Autonomous Robots

Representing the Robot's Workspace through Constrained Manipulability Analysis

Nikolaus Vahrenkamp · Tamim Asfour

Received: 10 December 2013 / Accepted: 12 June 2014

Abstract Quantifying the robot's performance in terms of dexterity and maneuverability is essential for the analysis and design of novel robot mechanisms and for the selection of appropriate robot configurations in the context of grasping and manipulation. It can also be used for monitoring and evaluating the current robot state and support planning and decision making tasks, such as grasp selection or inverse kinematics (IK) computation. To this end, we propose an extension to the well-known Yoshikawa manipulability ellipsoid measure [40], which incorporates constraining factors, such as joint limits or the self-distance between manipulator and other parts of the robot. Based on this measure we show how an extended capability representation of the robot's workspace can be built in order to support online queries like grasp selection or inverse kinematics solving. In addition to single handed grasping tasks, we discuss how the approach can be extended to bimanual grasping tasks. The proposed approaches are evaluated in simulation and we show how the extended manipulability measure is used within the grasping and manipulation pipeline of the humanoid robot ARMAR-III.

Keywords Manipulability · Reachability · Redundant Manipulators · Capability Representation

1 Introduction

For advanced robot systems operating in such different domains as industrial applications, service or humanoid robotics the ability of supervising own movements is essential for successful task execution. Hence, the systems must be equipped with methods to measure their performance, accuracy or success rate. In this context, measuring the manipulability is a well-known technique for determining the ability to maneuver in workspace. This measure, introduced by Yoshikawa in 1985 [40], can be seen as a quality index that gives information about the distance to singular configurations, which limit the possibility to adjust the end effector pose in workspace. By exploiting the robot's redundancy, poses which result in higher manipulability can be favored in order to allow better adaption during execution. Additionally, manipulability information can be used to support planning and decision processes, e.g. when an optimal grasp has to be selected out of a set of potential grasps. In this context a precomputed capability representation can be useful for efficient processing of online queries. Such online queries may cover extended inverse kinematics (IK) problems, including robot placement queries and grasp selection tasks. Due to the extended problem formulation, integrated algorithms are preferable, which rely on offline generated data that represents the robot's capabilities in terms of reaching and manipulation. Since during online processing IK solving is just one processing stage in a usually complex grasping pipeline, efficiency is highly desirable. To achieve the required performance, a representation of the manipulator's capabilities can be generated in an offline step, whereby different types of information can be addressed. Usually reachability information is of high interest since it gives information about workspace poses which are within reach of the manipulator. As we showed in earlier work [33] additional quality information, such as the manipulability, can be addressed in order to achieve an extended representation of the robot's capabilities.

1.1 Related Work

Yoshikawa introduced the manipulability index [40] as a quality index for redundant manipulators, which describes the distance of a given pose to a singular configuration. Since the ability to maneuver in workspace decreases at singular configurations this quality index is useful for online control or IK solving tasks. The quality measure is based on analyzing the manipulability ellipsoid which is spanned by the singular values of the Jacobian matrix. The manipulability measure is defined as

$$w = \sqrt{det(JJ^T)} = s_1 s_2 \cdots s_n. \qquad (1)$$

High Performance Humanoid Technologies (H^2T)
Karlsruhe Institute of Technology (KIT)
Adenauerring 2, 76131 Karlsruhe, Germany
E-mail: {vahrenkamp,asfour}@kit.edu

As shown in Eq. 1, w can be rewritten by multiplying the singular values s_i, which means the measure is proportional to the volume of the manipulability ellipsoid. In [28] a measure related to the Condition Number has been introduced as the relation of the smallest to the largest singular value

$$c = \frac{1}{cond(J)} = \frac{s_n}{s_1}. \quad (2)$$

In addition to singularities, joint limits have a major impact on the end effector's maneuverability in workspace [13,26]. This fact has been addressed by Tsai et al. [29] by introducing a penalization term that considers the distance to the lower (l_j^-) and the upper (l_j^+) joint limits:

$$P(\theta_j) = 1 - \exp(-k \prod_{j=1}^{n} \frac{(\theta_j - l_j^-)(l_j^+ - \theta_j)}{(l_j^+ - l_j^-)^2}). \quad (3)$$

In this penalty function k is a scaling factor that can be used to adjust the behavior near joint limits. By multiplying the manipulability with P, configurations which are near joint limits are penalized, but redundancy is not considered, which can lead to an erroneous quality information (e.g. one joint is close to a boundary while the end effector's maneuverability is not affected due to redundancy). In [38] constraining force effects such as gravity or external forces exerted on the end effector are investigated.

A dexterous performance measure is presented in [1]. By analyzing the so-called augmented Jacobian matrix, which combines information about the position, orientation, and joint limits of the end-effector, the manipulability of joint configurations can be measured. It is shown that this approach is more accurate than Yoshikawa's manipulability measure and exemplary applications for optimizing base placements are given. However, the approach relies on the generation of surface patches to represent the reachable workspace, which can lead to challenging computational problems.

As shown in [17] or [12], the use of manipulability ellipsoids can lead to issues that are caused by mixing translational and rotational sub spaces for manipulability measurement. By using manipulability polytopes the described problems can be avoided, but higher computational costs have to be taken into account.

In this work, we use a weighting matrix W in order to set the translational components of the Jacobian matrix in relation with the rotational components. The influence of the translational and the rotational components on the quality values can be adjusted with these weights. In our experiments we setup the weighting matrix in a way that a translation of 1 mm corresponds to a rotation of 8 degree. More details on the effects of translational and rotational components on the manipulability calculation can be found in [12].

Task constraints are considered by the directional manipulability approach for single arm [21] or bimanual [8] manipulators. Manipulability analysis of multi-contact setups is related to grasp stability computation, which is usually done by relating fingertip contact forces to the net wrench applied to the grasped object. This relation, expressed by the grasp matrix G, can be analyzed to identify stability features and to get manipulability information [23]. The manipulability of bi- or multimanual setups has been investigated in [6, 9, 16, 18, 22, 27]. Most works consider two or more end effectors cooperatively grasping an object and a distinction is made between internal (relative) and external (absolute) forces. Internal forces describe the possibility of a multi arm system to exert forces on the object while resting at the given workspace position. Based on these considerations an analysis of disturbance absorption and grasping force can be performed. External forces are considered when the object's Cartesian maneuverability is of interest.

An analysis of the manipulability of parallel robots is presented in [20].

The polytope approach for manipulability computation [12] is used in [16] to build up a polytope for multi arm systems by applying set and geometric operations on the polytopes of the involved single arm manipulators. As for the single arm case, the multi arm approach lacks of the possibility to derive the polytopes via a closed-form analytic method. In [18] a dual arm manipulability ellipsoid is introduced as an ellipsoid that approximates the intersection of both manipulator's manipulability ellipsoids. As stated in [9], it may be difficult to generalize this representation of the bimanual manipulability.

In [19], the so-called *Force Workspace*, a representation of the feasible workspace in terms of maximum forces and torques, has been introduced. The approach relies on discretizing the configuration space, which results in an exponential growth with the number of involved joints and hence this approach is not suitable for robots with a high number of DoF.

