6

Efficient and Precise Grasp Planning for Real World Objects

Christoph Borst, Max Fischer, and Gerd Hirzinger*

Summary. With the development of flexible and highly integrated dexterous gripping devices (e.g. fig.6.1) the research results on grasp and manipulation planning can be applied to realize systems with autonomous grasping capabilities. The need for efficient methods to perform grasp analysis and planning for real world applications, therefore increases.

The so far proposed grasp planning and grasp evaluation methods made big contributions on the understanding of the structure of the grasping problem. However, not too many grasp planning systems are known that are able to cope with the constraints of planning grasps in reality, like short planning times, complex and incomplete object models and physical relevance of the planning results.

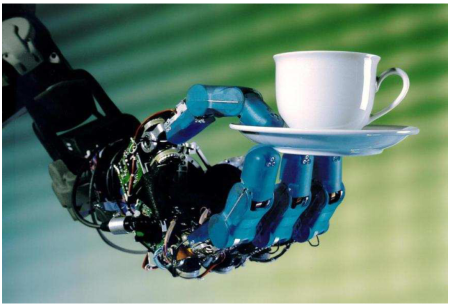

Fig. 6.1. DLR Hand II

In this chapter we summarize different grasp qualification methods and outline a physically well motivated grasp quality measure using wrench spaces. We present an algorithm, based on a physically well motivated grasp quality measure to qualify a given grasp with negligible approximation errors. Justified by statistic evaluations for some real world objects based on this grasp quality measure, we suggest a very effective generate and test grasp planner architecture. The proposed planner allows for planning high quality grasps for realistic object models extremely fast and thus can be used for online autonomous grasping systems.

6.1 Introduction

The main advantages of robotic hands compared to simple industrial grippers are their flexibility and enhanced grasping and manipulation potentials.

* C. Borst, M. Fischer and G. Hirzinger are with the Institute of Robotics and Mechatronics - German Aerospace Center (DLR), Wessling, Germany

F. Barbagli et al. (Eds.): Multi-point Interaction with Real and Virtual Objects, STAR 18, pp. 91–111, 2005.

Therefore many research activities have been carried out to design and develop new artificial hands. Also many aspects of grasp analysis, synthesis and control problems have been addressed. Nevertheless not too many complete grasp planning and grasp execution systems are known. This may result from the fact that the proposed approaches cannot be easily combined as the algorithms used are quite complex and there are rarely reference implementations available. Another issue is that many approaches need rich physical and geometric information about the object to grasp. Such information may not be available if it should be collected autonomously by the system.

For a grasp planning system to be used in an autonomous grasping scenario it is important to plan sufficiently good grasps in a limited processing time with imperfect input data. Especially the geometric and physical information about the objects to grasp may be only partial. These requirements have to be considered when determining which quality properties for a grasp should be considered in the planning system.

Informal, the main aspect for a planned grasp is to hold an object firmly and save, also in the presence of disturbances acting on the object. The analysis of the static qualities of a grasp represented by a set of contacts has been an intensively studied field in the past. As a basic quality criterion for a grasp, the force closure property has been identified. Efficient tests for this property have been developed using different modeling techniques, like the grasp matrix [29], the grasp wrench spaces [82, 171] or linear matrix inequalities [103]. Accordingly there are also many approaches dealing with the construction of a force closure grasp for 2D or 3D objects [197, 215, 225].

The force closure property, however, is only a minimal quality requirement for a grasp. Here unbound contact forces are assumed to balance disturbances in any direction. The magnitude of the contact forces that have to be applied are not considered. From a practical view it is more relevant, how efficiently a grasp can compensate for arbitrary disturbances or how efficiently it can balance a special set of disturbances that is expected when executing a desired task. To quantify this efficiency of a grasp, the concept of wrench spaces can be used. The set of all wrenches that can be applied to the object through the grasp contacts is called the *Grasp Wrench Space (GWS)*.

A commonly used and efficient way to approximate the GWS is to calculate the convex hull over the discretized friction cones [193, 197, 224]. The problem with all these approaches is the discretization of the friction cones, where significant errors may be introduced when approximating the cone with only a few vectors to achieve fast computation (e.g. 4 vectors lead to an error of $\sim$ 30 %, 8 vectors still $\sim$ 8 % [38]). Moreover, Teichmann and Mishra [273] showed that there are problems to be expected with this method for large friction coefficients. This GWS approximation corresponds to the idea that the sum of all applied forces is constrained to one, which has only a weak physical interpretation for multi-fingered grippers.

Ferrari and Canny [82] proposed a method for calculating a physically well interpretable GWS approximation, where the forces applied in the contact sum up to the number of contacts, by calculating the convex hull over the Minkowski sum of the friction cones. The drawback with directly calculating this GWS approximation is that it is computationally expensive. A more detailed discussion on the physical interpretation of these different GWS models follows in section 6.3.1.

To rate the quality of a grasp, task directed and task independent measures were introduced. Kirkpatrick et al. [144] use the largest wrench sphere that just fits within the GWS as a task independent quality measure of the grasp. The measure is not scale invariant and depends on the selection of the torque origin (**r** in fig. 6.2). To achieve invariance to the selection of the reference point, Li and Sastry [161] propose to use the volume of an ellipsoid generated by the grasp matrix as a measure of grasp quality. They also suggest to model the task wrench space as a six dimensional ellipsoid and fit it in the GWS. The problem with this approach is how to model the task ellipsoid for a given task, which they state to be quite complicated.

Pollard [224] introduces the Object Wrench Space (OWS) which incorporates the object geometry into the grasp evaluation. With her approach, however, this OWS represents an upper bound for the best grasp that can be achieved for an object with an infinite number of contacts. The OWS is not used to rate the quality of a grasp. Therefore the largest inscribed sphere is used. To achieve scale invariance, the torque component of the wrenches is scaled with the length of the longest object axis.

Strandberg [265] proposes to evaluate grasps using disturbance forces in order to overcome the problem of torque origin selection and to take the geometry of the object into account. The quality evaluation method is very reliable. However, the complexity of this approach is very high as the geometric information has to be evaluated for each grasp candidate.

