Fast Computation of Optimal Contact Forces

Stephen P. Boyd, *Fellow, IEEE*, and Ben Wegbreit, *Member, IEEE*

***Abstract*—We consider the problem of computing the smallest contact forces, with point-contact friction model, that can hold an object in equilibrium against a known external applied force and torque. It is known that the force optimization problem (FOP) can be formulated as a semidefinite programming problem (SDP) or a second-order cone problem (SOCP), and thus, can be solved using several standard algorithms for these problem classes. In this paper, we describe a custom interior-point algorithm for solving the FOP that exploits the specific structure of the problem, and is much faster than these standard methods. Our method has a complexity that is linear in the number of contact forces, whereas methods based on generic SDP or SOCP algorithms have complexity that is cubic in the number of forces. Our method is also much faster for smaller problems. We derive a compact dual problem for the FOP, which allows us to rapidly compute lower bounds on the minimum contact force and certify the infeasibility of a FOP. We use this dual problem to terminate our optimization method with a guaranteed accuracy. Finally, we consider the problem of solving a family of FOPs that are related. This occurs, for example, in determining whether force closure occurs, in analyzing the worst case contact force required over a set of external forces and torques, and in the problem of choosing contact points on an object so as to minimize the required contact force. Using dual bounds, and a warm-start version of our FOP method, we show how such families of FOPs can be solved very efficiently.**

***Index Terms*—Convex optimization, force closure, friction cone, grasp force, interior-point method, second-order cone program (SOCP).**

I. Introduction

A FUNDAMENTAL problem in robotics is choosing a set of grasping (contact) forces for an object (see, e.g., the survey [1]). The most basic requirement is the ability to restrain an object against a specified external wrench [2], such as that due to gravity. A generalization is the ability to resist external wrenches in a "task wrench space" [3] or any wrench due to a force applied at the boundary of the object [4]. The ability to resist an arbitrary external wrench is called force closure [5], [6]. In this paper, we first focus on the basic requirement, i.e., the ability of the contact forces to resist a specified external wrench; we then show how to efficiently handle some of these generalized contact force requirements, such as determining force closure.

Among the contact forces that can hold the object in equilibrium against the external wrench, we seek one with minimum force, as measured by the maximum magnitude of the contact forces. The problem of finding such a set of forces is the *force optimization problem* (FOP). Early papers on this topic include [2] and [7], where the FOP is formulated (approximately) as a linear program (LP), by approximating the friction force limit constraints as linear inequalities; other papers taking the same approach include [8] and [9]. Mishra [10] describes several types of FOPs, algorithms for determining feasible or optimal forces, and points out the underlying convexity of the problem. Other researchers have proposed neural network methods [11], [12], probabilistic algorithms [13], and various other methods.

The FOP comes up in several applications and settings, such as grasp optimization (which might involve the selection of the contact points as well as forces) [14]–[18], or real-time grasp control [19], [20], and force optimization for the legs of a quadruped robot [21]. Some experimental results for force optimization are reported in, e.g., [22].

In the late 1990s, several exact formulations of the FOP were obtained by expressing it as a convex optimization problem involving matrix inequalities or second-order cone inequalities. The FOP is expressed in [24]–[27] using (linear) matrix inequalities, so the resulting optimization problem is a semidefinite programming problem (SDP). The FOP is formulated as a second-order cone problem (SOCP) in [28].

These formulations reduce the problem to (what is now) a standard convex optimization problem. This means the problems can be solved, globally and efficiently, by a variety of methods for nonlinear convex optimization developed in the 1990s (see, e.g., [29]). General-purpose LP, SOCP, and SDP software is now widely available (see, e.g., [30] for comparative benchmarking of some recent codes). These solvers can reliably and efficiently solve FOPs. A typical FOP, with five contact points and one external wrench, can be solved in well under a second, on the order of 100 ms on a current typical desktop PC (for example, a 3-GHz Pentium IV). Several authors have developed custom solvers for the FOP that are faster than generic SDP or SOCP solvers. Buss *et al*. [25] developed a Dikin-type algorithm for the FOP, and Helmke *et al*. [27] developed a quadratically convergent algorithm. Our research is similar in spirit to these.

The need to solve the FOP quickly arises in several applications, for example, when the FOP must be solved many times. Suppose we are given a lower and an upper bound on the external force and torque components, i.e., a box in wrench space, and wish to find the maximum value of minimum contact force required to resist any wrench in this wrench box. (This includes the more basic problem of determining whether or not each wrench in the box can be resisted by some contact forces; by finding the maximum value of minimum force required, we obtain a quantitative measure of the ability to resist wrenches in the box.) We can do this by solving the FOP for each of the $2^6 = 64$ vertices of the wrench box. The maximum of the optimal forces over these vertices is, in fact, the maximum contact force required over all wrenches in the box, since the optimal

Manuscript received May 25, 2007. This paper was recommended for publication by Associate Editor F. Lamiraux and Editor F. Park upon evaluation of the reviewers' comments.

S. P. Boyd is with the Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA 94305-9510, USA (e-mail: boyd@stanford.edu).

B. Wegbreit is with Strider Labs, Palo Alto, CA 94303, USA (e-mail: wegbreit@ix.netcom.com).

Digital Object Identifier 10.1109/TRO.2007.910774

force required is a convex function of the wrench. So, here, we have an example where 64 FOPs must be solved. A similar example is provided by the problem of determining force closure. As we will see in Section II-F, this can be done by solving a set of seven (or more) FOPs.

As another example where many FOPs must be solved, consider the problem of optimizing the position and orientation of a manipulator, relative to an object, using the minimum force required to grasp the object as the objective (see, e.g., [6]). This is the (nonconvex) problem of optimizing the contact points at which to grasp a given object. This can be done using an outer search loop that generates candidate manipulator positions and orientations; for each candidate, we find the resulting contact points, and then, solve the associated FOP to determine the minimum force required to grasp the object. Such an algorithm can require the solution of hundreds of FOPs.

Applications that involve the solution of many FOPs, such as finding the worst case contact force over a wrench box or contact point optimization, benefit directly from a very fast FOP solver like the one we describe in this paper. We will also consider methods for obtaining even more efficiency when solving a family of related FOPs, using warm-start optimization techniques, and "short-circuiting," i.e., early termination.

In this paper, we describe a method for solving the FOP that is substantially faster than a general-purpose solver for SDP or SOCP. Despite the speed of our method, its termination is entirely nonheuristic; our method terminates with a provable bound on the suboptimality of the computed contact forces (when the problem is feasible), or with a certificate proving infeasibility of the problem, when the problem is infeasible.