Spatial reachability analysis for redundant manipulators has been addressed in [11, 31, 41] and [25]. In these works a discretized representation of the Cartesian space is used to encode the reachable workspace of the robot. The reachability of a manipulator is thereby loosely defined as the volume of the workspace that can be reached by the end effector. Usually the Cartesian space (three- or six-dimensional) around the robot is investigated, whereas only the position (3D) or position and orientation (6D) is considered. The reachability data is created during an offline step by filling a voxelized data structure. The grid entries represent either a probability that a given pose that lies within the corresponding workspace voxel is reachable or binary information is used to indicate that a voxel lies (partly) within the reachable workspace. Extensions to bimanual applications have been presented in [35, 42]. In [14] an ego-centered representation of the reachability of a humanoid robot's manipulator is learned autonomously. An extension of this work towards whole-body reaching tasks while considering joint limits is presented in [15].

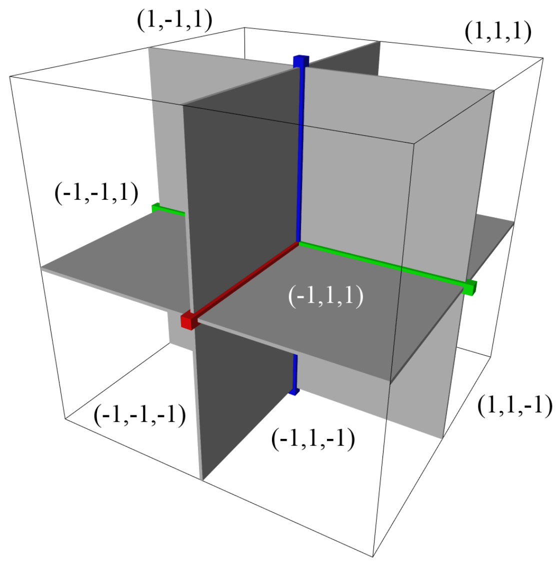

Fig. 1 In a 3D workspace, the space of potential movements can be partitioned by $2^3 = 8$ octants.

2 Extended Manipulability Measure

In the following we will develop an extended measure that can be used to describe the maneuverability of an end effector in workspace. It is based on existing methods for determining the manipulability, which we extended by methods for incorporating limiting constraints. Such constraints arise from joint limits and workspace (self) distance information and hence they have to be considered in order to get more realistic information about the end effector's possibility to maneuver in workspace. Note, that any other constraint can be incorporated as long as the derivation with respect to joint movements can be built.

In the context of optimal control, several approaches are known to avoid joint limits and obstacles [7, 10]. Compared to these works, where an optimal control strategy for a requested end effector movement is of interest, we want to investigate the limitations that are given for any possible movement in workspace. As shown later, we use several related methods, such as gradient computations, in order to determine penalization terms that lower the measure according to the given constraints.

2.1 Constraints Analysis

Throughout this work we consider the two constraints that are introduced by joint limits and the manipulator's self-collision distance. By incorporating these constraints into the manipulability calculation, we have to consider the fact that the symmetry of the manipulability ellipsoid computation cannot be preserved. This artefact is obvious since the possibility to move in one direction is limited differently compared to a movement in the opposed direction. E.g. one can think of a joint which is near to the lower joint limit, and hence penalization terms must have a higher influence for movements towards the lower limit compared to movements towards the upper limit. The same is true for movements in Cartesian space, where movements in some directions are limited due to obstacles, while movements in other directions are not.

Since the manipulability computations are based on an analysis of the Jacobian matrix J, we consider the constraints by introducing penalization terms that are applied on each entry of J. As shown above, these penalization terms change depending on the investigated movement. Hence, the space of potential movements has to be partitioned in order to distinguish between the different penalization factors. This is done by computing the corresponding manipulability independently for each hyperoctant in the 6 DoF movement space which consists of a 3 DoF translational and a 3 DoF rotational part. Therefore, the space of workspace movements is partitioned by 2^6 hyperoctants, which are identified by $\Gamma \in \{-1, +1\}^6$. A visualization of the 3D case is shown in Figure 1.

For each Γ an augmented Jacobian $\tilde{J}$ is constructed by applying the two penalization terms L (joint limits) and O (obstacle/self distance) to the corresponding entry of the Jacobian matrix J:

$$\tilde{J}_{i,j}(\Gamma, \theta) = L_{i,j}(\Gamma, \theta) O_{i,j}(\Gamma, \theta) J_{i,j}(\theta). \tag{4}$$

By applying Eq. 4 to all possible permutations of Γ, 64 Jacobians are generated, each of which describes the maneuverability in the corresponding hyperoctant of the space of potential movements.

2.1.1 Joint Limit Constraints

With $L_{i,j}$ of Eq. 4, the entries of J are penalized according to the quality measure that arises from the current distance to joint limits (l_j^- and l_j^+). Since the distance to lower and upper joint limit usually differs, the corresponding penalization terms also differ and hence, the quality measure depends on the investigated workspace movement. By exploiting the information that is served by the entries of J, we are able to determine which joint limit has to be considered for each Γ. For a given joint θ_j, the j-th column of J provides information about the Cartesian movement direction that is caused by moving the joint. Hence, for each Γ, which defines the Cartesian direction, we can identify which joint limits have to be considered.

In the following, we derive two penalization terms p_j^- and p_j^+ for each joint, which are used to build the joint limit penalization matrix L. In [7] a joint limit potential function $h(\theta)$ is proposed and the joint limit gradient function $\bigtriangledown h(\theta)$ is derived. For each joint θ_j the corresponding entry of $\bigtriangledown h(\theta)$ is given with

$$\bigtriangledown h(\theta)_j = \frac{\partial h(\theta)}{\partial \theta_j} = \frac{(\theta_j - l_j^-)^2(2\theta_j - l_j^+ - l_j^-)}{4(l_j^+ - \theta_j)^2(\theta_j - l_j^-)^2}. \tag{5}$$

This gradient is equal to zero, if the joint is at the middle of its range and goes to infinity at either limit. It can be used to avoid joint limits during online control as shown in [7] or [10]. In order to construct an according penalization term we have to distinguish the different potential movement directions in workspace (see Section 2.1). Depending of the actual position of the joint (either located in the lower or the upper half of its range), the following two penalization terms are built:

$$\begin{aligned} p_j^- &= \begin{cases} 1, & |\theta_j - l_j^-| > |l_j^+ - \theta_j| \\ \frac{1}{\sqrt{1+|\bigtriangledown h(\theta)_j|}}, & otherwise \end{cases} \\ p_j^+ &= \begin{cases} \frac{1}{\sqrt{1+|\bigtriangledown h(\theta)_j|}}, & |\theta_j - l_j^-| > |l_j^+ - \theta_j| \\ 1, & otherwise \end{cases} \end{aligned} \tag{6}$$

In Eq. 6, the penalization term p_j^- stands for the penalization that has to be applied when investigating the joint's movement in negative direction. If the current joint value is located in the upper half of its range, the value is 1, which means that the movement does not underlie any penalizations. Otherwise, the movement is penalized according to Eq. 5. The second part of Eq. 6, describes the construction of p_j^+, the penalization term that is applied when a movement in positive joint direction is investigated. In this case, the penalization is only considered when the current joint position is located in the upper half of its range.

Depending on the investigated movement of θ_j, either p_j^- (move towards lower joint limit) or p_j^+ (move towards upper joint limit) can be applied in order to penalize the computed maneuverability. Therefore, the following penalization matrix is computed:

$$L_{i,j}(\Gamma, \theta) = \begin{cases} p_j^-, & \operatorname{sgn}(J_{i,j}(\theta)) \operatorname{sgn}(\Gamma_i) < 0 \\ p_j^+, & otherwise \end{cases} \tag{7}$$

The selection which penalization has to be applied depends on two facts. First, the sign of the corresponding Jacobian entry $J_{i,j}$ indicates the direction a positive Cartesian movement causes in joint space. Further, the sign of i-th entry of Γ indicates which hyperoctant is considered. E.g. a positive sign of $J_{i,j}$ and a negative sign of the i-th entry of the hyperoctant description Γ results in a joint space movement towards the negative joint limit and hence, p_j^- has to be considered for penalization.

