

This is the author's copy of the publication as archived with the DLR's electronic library at <http://elib.dlr.de>. Please consult the original publication for citation.

Planning Realistic Force Interactions for Bimanual Grasping and Manipulation

A. M. Sundaram; O. Porges; M. A. Roa

Copyright Notice

©2016 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Citation Notice

```
@INPROCEEDINGS{sundaram2016planning,
  title={Planning Realistic Force Interactions for Bimanual Grasping and Manipulation},
  author={A. M. Sundaram and O. Porges and M. A. Roa},
  booktitle={2016 IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS)},
  pages={987-994},
  year={2016},
  month={Nov},
  organization={IEEE}}
```

Planning Realistic Interactions for Bimanual Grasping and Manipulation

Ashok M. Sundaram, Oliver Porges, Máximo A. Roa

Abstract—This work presents a dual arm grasp planning architecture that includes two relevant aspects often neglected: differences in hand actuation, and realistic forces applicable by the end effectors. The introduction of an actuation matrix allows considering differences in contact forces that can be generated between, for instance, a fully actuated and an underactuated hand. The consideration of realistic forces allows the computation of real magnitudes of forces and torques that can be resisted by the grasped object. The manipulability workspace can also be computed based on the capability maps, thus providing all the possible motions that can be imparted on the grasped object while respecting the dual hand grasp constraints. The joint consideration of these factors allow the selection of a good grasp for a desired bimanual manipulation.

I. INTRODUCTION

Robots can play a major role in areas like manufacturing, household service or healthcare. It is thus desirable to have robots able to work in human environments, rather than creating robot-specific environments. This has led to the development of anthropomorphic and dual-arm robots, many of them introduced in the last few years. Although dual arm grasping and dexterous manipulation have been intensively studied in the last years [1], the cognitive abilities for these robots are still behind their hardware capabilities.

Humans choose different manipulation strategies depending on the task, grasp requirements or environmental constraints. Finding the appropriate grasp for an object is a first step, commonly solved in robotics using a planner that relies on some grasp quality measure [2]. Traditional approaches for grasp analysis [3] usually rely on quality measures that help to compare grasps for the same object but are physically meaningless [4] due to considerations like unitary contact forces, linearization of friction cones, and lack of consideration of hand actuation and arm manipulability. The most common grasp quality measure is the largest perturbation wrench that can be resisted along any direction inside the Grasp Wrench Space, GWS (set of wrenches that can be applied on the object through forces at the contact points) [5]. For this computation, the contact forces are assumed to be of unitary magnitude. However, due to the joint and torque limits, and depending on the finger configuration, forces that can be produced by the robotic hand are bounded and change drastically within the workspace of the hand. A few works have considered more physically meaningful

measures. In [6], the grasp quality is calculated based on the distance between the Object Wrench Space, OWS (set of wrenches obtained by applying unitary forces at every possible contact point on the object surface) and the GWS. The GWS is constructed by considering realistic contact forces. A similar approach is taken in [7], but additionally dropping the linearization of the friction cone. The ray shooting algorithm proposed in [8] was extended in [9] to consider physically achievable contact forces for computing the grasp quality in a particular direction of interest.

For computing bimanual grasps, the simplest approach divides the object into two parts with a bisecting plane, and then grasps are synthesized separately for both parts [10]. Bimanual grasps can also be generated by considering all possible pairs of non-colliding grasps retrieved from the respective single hand grasp databases [11]. The grasp pairs can be further filtered based on the task or environmental constraints. For instance, in [12], the grasp synthesis is done for the first hand and then the grasp for the second hand is decided based on its reachability.

For planning bimanual grasps, a grasp quality measure should be physically meaningful and must help to answer: 1) when is a single hand grasp sufficient or a dual hand grasp required to successfully grab an object and resist the external disturbances produced by the task, and 2) which grasp amongst the set of available grasps is better, not only in terms of grasping with the hand but also in terms of reaching and manipulating the object with the arm. These questions are considered in this paper, and help to plan more realistic force interactions for grasping and manipulation tasks. To solve this problem, Section II presents the general architecture used for bimanual grasp planning. Section III describes the grasp analysis approach that considers realistic forces, and introduces the actuation matrix to account for the differences in the applicable forces, for instance, between a fully actuated and an underactuated hand. Section IV discusses the role of manipulability in the selection of a feasible grasp, and also proposes an online method to compute the manipulability workspace for a given bimanual grasp based on the capability maps (maps that represent the reachable workspace for a given robot). Finally, Section V concludes the paper.

II. BIMANUAL GRASP PLANNING ARCHITECTURE

The general architecture used in this work to simultaneously exploit the consideration of realistic forces and the manipulability analysis is presented in Fig 1. It consists of five major blocks: grasp synthesis, grasp analysis, manipulation analysis, grasp filtering and task-related information.

All the authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. Email: {ashok.meenakshisundaram, oliver.porges, maximo.roa}@dlr.de.

This work was partially funded through the project DPI2013-40882-P.

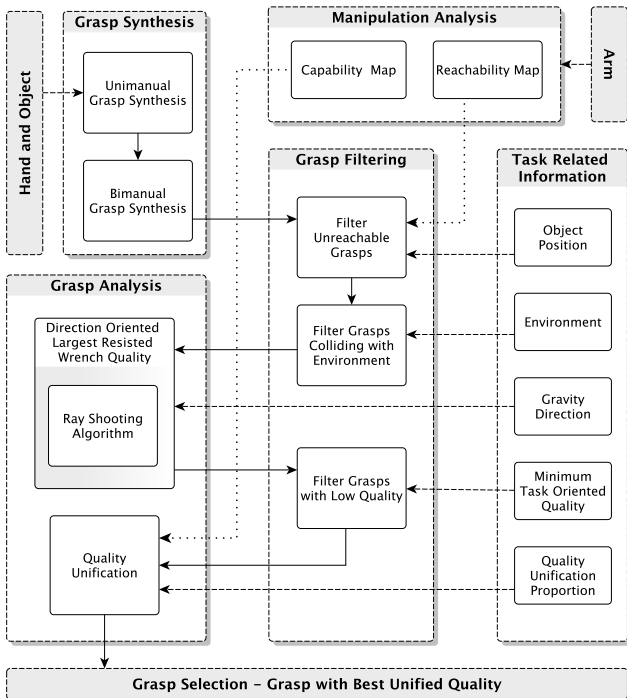


Fig. 1: Bimanual grasp planning architecture. Solid lines represent the flow of events using an updated grasp database. Dotted lines represent manipulability map information. Dashed lines represent robot and task related information.

