5
Fundamentals of Grasping

Robotic manipulation, where a robot *physically* interacts and changes the environment, is one of the most challenging tasks in robot autonomy from the perspectives of perception, planning, and control. Consider a simple pick-and-place problem: the robot needs to identify the object, find a good place to grasp, stably pick up the object, and move it to a new location, all while ensuring no part of the robot collides with the environment. In practice even this simple task can become much harder, for example if other objects are in the way and must be moved first, if the object does not have particularly good grasping features, if the weight, size, and surface texture of the object is unknown, or if the lighting is poor[1]. Manipulation tasks are also commonly composed of sequences of interactions, such as making a sandwich or opening a locked door. This chapter focuses on *grasping*[2], which is a fundamental component to all manipulation tasks.

Grasping

Grasping is a fundamental component of robotic manipulation that focuses on obtaining complete control of an object's motion (in contrast to other interactions such as pushing).

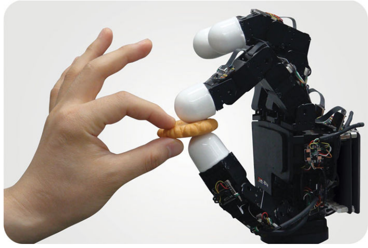

Figure 5.1: The Allegro Hand. Image retrieved from `wiki.wonikrobotics.com`.

Definition 5.0.1 (Grasp). *A grasp is an act of restraining an object's motion through application of forces and torques at a set of contact points.*

Grasping is challenging for several reasons:

1. The configuration of the gripper may be high-dimensional. For example the Allegro Hand (Figure 5.1) has 4 fingers with 3 joints each for a total of 12 dimensions. Plus there are an additional 6 degrees of freedom in the wrist posture (position and orientation), and all of these degrees of freedom vary continuously.

2. Choosing contact points can be difficult. An ideal choice of contact points would lead to a robust grasp, but the space of feasible contacts is restricted

[1] Generally speaking the infinite variability of the real world makes *robust* manipulation extremely difficult.

[2] D. Prattichizzo and J. C. Trinkle. "Grasping". In: *Springer Handbook of Robotics*. Springer, 2016, pp. 955–988

by the gripper's geometry. A rigid body object also has 6 degrees freedom, which affects where the contact points are located in the robot's workspace.

3. While the robot is attempting the grasp it must be sure that its entire body does not come into collision with the environment.

4. Once a grasp has been performed it is important to evaluate how robust the grasp is. While the grasp quality would ideally be optimized during the planning step, it may be important to also check retroactively in case uncertainty led to a different grasp than planned.

To address each of these challenges, the grasp can be subdivided into parts: planning, acquisition/maintenance, and evaluation. This chapter will focus on the fundamentals of how a grasp can be *modeled* and *evaluated* from a mathematical perspective, as well as how grasps can be *planned*[3] using grasp force optimization. Learning-based approaches to grasping and manipulations will also be discussed at a high level in Section 5.4 and 5.5.

5.1 Grasp Modeling

A grasp plan may be parameterized in several ways, including by the approach vector or wrist orientation of the gripper, by the initial finger configuration, or directly by points of contact with the object. However, regardless of the planning parameterization the resulting contacts between the gripper and the object will define the quality of the grasp. Therefore it is useful and convenient for grasp modeling to consider the contact points as the interaction interface between the gripper and object.

5.1.1 Contact Types

There are generally three types of contact that can occur in grasping scenarios:

1. *Point*: a point contact occurs when a single point comes in contact with either another point, a line, or a plane. A point contact is only stable if it is a point-on-plane contact[4], point-on-point or point-on-line contacts are unstable.

2. *Line*: line contacts occur when a line comes in contact with another line or a plane. Line-on-plane and line-on-nonparallel line contacts are stable, but line-on-parallel line contacts are unstable. Line contacts can also be represented as two point contacts.

3. *Plane*: plane-on-plane contacts are always stable. Plane contacts can also be represented as point contacts by converting a distribution of normal forces across a region into a weighted sum of point forces at the vertices of the region's convex hull.

[3] Part of grasp planning also includes the motion planning of the entire robot, but the focus of this chapter is on the grasp itself.

[4] Point-on-plane contacts are by far the most commonly modeled contact types and will almost always be used in grasp analysis.

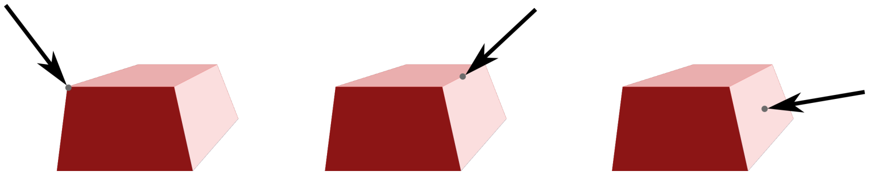

Figure 5.2: Grasping contacts are generally either point-on-point (left), point-on-line (middle), point-on-plane (right).

5.1.2 *Point-on-Plane Contact Models*

Point-on-plane contact models are by far the most commonly used for grasping since the possible contact points for most objects are almost always surface points (and not sharp edges or points). The purpose of the contact model is to specify the admissible forces and torques that can be transmitted through a particular contact. Considering a local reference frame defined at the contact point with the z direction pointing along the object's surface normal (with the positive direction defined as into the object), the force f can be written as:

$$f = f_{\text{normal}} + f_{\text{tangent}},$$

where $f_{\text{normal}} = [0, 0, f_z]^T$ is the vector component along the normal direction (with magnitude f_z) and $f_{\text{tangent}} = [f_x, f_y, 0]^T$ is the vector component tangent to the surface. For all types of contact only an inward force can be applied, therefore $f_z \geq 0$. Three types of contact models are commonly used, and each defines a set $\mathcal{F}$ of admissible forces that can be applied through the contact:

1. Frictionless Point Contact: forces can only be applied along the surface normal, no torques or forces tangential with the surface are possible ($f_{\text{tangent}} = 0$):

   $$\mathcal{F} = \{f_{\text{normal}} \mid f_z \geq 0\}.$$

   These types of contact models are more common in form closure grasps.