2.1.2 Obstacle and Self Distance Constraints

The second constraint that is considered in this work is caused by limitations arising from obstacles, such as environmental objects or parts of the robot. In order to take these constraints into account, the penalization matrix O is used as described in Eq. 4. To this end penalization terms are derived from the distance and the position of any limiting obstacles. First, the nearest points p_o and p_m on the surface of the obstacle and the manipulator are determined to build the corresponding obstacle vector $v' = p_o - p_m$ and the obstacle distance $d = |v'|$. The vector $v \in \mathbb{R}^3$ is extended to $\mathbb{R}^6$ by setting the rotational components to zero: $v = [v_0, v_1, v_2, 0, 0, 0]^T$. Additionally, a collision function $P(\theta, d)$ is needed with the following properties: $P(\theta, d)$ should go to infinity for $d \to 0$ and decays exponentially to zero as d increases. As proposed in [10], we choose

$$P(\theta, d) = e^{-\alpha d} d^{-\beta}. \tag{8}$$

The parameters α and β can be used to adjust the obstacle influence (see [10] for details). The gradient of P is the collision gradient function, which gives information about how each joint influences the obstacle distance:

$$\bigtriangledown P(\theta, d) = \frac{\partial P}{\partial \theta} = \left[\frac{\partial P}{\partial \theta_1}, \ldots, \frac{\partial P}{\partial \theta_n}\right] = \frac{\partial P}{\partial d}\frac{\partial d}{\partial \theta}. \tag{9}$$

As shown in [10] $\bigtriangledown P$ can be computed with

$$\begin{aligned} \tfrac{\partial P}{\partial d} &= -e^{-\alpha d} d^{-\beta}(\beta d^{-1} + \alpha), \\ \tfrac{\partial d}{\partial \theta} &= \tfrac{1}{d}\left[J^T v\right]^T. \end{aligned} \tag{10}$$

Finally, two penalization terms can be constructed similar to Eq. 6. Again, we are interested in all potential movements of the end effector and therefore a distinction is made, whether a workspace movement in negative ($o^-_{i,j}$) or positive ($o^+_{i,j}$) direction is investigated:

$$
\begin{aligned}
o^-_{i,j} &= \begin{cases} \frac{1}{\sqrt{1+|\bigtriangledown P_j|}}, & v_i \leq 0, i \leq 3 \\ 1, & otherwise \end{cases} \\
o^+_{i,j} &= \begin{cases} \frac{1}{\sqrt{1+|\bigtriangledown P_j|}}, & v_i > 0, i \leq 3 \\ 1, & otherwise \end{cases}
\end{aligned}
\tag{11}
$$

In Eq. 11 the terms $o^-_{i,j}$ and $o^+_{i,j}$ represent the penalizations that have to be applied when investigating movements in negative respectively positive Cartesian directions. Penalizations are only applied for translational dimensions ($i \leq 3$) and if the obstacle is located in the corresponding direction (v_i).

Finally, the collision penalization matrix O is computed for every Γ. Here, either o^-_i (movement in negative coordinate axis) or o^+_i (movement in positive coordinate axis) is considered:

$$
O_{i,j}(\Gamma,\theta) = \begin{cases} o^-_{i,j}, & \mathrm{sgn}(\Gamma_i) < 0 \\ o^+_{i,j}, & otherwise \end{cases} \tag{12}
$$

2.2 Extended Manipulability Measure

The matrices defined by Eq. 4 can be analyzed for manipulability in order to retrieve a quality index for a given configuration. Therefore, for a given configuration θ, all Jacobians are decomposed via Singular Value Decomposition (SVD) in order to retrieve the corresponding sets of singular values $\tilde{s_\Gamma}$. The extended manipulability is computed as:

$$
\tilde{c}_{ext} = \frac{min\{\tilde{s_\Gamma}\}}{max\{\tilde{s_\Gamma}\}}. \tag{13}
$$

The quality measure $\tilde{c}_{ext}$ implicitly considers redundancy, since a penalization due to joint limits can be compensated by redundant joints in the kinematic structure. The results of this measure applied to a 4 degrees of freedom (DoF) planar manipulator can be seen in Figure 2. The figure on the left shows a 2D visualization of the manipulability ellipsoid without considering joint limits, whereas the second visualization shows the scaled manipulability ellipsoids in each of the four quadrants. On the right, an obstacle is considered and the minimum distance is visualized. The according joint limit and obstacle penalization terms are applied as described in Eq. 4. Note, that only parts of the manipulability ellipsoids are visualized, so that one can see the penalization effects in the different quadrants. Due to this visualization technique, discontinuous transitions arise between the quadrants. Such artifacts only affect the visualization and have no influence on the computation, since for each quadrant (respectively hyperoctant in the general case) the whole manipulability ellipsoid is considered.

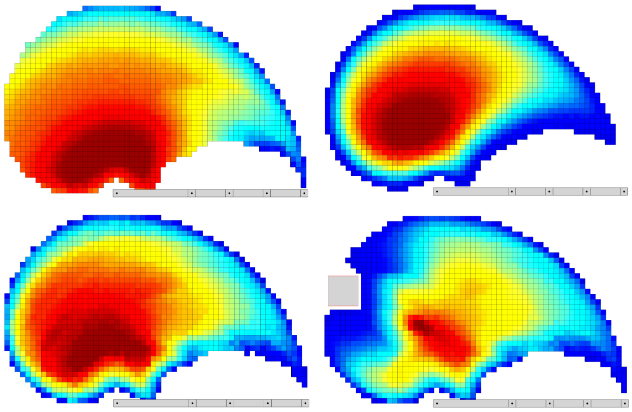

Fig. 3 A visualization of the SE(2) workspace of a 4 DoF planar manipulator that is analyzed by different approaches. The manipulability is encoded from red (high) to blue (low). Top row: (a) Yoshikawa's manipulability measure. (b) By applying a global penalization term, joint limits can be considered, but redundancy is not reflected by the measure. Bottom row:(c) The distribution of the extended manipulability measure $\tilde{c}_{ext}$. (d) The distribution of the extended manipulability measure when considering a workspace obstacle.

3 Manipulability Analysis and Representation

From reachability analysis, techniques are known to build a representation of the robot's operational workspace [11, 35, 41]. Usually these representations serve information about the reachable part of the Cartesian workspace that can be accessed by an end effector. By storing this information in the reachability map, a 6D voxel grid that covers position and orientation of the tool center point (TCP), the data can be accessed efficiently during online processing.

In [31], we showed how such discretized reachability data can be used to efficiently solve IK related tasks.

As we proposed in [33], such reachability maps can be extended in order to store extended quality information, i.e. manipulability data. This leads to a representation of the manipulability in workspace, which we introduced in [33] as manipulability distribution. The manipulability distribution is represented as a 6D voxel grid that holds quality information related to the manipulability that an end effector can achieve at the covered area of the workspace.