We obtain the speedup by exploiting special structure in the FOP to compute the search direction in each iteration of an interior-point method. The computational effort of our method scales linearly with M, the number of contact points; for a generic SDP or SOCP, however, the computational effort grows as M^3. Our method is much faster for small problems as well. For a typical grasping problem with $M = 5$ contact points, our method solves the FOP in around $400\ \mu$s, on a 3 GHz Pentium IV, with a not particularly optimized C++ implementation. This means that 2500 FOPs can be solved in 1 s. By exploiting warm-start techniques, the typical time to solve a FOP drops to around $200\ \mu$s, many hundreds of times faster than generic SDP or SOCP solvers.

It is difficult to make a direct speed comparison between our method and those of Buss *et al*. [25] and Helmke *et al*. [27], since these methods have different stopping criteria, not based on an explicit dual bound, and the run time depends very much on the details of the implementation. In fact, many of the methods in this paper can be used in both of these algorithms. For example, our dual-based stopping criterion is easily used in both algorithms; moreover, our method for a fast solution of the linear equations that must be solved in each step can be used to speed up the iterations in the Dikin method as well. In any case, there is not much room for significant speed improvement over our method, by any other method at all, since our method requires around eight or so iterations, each of which entails a few hundred M floating point operations. To put this in context, we note that our method requires only around a factor of 10 times more effort than simply verifying that equilibrium holds.

The outline of this paper is as follows. In Section II, we describe the basic contact FOP, formulated as a conic problem, i.e., a convex optimization problem with linear objective and equality constraints, and convex cone constraints on the variables. In Section III, we derive a compact dual problem for the FOP, which allows us to rapidly compute lower bounds on the minimum grasping force and certify the infeasibility of a FOP. We use this dual problem to terminate our optimization method with a guaranteed accuracy. In Section IV, we describe the barrier subproblem associated with a primal interior-point method for the FOP, and in Section V, we show how the special structure of the FOP can be exploited to compute the search direction very efficiently. We describe the overall algorithm in Section VI, and methods for efficiently solving a family of FOPs in Section VII. We give numerical results in Section VIII, and describe some variations and extensions on the problem in Section IX.

Finally, we describe our (fairly standard) notation. We denote the set of real numbers by $\mathbf{R}$, the set of real n-vectors as $\mathbf{R}^n$, and the set of real $m \times n$ matrices as $\mathbf{R}^{m\times n}$. We use the notation (a, b, c) (for example) to denote a column vector with components, and c, which we also write as

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix}.$$

When a, b, and c are reals, this is a vector in $\mathbf{R}^3$. We use the same notation when a, b, and c are themselves (column) vectors, in which case (a, b, c) is the column vector obtained by stacking a on top of b on top of c.

II. Grasp Force Optimization Problem

The rigid object is grasped at M contact points, which have positions $p^{(i)} \in \mathbf{R}^3$ (in the global coordinate system), for $i = 1, \ldots, M$. We will use a point contact with friction model. (In Section IX-C, however, we explain how our methods can be extended to handle other friction models, e.g., a soft contact point that can exert a torque on the object.) The force applied at a contact point $p^{(i)}$ will be denoted by $f^{(i)} \in \mathbf{R}^3$, and is given in a *local* coordinate system, with x-axis and y-axis tangent to the object surface at $p^{(i)}$, and z-axis normal to the object surface at $p^{(i)}$ and pointing inward. We denote its components as $f^{(i)} = (f_x^{(i)}, f_y^{(i)}, f_z^{(i)})$. Thus, $f_z^{(i)}$ is the normal (inward) force applied at contact point $p^{(i)}$, and $(f_x^{(i)}, f_y^{(i)})$ is the tangential force applied at contact point $p^{(i)}$.

A. Friction Cone Constraints

The point contact friction model requires the contact forces to satisfy the *friction cone constraints*

$$\|(f_x^{(i)}, f_y^{(i)})\| = \sqrt{f_x^{(i)2} + f_y^{(i)2}} \leq \mu_i f_z^{(i)}, \qquad i = 1, \ldots, M \tag{1}$$

where $\mu_i > 0$ is the friction coefficient at contact point $p^{(i)}$. This constraint states that the magnitude of the tangential force

does not exceed the friction coefficient times the normal force. (In particular, it implies that the normal forces $f_z^{(i)}$ must be nonnegative.) The friction cone constraints are *second-order cone constraints* [29, Sec. 4.4.2] [28]. We introduce the *friction cones* $K_1, \ldots, K_M \subseteq \mathbf{R}^3$, defined as

$$K_i = \left\{ x \in \mathbf{R}^3 \;\middle|\; \sqrt{x_1^2 + x_2^2} \leq \mu_i x_3 \right\}, \qquad i = 1, \ldots, M. \tag{2}$$

Using this notation, we can express the friction cone constraints (1) compactly as

$$f^{(i)} \in K_i, \qquad i = 1, \ldots, M. \tag{3}$$

B. Equilibrium Constraints

Let $Q^{(i)} \in \mathrm{SO}(3)$ be the 3×3 orthogonal matrix that transforms forces in the local coordinate system at $p^{(i)}$ into the global coordinate system. Thus, the force on the object due to the contact force at $p^{(i)}$ is $Q^{(i)} f^{(i)}$ (in the global coordinate system). The force equilibrium condition is

$$Q^{(1)} f^{(1)} + \cdots + Q^{(M)} f^{(M)} + f^{\mathrm{ext}} = 0 \tag{4}$$

where $f^{\mathrm{ext}} \in \mathbf{R}^3$ is the total external force that acts on the object (in the global coordinate system).

The torque applied to the object by the force at contact point $p^{(i)}$ is given by

$$p^{(i)} \otimes Q^{(i)} f^{(i)}$$

(in the global coordinate system). The torque equilibrium condition is

$$p^{(1)} \otimes Q^{(1)} f^{(1)} + \cdots + p^{(M)} \otimes Q^{(M)} f^{(M)} + \tau^{\mathrm{ext}} = 0$$

where $\tau^{\mathrm{ext}} \in \mathbf{R}^3$ is the total external torque that acts on the object (in the global coordinate system). Then, we can write the torque equilibrium condition as

$$S^{(1)} Q^{(1)} f^{(1)} + \cdots + S^{(M)} Q^{(M)} f^{(M)} + \tau^{\mathrm{ext}} = 0 \tag{5}$$

where

$$S^{(i)} = \begin{bmatrix} 0 & -p_z^{(i)} & p_y^{(i)} \\ p_z^{(i)} & 0 & -p_x^{(i)} \\ -p_y^{(i)} & p_x^{(i)} & 0 \end{bmatrix} \in \mathrm{skew}(3)$$

is the skew-symmetric matrix that satisfies $S^{(i)} x = p^{(i)} \otimes x$.