The major components in each block are discussed below, in the order of execution, and the novel components are discussed in greater detail in the next sections.

- 1) *Unimanual grasp synthesis*: For a given hand and object, a set of force closure grasps is generated and stored in a database, using standard approaches [13].
- 2) *Bimanual grasp synthesis*: Based on the grasp database of two individual hands (right and left, in the case of anthropomorphic hands), a set of bimanual grasps that do not collide with each other is found and stored in the database.
- 3) *Filter unreachable grasps*: By accessing the object position and the arm reachability map, all unreachable grasps are filtered out from the database [14].
- 4) *Filter grasps colliding with environment*: Discards all grasps colliding with the surrounding environment (e.g. the supporting table).
- 5) *Direction-oriented largest resisted wrench*: Depending on the object and task, external disturbances might be expected along certain direction of interest in the wrench space (e.g. opposite to the drilling direction while using a driller). The realistic largest resisted wrench along that direction is computed using the enhanced ray-shooting algorithm (Section III-C).
- 6) *Filter grasps with low quality*: Discards all grasps that cannot resist the minimum required wrench perturbation specified by the task.
- 7) *Quality unification*: By using the arm capability map,

the reachability index for both arms and the bimanual reachability volume (Section IV-A) are obtained for each bimanual grasp. The grasp quality and the arm capabilities are then unified according to a given proportion for quality unification (Section IV-B). The grasp with best unified quality is selected for execution.

III. GRASP ANALYSIS

A widely used grasp quality measure is the computation of the largest minimum resisted wrench, Q^ϵ , in the GWS [5]; it indicates the magnitude of the maximum perturbation wrench that a grasp can counteract in any direction. To find the quality of grasps in certain task specific directions, or to optimize the forces in order to resist a certain wrench, it is only required to analyze the GWS in a particular direction; this quality is hereafter referred to as Q^α . This section revises the different ways of constructing the GWS and the corresponding Q^ϵ measures, proposes adequate considerations for taking into account the actuation of the robotic hand, and describes an enhanced ray-shooting algorithm to compute a physically meaningful measure Q^α for a given grasp.

A. Influence of hand actuation on the grasp quality

Computation of Q^ϵ

The GWS describes the wrenches that can be applied on the object through the contact points. To obtain those wrenches, the friction cone that describes the contact between finger and object is usually approximated with a polygonal pyramid of m edges. The *primitive wrenches* generated by the discretized forces in the friction cone are grouped in a wrench set $\mathbf{w}_i^{d,c} = [\mathbf{w}_{i,1}^T, \mathbf{w}_{i,2}^T, \dots, \mathbf{w}_{i,m}^T]$. All the primitive wrenches for a given grasp with n_c contacts are collected in the set $\mathcal{W}^c = [\mathbf{w}_1^{d,c}, \mathbf{w}_2^{d,c}, \dots, \mathbf{w}_{n_c}^{d,c}]$.

The boundaries of the GWS can be obtained by $GWS = CH(\mathcal{P})$, where CH denotes the convex hull operation over the set \mathcal{P} , which is constructed from the primitive wrenches according to certain assumptions on the contact forces. The Q^ϵ measure is then calculated as the radius of the largest ball centered at the origin and fully contained in the GWS. Using normalized contact forces, two ways of constructing the set \mathcal{P} were proposed in [5]:

- 1) Limiting the sum of all contact forces (GWS_S): the sum of modules of the forces applied by all the contacts has an upper limit, and the corresponding GWS_S is constructed by using $\mathcal{P} = \mathcal{W}^c$. This assumption corresponds to having a limited power source (actuator) for all the fingers. Although this is not the case for most designs of multi-fingered hands, the simplicity and fast computation of GWS_S has made it the most common method for computing grasp quality [2].
- 2) Limiting the individual contact forces (GWS_F): each contact force has an upper limit, and the set \mathcal{P} for constructing GWS_F is given by

$$\mathcal{P} = \bigcup_{i=1}^{n_c} \left(\bigcup_{j=1}^i C_{n_c} \left(\bigoplus \mathcal{W}_j^c \right) \right) \quad \text{with } \mathcal{W}_j^c \in \binom{\mathcal{W}^c}{i} \quad (1)$$

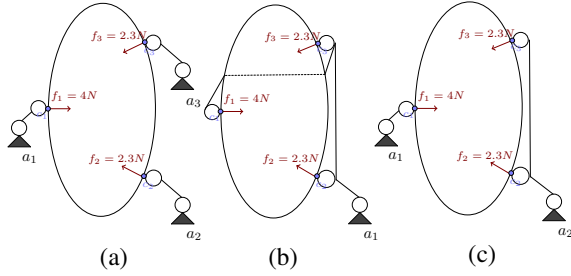


Fig. 2: Grasping an elliptical object with three contacts with the same forces and contact configuration and using: a) a fully actuated hand; b) an underactuated hand with a single actuator; c) an underactuated hand with two actuators.

where \mathcal{W}_j^c represents the j^{th} combination set of \mathcal{W}^c in the i^{th} iteration. This assumption corresponds to a limited independent power source for each finger. This is a more realistic assumption for the case of multifingered hands, as the individual contributions of different contact forces as well as their possible combinations are considered in (1). However, due to the Minkowski sum operation, the complexity is in the order of $O(m^{n_c})$, while the complexity for the previous case is $O(mn_c)$. As the CH computation will explode for a large number of contacts, this assumption is rarely used despite being physically more relevant.