For building the representation of the manipulability in workspace, a 6D voxelized data structure has to

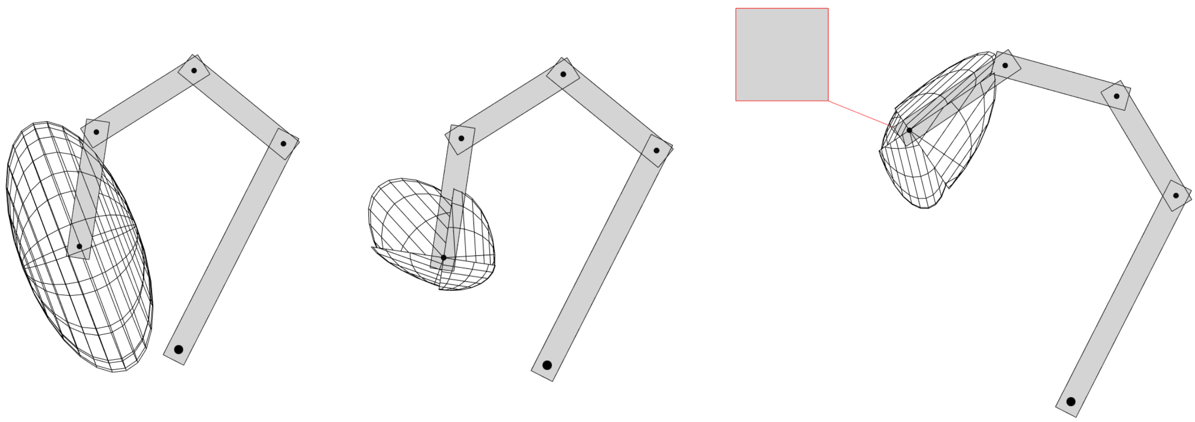

Fig. 2 (a) The basic manipulability ellipsoid without joint limit penalization ($c = 0.170$). (b) For each quadrant different penalization terms are applied according to the corresponding joint limit influence ($\tilde{c}_{ext} = 0.047$). (c) The obstacle influences the penalization terms ($\tilde{c}_{ext} = 0.067$).

be filled with reference manipulability values. Since the mapping between C-space and workspace is not unique for redundant manipulators, either the average manipulability or the lower respectively the upper bound of the achievable manipulability within a voxel can be stored. Hence, the manipulability data cannot be seen as an exact representation of the manipulators capabilities, but it can be used to efficiently serve an approximation of the expected manipulability of a given pose in workspace. This can be useful in several situations, e.g. when the selection of a grasp should be made and a large number of potential grasping poses are available. Further, the expected manipulability can help in choosing a suitable location for a hand-over process. As we showed in [32], robot placement problems for grasping and manipulation can be solved efficiently based on manipulability distributions.

To generate the manipulability representation, either work or joint space can be discretized and for each configuration the manipulability of the corresponding voxel can be updated. To speed up this process, a randomized approach can be used where a large set of joint configurations is sampled randomly. For each sample, the corresponding $\tilde{c}_{ext}$ value is computed and the 6D voxel v is determined by computing the forward kinematics of the manipulator in order to calculate the location of the end effector. If the actual entry of v is lower than $\tilde{c}_{ext}$, the value is updated. This leads to an upper-bound representation of the achievable manipulability which can be seen as an optimistic view on the robot's possibility to maneuver in workspace.

Figure 3 shows the visualization of several manipulability distributions of a 2D planar robot with 4 DoF. Note, that the workspace of the investigated manipulator is given by SE(2), the cross-space consisting of 2D position and 1D orientation. For visualization purposes the manipulability is depicted as a 2D grid, in which the maximum manipulability is shown for each position. The color indicates the magnitude of the achievable manipulability (blue:low, red:high).

On the top left the well known manipulability index c is used, whereas the top right figure shows the distribution when a penalization term according to Eq. 3 is applied to the manipulability computation. Since redundancy is not considered in Eq. 3, the resulting distribution gives only limited information about the achievable manipulability of a redundant manipulator.

When applying the proposed penalizations, the corresponding distribution in a 2D workspace can be seen in the bottom of Figure 3. The bottom left figure shows the result when joint limits are penalized according to Eq. 7. On the bottom right figure, the obstacle penalization (Eq. 12) is additionally applied and the combined Jacobians are analyzed as described in Section 2.1.

In Figure 4 a 3D visualization of the manipulability distributions of the kinematic chain covering three torso and seven arm joints of ARMAR-III [2] can be seen. The underlying manipulability measure incorporates penalizations due to joint limits and self distance as described in Section 2.1. The left figure shows a 3D visualization of the 6D distribution, which was generated by showing the orientation with maximum manipulability at each 3D position. On the right, a cut through the 6D distribution is shown.

4 Bimanual Manipulability

In this section we will discuss how suitable manipulability representations of bimanual robots can be generated. According to [9], the manipulability of multi-grasp setups with two or more end effectors cooperatively

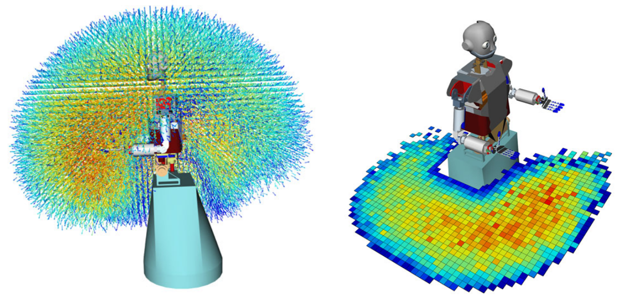

Fig. 4 (a) A 3D visualization of the 6D manipulability distribution of ARMAR-III's 10 DoF kinematic chain covering torso and right arm. (b) A cut through the manipulability distribution of the ARMAR-III's right manipulator (torso and arm). Joint limits and self-distance are considered by the underlying manipulability measure.

grasping an object, has to be computed separately for internal (relative) and external (absolute) forces. The internal forces give information about how much force can be applied by the grasp w.r.t. the other contacts. The external forces can be analyzed in order to retrieve information about how much force can be applied co-operatively in order to move the object. In this work we will only consider external forces, which give us information about the maneuverability of a grasped object in workspace.

Further, we use a simplified contact model by assuming point contacts at the TCP locations. Similar to the single handed case, we assume that the generated data structures serve hints for online processing components. The fact that inaccuracies arise in the calculation is compensated by the ability to quickly query a large subset of the workspace while assuming that the resulting solutions will be investigated by the corresponding online algorithms.

4.1 Extended Bimanual Manipulability

Manipulability analysis for multi arm systems with k manipulators can be performed by analyzing a composed Jacobian J_a that incorporates all involved joints and the grasp matrix [9]:

$$J_a^T = J_{1k}^T G^\dagger, \tag{14}$$

where $J_{1k} = diag(J_1 \cdots J_k)$ is the extended Jacobian that is constructed by combining the manipulator's Jacobian matrices J_i and $G^\dagger$ is the Pseudoinverse of the grasp matrix.

The grasp matrix G maps from contact to task space by converting the generalized contact forces $h = [h_1, \ldots, h_k]^T$ with $h_i = [f_i, \mu_i]^T$ (consisting of forces f_i and moments μ_i) to the net object force $h_a = [f_a, \mu_a]^T$:

$$h_a = Gh. \tag{15}$$

The grasp matrix is composed as $G = [G_1 \cdots G_k]$, where the grasp submatrices G_i are given by

$$G_i = \begin{bmatrix} I & O \\ R_i & I \end{bmatrix}. \tag{16}$$

The matrix R_i can be computed form the relation $r_i \times f_i = R_i f_i$, where r_i defines the distance vector of the i-th contact to the object's center of mass.