2. Point Contact with Friction[5]: it is possible to apply forces in directions other than just the surface normal. The admissible forces (i.e. forces that don't lead to slipping) are typically defined by a *friction cone*:

   $$\mathcal{F} = \{f \mid \|f_{\text{tangent}}\| \leq \mu_s \|f_{\text{normal}}\|, \quad f_z \geq 0\}.$$

   where μ_s is the static friction coefficient associated with the surface (see Figure 5.3).

   A pyramidal inner-approximation of the friction cone is often more useful from a computational standpoint, since its definition only requires a *finite* set of vectors (see Figure 5.4). The point contact with friction model is more common in force closure grasps.

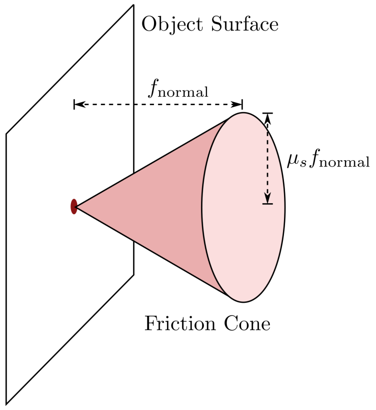

Figure 5.3: Friction cone defined by a static coefficient of friction μ_s.

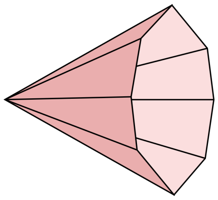

Figure 5.4: Linearized friction cone to *inner* approximate the true cone.

3. Soft-finger Contact Model: allows for a torque τ_{normal} around the surface normal axis and also includes a friction cone for the forces as in the point

[5] Also referred to as the *hard finger* model.

contact with friction model. The admissible torques are also constrained by friction:

$$\mathcal{F} = \{(\boldsymbol{f}, \tau_{\text{normal}}) \mid \|\boldsymbol{f}_{\text{tangent}}\| \leq \mu_s \|\boldsymbol{f}_{\text{normal}}\|, \quad f_z \geq 0, \quad |\tau_{\text{normal}}| \leq \gamma f_z\}.$$

where $\gamma > 0$ is the torsional friction coefficient.

5.1.3 *Wrenches and Grasp Wrench Space*

Under the assumption of a specific contact model, a grasp (defined by a set of contact points) can be quantified and evaluated by determining the *grasp wrench space*[6], which defines how the grasp can influence the object through an applied wrench.

Definition 5.1.1 (Wrench). *A wrench is a vector valued quantity that describes the forces and torques being applied to an object. For a force $\boldsymbol{f} \in \mathbb{R}^3$ and torque $\boldsymbol{\tau} \in \mathbb{R}^3$ applied at the object's center of mass, the wrench is the stacked vector:*

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\tau} \end{bmatrix} \in \mathbb{R}^6,$$

and is typically written with respect to a frame fixed in the body.

Each contact point i in a grasp applies a wrench to the object. Additionally the torque $\boldsymbol{\tau}_i$ can be computed by $\boldsymbol{\tau}_i = \boldsymbol{d}_i \times \boldsymbol{f}_i$ where $\boldsymbol{d}_i$ is the vector defining the position of the i-th contact point with respect to the object's center of mass. The wrench can then be written as:

$$\boldsymbol{w}_i = \begin{bmatrix} \boldsymbol{f}_i \\ \lambda(\boldsymbol{d}_i \times \boldsymbol{f}_i) \end{bmatrix}, \tag{5.1}$$

where the constant $\lambda \in \mathbb{R}$ is arbitrary but can be used to scale the torque magnitude if desired[7].

Using this definition of a wrench[8], a grasp can be defined as the set of all possible wrenches that can be achieved by the grasp's contact points. Mathematically, an admissible force $\boldsymbol{f}_i$ applied at the i-th contact point can be linearly mapped into the corresponding wrench on the object as $G_i \boldsymbol{f}_i$, where G_i is a wrench basis matrix that also includes a transformation from the local contact reference frame to an object-defined global reference frame. Therefore the total wrench on the object from all contacts is:

$$\boldsymbol{w} = \sum_{i=1}^{k} G_i \boldsymbol{f}_i = G \begin{bmatrix} \boldsymbol{f}_1 \\ \vdots \\ \boldsymbol{f}_k \end{bmatrix}, \quad G = \begin{bmatrix} G_1 & \dots & G_k \end{bmatrix}, \tag{5.2}$$

where the combined matrix G is referred to as the *grasp map* (which varies depending on the type of contact model used).

The *grasp wrench space* can then be defined as:

[6] The grasp wrench space is a *subset* of the *wrench space* $\mathbb{R}^n$, where $n = 6$ in 3D settings and $n = 3$ in 2D settings.

[7] If the forces $\boldsymbol{f}_{i,j}$ are dimensional a value of $\lambda = 1$ is common. When the forces are unit-dimension (i.e. scaled by their maximum magnitude), λ could be chosen to non-dimensionalize the entire wrench $\boldsymbol{w}_i$ by non-dimensionalizing the distance vector $\boldsymbol{d}_i$ (i.e. scale by an object size metric).

[8] For a soft-finger contact model the additional torque term must also be included.

Definition 5.1.2 (Grasp Wrench Space)**.** *The grasp wrench space $\mathcal{W}$ for a grasp with k contact points is the set of all possible wrenches $\boldsymbol{w}$ that can be applied to the object through admissible forces:*

$$\mathcal{W} := \{\boldsymbol{w} \mid \boldsymbol{w} = \sum_{i=1}^{k} G_i \boldsymbol{f}_i, \quad \boldsymbol{f}_i \in \mathcal{F}_i, \quad i = 1, \dots, k\}. \tag{5.3}$$

In other words, the grasp wrench space is defined by the output of (5.2) over all possible applied force combinations $\{\boldsymbol{f}_i\}_{i=1}^k$. If the grasp wrench space is large the grasp can compensate for a bigger set of external wrenches that might be applied to the object, leading to a more robust grasp.