We now introduce some more compact notation for the equilibrium constraints. We define the *contact force vector* $f \in \mathbf{R}^{3M}$ as

$$f = (f^{(1)}, \ldots, f^{(M)})$$

the vector of all the contact forces. We define *contact matrices* $A_i \in \mathbf{R}^{6 \times 3}$ as

$$A_i = \begin{bmatrix} Q^{(i)} \\ S^{(i)} Q^{(i)} \end{bmatrix}, \qquad i = 1, \ldots, M$$

and the (overall) *contact matrix* $A = [A_1 \cdots A_M] \in \mathbf{R}^{6 \times 3M}$. We collect the external force and torque into a single external wrench $\omega^{\mathrm{ext}} = (f^{\mathrm{ext}}, \tau^{\mathrm{ext}}) \in \mathbf{R}^6$. Note that $A^{(i)} f$ gives the wrench on the object due to the force at contact point $p^{(i)}$ and Af gives the total wrench on the object from all the contact forces. Thus, we can write the equilibrium conditions (4) and (5) as

$$Af + \omega^{\mathrm{ext}} = 0. \tag{6}$$

This is a set of six linear equations in the $3M$ contact forces f.

C. Contact Force Constraints

In many practical problems, there are further constraints on the contact forces due to actuator limits, kinematic constraints, and other limits, which we can describe as

$$f \in \mathcal{C}^{\mathrm{other}}.$$

These constraints are typically convex, and often polyhedral, i.e., $\mathcal{C}^{\mathrm{other}}$ is described by a set of linear equalities and inequalities on the contact forces. In the main development of this paper, we will ignore these constraints; but we will make occasional comments about how our methods can be extended to handle these constraints.

D. Force Optimization Problem

The FOP is to find a set of contact force vectors $f^{(i)}$'s that are as small as possible, while satisfying the friction cone constraints (3) and the force and torque equilibrium conditions (6). We will measure the size of the set of contact forces by the maximum magnitude of the M contact forces:

$$\begin{aligned} F^{\max} &= \max\{\|f^{(1)}\|, \ldots, \|f^{(M)}\|\} \\ &= \max_{i=1,\ldots,M} \sqrt{f_x^{(i)2} + f_y^{(i)2} + f_z^{(i)2}}. \end{aligned}$$

Our problem can be expressed as

$$\begin{array}{ll} \text{minimize} & F^{\max} \\ \text{subject to} & f^{(i)} \in K_i, \qquad i = 1, \ldots, M \\ & Af + \omega^{\mathrm{ext}} = 0. \end{array} \tag{7}$$

The optimization variables are the contact forces, i.e., $f \in \mathbf{R}^{3M}$; the problem data are A (which describe the geometry of the contact points), ω^{ext} (the externally applied wrench), and K_i (the friction cones, which depend on problem data μ_i, the friction coefficients). We note for future reference that A has rank 6, provided $M \geq 3$, and the contact points are not collinear (which we assume).

The problem (7) is convex, since the objective is a convex (although nondifferentiable) function of f, and the constraints consist of linear equalities and (convex) cone constraints. The problem remains convex if we add additional (convex) force constraint $f \in \mathcal{C}^{\mathrm{other}}$.

E. Conic Formulation

The FOP (7) can be expressed in conic form, indeed, as an SOCP, by introducing a new scalar variable F that bounds the magnitude of the contact forces:

$$\begin{array}{ll} \text{minimize} & F \\ \text{subject to} & (f^{(i)}, F) \in K, \qquad i = 1, \ldots, M \\ & f^{(i)} \in K_i, \qquad i = 1, \ldots, M \\ & Af + \omega^{\mathrm{ext}} = 0. \end{array} \tag{8}$$

Here, K denotes the standard second-order cone in $\mathbf{R}^4$

$$K = \left\{ x \in \mathbf{R}^4 \;\middle|\; \sqrt{x_1^2 + x_2^2 + x_3^2} \leq x_4 \right\}$$

so $(f^{(i)}, F) \in K_i$ means

$$\sqrt{f_x^{(i)2} + f_y^{(i)2} + f_z^{(i)2}} \leq F.$$

In the problem (8), there are $3M + 1$ (scalar) variables: the contact forces $f \in \mathbf{R}^{3M}$ and F, the new scalar variable introduced to bound the maximum force magnitude.

Although we will not use the SDP formulation, we mention that it is readily derived from the conic formulation of the FOP, by expressing the second-order cone constraints as linear matrix inequalities (LMIs). For example, we can express $f^{(i)} \in K_i$ as the LMI

$$\begin{bmatrix} \mu_i f_z^{(i)} & 0 & f_x^{(i)} \\ 0 & \mu_i f_z^{(i)} & f_y^{(i)} \\ f_x^{(i)} & f_y^{(i)} & \mu_i f_z^{(i)} \end{bmatrix} \succeq 0$$

where $\succeq$ denotes matrix inequality. This gives the SDP (or LMI) formulation of the FOP [26]. A more compact formulation, which uses only 2×2 LMIs, is given in [27].

F. Force Closure

Force closure occurs if, for every wrench $\omega \in \mathbf{R}^6$, there exists a set of contact forces $f^{(i)} \in K_i$ for which $Af + \omega = 0$. We can describe this condition in terms of cones. Define

$$W = \{-Af \mid f^{(i)} \in K_i\} \tag{9}$$

which is the cone of wrenches that can be resisted. Force closure is equivalent to $W = \mathbf{R}^6$.

We can determine force closure by solving a set of FOPs. Let $\omega_1, \ldots, \omega_K \in \mathbf{R}^6$ be a set of wrenches with 0 in the interior of their convex hull, i.e., there exist positive $\lambda_1, \ldots, \lambda_K$ such that

$$\lambda_1 \omega_1 + \cdots + \lambda_K \omega_K = 0. \tag{10}$$

(The minimum possible value of K is 7.) Force closure occurs if and only if each of these wrenches can be resisted, i.e., the associated K FOPs are all feasible.

If any of these FOPs is infeasible, then evidently force closure does not occur. On the other hand, suppose that each of these K FOPs is feasible, with optimal force vectors $f_1, \ldots, f_K \in \mathbf{R}^{3M}$. Given *any* $\omega \in \mathbf{R}^6$, we can write it as a nonnegative linear combination of the vectors $\omega_1, \ldots, \omega_K$:

$$\omega = \beta_1 \omega_1 + \cdots + \beta_K \omega_K, \qquad \beta_1, \ldots, \beta_K \geq 0.$$

It follows that the force vector

$$f = \beta_1 f_1 + \cdots + \beta_K f_K$$

is feasible, i.e., satisfies the friction cone constraints and resists the external wrench ω.

We describe two simple methods for constructing a set of wrenches with the required property. The first method constructs a minimal set, i.e., a set of seven wrenches. In particular, it constructs the wrenches as the vertices of a regular simplex in $\mathbf{R}^6$, centered at 0, so the sum of the wrenches is zero. Moreover, the wrenches all have unit norm, and are maximally equidistant on the unit sphere.