A different model on which quality measures for grasps are based is the stiffness matrix. Approaches using this method combine the contact stiffness matrices to a grasp stiffness matrix [117]. The magnitudes of the eigenvalues of this matrix can be used to rate the quality of a grasp. However, the problem of choosing a reference frame also occurs here, as the magnitudes of the eigenvalues are not frame invariant. Therefore methods have been proposed to overcome this problem [46, 167].

In the next section we give a description of exact Grasp and Task Wrench Spaces that are physically motivated and propose a very intuitive grasp measure using these wrench space definitions. In section 6.3 we present a very efficient algorithm to calculate this measure. Based on some statistical evaluations with this quality measure we propose in section 6.6 a generate and test planner to plan sufficiently good grasps for real world objects in very short time.

6.2 A Physically Motivated Specification of Properties, Requirements and the Quality of Grasps

There are three main questions regarding the static part of grasping: What are the forces/torques that can be applied to the object by the grasp? Which disturbances are expected to act on the object? The third question is about the quality of the chosen grasp. A good quality measure for a grasp is a scalar that describes how well the grasp can resist the expected disturbances.

For all three questions above various models and measures have been developed. From a physical or mechanical point of view, however, all can be modeled similarly and in a simple manner.

There are only forces and torques acting on the object, either as a disturbance anywhere on the object or in the grasp contacts to counteract the disturbances. Both, the set of disturbance forces/torques and the set of possible grasp forces/torques, are usually represented in a vector space [82, 197, 224].

6.2.1 The Grasp Wrench Space

Let us assume that the grasp consists of k point contacts with friction. So in each contact a force within the friction cone can be applied to the object (Fig. 6.2).

The length of the applied force vector is normalized to a unit force as we assume that each finger can apply the same magnitude of force and only one contact arises for each finger (precision grasp).

The direction of the force $\mathbf{f}_i$ that can be applied at contact point $\mathbf{c}_i$ is constrained by the friction cone specified by the friction coefficient μ, the contact point $\mathbf{c}_i$, and the contact normal $\mathbf{n}_i$. The constraint can be written as:

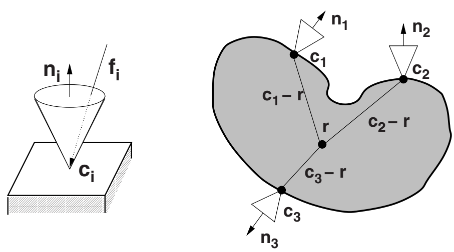

Fig. 6.2. A single contact point in 3D illustrating the friction cone and a sample k-contact grasp (k=3) on a planar object.

$$\|\mathbf{f}_i - (\mathbf{f}_i \dot{\mathbf{c}} \mathbf{n}_i)\mathbf{n}_i\| \quad \leqslant \quad -\mu(\mathbf{f}_i \dot{\mathbf{c}} \mathbf{n}_i). \tag{6.1}$$

Any force acting at a contact point on the object also creates a torque relative to a reference point $\mathbf{r}$ that can be arbitrarily chosen. Often the center of mass is used as that reference point to give it a physical meaning. The torque τ_i corresponding to $\mathbf{f}_i$ is then $\tau_i = (\mathbf{c}_i - \mathbf{r}) \times \mathbf{f}_i$. For convenience, these force and torque vectors can be concatenated to a *wrench* :

$$\mathbf{w_i} = \begin{pmatrix} \mathbf{f}_i \\ \tau_i \end{pmatrix} = \begin{pmatrix} \mathbf{f}_i \\ (\mathbf{c}_i - \mathbf{r}) \times \mathbf{f}_i \end{pmatrix}$$

Next we specify the set of wrenches that can be created by friction cone unit forces acting in one contact. We call this set the *Cone Wrench Space (CWS)*. It is used to clarify the construction of the Grasp Wrench Space.

$$\mathrm{CWS}_{\mathbf{c}_i} = \left\{ \mathbf{w_i} \,\middle|\, \mathbf{w_i} = \begin{pmatrix} \mathbf{f}_i \\ \tau_i \end{pmatrix} \wedge \quad \|\mathbf{f}_i - (\mathbf{f}_i \dot{\mathbf{c}} \mathbf{n}_i)\mathbf{n}_i\| \leqslant -\mu(\mathbf{f}_i \dot{\mathbf{c}} \mathbf{n}_i) \wedge \|\mathbf{f}_i\| \leqslant 1 \right\} \tag{6.2}$$

The *Grasp Wrench Space (GWS)* should contain all wrenches that a given grasp can counterbalance by applying forces in its k contacts. This set can be described as all wrenches that can be composed by summation of a single wrench for each contact:

$$\mathrm{GWS} = \left\{ \mathbf{w} \,\middle|\, \mathbf{w} = \sum_{\mathbf{i=1}}^{\mathbf{k}} \mathbf{w_i} \quad \wedge \quad \mathbf{w_i} \in \mathrm{CWS}_{\mathbf{c}_i} \right\} \tag{6.3}$$

Note that equation 6.3 is an exact description of the GWS. It corresponds to the idea that each finger of the manipulator is capable to exert a unit magnitude of force to the object. The only drawback of this kind of definition is that it is only descriptive but not constructive. To find the linear combination of finger forces to counterbalance a disturbance wrench is difficult. This problem is addressed in section 6.3.

6.2.2 The Task Wrench Space

The wrenches that are expected to occur for a given task can be specified as a so-called *Task Wrench Space (TWS)*. For the TWS two cases can be distinguished: Either the task to be executed is known and a specification in the wrench space is given or the task is unknown and no specification exists.

Given Task Specification

If there is a detailed description of the task given by a set of wrenches that are applied to the object during the manipulation one can use the convex hull over these task disturbance wrenches as a Task Wrench Space. Li and Sastry [161] propose to approximate the task wrench space by a task ellipsoid.

Unknown Task Specification

If one knows nothing about a grasping task, one at least can assume that a grasp should hold an object (1) against gravity, (2) against forces and torques arising from accelerating the object and (3) against forces that result from contacts of the object with the environment.

A commonly used approach to model an unknown task wrench space is to use a unit sphere in the wrench space. With this approach it is assumed that the probability for every wrench direction to occur as a disturbance is equal. However, this has no physical or mechanical interpretation. Torques are typically caused by forces acting on the boundary of the object and therefore a general task wrench space is not uniform for most objects.