In the following, we will discuss how this general formalism can be applied for bimanual setups while incorporating the extended manipulability measure of Section 2. First the extended Jacobian $\tilde{J}_{1k}$ that considers joint limit and obstacle constraints is defined for the bimanual case:

$$\tilde{J}_{12} = \begin{bmatrix} \tilde{J}_1 & O \\ O & \tilde{J}_2 \end{bmatrix}. \tag{17}$$

Here, $\tilde{J}_1$ and $\tilde{J}_2$ are the augmented Jacobians for the left and the right manipulator as defined in Eq. 4.

Furthermore, the grasp matrix is constructed for two points as described above. At this point we do not have any real contact information, but we are interested in the achievable manipulability for a given setup. In order to generate comparable data, we assume a virtual object grasped with both hands and derive the center of mass as the center point between both TCPs. Since no grasp contact data is available, we simplify the contact information by considering both TCP positions as point contacts. With this information we are able to generate structural data that can be analyzed for bimanual capability representation. The final bimanual Jacobian

$$\tilde{J}_a = (\tilde{J}_{12}^T G^\dagger)^T \tag{18}$$

is processed as described in Section 2.2 in order to compute the extended bimanual manipulability $\tilde{c}_{ext}^{bi}$.

4.2 Bimanual Manipulability Representation

The single arm approaches presented in Section 3 can be extended in order to represent the bimanual manipulability, resulting in a twelve dimensional space that consists of the two 6 DoF pose spaces of both end effectors. Since the memory consumption of such a representation would be too high for fine grained approximations, this representation would only allow a coarse discretization. A simple calculation illustrates the exponential memory consumption: A $3m^3$ voxelized robot workspace with a discretization of $5cm$ and $20°$ results in $60 \times 60 \times 60 \times 18 \times 18 \times 18 \approx 1.3$ billion voxels. When storing 1 byte per voxel, the memory consumption exceeds 1 GB. Although the memory usage can be darstically reduced by compression and/or hierarchical representations, a 12D cross space would cover $1.6 \cdot 10^{18} \approx 1600$ petabyte, which cannot be represented efficiently.

To this end, we propose a different way of representing the bimanual manipulability of a dual arm system: Instead of lowering the pose discretization, which would make it difficult to compare different poses, our approach relies on reducing the encoded information about the orientational relation of both end effectors. This allows us to generate a 6D data structure with limited information about the relative end effector orientations but with fine grain position resolution. As stated in the introduction, this representation does not represent the accurate bimanual manipulability distribution, but it serves as an information source that can be efficiently queried during online processing in order to determine promising areas or sub sets of data structures (e.g. grasps) for further processing.

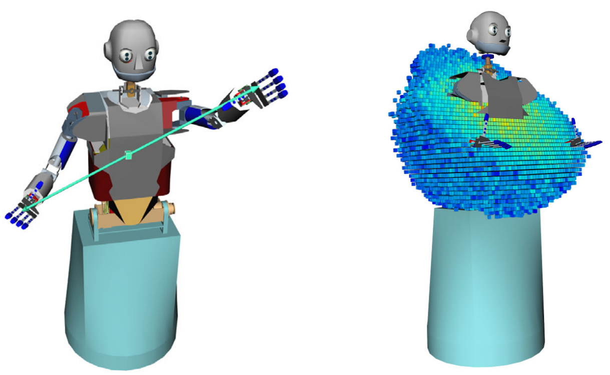

Fig. 5 (a) The voxel encodes the center between both end effectors and their relative position. (b) A cut through the 3D visualization of the bimanual manipulability distribution of ARMAR-III.

We propose to store an approximation of the spatial bimanual manipulability distribution consisting of a 3D position vector that describes the center between both end effectors in combination with a 3D spherical coordinate which encodes the relative position and distance of both end effectors. This results in a 6D description vector which can be stored efficiently, similar to the single arm case. Note, that this description does not serve any information about the end effector orientations, so that different oriented end effectors, which are located at the same position, result in the same voxel. A visualization of the voxel data can be seen in Figure 5(a). In Figure 5(b) the bimanual manipulability distribution of ARMAR-III is depicted. Similar to the single arm case (see Section 3), this spatial 6D grid is filled by determining the maximum $\tilde{c}^{bi}_{ext}$ for each voxel. Therefore, we need to compare the extended bimanual manipulability for all configurations of the robot which map into one voxel in order to determine the upper bound. As in the single arm case, we can use discretization or sampling techniques to approximate the manipulability distribution, which allows an efficient realization.

During buildup, self-collisions between the arms and the robot's body have been considered in order to discard any self-colliding configurations. In addition, the self-distance of end effector and body was considered to determine the penalization terms according to Section 2.1.2 in order to compute the extended Jacobians $\tilde{J}_1$ and $\tilde{J}_2$ as described in Section 4.1.

4.2.1 Data covered by the Bimanual Representations

To summarize the bimanual approach, a description of the bimanual data is given below:

a) End effector position
The 6D position vector v of a voxel indirectly encodes the relation of both end effectors. The positions of the TCPs can be reconstructed (w.r.t. discretization). Initially, the 3D position p of the voxel is determined:
$p = (v_0, v_1, v_2)$.

From the spherical coordinates $s = (r, \theta, \varphi) = (v_3, v_4, v_5)$ a Cartesian position $p' = (x', y', z')$ can be derived:
$x' = r \sin\theta \cos\varphi$, $y' = r \sin\theta \sin\varphi$, $z' = r \cos\theta$.

Finally, the Cartesian positions of the TCPs are
$TCP_{left} = p - p'$, $TCP_{right} = p + p'$.

b) Bimanual manipulability
The bimanual distribution's entry $\tilde{c}^{bi}_{v}$ at a given voxel v encodes the upper bound of the achievable bimanual manipulability for the encoded relation of both end effectors. Due to the voxelized data structure, this value can be efficiently queried for a given end effector posture in order to support online tasks. E.g. it is possible to quickly discard TCP postures and/or bimanual grasp setups when the corresponding quality entry is low, as shown in Section 5.2.

5 Evaluation and Applications

In this section we will evaluate the proposed method and several applications in the context of mobile manipulation will be presented.

5.1 Performance Evaluation

One of the main advantages of having available a representation of the robot's manipulability is the fact that online information retrieval can be efficiently realized by using offline structured data sources. In the following, we evaluate the effect of using precomputed manipulability representations in the context of grasp selection[1].

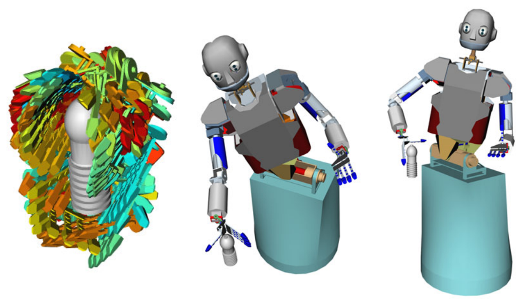

Fig. 6 (a) A subset of reachable grasps. The manipulability is encoded by color (blue:low, red:high). (b) A grasping configuration with $\tilde{c}_{ext} = 0.02$. (c) The optimal IK solution in terms of manipulability ($\tilde{c}_{ext} = 0.12$).

A model of a water bottle with 1000 precomputed grasps for the right end effector of ARMAR-III can be seen on the left of Figure 6. An exemplary subset of reachable grasps is depicted, whereas the manipulability is encoded by color (blue:low, red:high). The performance measurement is realized by placing the bottle randomly in front of ARMAR-III and evaluating several strategies for grasp selection.