We start with the 7×7 matrix $Z = I - (1/7)\mathbf{1}\mathbf{1}^T$, where $\mathbf{1} \in \mathbf{R}^7$ is the vector with all entries one. This matrix has value 6/7 on its diagonal and $-1/7$ for each off-diagonal entry. Its eigenvalues are 0 (with multiplicity 1) and 1 with multiplicity 6. Therefore, we can factor it as $V^T V = Z$, where $V \in \mathbf{R}^{6\times 7}$ (for example, using the eigenvalue decomposition). Let $v_1, \ldots, v_7$ denote the seven columns of V. From $V^T V = Z$, we conclude that $\|v_i\| = \sqrt{6/7}$, and that $v_i^T v_j = -1/7$ for $i \neq j$, so $\angle(v_i, v_j) = \cos^{-1}(-1/6) \approx 100^\circ$ for $i \neq j$. We can take $\omega_i = \sqrt{7/6} v_i$.

There are several ways to express a general ω as a nonnegative linear combination of $\omega_1, \ldots, \omega_K$. One method is to first express ω as a linear combination

$$\omega = \hat{\beta}_1 \omega_1 + \cdots + \hat{\beta}_K \omega_K$$

for example, as $\hat{\beta} = [\omega_1 \cdots \omega_K]^\dagger \omega$, where $\dagger$ denotes pseudo-inverse. Then, we take $\beta = \hat{\beta} + \rho\lambda$, where λ is from (10), and ρ is chosen large enough such that all entries of β are nonnegative.

The second method we describe involves a set of 12 wrenches, but has the advantage that no computation is involved in expressing a general wrench as a positive linear combination of the given wrenches. The wrenches are

$$\pm e_1, \ldots, \pm e_6$$

where e_i is the ith unit vector in $\mathbf{R}^6$. The sum of these wrenches is zero, so the condition (10) holds with $\lambda_i = 1$. Suppose we solve the 12 associated FOPs, which are all feasible (so force closure occurs). Let f_i^+ be an optimal contact force for wrench $\omega^{\text{ext}} = e_i$, and let f_i^- be an optimal contact force for wrench $\omega^{\text{ext}} = -e_i$. Once we have computed this set of 12 contact forces, we can very easily construct a feasible contact force vector for *any* wrench $\omega \in \mathbf{R}^6$, as follows:

$$f = (\omega_1)_+ f_1^+ + (\omega_1)_- f_1^- + \cdots + (\omega_6)_+ f_6^+ + (\omega_6)_- f_6^- \tag{11}$$

where $(u)_+ = \max\{0, u\}$ and $(u)_- = \max\{0, -u\}$ are the positive and negative parts of $u \in \mathbf{R}$, respectively, and here, ω_i denotes the ith component of ω. Thus, once we have solved the 12 FOPs, we can instantly generate a feasible (but suboptimal) contact force vector for any external wrench, using the formula (11) that describes a piecewise-linear mapping from ω into f, with 12 regions, i.e., the orthants in $\mathbf{R}^6$.

As a consequence, we find that the minimum grasping force required to resist the wrench ω is no more than $G\|\omega\|_1$, where G is the maximum of the optimal value of the 12 FOPs, and $\|\omega\|_1 = |\omega_1| + \cdots + |\omega_6|$. Thus, the number G, the maximum of the optimal grasping forces for the wrenches $\pm e_i$, gives a quantitative measure of force closure.

There are several other uses of the formula (11). Suppose we have computed an optimal contact force vector f for a wrench ω, and we wish to compute a contact force vector $\tilde{f}$ for a nearby wrench $\tilde{\omega}$. (In fact, $\tilde{\omega}$ does not have to be close to ω.) To do this, we use the formula (11) to find a feasible contact force vector δf that resists the external wrench $\tilde{\omega} - \omega$. Then, the contact force vector $f + \delta f$ satisfies the cone constraints, and

resists the wrench $\tilde{\omega}$. This contact force vector can be computed almost instantly; moreover, if ω and $\tilde{\omega}$ are not too far from each other, the force vector will not be too suboptimal.

We mention one important feature of (11). The original FOP is invariant under a change of the global geometric coordinate system. If we change the global coordinate system, we will change ω and the matrices A_i, and therefore, the basic FOP also. But the solution of this FOP will be the same as the solution of the original FOP. This property does not hold for (11), i.e., a change of the global coordinate system will, in general, change the force vector f computed from (11). (This is obvious since (11) depends on an expansion of the external wrench in a specific basis.) We have found that (11) gives good results when the zero coordinate of the global coordinate system is chosen to be near the contact point locations, e.g., their average.

III. Dual Problem and Infeasibility Conditions

A. Dual Force Optimization Problem

In this section, we derive a dual problem for the FOP (7). (For a general reference on Lagrange duality, see [29, Ch. 5].) We introduce Lagrange multiplier vectors $z_i \in \mathbf{R}^3$ for the friction cone constraints and Lagrange multiplier vectors $(u_i, v_i) \in \mathbf{R}^4$ with $u_i \in \mathbf{R}^3$ and $v_i \in \mathbf{R}$, for the constraints involving K, and $\nu \in \mathbf{R}^6$ for the equality constraint. The Lagrangian is then

$$\begin{aligned} &L(f, F, z, u, v, \nu) \\ &\quad = F - \sum_{i=1}^{M} z_i^T f^{(i)} - \sum_{i=1}^{M} (u_i, v_i)^T (f^{(i)}, F) \\ &\qquad + \nu^T (Af + \omega^{\mathrm{ext}}) \\ &\quad = \left(1 - \sum_{i=1}^{M} v_i\right) F - \sum_{i=1}^{M} (z_i + u_i - A_i^T \nu)^T f^{(i)} + \nu^T \omega^{\mathrm{ext}}. \end{aligned}$$

To obtain the dual function, we minimize L over the primal variables F and f. When we minimize L over F, we find that the minimum is $-\infty$, unless $\sum_{i=1}^{M} v_i = 1$. Minimizing L over $f^{(i)}$ yields $-\infty$ unless $A_i^T \nu = u_i + z_i$ for $i = 1, \ldots, M$. The dual function is, therefore, given by

$$\begin{aligned} &g(z, u, v, \nu) \\ &= \inf_{f,\, F} L(f, F, z, u, v, \nu) \\ &= \begin{cases} \nu^T \omega^{\mathrm{ext}}, & \sum_{i=1}^{M} v_i = 1, A_i^T \nu = u_i + z_i, i = 1, \ldots, M \\ -\infty, & \text{otherwise.} \end{cases} \end{aligned}$$