A more natural way to describe an unknown TWS that takes the object geometry into acount is the *Object Wrench Space (OWS)* as introduced by Pollard [224].

6.2.3 The Object Wrench Space

The OWS should contain any wrench that can be created by a distribution of n disturbance forces acting anywhere on the surface of the object. As we are interested in the effect of a normalized disturbance on the object, the sum of the length of all n forces should be 1. The number of forces that act on the object is unlimited (so $n \in \{1..\infty\}$; see fig. 6.3 for illustration). The OWS can again be composed of the union of cone wrench spaces, in the following way:

$$\mathrm{OWS} = \left\{ \mathbf{w} \middle| \mathbf{w} = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \alpha_{\mathbf{i}} \mathbf{w}_{\mathbf{i}} \quad \wedge \sum_{i=1}^{n} \alpha_i = 1 \wedge \mathbf{w}_i \in \mathrm{CWS}_i \wedge n \in \{1..\infty\} \right\} \tag{6.4}$$

This description represents the resulting wrenches of any possible disturbance of a certain magnitude that act on the surface. To add gravity, which acts not on the surface but in the center of mass, one can merge this OWS with the wrench space that is produced by forces of any direction acting in the center of mass. If the reference point $\mathbf{r}$ is equal to the center of mass, then one merges a sphere in the force domain to the OWS, scaled with the mass of the object. For the general case, the "mass wrench space" (MWS) generated by gravity $\mathbf{g}$ acting in any direction in the center of mass ($\mathbf{m}$) (dependent on the object rotation) can be written as:

$$\mathrm{MWS} = \left\{ \mathbf{w} \middle| \mathbf{w} = \begin{pmatrix} \mathbf{f} \\ (\mathbf{m} - \mathbf{r}) \times \mathbf{f} \end{pmatrix} \quad \wedge \|\mathbf{f}\| \leqslant \mathbf{m}\dot{\mathbf{c}}\mathbf{g} \right\} \tag{6.5}$$

Such an TWS = OWS $\cup$ MWS describes the general case where nothing about the task is known very naturally, as any possible disturbance and also gravity is represented. The drawback is again that there is no constructive description to calculate the set.

6.2.4 The Physically Motivated Grasp Quality Measure QM_{BF}

As stated in the introduction, many different metrics for grasp quality have been introduced but almost all of them have drawbacks that arise from the different units or scaling in the force/torque dimensions in the grasp wrench space or are even dependent on the selection of the reference point. With the OWS defined above, we can propose a quality metric that overcomes all

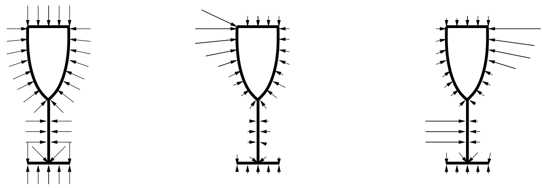

Fig. 6.3. Illustration of different force distributions that produce the wrench set of the OWS. Each distribution contributes one single wrench to the OWS set. The length of all force vectors sum to the unit length

these drawbacks and rates grasps in a physically interpretable and intuitive way. Of course, the non-uniformity of the wrench space remains. However, our concept generates a physically interpretable scaling between forces and torques automatically.

We take the ability of a grasp to counteract the possible disturbances on an object as a measure of the grasp quality. The Grasp Wrench Space (GWS) of a given grasp $\mathbf{C}_k$ represents the capabilities of the grasp, while the Object Wrench Space (OWS) of a given object $\mathbf{o}$ defines which disturbances may occur. So the largest factor by which we can scale the OWS to just fit in the GWS gives us a measure of the grasp quality. Formally expressed, we get

$$\mathrm{QMS}_{\mathrm{BF}}(\mathbf{C}_k, \mathbf{o}) = \Big(k; k \rightarrow \mathrm{Max} \,|\, \forall \mathbf{x} \in \mathrm{OWS}_{\mathbf{o}} : k\dot{\mathbf{c}}\mathbf{x} \in \mathrm{GWS}_{\mathbf{C}_k}\Big) \quad (6.6)$$

With this measure, we are independent of the selection of the reference point as we use it for the creation of both wrench spaces. Moving the reference point can be expressed by a linear transformation that is norm conserving and so it takes no effect on the scaling factor.

It should be noted that the above descriptions are an exact view on the static grasp situation which is the base for the grasp planning problem that we want to address with our grasp quality measure. The question of how to control the forces exerted by the manipulator to resist certain disturbances in a real dynamic grasp situation which is known as *force optimization problem* (see [103]) is not addressed. For a review on algorithms solving these problems we refer to [172].

The problem with our described measure is to find a way to efficiently calculate the scaling factor of the task wrench space. This problem is considered in the following sections.

6.3 An Efficient Algorithm to Quantify Grasp Quality

The problems to calculate the above described wrench spaces derive from their definitions as infinite wrench sets. The major problem is the non-linearity of the exact friction cone description. With an implicit description as above, the summation for assembling the Grasp and Object Wrench Space (eq. 6.3, 6.4) cannot be done directly. A widely used method to overcome this problem is the linearization of the friction cones (fig. 6.4). This allows for an easy construction of the wrench spaces, however, a large approximation error may be introduced if the friction cone is approximated with only a few vectors. On the other hand, a proper linearization with many vectors may considerably increase the problems dimension as we will show in the following.

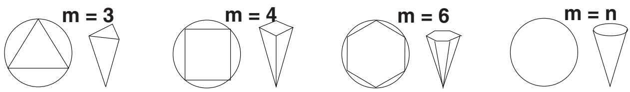

Fig. 6.4. The linearization of friction cones with 3, 4, 6 and infinite vectors.

6.3.1 Grasp Wrench Space Approximation Through Convex Hulls

Let's now assume that the friction cone of contact c_i is linearized with m vectors $\{\mathbf{f}_1, \mathbf{f}_2, .., \mathbf{f}_m\}_{c_i}$ and the corresponding wrench vectors $\{\mathbf{w}_1, \mathbf{w}_2, .., \mathbf{w}_m\}_{c_i}$. The discretized CWS (*DCWS*) (equivalent to the CWS defined above eq. 6.1) of a contact point $\mathbf{c}_i$ can then be written as any linear combination of friction cone wrench vectors:

$$\mathrm{DCWS}_{\mathbf{c_i}} = \left\{ \mathbf{w} \middle| \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i \mathbf{w}_i \quad \wedge \quad \sum_{i=1}^{m} \alpha_i \leqslant 1 \right\}. \tag{6.7}$$

Note that the boundary of this DCWS is equivalent to the convex hull over the friction cone spanning wrenches.