In the first run (Table 1, row 1), no precomputed data is used and the IK problem is solved for all grasps for the kinematic chain covering torso and arm (10 DoF). In case an IK solution could be computed, the manipulability is determined for the corresponding configuration in order to determine the grasp with maximum $\tilde{c}_{ext}$.

The second approach (Table 1, row 2) incorporates reachability data in order to discard all grasps which are not reachable. Hence, a binary representation of the 6D reachability similar to the approaches of [11, 31, 41] is used. In such reachability representations the entry of a 6D voxel indicates whether the pose is reachable or not without providing any additional quality information. For the remaining sub-set of reachable grasps, the IK solutions are computed in order to determine the corresponding manipulability.

The third approach (Table 1, row 3) takes full advantage of the precomputed manipulability data[2]. Due to the efficient data structures an ordered set of reachable grasps with corresponding manipulability values can be retrieved in less than one millisecond and the IK problem has only to be solved for the first entry of this list.

Table 1 Comparison of different approaches for grasp selection (Average of 100 test runs).

	Overall	Manip. Access	IK Solving	Manip. Comp.
No precomp. data	9.97s	-	9.07s	0.90s
Reach. data	2.24s	<0.001s	1.75s	0.49s
Manip. data	0.01s	<0.001s	0.01s	-

5.2 Bimanual IK tasks

The bimanual manipulability distribution (see Section 4) can be used to efficiently solve bimanual IK queries for a humanoid robot. In the evaluated setup, an object for bimanual manipulation (a wok) is given with 100 predefined grasps for each hand. The IK task comprises the selection of a suitable bimanual grasp set together with a collision-free joint configuration for both arms. In case of ARMAR-III each arm covers 7 DoF, resulting in an 14 DoF IK problem.

As showed in [31], precomputed reachability information can be used to speed up IK-related queries for single arm and bimanual tasks. Although a similar approach for IK-solving is used in this work, we extended the quality measurement that is used for building the reachability representation. Instead of using a quality value that is proportional to the part of the configuration space (C-space) that maps to the corresponding volume in workspace, we use the extended manipulability information as described in this article. This measure produces a better representation of the robot's capabilities, since it serves elaborate information about the possibility to maneuver in workspace.

The bimanual IK solver is related to the approaches of [31], which we summarize briefly in the following.

– The input of the bimanual IK query consists of the current robot configuration and the pose of the target object. The object definition covers a set of precomputed grasps for each end effector (G_{left} and

[1] All performance evaluations have been carried out on a single core 3GHz Linux PC.

[2] The offline step of building the corresponding manipulability data took 6 hours on a standard Linux PC.

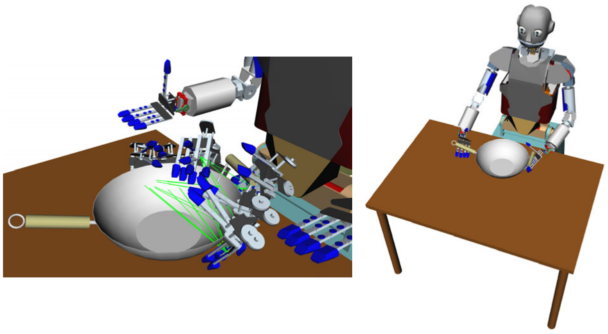

Fig. 7 (a) A subset of reachable grasps with potential bimanual combinations, indicated by green lines. (b) An exemplary 14 DoF IK solution.

G_{right}) and manipulability distributions for the left and right arm together with a bimanual representation (see Section 4.2) are assumed to be present.
- An IK query is processed, by first determining the reachable subset R_{left} and R_{right} of G_{left} and G_{right} at the current object location.
- Then, the bimanual manipulability distribution is queried in order to retrieve an ordered list of all reachable bimanual combinations, starting with the pair with highest bimanual manipulability.
- This list is processed until a pair of grasps is found for which both IK sub-queries for left and right arm can be solved. If the result is not in self-collision, a solution is reported.

For this evaluation, a wok with 100 predefined grasps for each hand is randomly placed in front of the robot (see Figure 7) and the bimanual IK solver is queried in order to determine a bimanual grasping configuration together with a collision-free 14 DoF IK solution. Figure 7(a) shows an intermediate result of the IK query process. All reachable grasps are shown together with a visualization of the reachable pairs (green lines). An exemplary IK solution is shown in Figure 7(b). The results of this evaluation are summarized in Table 2. The average runtime of a bimanual IK request is shown in the first column (20.2 ms), followed by the average runtime when an IK solution is available (20.8 ms). The last column shows the time that was needed by the algorithm to detect that no IK solution exists for the randomly placed object with the given grasps (18.2 ms).

Table 2 Results of the Bimanual IK tasks (Average of 100 test runs).

	Runtime Overall	Runtime IK found	Runtime no IK avail.
Bimanual Query	20.25 ms	20.83 ms	18.18 ms

5.3 Grasping Pipeline on ARMAR-III

Manipulability analysis is used to support the grasping and manipulation pipeline of the humanoid robot ARMAR-III. The overview of Figure 10 shows how several offline and online components play together in order to realize grasping and manipulation capabilities in a robust, generic and efficient way. As depicted in Figure 10, the following offline components are used to generate precomputed representations which are used to support online queries:

- **Grasp Planning** is used to automatically build object and hand related sets of feasible grasps. Based on an object's database entry, which serves 3D shape information, visual features and additional properties, the medial axis planner [24] is used to build sets of force closure grasps for both hands of ARMAR-III.
- **Hand Eye Calibration** is used to weaken errors that arise due to a misaligned internal robot representation. With the approaches of [30], the model of ARMAR-III's forward kinematics is learned and used during online control.
- **Manipulability Analysis** is performed as described in this article. The data structures, generated by this offline component, serve a manipulability distribution of the robot's 6D workspace. As discussed below, this data is available during online processing for efficient IK solving or grasp evaluation.

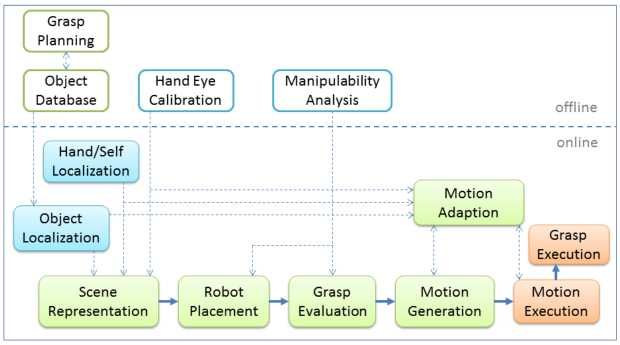

Fig. 10 The grasping and manipulation pipeline of ARMAR-III.

The discussed offline components are used to support online queries that are processed by several online processing stages, which are executed in order to efficiently serve a robust and generic robot skill that can be used for grasping known objects:

- **Localization** of hand and object(s) is performed by the vision system of ARMAR-III [4,5] supported by gaze selection strategies [39].