Thus, we can write the dual of the FOP (see, e.g., [29, Sec. 5.9]) as

$$\begin{array}{ll} \text{maximize} & \nu^T \omega^{\mathrm{ext}} \\ \text{subject to} & z_i \in K_i^*, \qquad i = 1, \ldots, M \\ & (u_i, v_i) \in K^*, \qquad i = 1, \ldots, M \\ & \sum_{i=1}^{M} v_i = 1 \\ & A_i^T \nu = u_i + z_i, \quad i = 1, \ldots, M \end{array}$$

with variables z, u, v, and ν; K^* denotes the dual of the standard second-order cone in $\mathbf{R}^4$ and K_i^* denotes the dual of the friction cone K_i. The standard second-order cone is self-dual, i.e., we have $K^* = K$; the dual friction cones are given by

$$K_i^* = \left\{ y \in \mathbf{R}^3 \;\middle|\; \sqrt{y_1^2 + y_2^2} \le (1/\mu_i) y_3 \right\}. \tag{12}$$

(This follows from the general fact that, for any cone K and any nonsingular matrix C, we have $(CK)^* = C^{-T} K^*$.) We can considerably simplify the basic Lagrange dual problem mentioned earlier, by eliminating all variables except $\nu \in \mathbf{R}^6$. Since $(u_i, v_i) \in K^* = K$ is the same as $\|u_i\| \le v_i$, we can eliminate the variables v_i and express the dual as

$$\begin{array}{ll} \text{maximize} & \nu^T \omega^{\mathrm{ext}} \\ \text{subject to} & z_i \in K_i^*, \qquad i = 1, \ldots, M \\ & \sum_{i=1}^{M} \|u_i\| \le 1 \\ & A_i^T \nu = u_i + z_i, \quad i = 1, \ldots, M. \end{array}$$

We now eliminate the variables u_i, using $u_i = A_i^T \nu - z_i$, to obtain

$$\begin{array}{ll} \text{maximize} & \nu^T \omega^{\mathrm{ext}} \\ \text{subject to} & z_i \in K_i^*, \qquad i = 1, \ldots, M \\ & \sum_{i=1}^{M} \|A_i^T \nu - z_i\| \le 1. \end{array} \tag{13}$$

Finally, we eliminate the variables z_i. The optimal z_i in the aforementioned problem is the one that minimizes $\|A_i^T \nu - z_i\|$ over $z_i \in K_i^*$, i.e., the (Euclidean) projection of $A_i^T \nu$ onto K_i^*:

$$z_i = P_{K_i^*}(A_i^T \nu)$$

and the associated minimum value is

$$\min_{z_i \in K_i^*} \|A_i^T \nu - z_i\| = \mathbf{dist}(A_i^T \nu, K_i^*)$$

the distance from $A_i^T \nu$ to K_i^*. Thus, we can write the dual FOP as

$$\begin{array}{ll} \text{maximize} & \nu^T \omega^{\mathrm{ext}} \\ \text{subject to} & \sum_{i=1}^{M} \mathbf{dist}(A_i^T \nu, K_i^*) \le 1 \end{array} \tag{14}$$

with variables $\nu \in \mathbf{R}^6$.

We can give an explicit formula for the projection of a point (x, y), with $x \in \mathbf{R}^2$ and $y \in \mathbf{R}$, onto the cone K_i^*:

$$P_{K_i^*}(x, y) = \begin{cases} (x, y), & y \ge \mu_i \|x\| \\ \beta(x, \mu_i \|x\|), & -(1/\mu_i)\|x\| \le y \le \mu_i \|x\| \\ 0, & y \le -(1/\mu_i)\|x\| \end{cases}$$

where

$$\beta = \frac{\mu_i y + \|x\|}{(1 + \mu_i^2)\|x\|}.$$

The corresponding distance is given by

$$\begin{aligned} &\mathbf{dist}((x, y), K_i^*) \\ &\quad = \begin{cases} 0, & y \ge \mu_i \|x\| \\ (\mu_i \|x\| - y)/\sqrt{1 + \mu_i^2}, & -(1/\mu_i)\|x\| \le y \le \mu_i \|x\| \\ \|(x, y)\|, & y \le -(1/\mu_i)\|x\|. \end{cases} \end{aligned} \tag{15}$$

This gives us an explicit formula for $\mathbf{dist}(A_i^T \nu, K_i^*)$, so the dual (14) is completely explicit.

The dual FOP (14) satisfies the following properties.

1) *Weak duality:* Suppose ν is dual feasible, i.e., $\sum_{i=1}^{M} \mathbf{dist}(A_i^T \nu, K_i^*) \leq 1$. Then, the dual objective value $\nu^T \omega^{\mathrm{ext}}$ is a lower bound on $F^\star$, the optimal value of the (primal) FOP (7).
2) *Strong duality:* Suppose the primal FOP (7) is *strictly feasible*, i.e., there exists (f, F) with $Af + \omega^{\mathrm{ext}} = 0$, $(f^{(i)}, F) \in \mathbf{int} K$, and $f^{(i)} \in \mathbf{int}\ K_i$. ($\mathbf{int}\ S$ denotes the interior of the set S.) Then, there exists a dual feasible $\nu^\star$ for which $\nu^{\star T} \omega^{\mathrm{ext}} = F^\star$. (In fact, $\nu^\star$ is optimal for the dual FOP (14).)

Since the dual objective $\nu^T \omega^{\mathrm{ext}}$ and the dual constraint function $\sum_{i=1}^{K} \mathbf{dist}(A_i^T \nu, K_i^*)$ are both homogeneous, we can scale any ν for which $\sum_{i=1}^{K} \mathbf{dist}(A_i^T \nu, K_i^*) \neq 0$ so that $\sum_{i=1}^{K} \mathbf{dist}(A_i^T \nu, K_i^*) = 1$. This gives the lower bound

$$F^{\mathrm{dual}} = \frac{\nu^T \omega^{\mathrm{ext}}}{\sum_{i=1}^{K} \mathbf{dist}(A_i^T \nu, K_i^*)} \leq F^\star \tag{16}$$

valid for $\sum_{i=1}^{K} \mathbf{dist}(A_i^T \nu, K_i^*) \neq 0$. (This lower bound is interesting only when $\nu^T \omega^{\mathrm{ext}} > 0$; otherwise, it gives a worse lower bound than the trivial one $0 \leq F^\star$.)

B. Sensitivity Interpretation

The optimal dual variable $\nu^\star$ is useful as a measure of the sensitivity of the optimal grasping force with respect to changes in the external wrench. Assuming that $F^\star$ is a differentiable function of ω^{ext} (which is not always the case), we have

$$\frac{\partial F^\star}{\partial \omega^{\mathrm{ext}}{}_i} = \nu_i^\star.$$

For example, $\nu_1^\star$ gives us (approximately) the change in minimum required grasping force, per Newton of increase in the x-component of the externally applied force. Thus, if $\nu^\star$ is large, small changes in externally applied wrench can lead to large changes in the minimum required grasping force.