Consideration of the hand actuation

In the case of multifingered underactuated hands, there can be several contact points in one finger that are controlled through a single actuator, or even all the finger contacts can be individually controlled by one actuator [15]. To consider this combination in the computation of GWS , an actuation matrix $\mathbf{M} \in \mathbb{R}^{n_a \times n_c}$ is defined for each grasp, where n_a is the number of actuators. With each row representing one actuator, the columns, i.e. the contacts that are controlled through that actuator are filled with a value of one, while all other columns are filled with zeros to indicate that those contacts are not influenced by that actuator. Fig. 2 shows for instance the same grasp on a 2D object, obtained with three hands with different modes of actuation. Actuation matrices for the three cases are, respectively,

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}; \mathbf{M} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}; \mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

By using the actuation matrix, the discretized wrenches affected by the same actuator are given by

$$\mathbf{w}_i^{d,a} = \bigcup_{j=1}^{n_c} \begin{cases} \mathbf{w}_j^{d,c} & \text{if } \mathbf{M}_{i,j} = 1 \\ \text{empty} & \text{otherwise} \end{cases} \quad (2)$$

The set of actuator-wise discretized contact wrenches is then $\mathcal{W}^a = [\mathbf{w}_1^{d,a}, \mathbf{w}_2^{d,a}, \dots, \mathbf{w}_{n_c}^{d,a}]$. The corresponding GWS_U can be constructed using the set \mathcal{P} given by

$$\mathcal{P} = \bigcup_{i=1}^{n_a} \left(\bigcup_{j=1}^{i C n_a} \left(\bigoplus \mathcal{W}_j^a \right) \right) \quad \text{with } \mathcal{W}_j^a \in \binom{\mathcal{W}^a}{i} \quad (3)$$

Note that while (1) uses \mathcal{W}^c with discretized contact wrenches, (3) uses \mathcal{W}^a consisting of actuator-wise discretized contact wrenches.

The computation of the quality measure Q^ϵ can be efficiently implemented with little variations over the incremental grasp wrench space algorithm proposed in [16], which already removes the need for a fixed discretization of the friction cone, and only considers the addition of primitive wrenches in the direction of the weakest facet of GWS . Fig. 3 shows the GWS and the computed quality Q^ϵ for the grasps depicted in Fig. 2. A traditional grasp analysis will lead to the GWS_S shown in Fig. 3b. Note that even if the same normal forces can be applied at the contacts in the three cases, the ability of the hand to apply these forces depending on its actuation has a large influence on the resulting grasp quality.

B. Realizable forces

Unitary forces at each contact point are traditionally assumed in the computation of grasp quality measures [5]. However, to obtain a realistic quality measure, the physically realizable magnitude of the force has to be considered for the construction of the GWS [6], [9].

The maximum realizable contact force by each finger can be obtained as follows. For each contact i , the unit contact normal force is transformed to the finger base coordinate frame, and represented by \mathbf{f}_i . The sum of all such contact forces generated by the finger j and expressed in the finger base coordinate frame, is given by \mathcal{F}_j . If $\mathbf{J}(\mathbf{q}_j)$ is the body Jacobian, with \mathbf{q}_j representing the joint states of the finger, then the torque $\boldsymbol{\tau}_j$ required to achieve \mathcal{F}_j is given by $\boldsymbol{\tau}_j = \mathbf{J}(\mathbf{q}_j)^T \mathcal{F}_j$. The resulting $\boldsymbol{\tau}_j$ is scaled by a factor k until one of the components reaches its corresponding torque limit. For each contact i , the force \mathbf{f}_i is multiplied by the scaling factor k and transformed back to the contact reference frame, to get the maximum realizable normal force. With the use of this realizable force for each contact, the calculated quality measure will represent the physical magnitude of wrench that can be counteracted by the grasp in any direction. It has to be noted that the calculated maximum contact force is the physical limit of the force, and the real force applied at each contact depends on the particular grasp situation.

C. Enhanced Ray Shooting Algorithm

Ray-shooting problem in OWS

The ray shooting problem, previously introduced in [8] for computation of grasp quality, is formalized as follows. For a compact set $\omega_i \in \mathcal{P}$ of points in \mathbb{R}^n , a point $\mathbf{r} \in \mathbb{R}^n$ (other than the origin $\mathbf{0}$), and a ray R originating from the origin and passing through the point \mathbf{r} , the ray shooting problem is defined as finding the intersection point \mathbf{s} between the ray R and the boundary of $CH(\mathcal{P})$.

If $CH(\mathcal{P})$ contains the origin, then based on duality considerations $CH(\mathcal{P})$ can be transformed into a convex polytope $CP(\mathcal{P})$. Each point $\omega_i \in \mathcal{P}$ can thus be represented as a hyperplane H

$$H : \omega_i X = \omega_{i,1}x_1 + \dots + \omega_{i,n}x_n = 1 \quad (4)$$

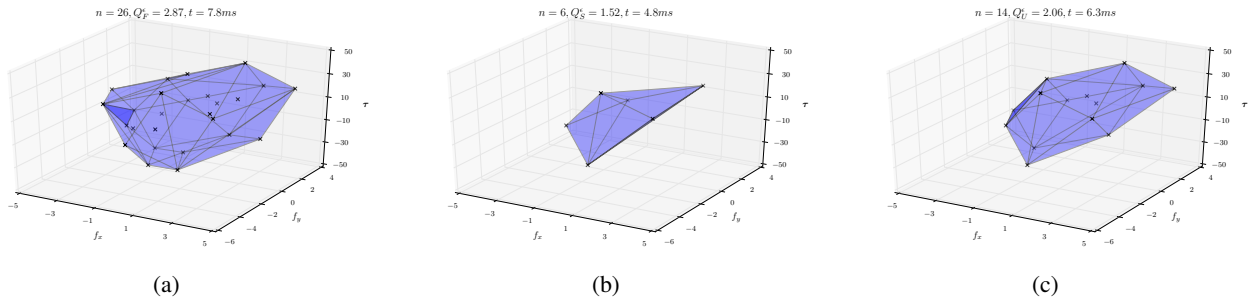


Fig. 3: Computation of GWS for the grasps shown in Fig. 2, with different type of actuation: a) GWS_F for the grasp in Fig. 2a; b) GWS_S for the grasp in Fig. 2b; c) GWS_U for the grasp in Fig. 2c. The number of wrench vectors n , quality Q^c , and computational time t are displayed. The weakest facet is highlighted in dark blue.