The discretized GWS (*DGWS*) for n contacts can then be written as:

$$\mathrm{DGWS} = \left\{ \mathbf{w} \middle| \mathbf{w} = \sum_{i=1}^{n} \mathbf{w}_i \quad \wedge \quad \mathbf{w}_i \in \mathrm{LCWS}_{c_i} \right\} \tag{6.8}$$

Ferrari and Canny [82] proposed two methods to calculate the boundary of such a discretized GWS with convex hulls. The simpler method, which we define as DGWS_{Q1} just calculates the convex hull over the wrench vectors of all contacts and thus can be written as:

$$\begin{aligned} \mathrm{DGWS}_{Q1} &= ConvHull\left(\bigcup_{i=1}^{n} \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m\}_{c_i}\right) \qquad (6.9) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_{i,j} \mathbf{w}_{i,j} \quad \text{with} \quad \sum_{i=1}^{n}\sum_{j=1}^{m} \alpha_{i,j} = 1. \end{aligned}$$

One can easily see that this does not correspond to our GWS definition. The resulting wrench is linear combined from contact wrenches created by unit forces. The parameters $\alpha_{i,j}$ limit the sum of the generating contact forces to the unit force. For a gripper with n fingers that means that all fingers together exert a unit force on the object which can be seen as a bound for the engergy consumption of the gripper. For a n-contact grasp our GWS definition limits the sum of the generating forces to n unit forces. This corresponds to the more relevant case where the force that each finger can exert on the object is bound to a unit force.

Ferrari and Canny also proposed a calculation method that deals with the second case. The DGWS_{Q2} is the convex hull over the Minkowski sum over all contact wrench vectors. The idea to construct the hull is: There is a convex wrench space that is spanned by each contact and the whole GWS is the sum of all these spaces. The sum sets can be constructed by the Minkowski sum and for the special case of convex sets the convex hull can be calculated before or after the Minkowski summation. So the DGWS_{Q2} can be written as:

$$\begin{aligned} \text{DGWS}_{Q2} &= ConvHull(\bigoplus_{i=1}^{n}\{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m\}_{c_i}) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{i,j}\mathbf{w}_{i,j} \quad \text{with} \quad \forall i : \sum_{j=1}^{m}\alpha_{i,j} = 1. \end{aligned} \tag{6.10}$$

Although the second method is physically more relevant it is almost never used to calculate a GWS or to evaluate the quality of a grasp. The reason for this is, that the Minkowski sum operation increases the number of vectors to process exponentially in the number of contacts. To calculate DGWS_{Q1} (simple case) for a grasp with 4 contacts where each friction cone is linearized by e.g. 10 vectors one has to calculate the convex hull over $4 \times 10 = 40$ vectors. In the second, more relevant case one has to calculate the convex hull over $10^4 = 10000$ vectors.

The calculation of the whole DGWS_{Q2} is complex and hard to simplify. However, for the calculation of quality measures, that are based on the determination of the direction of the smallest wrench magnintude in DGWS ("weakest" wrench direction) the computational cost can be reduced.

6.3.2 Evaluating the Grasp Quality Using Convex Hulls

In section 6.2 we defined a grasp quality measure with respect to a GWS and a required set of wrenches, represented either by a TWS or by a more general OWS. Now we want to show that we can also use the representation of convex hulls to efficiently calculate our proposed quality measure.

The representation of the Grasp Wrench Space is clear. We use the physically more relevant hull over the Minkowski sum of the friction cones ($DGWS_{Q2}$). Additionally, we want to overcome the large friction cone approximation errors.

For the representation of a Task Wrench Space we have to consider two different cases:

A TWS can be specified by a small set of wrench vectors that depict the maximal expected load on the grasp for a desired task. So the distance of these wrench vectors to the boundary of the GWS reflect the quality of the grasp (fig. 6.5 A).

For the case where no TWS is specified, we want to use the concept of Object Wrench Spaces (OWS). To calculate an OWS representation one can sample the object to grasp with applying a unit force in normal direction of the adjacent faces at every object corner. Though the produced very large set of wrench vectors can be reduced by calculating the convex hull, the computational complexity of such an approach is far too high to receive a fast computable grasp quality measure (fig. 6.5 B).

Therefore often a sphere in wrench space is used to represent a general TWS. This, however, introduces dependencies on reference frames and is also physically not so well interpretable. From a computational point of view indeed it is a nice representation. For the determination of the radius of the largest inscribed sphere (representing the grasp quality) one has to find the shortest distance from the GWS origin to its boundary. This can be done with simply sorting the facets of the convex hull with respect to their distance from the origin (fig. 6.5 C).

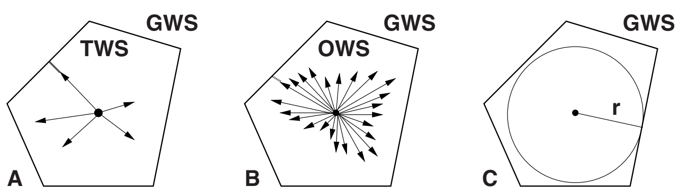

Fig. 6.5. The quality measure applied to different task wrench space representations.

To keep the explanations more comprehensible we first present an efficient algorithm to compute the grasp quality measure for a given set of task wrenches and for a sphere representing a general TWS. Then we will show, that the calculation of the measure with the OWS concept can be transformed to one of the above described two cases. So the algorithm can be used also with an OWS as general Task Wrench Space and the problems arising with the use of a sphere can be avoided.