C. Infeasibility Conditions

The FOP (7) is feasible when there exists a force vector f that satisfies the friction cone and equilibrium constraints, i.e.,

$$f^{(i)} \in K_i, \qquad i = 1, \ldots, M, \qquad Af + \omega^{\mathrm{ext}} = 0 \tag{17}$$

a set of M second-order cone constraints and six linear equality constraints. The associated set of *alternative constraints* give the conditions under which (17) is infeasible:

$$A_i^T \nu \in K_i^*, \qquad i = 1, \ldots, M, \qquad \nu^T \omega^{\mathrm{ext}} > 0 \tag{18}$$

a set of M second-order cone constraints for the dual variable $\nu \in \mathbf{R}^6$. The conditions (17) and (18) are strong alternatives: for any problem data, exactly one of them is feasible [29, Sec. 5.9.4]. Thus, feasibility of either set of conditions implies infeasibility of the other set of conditions.

It is easy to verify weak duality, i.e., that (17) and (18) cannot both be feasible. If they were both feasible, say with f and ν, then we would have

$$0 = \nu^T (Af + \omega^{\mathrm{ext}}) = \sum_{i=1}^{M} (A_i^T \nu)^T f^{(i)} + \nu^T \omega^{\mathrm{ext}} > 0$$

since $A_i^T \nu \in K_i^*$ and $f^{(i)} \in K_i$ imply $(A_i^T \nu)^T f^{(i)} \geq 0$.

The infeasibility conditions (18) are closely related to the dual FOP (14): if ν satisfies the alternative constraints (18), then for any $\alpha > 0$, $\alpha\nu$ is feasible for the dual problem (14), since $\mathbf{dist}(A_i^T (\alpha\nu), K_i^*) = 0$. Therefore, $\alpha \nu^T \omega^{\mathrm{ext}}$ is a lower bound on $F^\star$, for any $\alpha > 0$. Since this can be made arbitrarily large, we must have $F^\star = \infty$, i.e., the FOP (7) is infeasible.

We can interpret the infeasibility conditions (18) in terms of energy supplied to the object by the external wrench, when the object is displaced by a small amount. We think of the dual variable $\nu \in \mathbf{R}^6$ as representing a small displacement and rotation of the object; $\nu^T \omega^{\mathrm{ext}} > 0$ means that some positive energy is supplied to the object by the external force. The vectors $A_i^T \nu$ give the displacement, to first order, of the contact points, in the local coordinates. Since $A_i^T \nu$ and $f^{(i)}$ are in dual cones, we have $(A_i^T \nu)^T f^{(i)} \geq 0$ for any feasible set of forces. This means that if the contact forces satisfy the friction cone constraints, each one provides zero or positive energy to the object. But the total energy supplied to the object, by the external wrench and the contact forces, must be zero. We conclude that no feasible contact forces exist.

D. Force Nonclosure

We can now give similar necessary and sufficient conditions for the absence of force closure. This is the condition that the FOP is infeasible for *some* external wrench; by the earlier results, this is equivalent to

$$A_i^T \nu \in K_i^*, \qquad i = 1, \ldots, M, \qquad \nu \neq 0. \tag{19}$$

Given such a ν, we can easily construct an external wrench that cannot be resisted; for example, $\omega^{\mathrm{ext}} = \nu$. More generally, a ν that satisfies the no-force-closure conditions (19) gives us an entire half-plane of wrenches that cannot be resisted, i.e., any ω^{ext} with $\nu^T \omega^{\mathrm{ext}} > 0$.

We also note that $-\nu$ can be interpreted as a tangent plane, at 0, to the cone W of resistable wrenches [defined in (9)], i.e., for each nonzero $\omega \in W$, we have $\nu^T \omega \leq 0$. Indeed, $-\nu \in W^*$, the dual of the cone of resistable wrenches; this establishes that force closure fails, since force closure is equivalent to $W^* = \{0\}$.

IV. Barrier Subproblem

In this section, and then, next, we describe and analyze the barrier subproblem associated with the FOP (8), and Newton's method, which is used to solve it. This subproblem arises directly in interior-point methods such as primal barrier methods, including the algorithm we will describe, and also indirectly in all others, e.g., primal-dual methods. The Dikin search direction [25] can also be expressed in terms of the barrier subproblem: it is the Newton search direction for the subproblem, in the limit $t \to \infty$ (and suitably scaled).

The barrier subproblem is

$$\begin{array}{ll} \text{minimize} & tF + \phi \\ \text{subject to} & Af + \omega^{\mathrm{ext}} = 0 \end{array} \tag{20}$$

where $t > 0$ is a parameter and ϕ is the *log barrier* for the cone constraints. The log barrier is given by

$$\phi(f, F) = \sum_{i=1}^{M} \phi_i(f^{(i)}, F)$$

where

$$\begin{aligned}\phi_i(f^{(i)}, F) = &-\log(F^2 - f_x^{(i)2} - f_y^{(i)2} - f_z^{(i)2}) \\ &-\log(\mu_i^2 f_z^{(i)2} - f_x^{(i)2} - f_y^{(i)2})\end{aligned}$$

if $(f^{(i)}, F)$ satisfies the cone constraints strictly, i.e.,

$$(f(i), F) \in \mathbf{int}K, \qquad f(i) \in \mathbf{int}K_i \tag{21}$$

and $+\infty$ otherwise. The log barrier is a smooth convex function, so the subproblem (21) is a smooth convex problem. (For more on log barriers for convex cones, see [29] and [31].)

Unless stated otherwise, we will assume that the subproblem is feasible, i.e., there exists a point (f, F) that satisfies $Af + \omega^{\text{ext}} = 0$ and (21). This assumption is slightly more than assuming that the original FOP (8) is feasible; it means that the original FOP is strictly feasible. This assumption can be simplified as follows: there exists $f \in \mathbf{R}^{3M}$ that satisfies

$$Af + \omega^{\text{ext}} = 0, \qquad f^{(i)} \in \mathbf{int}K_i, \qquad i = 1, \ldots, M.$$

(We can always choose F large enough to satisfy the other cone constraints strictly.)

Before getting into details, we mention the most important facts about the barrier subproblem (20). First, it is a convex optimization problem, with smooth objective and equality constraints, so it can be solved by Newton's method. Second, it gives an approximate solution of the FOP (8): the solution (f, F) of the barrier subproblem is guaranteed to be at most $4M/t$-suboptimal for the original FOP (8).

A. Barrier Subproblem Optimality Conditions

Suppose (f, F) strictly satisfies the cone constraints, i.e., (21) holds. The *primal residual* is

$$r_p = Af + \omega^{\text{ext}} \in \mathbf{R}^6$$

which is the error in the equilibrium force and torque balance condition. The *dual residual* is

$$r_d = \begin{bmatrix} r_f \\ r_F \end{bmatrix} = \begin{bmatrix} \nabla_f \phi(f, F) + A^T \nu \\ t + \partial\phi(f, F)/\partial F \end{bmatrix} \in \mathbf{R}^{3M+1}$$

where $\nu \in \mathbf{R}^6$ is the Lagrange multiplier associated with the equality constraints. Here, $r_f \in \mathbf{R}^{3M}$ is the component of the dual residual associated with the force vector f, and $r_F \in \mathbf{R}$ is the component of the dual residual associated with F.