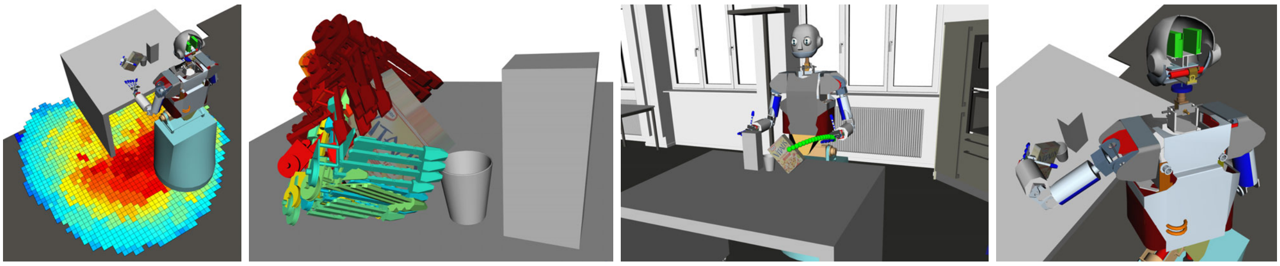

Fig. 8 Armar-III's grasping pipeline is supported by precomputed manipulability information. (a) The virtual representation of the scene and a 2D map of potential robot positions for grasping. (b) All reachable grasps are colorized according to their manipulability. (c) A collision-free grasping motion. (d) The target grasping pose.

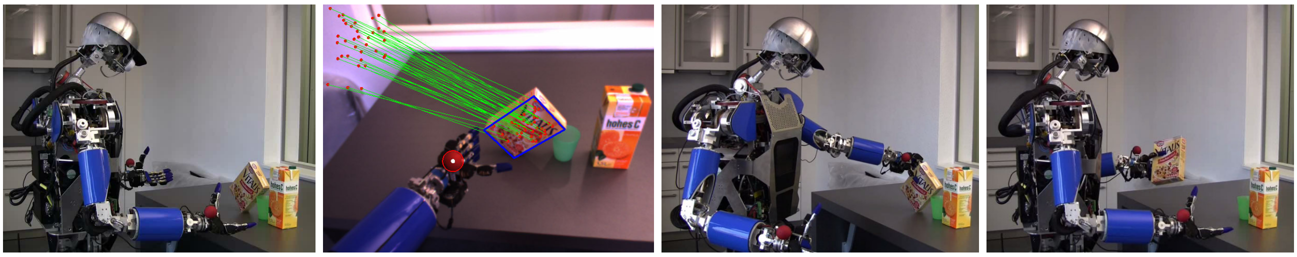

Fig. 9 The execution of the planned grasping motion is supported by ARMAR-III's visual servoing system.

– The **Robot Placement** problem has to be solved when the target object is too far away and no reachable grasp is available. In order to efficiently find suitable robot placements, the reachability inversion technique [32] is used. This approach takes advantage of precomputed manipulability distributions and generates manipulability-related maps in SE(2) which can be queried to find appropriate robot base poses (see Figure 8(a)).
– **Grasp Evaluation** is used for generating an ordered subset of reachable grasps. To this end, the manipulability distribution is queried for all available grasps which are stored in the object database. Since the manipulability data is organized as a 6D voxel grid, such queries can be realized as look-up table requests and hence, efficiency is provided even for large grasp sets. The manipulability query serves the reachability status together with a quality value that describes the manipulability of the target grasping pose. As shown in Figure 8(b), all reachable grasps can be ordered according to their manipulability in order to support the grasp selection process.
– The **Motion Generation** component generates a collision-free grasping motion for the hip-arm system of the robot. The set of reachable grasps is used as input for the IK-RRT [35] approach, which combines the IK search with the search for a collision-free motion. Since the planner relies on probabilistic techniques, the probability of selecting a target grasp is proportional to its manipulability value. The result of this planning step is a grasping motion as depicted in Figure 8(c). The corresponding target configuration is shown in Figure 8(d).
– **Motion and Grasp Execution** is provided by the control system of ARMAR-III [2].
– With **Motion Adaption** techniques, errors that are introduced by calibration imprecisions, inaccurate sensing and actuation or modeling errors, are reduced. The approach for multi-sensory motion adaption [37] is related to position-based visual servoing and continuously aligns the visually determined hand-object relation with the expected pose relation between hand and target object [34] (see Figure 9).

5.4 Manipulability Analysis of ARMAR-IV

We applied the proposed methods to a model of the humanoid robot ARMAR-IV [3] in order to gather information about the reachability of the arms and the possibility to manipulate with the end effectors. The visualization of the reachable workspace and the corresponding quality information as shown in Figure 11 was used to visualize and to analyze the effect of the robot's kinematic design and to investigate the influence of joint limits. Hence, manipulability analysis provided useful information during early stages of the robot design to support mechanical decision processes and for evaluating the performance of kinematic structures. In addition, the manipulability data is used to implement grasping and manipulation tasks on ARMAR-IV, similar to the proposed methods of Section 5.3.

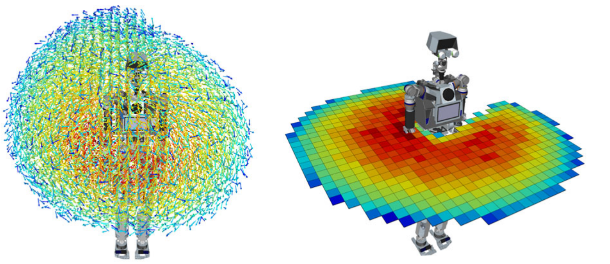

Fig. 11 The distribution of the manipulability of ARMAR-IV's right arm (8 DoF). The left image visualizes the end effector orientation with highest manipulability at each 3D position. A cut through the manipulability distribution is shown on the right.

6 Conclusion

With the proposed approaches, extended manipulability computations can be performed for redundant manipulators. We showed how constraining characteristics like joint limits or the robot's self distance can be incorporated into the manipulability measure, allowing to determine meaningful quality information about the possibility to maneuver a manipulator in workspace. Based on this measure, manipulability distributions are generated as comprehensive capability representations which allow to efficiently process online queries such as grasp evaluation or IK solving. We extended the approach for bimanual setups and showed how bimanual capability representations can be built. The performance of the proposed approaches have been evaluated in simulation and we discussed how manipulability analysis is used within the grasping pipeline of ARMAR-III. The kinematic capabilities of ARMAR-IV have been analyzed in order to investigate the effects of the robot's kinematic design on the reachability and dexterity of the manipulators. Further, we provide an open source implementation of the extended manipulability measure with the robot simulation toolbox Simox[3] [36].

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement no 611832 (WALK-MAN) and grant agreement no 611909 (KoroiBot).