6.3.3 An Incremental Algorithm for Calculating the Grasp Quality Using Convex Hulls

As described above, the calculation of the whole $DGWS_{Q2}$ is complex and cannot be simplified. For the quality calculation, however, only the part where the “weakest” wrench direction of the grasp is found is of interest. The idea of the algorithm is to incrementally add vectors to the DGWS at its “weakest” wrench direction to refine the the hull. The added wrenches must be composed by contact wrenches. It can be shown that one has to calculate only a subset of the DGWS in high precision to get the quality of a grasp (see fig. 6.6). In the following we describe the algorithm in detail:

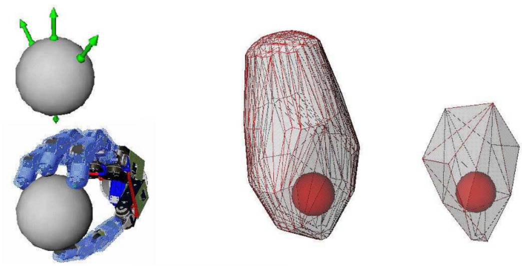

Fig. 6.6. Visualization of a $DGWS_{Q2}$ (middle) for the shown grasp on a sphere (left) and the result when incrementally calculating the DGWS at its “weakest” wrench direction (right)

Algorithm Outline

1. Calculate an initial convex hull over a small set of contact wrench vectors, e.g. 4 per contact (starting hull with 16 wrenches)
2. Determine the facet of the convex hull that is next to the origin. The facet normal gives the direction of the smallest wrench of the so far calculated DGWS.
3. Determine the wrench vector that expands the hull most in the given direction. The vector must be linearly composed from the set of contact wrenches.
4. Add the new wrench to the convex hull.
5. Start with the second step as long as the approximation error exceeds the given limit.

The Algorithm for a Given TWS

We assume the TWS to be given as a set of wrench vectors representing a special task. To evaluate the quality of a grasp one has to calculate the

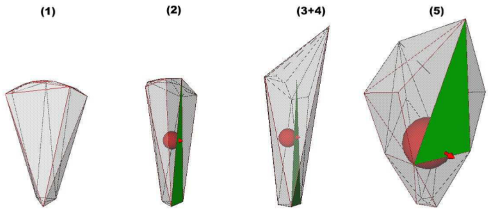

Fig. 6.7. A visualization of the algorithm steps: (1) small initial hull, (2,3+4) one step in the incremental hull expanding procedure, (5) last step: the hull is complete

scaling factor of each given task wrench vector. The smallest scaling factor is the quality of the grasp with respect to the given TWS (see fig.6.5 A). To find out the smallest scaling factor one has to calculate step 3 of the above algorithm for each given TWS wrench. So for a small number of wrenches this evaluation is very fast.

Weakest Wrench Calculation

One important step of the algorithm is still missing: How to calculate the new wrench in step (3) that expands the hull most in a given direction?

The problem is to find a new wrench $\mathbf{w}_{max}$, composed of contact wrenches $\mathbf{w}_{c_i}$, that spans the hull most in a given direction $\mathbf{w}_{min}$. This can be written as an optimization problem:

$$\text{WMax} = \left(\mathbf{w}_{min} \dot{\mathbf{c}} \mathbf{w}_{max} \rightarrow \text{Max} \,|\, \mathbf{w}_{max} = \sum_{i=1}^{n} \mathbf{w}_{c_i} : \mathbf{w}_{c_i} \in DCWS_{c_i} \right) \tag{6.11}$$

The dot product gives the scalar value we want to optimize. It makes no difference if it is applied to $\mathbf{w}_{max}$ or to each of the contact wrenches $\mathbf{w}_{c_i}$ (see eq. 6.12). This means that we can solve the problem for a single contact, which is much more simple than optimizing for the whole grasp.

$$\mathbf{w}_{min} \dot{\mathbf{c}} \mathbf{w}_{max} = \mathbf{w}_{min} \dot{\mathbf{c}} \sum_{i=1}^{n} \mathbf{w}_{c_i} = \sum_{i=1}^{n} \left(\mathbf{w}_{min} \dot{\mathbf{c}} \mathbf{w}_{c_i} \right) . \tag{6.12}$$

So the new problem is to find the wrench $\mathbf{w}_{c_i}$ for a single contact that maximizes the dot product with the given wrench $\mathbf{w}_{min}$.

$$\mathbf{w}_{min} \dot{\mathbf{c}} \mathbf{w}_{c_i} = \mathbf{w}_{min} \dot{\mathbf{c}} \begin{pmatrix} \mathbf{f_i} \\ (\mathbf{c_i} - \mathbf{r}) \times \mathbf{f_i} \end{pmatrix} \tag{6.13}$$

From a geometric view equation 6.13 means that we project the minimal wrench direction on a special wrench that can be exerted through contact

$\mathbf{c}_i$. Instead of projecting $\mathbf{w}_{min}$ to the solution wrench vector $\mathbf{w}_{max}$, we can project $\mathbf{w}_{min}$ to the 3D manifold spanned by the contact $\mathbf{c}_i$. With choosing a clever coordinate basis we can solve the problem of finding the best contact wrench geometrically in the force domain.

Let's take the following base vectors for the 3D manifold defined by contact $\mathbf{c}_i$:

$$\mathbf{e}_1 = \begin{pmatrix} \mathbf{f}_{e1} \\ \tau_{e1} \end{pmatrix}, \mathbf{e}_2 = \begin{pmatrix} \mathbf{f}_{e2} \\ \tau_{e2} \end{pmatrix}, \mathbf{e}_3 = \begin{pmatrix} \mathbf{f}_{e3} \\ \tau_{e3} \end{pmatrix} \tag{6.14}$$

$$\mathbf{f}_{e1} = \begin{pmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \mathbf{f}_{e2} = \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \end{pmatrix}, \mathbf{f}_{e3} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{pmatrix} \quad \text{and} \quad \tau_e = (\mathbf{c_i} - \mathbf{r}) \times \mathbf{f_e}. \tag{6.15}$$

With the projection of the minimal wrench direction $\mathbf{w}_{min}$ into the manifold of the contact $\mathbf{c}_i$ we get a force vector that represents the force that makes the most contribution to $\mathbf{w}_{min}$ if applied in contact $\mathbf{c}_i$. With this force vector $\mathbf{f}_{min}$ we can solve the problem now geometrically.