Example 5.1.1 (Computing a Grasp Wrench Space from Friction Cones)**.** Consider a grasping problem with k contact points with friction, and let contact point i be associated with a linearized friction cone $\mathcal{F}_i$ whose edges are defined by the set of m forces:

$$\{\boldsymbol{f}_{i,1}, \boldsymbol{f}_{i,2}, \dots \boldsymbol{f}_{i,m}\},$$

such that any force $\boldsymbol{f}_i \in \mathcal{F}_i$ can be written as a positive combination of these vectors:

$$\boldsymbol{f}_i = \sum_{j=1}^{m} \alpha_{i,j} \boldsymbol{f}_{i,j}, \quad \alpha_{i,j} \geq 0.$$

The condition $\sum_{j=1}^{m} \alpha_{i,j} \leq 1$ will also be imposed to constrain the overall magnitude[9]. Geometrically, this means that the friction cone $\mathcal{F}_i$ is the convex hull of the points $\boldsymbol{f}_{i,j}$ and the origin of the local contact reference frame (see Figure 5.5).

This friction cone can then be mapped into the wrench space using (5.1). Assuming the forces $\boldsymbol{f}_{i,j}$ and position vector $\boldsymbol{d}_i$ are already expressed in a reference frame fixed in the object that is common to all i contact points, the grasp wrench space $\mathcal{W}$ can be written as:

$$\mathcal{W} = \{\boldsymbol{w} \mid \boldsymbol{w} = \sum_{i=1}^{k} \boldsymbol{w}_i, \quad \boldsymbol{w}_i = \sum_{j=1}^{m} \alpha_{i,j} \boldsymbol{w}_{i,j}, \quad \boldsymbol{w}_{i,j} = \begin{bmatrix} \boldsymbol{f}_{i,j} \\ \lambda(\boldsymbol{d}_i \times \boldsymbol{f}_{i,j}) \end{bmatrix}, \\ \sum_{j=1}^{m} \alpha_{i,j} \leq 1, \quad \alpha_{i,j} \geq 0\}.$$

In other words, the grasp wrench space is defined by taking the Minkowski sum over the sets of wrenches that can be generated from each individual contact!

For example, consider the 2D problem shown in Figure 5.6 where there are $k = 2$ contact points with friction. The friction cones are defined by the convex hull of the vectors $\{\boldsymbol{f}_{1,1}, \boldsymbol{f}_{1,2}\}$ and $\{\boldsymbol{f}_{2,1}, \boldsymbol{f}_{2,2}\}$ (and their origins) and the distance vectors from the center of mass to the contact points are $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$. The force vectors $\boldsymbol{f}_{i,j}$ are then mapped into the wrenches $\boldsymbol{w}_{i,j}$ (shown on a 2D plot of vertical force f_y and torque τ in Figure 5.6, ignoring the horizontal force components f_x). The grasp wrench space $\mathcal{W}$ is then shown in the grey region of the wrench space, where the solid grey line is the boundary of $\mathcal{W}$.

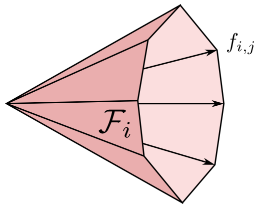

Figure 5.5: Any force $\boldsymbol{f}_i \in \mathcal{F}_i$ can be written as a convex combination of the forces along the edge vectors $\boldsymbol{f}_{i,j}$.

[9] In practice the physical hardware has limitations on the magnitude of the normal forces that can be applied.

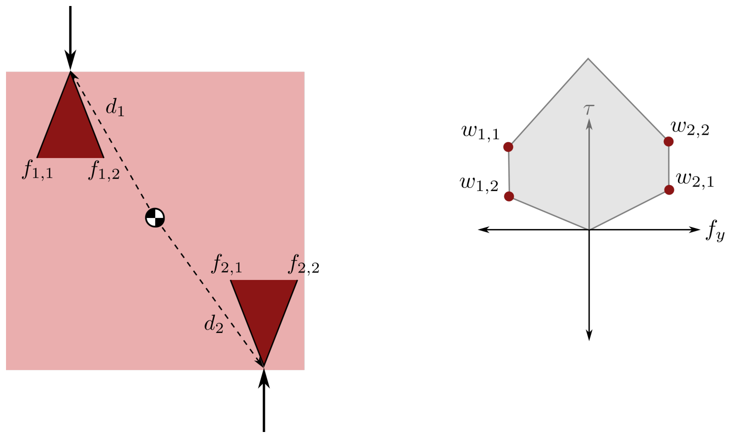

Figure 5.6: An example 2D grasp consisting of two point contacts with friction. The friction cones shown in the figure on the left yield the grasp wrench space in the figure on the right (showing only the vertical force and torque dimensions). Note that the grasp wrench space is bounded because it is assumed the magnitude of the contact forces are bounded. The solid grey line represents the boundary of $\mathcal{W}$.

5.2 *Grasp Evaluation*

Now that the basics of grasp modeling have been introduced[10] it is possible to explore techniques for evaluating whether a grasp is "good". In particular, an ideal grasp is one that has *closure*.

Definition 5.2.1 (Grasp Closure)**.** *Grasp closure occurs when the grasp can be maintained for every possible disturbance load.*

For example having grasp closure on a book would enable the gripper to maintain its grasp even if the book was hit by another object or if another book was suddenly stacked on top of it. In practice it may not be reasonable to assume that every *magnitude* disturbance load could be accounted for, but the concept of closure is useful nonetheless.

It can also be helpful to distinguish between two types of grasp closure. A *form closure*[11] grasp typically has the gripper joint angles locked and there is no "wiggle" room for the object (i.e. the object is kinematically constrained). Alternatively, a *force closure*[12] grasp uses forces applied at contact points to be able to *resist* any external wrench. Force closure grasps typically rely on *friction* and generally require fewer contact points than are required for form closure, but may not be able to actually cancel all disturbance wrenches if the friction forces are too weak. This chapter will primarily focus on evaluating force closure grasps since these are most common in robotics.

5.2.1 *Force Closure Grasps*

The concept of force closure can be related to the grasp modeling concepts from Section 5.1:

Definition 5.2.2 (Force Closure Grasp)**.** *A grasp is a force closure grasp if for any*

[10] contact types, contact models, grasp wrench spaces

[11] Also called power grasps or enveloping grasps. A grasp must have at least seven contacts to provide form closure for a 3D object.