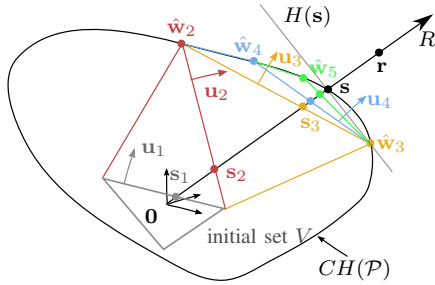


Fig. 4: Ray shooting algorithm. The different colors mark each iteration j towards the intersection point s . In each iteration, the normal \mathbf{u}_j defines the search direction used to find $\hat{\omega}_{j+1}$.

and $CP(\mathcal{P})$ is the intersection of all these finite number of halfspaces

$$CP(\mathcal{P}) : \quad \omega_i^T X \leq 1 \quad \forall \omega_i \in \mathcal{P} \quad (5)$$

By using the duality property, the ray shooting problem can be redefined as finding the hyperplane $H(\mathbf{s})$ such that (6) is satisfied.

$$\mathbf{s}^T X = 1 \quad (6)$$

The ray shooting algorithm proposed by [8] is based on this duality property (Fig. 4). It starts with a initial set $V = [\mathbf{a}_1, \dots, \mathbf{a}_{n+1}]$ containing $n+1$ affinely independent points, with n the dimension of the space, such that the origin is within its CH . In each iteration of the algorithm, it finds a point $\hat{\omega} \in CH(\mathcal{P})$ such that one of the facets of $CH(V)$ intersects with the ray R at a point s , which gets closer in every iteration to the farthest point of $CH(\mathcal{P})$ along R . Since the origin is within $CH(V)$, there exists a facet definitely intersected by the ray R . Also, $CH(V)$ forms a n -simplex with $n+1$ facets. Each facet can be represented as the CH of the set F_j with a subset of n points from the set V ,

$$\text{Facet}_j = CH(F_j) \quad (7)$$

$$F_j = [a_1, \dots, a_{j-1}, a_{j+1}, \dots, a_{n+1}] \quad (8)$$

Corresponding to each facet set F_j , using the duality property ((4) and (5)), the facet can be represented as a CP with n

hyperplanes

$$\mathbf{A}_j^T \cdot \mathbf{u}_j = [1 \ 1 \ \dots \ 1]^T \quad (9)$$

$$\mathbf{A}_j = [a_1 \ \dots \ a_{j-1} \ a_{j+1} \ \dots \ a_{n+1}] \in \mathbb{R}^{n \times n} \quad (10)$$

The point \mathbf{r} that defines the ray R can be expressed as a convex combination of the vertex points \mathbf{A}_j of each facet F_j

$$\mathbf{c}_j = \mathbf{A}_j^{-1} \mathbf{r} \quad (11)$$

The facet F_j intersects with the ray R at the point s_j if the following condition holds true,

$$s_j = \begin{cases} \frac{\mathbf{r}}{\sum \mathbf{c}_j} & \text{if } \min(\mathbf{c}_j) \geq 0 \\ \text{null} & \text{otherwise} \end{cases} \quad (12)$$

All the facets are iterated until the facet intersecting with the ray R is found. For the intersecting facet, by using the duality property again, the normal of the facet is found as

$$\mathbf{u}_j = (\mathbf{A}_j^T)^{-1} [1 \ 1 \ \dots \ 1]^T \quad (13)$$

Using this normal \mathbf{u}_j as the search direction, a new point $\hat{\omega}_{j+1}$ from the original set \mathcal{P} is found, such that the chosen point has the maximum projection \hat{p} in that direction. The initial set V is updated with the new point $\hat{\omega}_{j+1}$. The point \mathbf{a}_j that does not contribute to the facet F_j is removed from the original set V and the new point $\hat{\omega}_{j+1}$ is added,

$$V = F_j \cup \hat{\omega}_{j+1} \quad (14)$$

The iterative process continues until the exit condition is met:

$$\hat{p} - 1 \leq \epsilon \quad (15)$$

This can be interpreted as, the new point $\hat{\omega}_{j+1}$ lies on the plane described by the search direction \mathbf{u}_j , and therefore, the maximum projection is $\hat{p} = 1$. This ensures that the intersection point s_j is the farthest point possible on the set \mathcal{P} along the direction of the ray R . A very small threshold is denoted by ϵ .

On exit of the algorithm, the final farthest intersection point s corresponds to the Q^α measure. The computation does not require linearization of the friction cone [9]. In the discussion of the algorithm so far, the problem of finding the point $\hat{\omega}_{j+1} \in \mathcal{P}$ which has the maximum projection in the

search direction \mathbf{u}_j is not addressed yet. To find this point from the OWS, since only normal force is allowed at each contact point, the resulting projection is directly considered as the maximum projection for that particular point. To find the maximum projection point from the OWS [6], a support function $h_{\mathcal{P}}(\mathbf{u})$ is defined as

$$h_{\mathcal{P}}(\mathbf{u}) = \hat{p} = \max_{\hat{\omega} \in \mathcal{P}}(\mathbf{u}^T \hat{\omega}) \quad (16)$$

The set \mathcal{W} is revised to hold only the maximum projection wrench vectors from each contact, $\mathcal{W} = [\hat{\omega}_1, \hat{\omega}_2, \dots, \hat{\omega}_{n_c}]$. A new projection set \mathcal{W}^p is also introduced to hold the corresponding projection values $\mathcal{W}^p = [\hat{p}_1, \hat{p}_2, \dots, \hat{p}_{n_c}]$. Alg. 1 shows the pseudocode of the ray shooting algorithm in the OWS.