$$\mathbf{f}_{min} = \begin{pmatrix} \mathbf{w}_{min}\dot{\mathbf{c}}\mathbf{e}_1 \\ \mathbf{w}_{min}\dot{\mathbf{c}}\mathbf{e}_2 \\ \mathbf{w}_{min}\dot{\mathbf{c}}\mathbf{e}_3 \end{pmatrix} \tag{6.16}$$

There are four different cases to consider (fig. 6.8):

1. The projection of $\mathbf{w}_{min}$ results in the null vector. Then the contact cannot contribute to this weakest wrench direction. (fig. 6.8A)
2. The dot product of the resulting force vector $\mathbf{f}_{min}$ with the contact normal is negative or zero. Then the contact also cannot contribute to this weakest wrench direction. (fig. 6.8B)
3. The dot product of the resulting force vector $\mathbf{f}_{min}$ with the contact normal is positive, but $\mathbf{f}_{min}$ does not ly in the friction cone of the contact. The best force vector to choose is then the projection of $\mathbf{f}_{min}$ onto the boundary of the friction cone. (fig. 6.8C)
4. The resulting force $\mathbf{f}_{min}$ lies within the friction cone of the contact, so it is already the best contributing force vector for this contact. (fig. 6.8D)

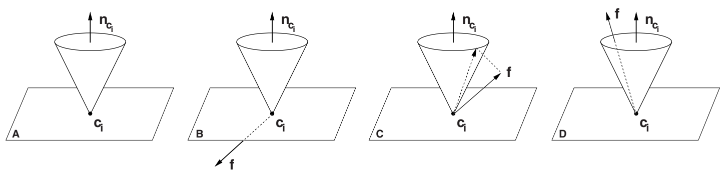

Fig. 6.8. 4 cases for the projection of the "weakest" wrench direction:

Result

The incremental algorithm solves the problem of measuring the quality of a grasp if the Task Wrench Space is modeled as a sphere. If there is a set of wrenches given that specify a special task, a part of the algorithm can also be used. Assumed, that the task specification is given by a small number of wrenches this method is very fast. If we take the OWS as a general task specification as suggested in section 6.2.3 and therefore sample the object, the OWS representation will be composed of many vectors and so the evaluation of the grasp quality may be slow. To overcome this problem we present in the following a method to approximate the OWS representation. This approximation can be used to transform the grasp quality evaluation with respect to an OWS to the case where a sphere is used as general task specification.

6.4 An Approximated OWS Representation for Efficient Integration in the Grasp Qualification Approach

Now we want to compare the OWS (and not the largest inscribed sphere) with the GWS of the grasp that is actually evaluated. That means we search the largest scaling factor for a given OWS to fit it into a GWS. In order to keep this algorithm of the same complexity as the one described above, we cannot use the sampled OWS directly. Instead, we circumscribe the OWS with an ellipsoid and use the corresponding inverse linear mapping for the GWS. Thus we reduce the problem to the "sphere fitting" problem with an additional linear mapping per GWS vector (see fig. 6.9).

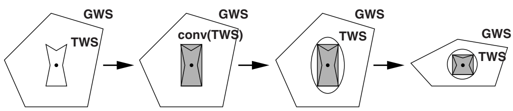

Fig. 6.9. Approximating the OWS with an ellipsoid:
1. The sampled OWS (exact space, exact quality measure)
2. Convex Hull over the sampled OWS (approx. space, exact quality measure)
3. Enclosing ellipsoid (approx. space, approx. quality measure)
4. Linear transformation of ellipsoid and GWS (sphere algorithm applicable)

6.4.1 Calculating a Minimal OWS Enclosing Ellipsoid

The problem may be stated as finding the smallest ellipsoid (spanned by the quadratic form $\mathbf{x}^T\mathbf{Q}\mathbf{x} \leqslant 1$, $\mathbf{Q}$ symmetric and positive definite) that encloses the OWS. More formally, we look for $\mathbf{Q}$ that fulfills

$$\forall \mathbf{x} \in \mathbf{OWS} : \mathbf{x}^T \mathbf{Q} \mathbf{x} \leqslant 1 \quad \wedge \quad \forall \mathbf{Q}' \neq \mathbf{Q} : V(\mathbf{Q}') < V(\mathbf{Q})$$

$V(\mathbf{Q}) = 1/\sqrt{det(\mathbf{Q})}$ being the volume of the ellipsoid spanned by $\mathbf{Q}$.

An efficient randomized algorithms to solve this problem in expected linear time (with fixed dimension) is known [297]. There are two problems with such an approach: The major problem is, that with the minimal enclosing ellipsoid we may introduce a non uniform scaling for the force vectors. That means, forces in certain directions may be weighted more than in other directions which is definitely unwanted. The second problem is that these algorithms are quite delicate to implement and only reference implementations in 2D or 3D exist.

Furthermore we are not necessarily interested in the smallest enclosing ellipsoid. The ellipsoid approximation of the OWS is used to get a reliable proportion of forces and torques and for the representation of the basic geometry of the object to grasp. This significantly affects the grasps quality. So we can use some knowlege of the force and torque domain structure of the OWS to make the ellipsoid construction easier and more adequate for our needs.

6.4.2 The Construction of a Small Enclosing Ellipsoid

By definition, the forces generating the OWS have all unit length and can have any direction (on most objects), thus the OWS projection to the force dimensions can be tightly enclosed by a unit sphere. To add the effects of gravity for the (most relevant) case that we use the center of mass as our reference point, we add a scaling factor $c = Max(\mathbf{f}_{contact}, \mathbf{f}_{gravity})$ and get

$$Hull(\mathrm{OWS}|_{\mathrm{forces}}) = \left\{ \mathbf{f} \;\middle|\; \frac{1}{c}\|\mathbf{f}\|_2 = 1 \right\}$$

The form and size of the OWS projected to its torque dimensions $OWS|_{torques}$ is determined by the object geometry. From examples with different test objects we can see that this projection can be approximated by a 3 dimensional ellipsoid without introducing a large error (see fig. 6.10).

Assuming that we already have given the ellipsoid which encloses $OWS|_{torques}$ as a quadratic form with the symmetric matrix $\mathbf{W}$ we get

$$Hull(\mathrm{OWS}|_{\mathrm{torques}}) = \left\{ \mathbf{t}' \mid \mathbf{t}'^T \mathbf{W} \mathbf{t}' = 1 \right\}$$

with $\mathbf{t}' = \mathbf{t} - \mathbf{t}_{origin}$ where $\mathbf{t}_{origin}$ is the center of the torque enclosing ellipsoid.

