space path planners or evaluate and compare robot designs.

Some of the shapes fitted to the data still include a few points on the sphere for which no inverse kinematics solution was found. It remains to be evaluated how often these positions are encountered by e.g. grasp planners using the map to preselect locations for the generation of grasps.

The generation process of the capability map or the inverse kinematics needs to discard configurations that cause self-collisions or collision with the table that Justin is permanently attached to. Considering that our robot arms are redundant, it remains an open question whether this should be done by or external to the inverse kinematics.

The capability map does not discriminate regions containing singularities. These situations can either be resolved by the control algorithm driving through a singularity. Or these situations can be addressed by combining the capability map with a singularity map based on the manipulability measure, e.g. shown in figure 2.

VII. ACKNOWLEDGMENTS

This work was funded by the German Research Foundation (DFG) within the cluster of excellence CoTeSys (Cognition for Technical Systems).

REFERENCES

- [1] M. Kawato, "Internal models for motor control and trajectory planning," *Curr. Opin. Neurobiol.*, vol. 9, no. 6, pp. 718–727, 1999.
- [2] C. Ott, O. Eiberger, W. Friedl, B. Büml, U. Hillenbrand, C. Borst, A. Albu-Schäffer, B. Brunner, H. Hirschmüller, S. Kielhöfer, R. Konietschke, M. Suppa, T. Wimböck, F. Zacharias, and G. Hirzinger, "A humanoid two-arm system for dexterous manipulation," in *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 276–283.
- [3] F. Zacharias, C. Borst, and G. Hirzinger, "Bridging the gap between task planning and path planning," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006, pp. 4490–4495.
- [4] T. Yoshikawa, *Foundations of Robotics: Analysis and Control*. Cambridge, Massachusetts: MIT Press, 1990.
- [5] J.-Y. Park, P.-H. Chang, and J.-Y. Yang, "Task-oriented design of robot kinematics using the grid method," *Advanced Robotics*, vol. 17, no. 9, pp. 879–907, 2003.
- [6] F. Park and R. Brockett, "Kinematic dexterity of robotic mechanisms," *J. Robotics Research*, vol. 13, no. 1, pp. 1–15, 1994.
- [7] R. Sturges, "A quantification of machine dexterity applied to an assembly task," *J. Robotics Research*, vol. 9, no. 3, pp. 49–62, 1990.
- [8] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *J. Robotics Research*, vol. 6, no. 2, pp. 72–83, 1987.
- [9] L. Stocco, S. E. Salcudean, and F. Sassani, "Fast constrained global minimax optimization of robot parameters," *Robotica*, vol. 16, no. 6, pp. 595–605, 1998.
- [10] K. Abdel-Malek and W. Yu, "Placement of robot manipulators to maximize dexterity," *J. Robotics and Automation*, vol. 19, no. 1, pp. 6–15, 2004.
- [11] L. Guillemo, J. Kuffner, K. Nishiwaki, and S. Kagami, "Manipulability optimization for trajectory generation," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2006, pp. 2017–2022.
- [12] —, "Efficient prioritized inverse kinematic solutions for redundant manipulators," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005, pp. 1905–1910.
- [13] R. Konietschke, G. Hirzinger, and Y. Yan, "Singularities of the 9-dof dlr medical robot setup for minimally invasive applications," in *Advances in Robot Kinematics*, 2006, pp. 193–200.
- [14] Y. Guan and K. Yokoi, "Reachable space generation of a humanoid robot using the monte carlo method," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006, pp. 1984–1989.
- [15] E. Saff and A. Kuijlaars, "Distributing many points on the sphere," *Mathematical Intelligencer*, vol. 19, no. 1, pp. 5–11, 1997.
- [16] J. Craig, *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.
- [17] R. Konietschke, S. Frumento, T. Ortmaier, U. Hagn, and G. Hirzinger, "Kinematic Design Optimization of an Actuated Carrier for the DLR Multi-Arm Surgical System," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) 2006*, Beijing, China, 2006.
- [18] C. Borst, M. Fischer, and G. Hirzinger, "A Fast and Robust Grasp Planner for Arbitrary 3D Objects," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 1999, pp. 1890–1896.