[12] Also called a precision grasp. Under a point contact with friction model, a grasp must have at least three contacts to provide force closure for a 3D object.

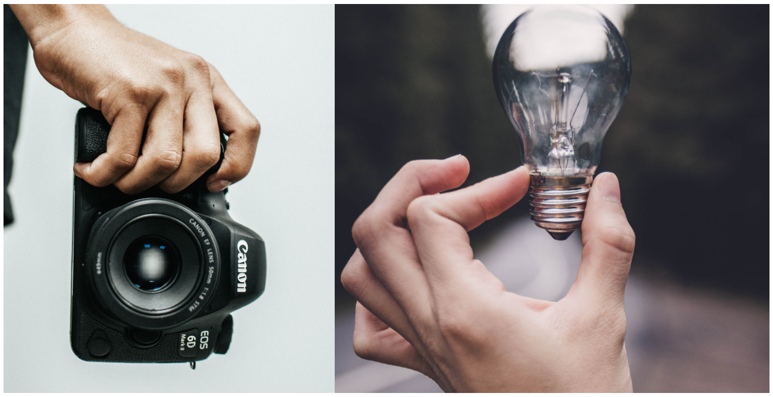

Figure 5.7: Examples of grasps with form closure (left) and force closure under the soft-finger contact model (right).

external wrench $\boldsymbol{w}$ *there exist contact forces* $\{\boldsymbol{f}_i\}_{i=1}^k$ *such that:*

$$-\boldsymbol{w} = \sum_{i=1}^{k} G_i \boldsymbol{f}_i. \quad \boldsymbol{f}_i \in \mathcal{F}_i, \quad i = 1, \dots, k,$$

or equivalently such that:

$$-\boldsymbol{w} \in \mathcal{W}.$$

This definition implies that the grasp wrench space must satisfy $\mathcal{W} = \mathbb{R}^n$ for a force closure grasp, which implicitly assumes that the contact forces can be infinite in magnitude.[13] Since real hardware has limitations on the magnitude of the applied contact forces, a more practical definition of force closure is to be able to *resist* any external wrenches. The conditions for force closure can be summarized by the following theorem:

Theorem 5.2.3. *In an n-dimensional vector space with:*

$$\mathcal{W} := \{\boldsymbol{w} \mid \boldsymbol{w} = \sum_{k=1}^{N} \beta_k \boldsymbol{w}_k, \quad \beta_k \geq 0\},$$

$\mathcal{W} = \mathbb{R}^n$ *if and only if the set* $\{\boldsymbol{w}_k\}_{k=1}^N$ *contains at least* $n+1$ *vectors, n of the vectors are linearly independent, and there exists scalars* $\beta_k > 0$ *such that:*

$$\sum_{k=1}^{N} \beta_k \boldsymbol{w}_k = 0.$$

From a practical perspective this theorem specifies a minimum number of different wrenches that must be used as a basis for the grasp wrench space, and also states that it must be possible for the grasp to apply zero wrench *even when some of the contact forces are non-zero*. These conditions are equivalent to saying that grasp wrench space $\mathcal{W}$ must contain the origin in its *interior*[14].

Note that in the practical case where the applied contact forces are assumed to be bounded, the conditions of Theorem 5.2.3 must still hold to guarantee the origin is in the interior of $\mathcal{W}$, which is required to resist any external wrench. The implications of this theorem are explored further in the following examples.

[13] For 2D objects $n = 3$ and for 3D objects $n = 6$.

[14] The grasp is not in force closure if the origin is on the *boundary* of $\mathcal{W}$.

Example 5.2.1 (2D Object (Forces Only))**.** Consider a simplified 2D problem where instead of complete force closure (i.e. ability to withstand any *wrench*) it is sufficient to only require the cancellation of external *forces*. In this case $n = 2$ and Theorem 5.2.3 states that it must be possible for the grasp to generate 3 force vectors where 2 are linearly independent and where:

$$\beta_1 \boldsymbol{f}_1 + \beta_2 \boldsymbol{f}_2 + \beta_3 \boldsymbol{f}_3 = 0, \quad \beta_1, \beta_2, \beta_3 > 0.$$

Two examples grasps are shown in Figures 5.8 and 5.9. In Figure 5.8 the three contacts are frictionless, but even though there are 3 possible force vectors with 2 linearly independent, there is no way to generate zero force with non-zero forces at each contact! Therefore this grasp does not have force closure.

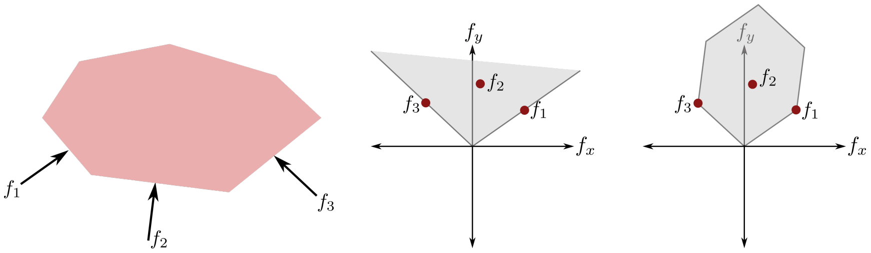

Figure 5.8: A 2D grasp with frictionless contacts that cannot compensate for all possible external forces on the object. The middle and right-side figures show the space of all possible applied forces for the cases of unbounded and bounded contact force magnitude, respectively.

Alternatively, Figure 5.9 shows a case where it is possible to have force closure using a point contact without friction and a point contact with friction (a hypothetical example). In this case all of the conditions in Theorem 5.2.3 are satisfied, and it can be seen that the origin is contained in the interior of the space of possible applied forces.