[3] http://simox.sf.net

References

1. Abdel-Malek, K., Yu, W., Yang, J.: Placement of robot manipulators to maximize dexterity. International Journal of Robotics and Automation, vol. 19(1), pp. 6–14 (2004)
2. Asfour, T., Regenstein, K., Azad, P., Schröder, J., Bierbaum, A., Vahrenkamp, N., Dillmann, R.: Armar-III: An integrated humanoid platform for sensory-motor control. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006), pp. 169–175 (2006)
3. Asfour, T., Schill, J., Peters, H., Klas, C., Bücker, J., Sander, C., Schulz, S., Kargov, A., Werner, T., Bartenbach, V.: ARMAR-4: A 63 DOF Torque Controlled Humanoid Robot. In IEEE/RAS International Conference on Humanoid Robots (Humanoids), 2013
Asfour, T., Regenstein, K., Azad, P., Schröder, J., Bierbaum, A., Vahrenkamp, N., Dillmann, R.: Armar-III: An integrated humanoid platform for sensory-motor control. In: IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006), pp. 169–175 (2006)
4. Azad, P., Asfour, T., Dillmann, R.: Accurate shape-based 6-dof pose estimation of single-colored objects. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2690–2695 (2009)
5. Azad, P., Asfour, T., Dillmann, R.: Combining harris interest points and the sift descriptor for fast scale-invariant object recognition. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4275–4280 (2009)
6. Bicchi A., Prattichizzo D., Melchiorri C.: Force and Dynamic Manipulability for Cooperating Robot Systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1479–1484 (1997)
7. Chan, T.F., Dubey, R.: A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. Robotics and Automation, IEEE Transactions on, vol. 11(2), pp. 286–292 (1995).
8. Chen, H., Lee, S.I., Do, J.H., Lee, J.M.: Directional manipulability to improve performance index of dual arm manipulator for object grasping. In: S. Lee, H. Cho, K.J. Yoon, J. Lee (eds.) Intelligent Autonomous Systems 12, Advances in Intelligent Systems and Computing, vol. 193, pp. 595–602. Springer Berlin Heidelberg (2013).
9. Chiacchio, P., Chiaverini, S., Sciavicco, L., Siciliano, B.: Global task space manipulability ellipsoids for multiple-arm systems. Robotics and Automation, IEEE Transactions on, vol. 7(5), 678–685 (1991).
10. Dariush, B., Bin Hammam, G., Orin, D.: Constrained resolved acceleration control for humanoids. In: Intelligent Robots and Systems, 2010 IEEE/RSJ International Conference on, pp. 710–717 (2010).
11. Diankov, R.: Automated construction of robotic manipulation programs. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2010)
12. Finotello, R., Grasso, T., Rossi, G., Terribile, A.: Computation of kinetostatic performances of robot manipulators with polytopes. In: Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, vol. 4, pp. 3241–3246 (1998).
13. Flacco, F., Luca, A.D., Khatib, O.: Prioritized multi-task motion control of redundant robots under hard joint constraints. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 3970–3977. Algarve, Portugal (2012)
14. Jamone, L., Natale, L., Metta, G., Nori, F., Sandini, G.: Autonomous online learning of reaching behavior in

a humanoid robot. International Journal of Humanoid Robotics, vol. 09(03) (2012)
15. Jamone, L., Brandao M., Natale L., Hashimoto K.,Sandini G.,Takanishi A.:Autonomous online generation of a motor representation of the workspace for intelligent whole-body reaching. Robotics and Autonomous Systems , vol. 62(04), pp. "556–567 (2014)
16. Kokkinis, T., Paden, B.: Kinetostatic performance limits of cooperating robot manipulators using force-velocity polytopes. In: ASME Annual Meeting, Rootics, pp. 151–155 (1989)
17. Lee, J.: A study on the manipulability measures for robot manipulators. In: Intelligent Robots and Systems, 1997. IROS '97., Proceedings of the 1997 IEEE/RSJ International Conference on, vol. 3, pp. 1458–1465 (1997)
18. Lee, S.: Dual redundant arm configuration optimization with task-oriented dual arm manipulability. Robotics and Automation, IEEE Transactions on, vol. 5(1), pp. 78–97 (1989).
19. Madhani, A., Dubowsky, S.: The force workspace: A tool for the design and motion planning of multi–limb robotic systems. ASME Journal of Mechanical Design pp. 218–225 (1997)
20. Merlet, J.: Jacobian, manipulability, condition number and accuracy of parallel robots. In: S. Thrun, R. Brooks, H. Durrant-Whyte (eds.) Robotics Research, Springer Tracts in Advanced Robotics, vol. 28, pp. 175–184. Springer Berlin / Heidelberg (2007)
21. Nait-Chabane, K., Hoppenot, P., Colle, E.: Directional manipulability for motion coordination of an assistive mobile arm. In: ICINCO 2007, Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics, pp. 201–208 (2007)
22. Nakamura, Y., Nagai, K., Yoshikawa, T.: Mechanics of coordinative manipulation by multiple robotic mechanisms. In: Robotics and Automation. Proceedings. 1987 IEEE International Conference on, vol. 4, pp. 991–998 (1987)
23. Prattichizzo, D., Trinkle, J.C.: Grasping. In: Handbook of Robotics, pp. 671–700. Springer (2008)
24. Przybylski, M., Asfour, T., Dillmann, R.: Planning grasps for robotic hands using a novel object representation based on the medial axis transform. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 1781–1788 (2011)
25. Porges O., Stouraitis T., Borst C., Roa M.: Reachability and Capability Analysis for Manipulation Tasks, In: Advances in Intelligent Systems and Computing, vol. 253, pp. 703–718 (2014)
26. Siciliano, B., Khatib, O. (eds.): Springer Handbook of Robotics. Springer (2008)
27. Ting-Yung Wen, J., Wilfinger, L.: Kinematic manipulability of general constrained rigid multibody systems. Robotics and Automation, IEEE Transactions on, vol. 15(3), pp. 558–567 (1999)
28. Togai, M.: An application of the singular value decomposition to manipulability and sensitivity of industrial robots. SIAM Journal on Algebraic and Discrete Methods, vol. 7(2), pp. 315–320 (1986)
29. Tsai, M.J.: Workspace geometric characterization and manipulability of industrial robots. Ph.D. thesis, Ohio State University (1986)
30. Ulbrich, S., Ruiz, V., Asfour, T., Torras, C., Dillmann, R.: Kinematic bezier maps. IEEE Transactions on Systems, Man, and Cybernetics, vol. 42(4), pp. 1215–1230 (2012)
31. Vahrenkamp, N., Asfour, T., Dillmann, R.: Efficient inverse kinematics computation based on reachability analysis. International Journal of Humanoid Robotics, vol. 9(4) (2012)
32. Vahrenkamp, N., Asfour, T., Dillmann, R.: Robot placement based on reachability inversion. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 1970 – 1975 (2013)
33. Vahrenkamp, N., Asfour, T., Metta, G., Sandini, G., Dillmann, R.: Manipulability analysis. In: Proceedings of IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 568–573. Osaka, Japan (2012)
34. Vahrenkamp, N., Barski, A., Asfour, T., Dillmann, R.: Planning and execution of grasping motions on a humanoid robot. In: Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on, pp. 639–645 (2009)
35. Vahrenkamp, N., Berenson, D., Asfour, T., Kuffner, J., Dillmann, R.: Humanoid Motion Planning for Dual-Arm Manipulation and Re-Grasping Tasks. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 2464–2470. St. Louis, USA (2009)
36. Vahrenkamp, N., Kröhnert, M., Ulbrich, S., Asfour, T., Metta, G., Dillmann, R., Sandini, G.: Simox: A robotics toolbox for simulation, motion and grasp planning. In: International Conference on Intelligent Autonomous Systems (IAS) (2012)
37. Vahrenkamp, N., Wieland, S., Azad, P., Gonzalez, D., Asfour, T., Dillmann, R.: Visual servoing for humanoid grasping and manipulation tasks. In: Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on, pp. 406–412 (2008)
38. Watanabe, T.: Effect of torque-velocity relationship on manipulability for robot manipulators (2011)
39. Welke, K., Schiebener, D., Asfour, T., Dillmann, R.: Gaze selection during manipulation. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 652 – 659 (2013)
40. Yoshikawa, T.: Manipulability of robotic mechanisms. The International Journal of Robotics Research **4**(2), 3–9 (1985).
41. Zacharias, F., Borst, C., Hirzinger, G.: Capturing robot workspace structure: representing robot capabilities. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 3229–3236 (2007)
42. Zacharias, F., Leidner, D., Schmidt, F., Borst, C., Hirzinger, G.: Exploiting structure in two-armed manipulation tasks for humanoid robots. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 5446–5452 (2010)