Algorithm 1 Ray shooting algorithm in OWS

Given: Object sampled with potential contacts c , each with a contact coordinate frame $\{\mathbf{C}\}$, defined by their surface normal and position \mathbf{p} , direction of interest R through the point \mathbf{r}

Output: Q_{OWS}^α measure

```

1: form initial set  $V$ 
2: repeat
3:    $j = 0$ 
4:   repeat
5:      $j = j+1$ 
6:     establish the facet  $\mathbf{A}_j$  (Eq. 10)
7:     find  $\mathbf{c}_j$  (Eq. 11)
8:   until ray  $R$  intersects with the facet (Eq. 12)
9:   find the normal of the facet  $\mathbf{u}_j$  (Eq. 13)
10:  for each contact  $c_i$  from the sampled object do
11:    consider unit contact normal force  $f_{i,n}$ 
12:    find the resulting wrench on the object
13:    find its projection along  $\mathbf{u}_j$ 
14:    save the projection in a set  $\mathcal{W}^p$ 
15:    save the wrench in a set  $\mathcal{W}$ 
16:  end for
17:   $\mathcal{P} = \mathcal{W}$ 
18:  find wrench  $\hat{\omega}$  with maximum projection (Eq. 16)
19:  update the set  $V$  (Eq. 14)
20: until  $\hat{p} - 1 \leq \epsilon$ 
21: return  $s$  as  $Q_{OWS}^\alpha$ 

```

Enhanced algorithm including hand actuation in GWS

In order to consider the hand actuation appropriately, the actuation matrix \mathbf{M} is used follows.

$$\mathcal{W}_i^{a,p} = \bigcup_{j=1}^{n_c} \begin{cases} \hat{p}_j & \text{if } \mathbf{M}_{i,j} = 1 \\ \text{empty} & \text{otherwise} \end{cases} \quad (17)$$

$$\mathcal{W}^a = [\hat{\omega}_1 \Rightarrow \max(\mathcal{W}_1^{a,p}), \dots, \hat{\omega}_{n_a} \Rightarrow \max(\mathcal{W}_{n_a}^{a,p})] \quad (18)$$

$$\mathcal{W}^{a,p} = [\max(\mathcal{W}_1^{a,p}), \dots, \max(\mathcal{W}_{n_a}^{a,p})] \quad (19)$$

$$\mathcal{V}^a = \bigcup_{i=1}^{n_c} \left(\bigcup_{j=1}^{i C_{n_c}} \left(\bigoplus \mathcal{W}_j^a \right) \right) \text{ with } \mathcal{W}_j^a \in \binom{\mathcal{W}^a}{i} \quad (20)$$

$$\mathcal{V}^{a,p} = \bigcup_{i=1}^{n_c} \left(\bigcup_{j=1}^{i C_{n_c}} \left(\bigoplus \mathcal{W}_j^{a,p} \right) \right) \text{ with } \mathcal{W}_j^{a,p} \in \binom{\mathcal{W}^{a,p}}{i} \quad (21)$$

The updated (18) and (19) hold the wrench with the maximum projection for each actuator. The contact forces are limited for each actuator individually using (20) and (21).

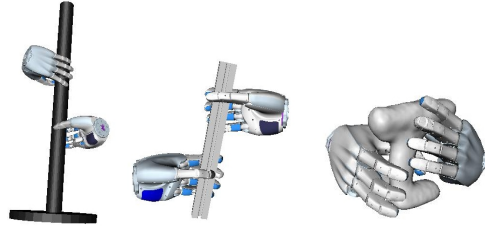


Fig. 5: Examples of bimanual grasps generated for different objects (tensator, profile, driller) and DLR HIT Hand II.

The final grasp quality is the distance between the intersection points of R with GWS and OWS. Alg. 2 shows the pseudocode of the enhanced algorithm.

Algorithm 2 Enhanced ray shooting algorithm

Given: Contacts c , with contact coordinate frame $\{\mathbf{C}\}$, defined by their surface normal and position \mathbf{p} , finger link associated with each contact, joint configuration of the hand along with the torque limits and kinematics, direction of interest R through the point \mathbf{r}

Output: Q^α measure

```

1: form initial set  $V$ 
2: define the actuation matrix  $\mathbf{M}$ 
3: repeat
4:    $j = 0$ 
5:   repeat
6:      $j = j+1$ 
7:     establish the facet  $\mathbf{A}_j$  (Eq. 10)
8:     find  $\mathbf{c}_j$  (Eq. 11)
9:   until ray  $R$  intersects with the facet (Eq. 12)
10:  find the normal of the facet  $\mathbf{u}_j$  (Eq. 13)
11:  for each contact  $c_i$  do
12:    consider the realizable contact normal force
13:    find maximum projection wrench  $\hat{\omega}_i$  along  $\mathbf{u}_j$ 
14:    save the maximum projection in a set  $\mathcal{W}^p$ 
15:    save the corresponding wrench in a set  $\mathcal{W}$ 
16:  end for
17:  collect all the actuator maximum projections  $\mathcal{W}^{a,p}$  (Eq. 19)
18:  collect all the corresponding wrenches  $\mathcal{W}^a$  (Eq. 18)
19:  limit individual actuator forces (Eq. 20 and 21)
20:   $\mathcal{P} = \mathcal{V}^a$ 
21:  find the overall maximum projection wrench  $\hat{\omega}$  (Eq. 16)
22:  update the set  $V$  (Eq. 14)
23: until  $\hat{p} - 1 \leq \epsilon$ 
24: save  $s$  as  $Q^\alpha$ 
25: find  $Q_{OWS}^\alpha$  (Algorithm 1)
26: return  $Q^\alpha - Q_{OWS}^\alpha$ 