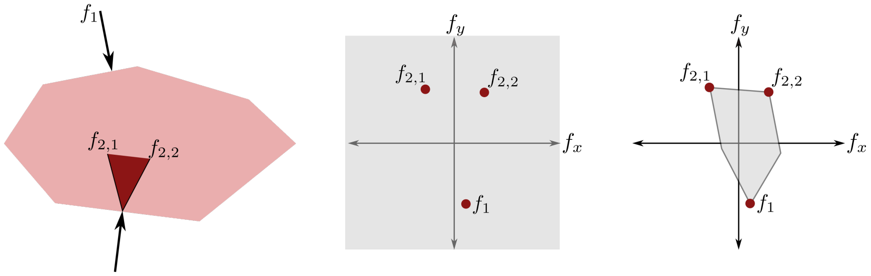

Figure 5.9: A 2D grasp consisting of a contact with friction and a contact without friction. The middle and right-side figures show the space of possible forces for the cases of unbounded and bounded magnitude, respectively. In the unbounded case it is possible to compensate for any external force, and in the bounded case it is possible to *resist* an arbitrary force.

Example 5.2.2 (2D Object)**.** In the more general case with 2D objects where the torque is also considered, the grasp wrench space is in 3D (i.e. $\mathcal{W} \subseteq \mathbb{R}^3$). Therefore, Theorem 5.2.3 states the grasp wrench space satisfies $\mathcal{W} = \mathbb{R}^3$ if and only if it is possible for the grasp to generate at least 4 different wrenches, with 3 being linearly independent, and where:

$$\beta_1 \boldsymbol{w}_1 + \beta_2 \boldsymbol{w}_2 + \beta_3 \boldsymbol{w}_3 + \beta_4 \boldsymbol{w}_4 = 0, \quad \beta_1, \beta_2, \beta_3, \beta_4 > 0.$$

If frictionless contacts are assumed these conditions require *at least* 4 contact points and in the friction case *at least* 2 contacts are required.

Consider again the grasp shown in Example 5.1.1 (Figure 5.6). The 4 edges of the friction cones create a set of 3 linearly independent wrenches, but there is no way to generate zero wrench with non-zero contact forces. This is evident in the fact that it is not possible to generate a negative torque, which means the grasp is not in force closure[15]. An alternative grasp that is in force closure is shown in Figure 5.10, which leverages a third contact point to ensure the grasp achieves stability.

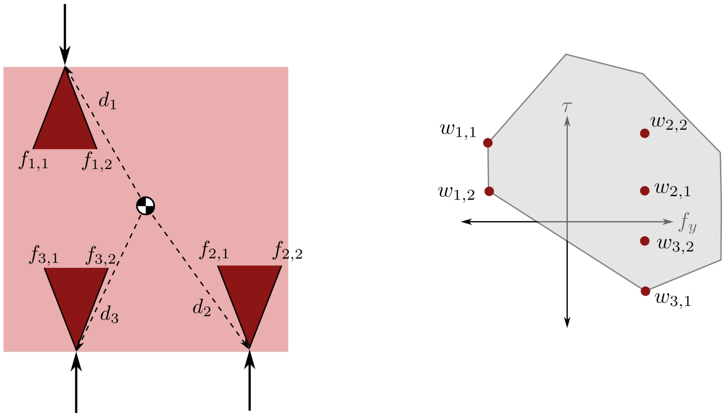

Figure 5.10: A 2D grasp consisting of three point contacts with friction. The friction cones shown in the figure on the left yield the grasp wrench space in the figure on the right (showing only the vertical force and torque dimensions and assuming bounded contact force magnitudes). This grasp is in force closure because it can resist any external wrench (the origin is contained in the interior of $\mathcal{W}$).

Example 5.2.3 (3D Object)**.** For 3D objects the grasp wrench space is in 6D (i.e. $\mathcal{W} \subseteq \mathbb{R}^6$). Theorem 5.2.3's conditions therefore require that the grasp to be able to generate at least 7 different wrenches, with 6 being linearly independent, and where:

$$\sum_{k=1}^{7} \beta_k \boldsymbol{w}_k = 0, \quad \beta_k > 0.$$

If frictionless contacts are assumed these conditions require *at least* 7 contact points and in the friction case *at least* 3 contacts are required.

5.2.2 *Grasp Wrench Hull*

The grasp wrench space $\mathcal{W}$ defines the set of all possible wrenches that can be applied to an object by a grasp, but unfortunately computing this set can be quite cumbersome in practice. One alternative approach for characterizing a grasp is through the definition of the *grasp wrench hull*, which can be efficiently computed. Given a set of linearized friction cones $\mathcal{F}_i$ defined by the set of bounded forces $\{\boldsymbol{f}_{i,1}, \boldsymbol{f}_{i,2}, \dots, \boldsymbol{f}_{i,m}\}$ for each contact in the grasp, the wrench hull $\tilde{\mathcal{W}}$ is mathematically defined as:

$$\tilde{\mathcal{W}} = \{\boldsymbol{w} \mid \boldsymbol{w} = \sum_{i=1}^{k}\sum_{j=1}^{m} \alpha_{i,j}\boldsymbol{w}_{i,j}, \quad \boldsymbol{w}_{i,j} = \begin{bmatrix} \boldsymbol{f}_{i,j} \\ \lambda(\boldsymbol{d}_i \times \boldsymbol{f}_{i,j}) \end{bmatrix}, \quad \sum_{i=1}^{k}\sum_{j=1}^{m} \alpha_{i,j} = 1, \quad \alpha_{i,j} \geq 0\},$$

where $\boldsymbol{d}_i$ is again the vector from the object center of mass to contact point i. Note that this is almost identical to the grasp wrench space definition ex-

[15] Notice again that the origin is not contained in the interior of the grasp wrench space!

cept that the constraint $\sum_{j=1}^{m} \alpha_{i,j} \leq 1$ for all i has been replaced by the constraint $\sum_{i=1}^{k} \sum_{j=1}^{m} \alpha_{i,j} = 1$. Put simply, the wrench hull is the convex hull of the wrenches $\boldsymbol{w}_{i,j}$! The difference between the grasp wrench space and the wrench hull is shown in Figure 5.11 for the grasp from Example 5.2.2.

Importantly the property $\tilde{\mathcal{W}} \subseteq \mathcal{W}$ holds by definition. Therefore grasp force closure is also guaranteed when the origin is in the interior of the wrench hull space. This fact, coupled with the fact that $\tilde{\mathcal{W}}$ is easier to compute than $\mathcal{W}$, makes it a useful characterization of grasps for evaluating grasp quality.