We can further divide r_f into components corresponding to the individual contact points, as

$$r_f = \begin{bmatrix} r^{(1)} \\ \vdots \\ r^{(M)} \end{bmatrix} = \begin{bmatrix} \nabla_{f^{(1)}} \phi_1(f^{(1)}, F) + A_1^T \nu \\ \vdots \\ \nabla_{f^{(M)}} \phi_M(f^{(M)}, F) + A_M^T \nu \end{bmatrix}.$$

Here, $r^{(i)} \in \mathbf{R}^3$ is the portion of the dual residual associated with contact point i. We will give more explicit formulas for these in Section IV-D.

The optimality conditions for the barrier subproblem (20) can be expressed in terms of the primal and dual residuals as

$$r_p = 0, \qquad r_d = 0.$$

(See, e.g., [Ch. 10].)

B. Dual Feasible Points From Subproblem Solution

If $(f^\star, F^\star, \nu^\star)$ are optimal for the barrier subproblem, then $\nu^\star/t$ is feasible for the dual FOP (14). The associated dual objective value is

$$\frac{\nu^{\star T} \omega^{\text{ext}}}{t} = F^\star - \frac{4M}{t}.$$

This can be shown by direct computation, or from general facts about conic problems and logarithmic barriers, since here we have $2M$ second-order cones, each with a θ-value of 2 (see [Sec. 11.6], [31]). It follows that the solution of the barrier subproblem is at most $4M/t$ suboptimal for the FOP (8).

C. Newton Method

We use an infeasible start Newton method to solve the barrier subproblem (20) [Sec. 10.3]. The Newton method starts with, and maintains, $(f, F, \nu) \in \mathbf{R}^{3M} \times \mathbf{R} \times \mathbf{R}^6$, with (f, F) strictly satisfying the cone constraints [i.e., (21)], but not necessarily the equality constraints $Af + \omega^{\text{ext}} = 0$. Within a finite number of steps, however, the equality constraints become satisfied, and once satisfied, they remain satisfied at all subsequent iterations. In particular, if the infeasible start Newton method is started at (f, F, ν) with the equality constraints $Af + \omega^{\text{ext}} = 0$ satisfied, the equality constraints will be satisfied at all future iterates.

At each iteration of the Newton method, we compute the Newton step $(df, dF, d\nu)$ at the current point (f, F, ν). We then carry out a backtracking line search to find a step length γ, based on the merit function $\|r_d\|^2 + \|r_p\|^2$. The backtracking line search algorithm chooses $\gamma = \beta^k$, where $\beta \in (0, 1)$ is an algorithm parameter, and k is the smallest nonnegative integer for which

$$\left(\|\hat{r}_d\|^2 + \|\hat{r}_p\|^2\right)^{1/2} \leq (1 - \alpha\gamma)\left(\|r_d\|^2 + \|r_p\|^2\right)^{1/2}$$

holds, where $\hat{r}_d$ and $\hat{r}_p$ are the primal and dual residuals evaluated at the tentative point $(f + \gamma df, F + \gamma dF, \nu + \gamma d\nu)$, and $\alpha \in (0, 1/2)$ is another algorithm parameter. (In particular, we require that $(f + \gamma df, F + \gamma dF)$ strictly satisfies the cone constraints.) The parameters must satisfy $0 < \beta < 1$ and $0 < \alpha < 1/2$. Common choices for the parameters are $\beta = 1/2$, $\alpha = 0.1$. (The Newton method is fairly insensitive to the choice of these parameters.)

After the line search, we update the new point as

$$f := f + \gamma\, df, \qquad F := F + \gamma\, dF, \qquad \nu := \nu + \gamma\, d\nu.$$

The Newton method is terminated when the norm of the residual is small enough, or when some other exit criterion is satisfied. Once a full Newton step is taken (i.e., the line search

gives $\gamma = 1$), the primal residual becomes zero, and remains zero (to within numerical accuracy) from that step on; see [29, Sec. 10.3.2].

The main effort of the algorithm is in computing the Newton step $(df, dF, d\nu) \in \mathbf{R}^{3M+7}$, which is given by the solution of the Karush–Kuhn–Tucker (KKT) system

$$\begin{bmatrix} \dfrac{\partial^2\phi}{\partial f^2} & \dfrac{\partial^2\phi}{\partial f \partial F} & A^T \\ \dfrac{\partial^2\phi}{\partial F \partial f} & \dfrac{\partial^2\phi}{\partial F^2} & 0 \\ A & 0 & 0 \end{bmatrix} \begin{bmatrix} df \\ dF \\ d\nu \end{bmatrix} = - \begin{bmatrix} r_f \\ r_F \\ r_p \end{bmatrix}$$

a set of $3M + 7$ equations in $3M + 7$ variables. Since the log barrier is a sum of functions of each $f^{(i)}$, the second derivative appearing in the upper left entry of the KKT matrix is block diagonal, so we can express the KKT system as

$$\begin{bmatrix} H_1 & & & q_1 & A_1^T \\ & \ddots & & \vdots & \vdots \\ & & H_M & q_M & A_M^T \\ q_1^T & \cdots & q_M^T & H_F & 0 \\ A_1 & \cdots & A_M & 0 & 0 \end{bmatrix} \begin{bmatrix} df_1 \\ \vdots \\ df_M \\ dF \\ d\nu \end{bmatrix} = - \begin{bmatrix} r_1 \\ \vdots \\ r_M \\ r_F \\ r_p \end{bmatrix} \tag{22}$$

where

$$\begin{aligned} H_i &= \frac{\partial^2 \phi_i}{\partial f^{(i)2}} \in \mathbf{R}^{3\times 3} \\ H_F &= \frac{\partial^2 \phi}{\partial F^2} \in \mathbf{R} \\ q_i &= \frac{\partial^2 \phi_i}{\partial f^{(i)} \partial F} \in \mathbf{R}^3. \end{aligned} \tag{23}$$

Explicit formulas for these will be given in the next section; for now, we note the sparsity pattern of the KKT system, which consists of a block diagonal part, with M 3×3 blocks (each associated with a contact point), bordered by seven dense rows and columns (corresponding to the objective variable F and dual variable ν).

The infeasible start Newton method always converges to the optimal point for the barrier subproblem, provided the original FOP is strictly feasible. When the original FOP problem is not strictly feasible, the barrier subproblem is not feasible, and the Newton method does not converge; the residuals do not converge to zero. Moreover, the Newton method cannot take a step size $\gamma = 1$ (because if it did, the next iterate would be strictly feasible).