```

D. Results

To illustrate the applicability of the algorithms above presented, we use three different objects: a tensator, aluminum profile and driller, shown with examples of bimanual grasps on Fig. 5. Fig. 6 shows a comparison of the Q^α measure when using normalized forces and when using the realizable forces, computed for the actuation specs of the DLR HIT hand II. This hand counts with 15 joints actuated via 12 motors (the two distal links in each finger are coupled). Suitable consideration of the actuation was used, according to Section III-A. Note that there are small differences in quality for 10 different bimanual grasps on the objects when

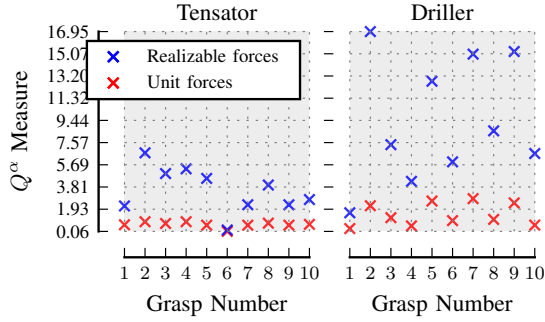


Fig. 6: Comparison of Q^α with and without the use of realizable forces, and employing the ray shooting algorithm. The quality is computed in the direction of gravity.

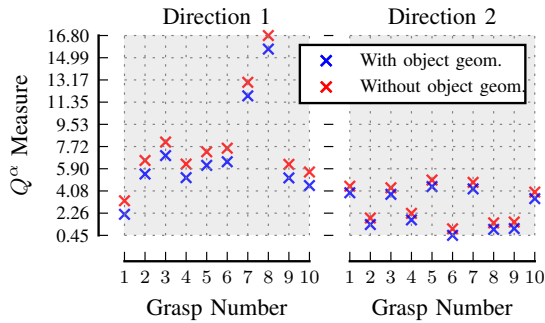


Fig. 7: Comparison of Q^α measure with and without the consideration of object geometry for the aluminum profile. The quality is computed in two different wrench directions.

using unitary forces; however, Q^α reveals greater differences in the amount of weight that the grasp can lift (Q^α is computed in the direction of gravity). Note that when the direction of wrench is not aligned with a pure force or a pure torque axis, the resultant magnitude combines units of force and torque, unless a suitable scaling factor is selected for the torque components; a deeper discussion on this issue is found in [17]. Fig. 7 compares the quality values obtained when considering OWS or not (i.e., when considering the influence of object geometry). Fig. 8 provides an idea of the average computational time for 10 different grasps computed on the three objects. The algorithm is fast enough for making real time computations of the realistic grasp quality.

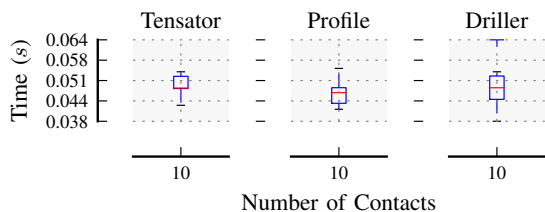


Fig. 8: Time required to compute Q^α using the ray shooting algorithm. The quality is computed in the direction of gravity.

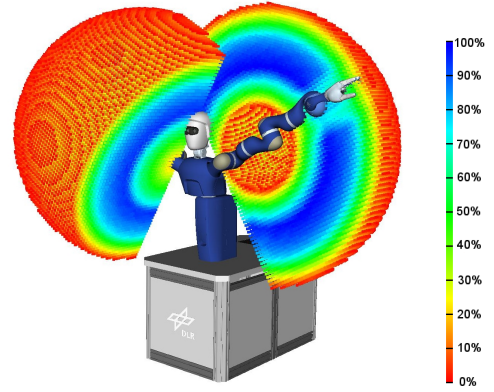


Fig. 9: The capability maps for the right and left arm of the SpaceJustin robot, dissected at different angles, are shown.

IV. BIMANUAL MANIPULATION ANALYSIS

Positioning of the hand is achieved by a robotic arm, therefore, we have to incorporate the arm capabilities into the grasp selection process. We used the Reachability [18], [19] and Capability maps in our previous works to filter feasible single handed grasps [14]. Reachability maps discretize the workspace around the robot into a 6D hierarchical structure that helps to identify reachable end-effector poses. By using the Reachability index to encode the arm's local dexterity into a scalar value, a Capability map is obtained. These maps are practical as on-line tools for quick reachability queries, and allow a comprehensive visualization of the workspace structures (Fig. 9). We extend here the use of Reachability maps for a multi-arm set up while maintaining computational efficiency.

A. Workspace model fusion

A bimanual grasp is defined as a relative pose of two end-effectors. This relative pose determines how large is the dual arm manipulation ability, i.e. the possible motions allowed by the closed kinematic chain. Our goal is to either maximize the manipulability volume or to verify the feasibility of a given trajectory. In contrast to [19], we do not need to precompute a new bimanual workspace model. The arms can have reconfigurable relative locations, which makes this method also suitable for collaborative mobile robot manipulation scenarios. More than two maps can be fused together, the only limit being the allowed computation time.

A Reachability map for each arm R_a is precomputed off-line, including collision avoidance with objects known at generation time (e.g. the robot body or mobile base). Fusion of the workspaces of two arms can be performed in a constrained $R_1 \cup_t R_2$ or unconstrained manner $R_1 \cup R_2$. The result of both operations is a new Reachability map, containing poses that the grasped object can reach. \cup_t only allows for translation of the object (for cases when orientations must be maintained, like moving a tray with both hands while carrying a beverage on the tray), while \cup also accounts for all possible rotations in space. Thus, we

can tell which grasp poses allow more freedom for object manipulation, or if a certain trajectory of the object is feasible or not using a particular grasp.