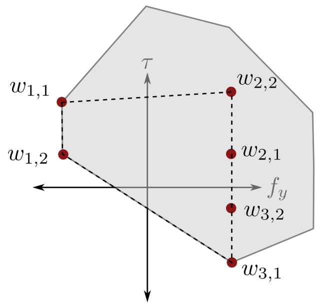

Figure 5.11: Difference between grasp wrench space $\mathcal{W}$ (grey area) and wrench hull $\tilde{\mathcal{W}}$ (area enclosed by black dashed line) for the grasp in Example 5.2.2.

5.2.3 *Grasp Quality*

If the gripper could apply contact forces with infinite magnitude then a "good" grasp could simply be defined as one that is in force closure. However a more practical definition of grasp quality should be based on the assumption that the magnitude of the contact forces is bounded. In other words, a metric for grasp quality should quantify *how well the grasp can resist external wrenches for a given bound on the contact force*.

To accomplish this, grasp quality metrics can be defined based on the definition of the grasp wrench hull $\tilde{\mathcal{W}}$. In particular, a useful metric is the radius of the largest ball centered at the origin that is completely contained in the grasp wrench hull (Figure 5.12). This metric is useful for the following reasons:

1. If the radius is zero, the origin is not contained in the interior of the wrench hull and therefore the grasp is not in force closure.

2. For a radius greater than zero, the metric represents the magnitude of the *smallest* external wrench that pushes the grasp to the limits. The direction from the origin to where the ball touches the boundary of $\mathcal{W}$ identifies the (opposite) direction in which the grasp is least able to resist external wrenches.

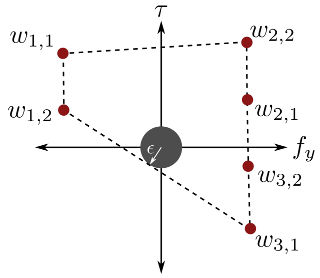

Figure 5.12: Grasp quality can be measured as the radius ϵ of the largest ball contained in the grasp wrench hull centered at the origin.

Another method for quantifying the grasp quality is to compute the volume of the grasp wrench hull $\tilde{\mathcal{W}}$. This approach provides more of an average-case metric rather than a worst-case metric, and can help differentiate between different grasp spaces that have the same worst-case metric. For example Figure 5.13 shows a grasp with the same worst-case metric as the grasp in Figure 5.12, but which would be considered worse with respect to the volumetric (average-case) metric.

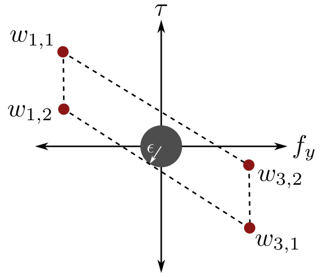

Figure 5.13: Using the volume of $\tilde{\mathcal{W}}$ as a metric for grasp quality can help differentiate between two grasps with the same worst-case performance. For example this grasp would be considered less robust than the grasp in Figure 5.12 since it has smaller volume.

5.3 *Grasp Force Optimization*

Recall from Section 5.1 that for a particular contact model a grasp map matrix G can be defined such that:

$$\boldsymbol{w} = G \begin{bmatrix} f_1 \\ \vdots \\ f_k \end{bmatrix}, \tag{5.4}$$

where $\boldsymbol{f}_i$ is the force vector associated with contact point i and $\boldsymbol{w}$ is the total wrench applied to the object. Additionally, recall that the matrix G also includes a rotational transformation such that the force vectors $\boldsymbol{f}_i$ are in local contact reference frames but $\boldsymbol{w}$ is represented in a common frame fixed in the object body (at the center of mass).

The next logical question to ask is how to compute force vectors $\{\boldsymbol{f}_i\}_{i=1}^k$ to achieve a desired wrench $\boldsymbol{w}$[16]. While one naive approach would be to just solve (5.4) using a least-squares method, this would fail to account for any constraints on the force vectors. In particular, this section will focus on the point contact with friction model where each force vector $\boldsymbol{f}_i = [f_x^{(i)}, f_y^{(i)}, f_z^{(i)}]^T$ must satisfy the friction cone constraint:

$$\sqrt{{f_x^{(i)}}^2 + {f_y^{(i)}}^2} \leq \mu_{s,i} f_z^{(i)}, \quad f_z^{(i)} \geq 0,$$

where $\mu_{s,i}$ is the static friction coefficient for contact i. In this section the compact notation:

$$\boldsymbol{f}_i \in \mathcal{F}_i, \quad \mathcal{F}_i := \{\boldsymbol{f} \in \mathbb{R}^3 \mid \sqrt{f_x^2 + f_y^2} \leq \mu_{s,i} f_z, \quad f_z \geq 0\},$$

for the friction cone constraint will again be used. It might also be desirable to include additional constraints on the force vectors, for example to account for hardware limitations (e.g. torque limits) or kinematic constraints. These additional constraints will be generally referred to by a convex constraint set C, such that $\boldsymbol{f}_i \in C$ is required for all $i = 1, \ldots, k$.

To summarize, the problem is to find a set of force vectors $\{\boldsymbol{f}_i\}_{i=1}^k$ such that (5.4) is satisfied[17] and such that $\boldsymbol{f}_i \in \mathcal{F}_i$ and $\boldsymbol{f}_i \in C$ for $i = 1, \ldots, k$. This problem can then be solved by formulating it as a convex optimization problem[18,19]:

$$\begin{aligned} \underset{\boldsymbol{f}_i,\ i\in\{1,\ldots,k\}}{\text{minimize}} \quad & J(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_k), \\ \text{s.t.} \quad & \boldsymbol{f}_i \in \mathcal{F}_i, \quad i = 1, \ldots, k, \\ & \boldsymbol{f}_i \in C, \quad i = 1, \ldots, k, \\ & \boldsymbol{w} - G\begin{bmatrix} \boldsymbol{f}_1 & \ldots & \boldsymbol{f}_k \end{bmatrix}^T = 0, \end{aligned} \tag{5.5}$$

where $J(\cdot)$ is the objective function that is convex in each $\boldsymbol{f}_i$. Simply choosing $J = 0$ would result in a convex feasibility problem, but a more common choice is:

$$J(\boldsymbol{f}_1, \ldots, \boldsymbol{f}_k) = \max\{\|\boldsymbol{f}_1\|, \ldots, \|\boldsymbol{f}_k\|\},$$

which is the maximum applied force magnitude among all contact points.