So we have to construct an OWS enclosing ellipsoid that is spanned by a 3D sphere in force and a 3D ellipsoid in torque domain. This can be done analytically.

The main idea can be described with a 3D analogon. Take a cylinder in 3D that should be enclosed by an ellipsoid as an example. The cylinder is bound by a circle (2D) and a height (1D). A small enclosing ellipsoid for the

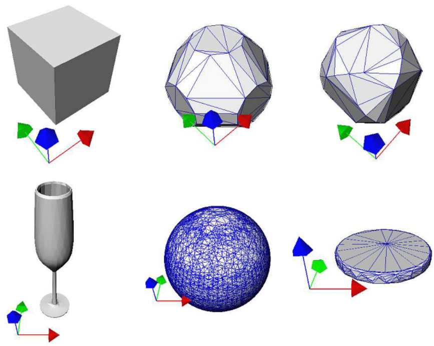

Fig. 6.10. The structure of the OWS projections in force and torque space for two sample objects a cube and a champagne glass. In force space (middle) one can see that not every direction can be generated by a single disturbance force, due to the limited surface normal directions on a cube. For the glass the force space is almost a perfect sphere.
The torque space for the cube is symmetric in all coordinate axes and the enclosing ellipsoid would be a sphere in this special case. For the glass the torque space is flat for torques round the symmetry axis of the glass.

cylinder needs to touch the cylinder's bounding circle. So the projection of the ellipsoid to the circle dimensions should also look like a circle. The projection of the cylinder to the height and one circle dimension gives a rectangle which should be touched by the ellipsoid in its corner points. This defines a band of ellipsoids where we select the one with minimal volume (fig. 6.11).

In our 6D case we have a sphere and an ellipsoid instead of a circle and height and we want to preserve the structure of the sphere in force projection. From the considerations above we can derive equations that define a band of ellipsoids in 6D. Of these we choose the ellipsoid with the smallest volume to approximate the OWS. For a detailed description of this method we refer to [41].

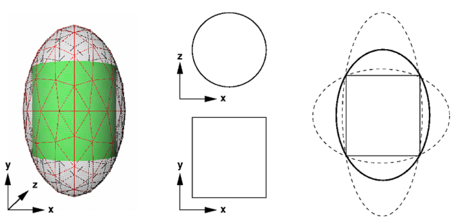

Fig. 6.11. Fitting an ellipsoid to a 3D cylinder

6.5 A Statistical Evaluation of Grasp Quality

Considering the algorithm described above one can measure the quality of a given contact set, representing a grasp, efficiently. Given a set of contacts and an associated task wrench space one can decide wether the grasp is valid for the operation or not. It would also be interesting to have an idea how well the

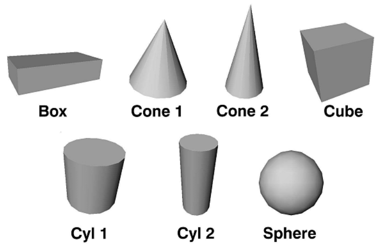

Fig. 6.12. Set of geometric primitives

grasp performs concerning other grasps on the object or how well it performs in comparison to human grasps. Last the question of how many good grasp on an object can be found ist also of interest as it gives an idea of how hard the grasp planning problem is.

A statistical analysis on a set of representative objects can provide answers to these questions. For the test objects we chose a set of geometric primitives (fig. 6.12) where many real world object shapes can be composed with. Also we selected some real world objects (fig. 6.13)to show that their results are similar to the geometric primitives.

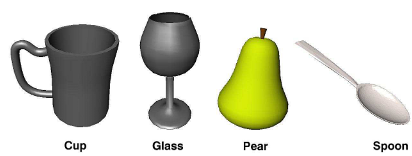

Fig. 6.13. Set of real world objects

There are two main topics that we want to address with this analysis: The number of grasp with an acceptable quality level that can be found and the comparison of a quality rated grasp with a typical human grasp on the same object.

6.5.1 The Ratio of Force Closure Grasps

A widely used minimal quality requirement for a grasp is the force closure property. To get an idea how hard the force closure grasp planning problem is, we generated 10^6 3-,4- and 5-contact grasp candidates randomly for each object and tested them for the force closure property. Fig. 6.14 shows the results for a friction coefficient of $\mu = 0.5$.

The worst case for the three contact case is the cone where only 2% of the generated candidates are force closure. That means after the generation of 350 candidates the probability is 99.9% that at least one force closure candidate is in the set. For the four and five contact case the problem is more relaxed. Here the worst cases are with the grasp candidates on the glass where 8% and 15%, resp. are force closure, which means the generation of 84 and 43, resp. candidates for one force closure grasp. The generation and force closure test

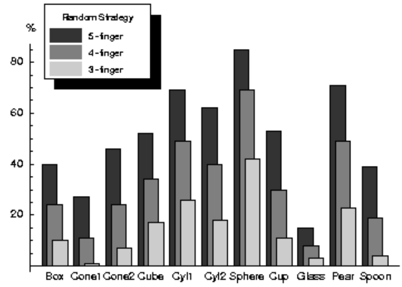

Fig. 6.14. The ratio of force closure grasps on the test objects generating grasps with 3, 4 and 5 contacts (friction $\mu = 0.5$)

of 100 candidates takes about one second on a Pentium III/900 computer. It seems that for a three finger manipulator random grasp generation may not be a very promising approach but it should be mentioned that there are simple and efficient heuristics that can improve these results considerably as we showed in [40].

6.5.2 The Distribution of Grasp Quality

It can be stated that normally many force closure grasps are possible for an object. What can be stated for the distribution of quality rated grasps on an object and how can the measured quality be intuitively rated?

To obtain answers to these questions we generated 10^6 force closure grasp candidates on each test object and evaluated their quality value with our quality measure. The normalized results (area of the histograms is equal to one) are shown in figure 6.15.