Algorithm 3 Constrained fusion of Reachability maps \cup_t

Given: Reachability maps R_1 and R_2 , hand poses G_1 and G_2

Output: Reachability map R_{nt}

```

1: for each Voxel  $V_i$  in  $R_1$  do
2:   Combine  $r(G_1)$  and  $t(V_i)$  into query  $q$ 
3:   if  $R_1$  contains  $q$  then
4:     Adjoint pose  $q_a = r(G_2) + t(G_2) - t(G_1)$ 
5:     if  $R_2$  contains  $q_a$  then
6:       Add  $q$  into  $R_{nt}$ 
7:     end if
8:   end if
9: end for
10: return  $R_{nt}$ 

```

Algorithm 4 Unconstrained fusion of Reachability maps \cup

Given: Reachability maps R_1 and R_2 , hand poses G_1 and G_2

Output: Reachability map R_n

```

1: for each Voxel  $V_i$  in  $R_1$  do
2:   for each Rotation bin  $r_j$  in  $V_i$  do
3:     Combine  $r_j$  and  $t(V_i)$  into query  $q$ 
4:     if  $R_1$  contains  $q$  then
5:       Adjoint pose  $q_a = q \times (G_2 - G_1)$ 
6:       if  $R_2$  contains  $q_a$  then
7:         Add  $q$  into  $R_n$ 
8:       end if
9:     end if
10:  end for
11: end for
12: return  $R_n$ 

```

Algorithms 3 and 4 describe the procedure of the fusion mechanism. The hand poses G_x can be represented by a homogeneous transformation matrix, where $r(G_x)$ is the rotational and $t(G_x)$ is the translational part. Note that for the constrained map, Alg. 3 allows only translational motions, but actually any constraint can be imposed on the grasped object and can be included here by suitable definition of the adjoint pose q_a . In both fusion cases, one traversal of the Reachability map is sufficient to create a new one. The manipulability volume R^v is then obtained by adding the volumes of occupied voxels in R_n or R_{nt} .

Fig. 10 and Fig. 11 show R_{nt} and R_n , respectively, for the tensor grasped case from Fig. 5 using SpaceJustin (Fig. 9). The figures depict the locations of the right hand frame that fulfill the grasp restriction with respect to the left hand. The reachable volume only for translations R_{nt} is $1.233m^3$ (Fig. 10), while the unconstrained reachable volume R_n is $2.149m^3$ (Fig. 11). Our implementation of Reachability maps typically handles around 3500 queries per millisecond. This implementation allows executing the presented fusion at reasonable timing of about 10 milliseconds for R_{nt} , and about 7 seconds for R_n on a single core processor, computed for a map of $5cm$ voxel size, 200 approach directions and 50 roll bins.

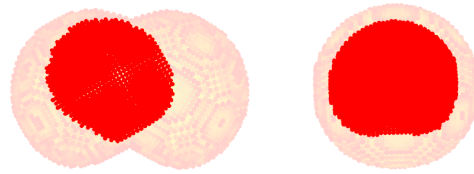


Fig. 10: Front and side view of R_{nt} , with original Reachability maps (with transparency) for scale reference.

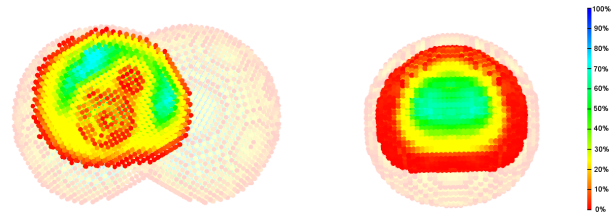


Fig. 11: Front and side view of R_n , with original Reachability maps (with transparency) for scale reference.

B. Unified Grasping-Manipulation Analysis

A unified grasping-manipulation quality is defined using a weighted proportion of the largest minimum resisted wrench (grasp quality) Q^α and the bimanual reachability volume R^v . First, to represent the two qualities in the same scale, both the grasp and the manipulation qualities are independently normalized across the grasp database (to standardize the range to $[0,1]$). The weighting factors (x_1 and x_2) can be decided based on the task to be executed, depending on whether it is more important to have maximum mobility of the grasped object, or higher robustness to perturbations.

$$U_i^Q = \frac{x_1 \cdot (Q_i^\alpha - \min(Q^\alpha))}{\max(Q^\alpha) - \min(Q^\alpha)} + \frac{x_2 \cdot (R_i^v - \min(R^v))}{\max(R^v) - \min(R^v)} \quad (22)$$

As an example for the tensor object, 10 random bimanual grasps are considered, and the different qualities are shown in Fig. 12. In the first row, the grasp quality Q^α (computed in the direction of gravity) and the bimanual reachability volume R^v are shown. According to these individual measures, grasps number 7 and 10 are found to be good for each individual criterion, respectively. After unifying these qualities in different proportions, the unified quality and the best grasp are shown in the second row.

V. FINAL DISCUSSION

This paper has presented new tools for including real limitations in hand actuation and specifications of task constraints in the planning process to obtain single and dual arm grasps. The assumptions commonly used in grasping literature for computing the grasp quality have been revisited. By introducing an actuation matrix, it is possible to consider different hand actuation mechanisms into the computation of the grasp quality. Moreover, considering the real limitations of finger actuators into the construction of the GWS leads to the computation of a physically meaningful quality measure.