Note that the fundamental disadvantage of this optimization-based approach is that the positions of all contacts with respect to the object's center of mass and the object's friction coefficients are assumed to be known. It is also assumed that the desired wrench $\boldsymbol{w}$ is known!

[16] The desired wrench may be used to counter an external disturbance (to maintain equilibrium) or to *manipulate* the object.

[17] In the case that the desired wrench is used to counter an external disturbance, the condition (5.4) is referred to as the equilibrium constraint.

[18] S. Boyd and B. Wegbreit. "Fast Computation of Optimal Contact Forces". In: *IEEE Transactions on Robotics* 23.6 (2007), pp. 1117–1132

[19] The problem is technically a second-order cone program because of the friction cone constraints.

5.4 *Learning-Based Approaches to Grasping*

Model-based methods for grasp evaluation and optimization require several assumptions that may be either difficult to validate in practice, or may not even be valid in all scenarios. These assumptions include:

1. A coulomb (static) friction model defines the friction cone, and the coefficient μ_s is known.

2. The object's geometry and mass distribution is known[20], such that given a contact point the vector $\boldsymbol{d}_i$ from the object's center of mass to the contact is known.

3. The object is a rigid body.

4. The desired forces $\boldsymbol{f}_i$ can be applied perfectly.

Learning-based methods for grasp analysis[21] can leverage data to decrease reliance on these assumptions, for example by not requiring explicit knowledge of the object's physical parameters. Learning-based methods can also combine the task of grasping with other parts of the manipulation pipeline, such as perception and motion planning.

This section will introduce some recent learning-based approaches to robotic grasping, which is still a very active area of research. Specifically, these examples will demonstrate several learning-based strategies including approaches that create synthetic training data from model-based simulators and approaches that use real hardware to generate data.

5.4.1 *Choosing a Grasp Point from an RGB Image*[22]

The objective of this supervised learning approach was to learn how to find a good grasp point in an RGB image of an object, and then generate a prediction of the point's 3D position. Since supervised learning techniques can require a lot of training data, this approach auto-generated training images *synthetically* using realistic rendering techniques (see Figure 5.14). The use of synthetic data also made it easier to collect a diverse training set including images with different lighting, object color, and object orientation and size.

Once a model was trained to produce good grasp point classifications, 3D predictions of the target grasp position were generated by *structure-from-motion*, where two images were used to triangulate the point in space. While there are certainly limitations to this approach, this work produced promising results, including good grasp success rates on novel objects (that weren't included in the training dataset). This work also had substantial influence on future learning-based grasping and manipulation approaches.

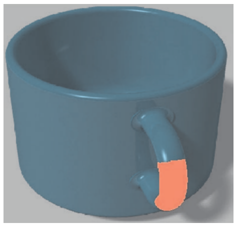

Figure 5.14: Synthetic image of a cup and its labeled grasp point from Saxena et al. (2008).

[20] One option would be to build a database of known objects, but this may not be scalable to real world problems.

[21] J. Bohg et al. "Data-Driven Grasp Synthesis—A Survey". In: *IEEE Transactions on Robotics* 30.2 (2014), pp. 289–309

[22] A. Saxena, J. Driemeyer, and A. Ng. "Robotic Grasping of Novel Objects using Vision". In: *The International Journal of Robotics Research* 27.2 (2008), pp. 157–173

5.4.2 *Exploiting Simulation and a Database of Objects to Predict Grasp Success*[23]

The Dex-Net approach for learning to grasp is another supervised learning approach that relies on simulations to generate training data. Specifically this approach assumes a parallel jaw gripper and contains a large ($> 10,000$) database of 3D object models. For each model in the Dex-Net database, the simulator uses analytical (model-based) techniques to evaluate a large number of potential grasps for probability of success[24]. This is accomplished by empirically sampling a grasp some number of times and determining (through simulation) the percentage that result in force closure[25]. The database of objects and potential grasps can then be used to train a model to predict the probability of force closure for a new grasp/object.

In practice a number of candidate grasps are generated for a given (potentially novel) object, are evaluated by the learned model to predict probability of success, and then a *multi-armed bandit*[26] approach is used to select which grasp to take. The learned models are then updated based on the outcome of the action for continuous improvement. This work showed that leveraging the prior information from the object database can significantly improve grasping for new objects (even if they are not in the database), and later improvements have enhanced the approach even further.

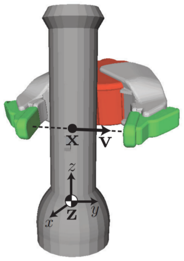

Figure 5.15: Dex-Net grasps are parameterized by the centroidal position of the gripper x and the approach direction v, Mahler et al. (2016).

5.4.3 *Learning to Grasp Through Real-World Trial-and-Error*[27]

Instead of leveraging simulators to generate synthetic data this work uses hardware experiments to generate real-world data. The resulting experiences are then used in a self-supervised approach to learn an *end-to-end* framework to grasp objects in cluttered environments. One of the reasons this work is significant is the lack of assumptions that are made: 3D object models are not needed, only RGB images are required, it does not use contact models or simulated data, no physical object information is used, and no hand-engineered path/grasp planning algorithms are used. Instead the system just learns through trial-and-error, exploring approaches to actuate the robot arm and gripper that eventually lead to robust grasps.

This approach showed impressive results over hand-designed or open-loop approaches, but at the cost that it took six months and a large number of robots to generate enough training data.