As expected there are only a few very well rated grasp for each object. An intuitive benchmark for these generated grasps it the comparison with a typical human grasp on these objects. We assume that any grasp that has a higher quality rating than the typical human grasp is adequate for usual tasks. The typical human grasp samples are derived from the grasp taxonomy proposed by Cutkosky and Howe [67] and the sample grasps showed in Napiers book on hands [210].

With this assumption, we can calculate the probability of getting such a grasp for our test objects. It ranges from 8% (Cone2) up to 83% (Sphere). Therefore with the generation of 12 (Sphere) up to 83 (Cone2) force closure grasp candidates the probability of getting at least one candidate better than the human like grasp is more than 99.9%. To give an idea of the calculation time, generating 100 force closure grasp candidates and rating their quality takes about 2 sec. on a Pentium III/900MHz.

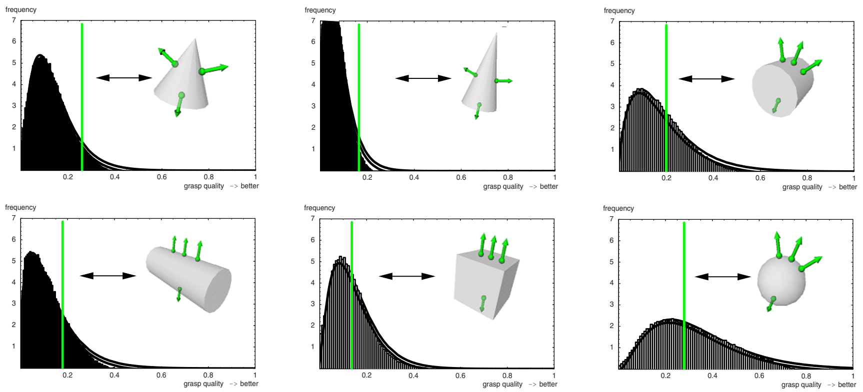

Fig. 6.15. Results from the evaluation of grasp quality on the test objects (4 contacts, friction $\mu = 0.5$). The quality of the human like grasps is marked with a vertical line.

6.6 An Effective Grasp Planner Architecture

The grasp qualification algorithm and the statistical analysis allow to *measure* and *classify* the quality of a given grasp. However, the question how to *generate* a grasp with a high quality rating is still left open.

Treating the grasp planning problem as a large optimization problem leads to objective functions with numbers of local minima and discontinuities in the case of imperfect object models. The global minimization of these functions is computational complex and therefore relatively slow.

For the geometric construction of force closure grasps methods have been developed for 2D object that can also be applied to 3D objects, but it's more difficult and unhandy. Grasps that are optimal with respect to our quality measure or measures of similar physical interpretation however cannot be constructed efficiently.

Taking the statistical data and the efficiency of our grasp qualification algorithm into account, it seems indicated to use a generate and test strategy to plan good although not optimal grasps. In the following we outline a generate and test architecture for a grasp planner and shortly mention the planning steps that are missing so far to get a complete planning system.

6.6.1 A Generate and Test Planner for Precision Grasps

From a functional view on the grasp planning problem we can identify the following modules for a generate and test grasp planner:

1. grasp contact generation
2. grasp configuration (calculating kinematic valid joint configuration for the gripper)

3. collision check (to avoid intrusions of the gripper through the object)
4. grasp quality evaluation

The main topic to keep the generate and test planner fast and effective is, to arrange these operations in order of their computational complexity. To increase the efficiency one can develop simple and fast tests to identify unpromising candidates very early. For our grasp planner the modules are arranged as follows:

First, based on the object geometry a set of contacts is chosen randomly. In the second step some constraints are checked, that come from the gripping device, as minimum and maximum contact distance. There is also a heuristic force closure test done, which we described in [40]. The third step is the calculation of the quality measure. The planner checks, if the contacts lie next to object edges, which we assume to be none robust contacts. In the fourth and fifth step a kinematic configuration is calculated to reach the grasp contacts with the fingertips (see [39] for details) and a collision check using RAPID [95] is performed to check intrusions of fingers through the grasp object. Fig. 6.16 shows a chart of the planner modules.

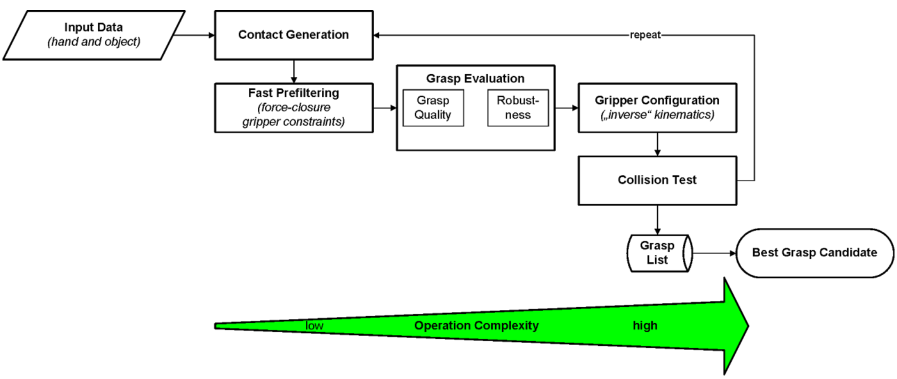

Fig. 6.16. Outline of the generate and test planner for planning precision grasps.

From a software engineering view this architecture is very attractive as the interfaces between the modules are very small. Therefore new modules can easily be plugged in or changed with existing ones. So one can change the gripper dependent parts and adapt to different grasping devices relatively easy.

6.7 Conclusion

In this chapter we give an exact description of wrench spaces that are relevant for grasp analysis and grasp planning. From these descriptions we derive a physically well interpretable, exact grasp quality measure. Starting from a discussion of so far proposed and usually used methods to determine the quality of a grasp and their known problems, we give an efficient incremental algorithm to calculate the quality of a grasp very fast and with negligible approximation errors. For the evaluation of a grasp with respect to an object wrench space we outline a method, using approximating ellipsoids to transform the problem to the fast computable "sphere fitting" problem.

Based on a statistical analysis on force closure ratio and grasp quality distributions on test objects we propose a fast and effective generate and test planner architecture. This planner allows us to plan, in comparison with human like grasps, sufficiently good grasps for arbitrary 3D objects online. So our planner can be used for autonomous grasping scenarios.