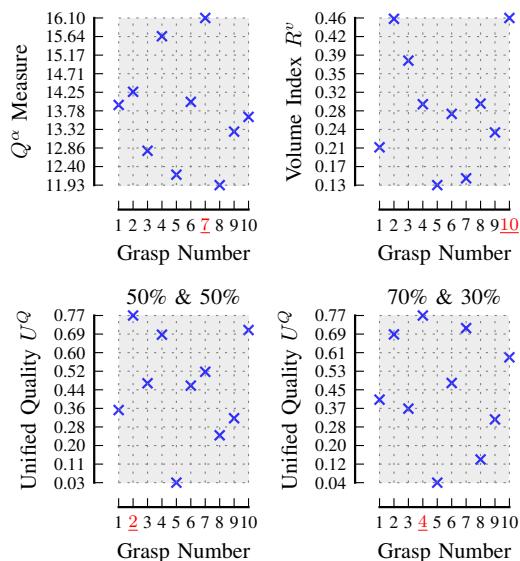


Fig. 12: Unified grasping-manipulation quality for different bimanual grasps of the tensor object. The original qualities for grasping and reachability are shown on the top. The unified qualities with two different proportions are shown in the bottom. The best grasp using each quality is highlighted on the x axis.

Additionally, two different modes of task constraints for bimanual grasps can now be considered at planning time. First, specifications on directions where wrenches are expected as potential perturbations can be verified using an enhanced ray-shooting approach to compute the grasp quality along the desired directions. This can be directly used to verify for instance if a given grasp can lift an object from a table, by computing the quality along the vertical axis. Second, when the object must be moved to a desired location, the feasibility of the motion, respecting the closed kinematic chain of the dual arm grasp, can be easily verified online through the computation of the bimanual manipulability space, based on the superposition of the capability maps for both arms.

With these considerations, the planning process can better reflect the restrictions of the real world application. As a next step, we want to verify the findings in the experimental setup with SpaceJustin (Fig. 13), including the validation of the physically meaningful grasp quality measure compared to the traditional one, and the verification of results for different fully actuated and underactuated hands.

ACKNOWLEDGMENT

The authors would like to thank Katharina Hertkorn (DLR) and Sven Schneider (Bonn-Rhein-Sieg University of Applied Sciences) for their support and advice for Ashok M. Sundaram during the course of this work.

REFERENCES

[1] C. Smith, Y. Karayiannidis, L. Nalpanitidis, X. Gratal, P. Qi, D. Dimarogonas, and D. Kragic, "Dual arm manipulation - a survey," *Robotics and Autonomous systems*, vol. 60, pp. 1340–1353, 2012.

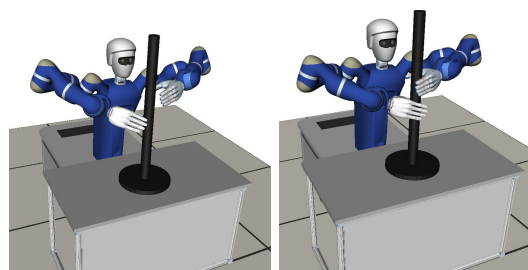


Fig. 13: SpaceJustin robot executing the planned grasp in the simulated environment based on OpenRave [20]. The pre-grasp and grasp configurations are shown.

[2] M. A. Roa and R. Suárez, "Grasp quality measures: Review and performance," *Autonomous Robots*, vol. 38, no. 1, pp. 65–88, 2015.

[3] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis - a survey," *IEEE Trans. Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[4] J. Kim, K. Iwamoto, J. Kuffner, Y. Ota, and N. Pollard, "Physically-based grasp quality evaluation under uncertainty," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012, pp. 3258–3263.

[5] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1992, pp. 2290–2295.

[6] H. Jeong and J. Cheong, "Evaluation of 3D grasps with physical interpretations using object wrench space," *Robotica*, vol. 30, no. 3, pp. 405–417, 2012.

[7] Y. Zheng and K. Yamane, "Evaluation of grasp force efficiency considering hand configuration and using novel generalized penetration distance algorithm," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2013, pp. 1572–1579.

[8] Y. Zheng, M. C. Lin, and D. Manocha, "A fast n-dimensional ray-shooting algorithm for grasping force optimization," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2010, pp. 1300–1305.

[9] K. Hertkorn, M. Roa, T. Wimboeck, and C. Borst, "Simultaneous and realistic contact and force planning in grasping," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 2291–2298.

[10] Z. Xue, A. Kasper, J. M. Zoellner, and R. Dillmann, "An automatic grasp planning system for service robots," in *Int. Conf. on Advanced Robotics*, 2009, pp. 1–6.

[11] D. Sidobre, X. Broquere, J. Mainprice, E. Burattini, A. Finzi, S. Rossi, and M. Staffa, "Human robot interaction," in *Advanced Bimanual Manipulation*. Springer, 2012, pp. 123–172.

[12] N. Vahrenkamp, M. Przybylski, T. Asfour, and R. Dillmann, "Bimanual grasp planning," in *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, 2011, pp. 493–499.

[13] M. A. Roa, M. Argus, D. Leidner, C. Borst, and G. Hirzinger, "Power grasp planning for anthropomorphic robot hands," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012, pp. 563–569.

[14] O. Porges, T. Stouraitis, C. Borst, and M. A. Roa, "Reachability and capability analysis for manipulation tasks," in *ROBOT2013: First Iberian Robotics Conference*. Springer, 2014, pp. 703–718.

[15] M. Pozzi, A. M. Sundaram, D. Praticchizzo, and M. A. Roa, "Grasp quality evaluation in underactuated robotic hands," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016.

[16] C. Borst, M. Fischer, and G. Hirzinger, "A fast and robust grasp planner for arbitrary 3D objects," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1999, pp. 1890–1896.

[17] M. A. Roa and R. Suárez, "Computation of independent contact regions for grasping 3D objects," *IEEE Trans. Robotics*, vol. 25, no. 4, pp. 839–850, 2009.

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

[19] N. Vahrenkamp and T. Asfour, "Representing the robots workspace through constrained manipulability analysis," *Autonomous Robots*, vol. 38, no. 1, pp. 17–30, 2015.

[20] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, 2010.