5.5 *Learning-Based Approaches to Manipulation*

The previous sections of this chapter have focused on the problem of grasping, but many robotic manipulation tasks involve more than simply grasping an object. For example it is possible to manipulate objects *without* force closure grasps, such as by pushing the object[28]. Many manipulation tasks that do involve grasping also involve other complex steps, such as using the grasped object to manipulate other objects (e.g. hitting a hammer with a nail) or placing

[23] J. Mahler et al. "Dex-Net 1.0: A cloud-based network of 3D objects for robust grasp planning using a Multi-Armed Bandit model with correlated rewards". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 1957–1964

[24] The Dex-Net database consists of 2.5 million tested grasps.

[25] There is simulated uncertainty in object and gripper pose, as well as the surface friction.

[26] A fundamental reinforcement learning problem focused on uncertain decision making.

[27] S. Levine et al. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection". In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 421–436

[28] Not only is it possible, but may be necessary if the object is too large or heavy to grasp.

the object in a certain position (e.g. inserting a key into a lock). This section will introduce at a high-level some interesting and foundational problems in manipulation, and present the high-level ideas found in some recent research on learning-based approaches to solving them.

5.5.1 *Planar Pushing*

Planar pushing is a fundamental manipulation task where the goal is to control the pose of an object in a 2D setting by only using "pushing" contacts. While the contact point models used for grasping can also be applied in this problem, the interaction of the object and the surface must also be accounted for.

Similar to the physics-based contact models for grasping, physics-based models can also be developed to predict the sliding interactions between the object and the surface. In particular, the concept of a *friction limit surface*[29] can be used to model the interaction between the object and the surface. The friction limit surface is a boundary in wrench space that separates wrenches that the surface can apply to the object through friction and those it can't. The part of the wrench space enclosed by this surface will contain the origin (i.e. the zero wrench), and most importantly whenever the object is *slipping* the wrench applied on the object lies *on the friction limit surface*. This surface can be determined numerically if the coefficient of friction, the contact area, and the pressure distribution are known. For simplicity, it is common to approximate this surface as an ellipsoid. To summarize:

1. If an external wrench applied to the object is within the region of the wrench space enclosed by the friction limit surface, friction between the object and the surface will cause the object to remain motionless.

2. If the part slides quasistatically[30], the pushing wrench must lie on the friction limit surface and the motion (velocity) of the object can be determined.

The friction limit surface provides the foundation for a physics-based model that predicts how an object will slide across a surface under external contact forces. Such a model could be used to design a controller (e.g. with model predictive control) for planar pushing tasks. However, these physics-based models are based on approximations and assumptions that may impact their accuracy or applicability to real problems. In fact some studies have been performed to evaluate the accuracy of physics-based pushing models[31].

While physics-based controllers such as model predictive control can handle some uncertainty via feedback control mechanisms, it is still desirable to improve the modeling accuracy and eliminate assumptions requiring knowledge of the parameters that define the models[32]. Learning-based approaches are one possible solution to some of these challenges, where real-world data can be used to either completely replace or augment the physics-based models.

In fact, recent work[33] has compared the use of physics-based, hybrid (physics + learning), and learning-based models for planar pushing tasks. In this work

Assumptions in physics-based pushing model: ellipsoidal friction limit surface, coulomb friction, perfectly planar object/surface, rigid body object, physical properties of object are known.

[29] I. Kao, K. Lynch, and J. Burdick. "Contact Modeling and Manipulation". In: *Springer Handbook of Robotics*. Springer, 2016, pp. 931–951

[30] Assumption that the part moves slowly enough that inertial effects are negligible.

[31] K. Yu et al. "More than a million ways to be pushed. A high-fidelity experimental dataset of planar pushing". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 30–37

[32] For example the physical properties of the object and surface.

[33] A. Kloss, S. Schaal, and J. Bohg. "Combining learned and analytical models for predicting action effects from sensory data". In: *The International Journal of Robotics Research* (2020)

the hybrid model learned a mapping from sensor measurements (RGB images) into a set of parameters that were required for the physics-based motion model and the learning-based model just directly learned a single neural network for mapping sensor measurements to motion predictions directly. As might be expected the hybrid approach achieved better generalization by leveraging the physics-based model's structure, while the learning-based approach overfit to the training data[34].

5.5.2 Contact-Rich Manipulation Tasks

Many 6D manipulation problems involve grasping an object and then using it to physically interact with the environment. Classic everyday examples include hitting a nail with a hammer, inserting a key into a lock, and plugging a cord into an electrical outlet. These types of *contact-rich* tasks tasks typically rely on multiple different sensing modalities including haptic and visual feedback. Consider the task of inserting a key into a lock: without sight it would be challenging to correctly position the key and without tactile sensing (e.g. force/torque sensing) it would be challenging to know when the key is perfectly aligned and can be inserted.

However, it can be quite challenging to *integrate* multiple sensing modalities toward a common task, especially when the sensing modalities are so different and since manipulation tasks can be quite complex. One approach may be to individually develop systems for different subtasks and manually find a common interface to stitch them together, however this could be challenging from a system engineering perspective. An alternative is to use machine learning techniques to automatically integrate the sensing modalities.

One learning-based approach to this problem is to design an end-to-end system that takes as input all sensor data streams and outputs actions for the robot to execute the task. However, when implemented in a naive way (e.g. a single massive neural network architecture) end-to-end approaches can be data inefficient. An alternative is to add additional structure to the learning-based approach by leveraging some insights into the problem, similar to how the physics-based motion model was used in the learning-based planar pushing example discussed in the previous section.

A structured approach for manipulation tasks relying on multiple sensing modalities is introduced by Lee et al.[35]. In this work an end-to-end system that takes sensor data streams as input and outputs robot actions is split into two parts: first transforming the multi-modal sensor data streams into a low-dimensional feature representation that contains task relevant information[36], and then using these features as the input to a learned policy that generates robot actions. In other words, the insight is that the learning process can be made more efficient by first learning a way to compress and summarize all of the sensor data, and then learning how to use the summarized information to generate a good policy. Another benefit to this approach is that the sensor

[34] This is a classic example of bias-variance tradeoff in modeling.

[35] M. Lee et al. "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks". In: *International Conference on Robotics and Automation (ICRA)*. 2019, pp. 8943–8950

[36] This is accomplished by training an autoencoder network.

data *encoder* can generalize more effectively to new tasks, meaning that only the policy portion needs to be retrained!