The convergence of Newton's method is quadratic, so terminal convergence is extremely fast. For the same reason, if the method is started at an initial point with primal and dual residuals not too large, convergence to high accuracy can be obtained within a few steps. The total number of Newton steps can be bounded using the theory of self-concordance [31], but the bounds obtained are usually far larger than the actual number required in practice, which is often fewer than ten, and rarely more than a few tens. When the starting point is good, i.e., the starting residuals are small, convergence typically occurs in just a few steps.

D. Barrier Gradient and Hessian Formulas

In this section, we give explicit expressions for the gradient and Hessian of the log barrier, which are needed to define the dual residuals, and the coefficient matrix in the KKT system (23).

The log barrier ϕ is a sum of $2M$ terms, each of which is the negative log of a quadratic form. So, we first give a general formula for the gradient and Hessian of the function

$$\psi(u) = -\log(u^T P u)$$

where P is a symmetric matrix. (We assume that $u^T P u > 0$.) We have

$$\begin{aligned} \nabla\psi &= -\frac{2}{u^T P u} P u \\ \nabla^2\psi &= -\frac{2}{u^T P u} P + \left(\frac{2}{u^T P u}\right)^2 (Pu)(Pu)^T. \end{aligned} \tag{24}$$

Using this formula (or by direct differentiation), we have

$$\begin{aligned} \frac{\partial \phi_i}{\partial f_x^{(i)}} &= \frac{2 f_x^{(i)}}{F^2 - f_x^{(i)2} - f_y^{(i)2} - f_z^{(i)2}} \\ &\quad + \frac{2 f_x^{(i)}}{\mu_i^2 f_z^{(i)2} - f_x^{(i)2} - f_y^{(i)2}} \\ &= (a_i + b_i) f_x^{(i)} \end{aligned}$$

where we define

$$\begin{aligned} a_i &= \frac{2}{F^2 - f_x^{(i)2} - f_y^{(i)2} - f_z^{(i)2}} \\ b_i &= \frac{2}{\mu_i^2 f_z^{(i)2} - f_x^{(i)2} - f_y^{(i)2}}. \end{aligned}$$

We have similar expressions for the partial derivative with respect to $f_y^{(i)}$, with $f_x^{(i)}$ changed to $f_y^{(i)}$ in the numerators. The partial derivative with respect to $f_z^{(i)}$ is

$$\begin{aligned} \frac{\partial \phi_i}{\partial f_z^{(i)}} &= \frac{2 f_z^{(i)}}{F^2 - f_x^{(i)2} - f_y^{(i)2} - f_z^{(i)2}} \\ &\quad - \frac{2\mu_i^2 f_z^{(i)}}{\mu_i^2 f_z^{(i)2} - f_x^{(i)2} - f_y^{(i)2}} \\ &= (a_i - \mu_i^2 b_i) f_z^{(i)}. \end{aligned}$$

Thus, the gradient is given by

$$\nabla_{f^{(i)}} \phi_i(f^{(i)}, F) = \begin{bmatrix} (a_i + b_i) f_x^{(i)} \\ (a_i + b_i) f_y^{(i)} \\ (a_i - \mu_i^2 b_i) f_z^{(i)} \end{bmatrix}.$$

The partial derivative of ϕ with respect to F is

$$\frac{\partial \phi}{\partial F} = \sum_{i=1}^{M} -\frac{2F}{F^2 - f_x^{(i)2} - f_y^{(i)2} - f_z^{(i)2}} = -\left(\sum_{i=1}^{M} a_i\right) F.$$

Therefore, the dual residuals can be explicitly expressed as

$$r_i = \begin{bmatrix} (a_i + b_i) f_x^{(i)} \\ (a_i + b_i) f_y^{(i)} \\ (a_i - \mu_i^2 b_i) f_z^{(i)} \end{bmatrix} + A_i^T \nu, \qquad i = 1, \ldots, M$$

$$r_F = t - (a_1 + \cdots + a_M) F. \tag{25}$$

Now, we work out the blocks in the barrier Hessian $\nabla^2 \phi$, using the notation in (23). Using the general formula (24), we have

$$H_i = a_i I + a_i^2 f^{(i)} f^{(i)T} + b_i \mathbf{diag}(1, 1, -\mu_i^2) + b_i^2 \begin{bmatrix} f_x^{(i)} \\ f_y^{(i)} \\ -\mu_i^2 f_z^{(i)} \end{bmatrix} \begin{bmatrix} f_x^{(i)} \\ f_y^{(i)} \\ -\mu_i^2 f_z^{(i)} \end{bmatrix}^T.$$

The vectors q_i that form the last row and column of the Hessian are given by

$$q_i = -a_i^2 F f^{(i)}, \qquad i = 1, \ldots, M.$$

Finally, we have

$$\begin{aligned} H_F &= \frac{\partial^2 \phi}{\partial F^2} \\ &= \sum_{i=1}^{M} (-a_i + a_i^2 F^2) \\ &= \sum_{i=1}^{M} a_i \frac{F^2 + \|f^{(i)}\|^2}{F^2 - \|f^{(i)}\|^2} \\ &= \sum_{i=1}^{M} \frac{a_i^2}{2} (F^2 + \|f^{(i)}\|^2). \end{aligned}$$

V. Efficient Newton Step Computation

In this section, we show how to compute the Newton step $(df, dF, d\nu)$, i.e., solve the equations (22), efficiently. These equations can be solved using standard methods for linear equations, such as Gaussian elimination. For example, we can compute an LDL^T factorization of the KKT matrix, and then, find $(df, dF, d\nu)$ by a back and forward substitution. The cost of this is around $(1/3)(3M + 7)^3 + 2(3M + 7)^2 \approx 9M^3 + 63M^2$ flops (floating point operations). While this can be fast for modest values of M (say, M smaller than 10 or so), we can exploit the special structure of the equations to solve the equations far faster. The method we describe in this section requires around $350M$ flops, and in addition, involves no conditionals (which are needed to pivot in the general case), leading to extremely fast execution time. For M large, our method is clearly far faster than a general method. It is also faster for small values of M such as $M = 4$, for which $(1/3)(3M + 7)^3 + 2(3M + 7)^2$ is around two and one-half times $350M$.

Our method is based on a sequence of two elimination steps, in which particular blocks of variables are eliminated, using Cholesky factorizations to compute the required inverses since their definiteness properties are known ahead of time. The leading $3M \times 3M$ block is a block diagonal, with 3×3 blocks that are positive definite, and so, can be inverted very efficiently. A basic elimination method (see, e.g., [Appendix C, 29]) applied to this block yields a 7×7 set of linear equations to solve. This reduced system of linear equations is indefinite, but it contains a 6×6 subblock that is positive definite, so a further elimination step can be taken, using a Cholesky factorization to invert the positive definite subblock.

Before giving the details of our elimination method, we comment on how it differs from using a good sparse solver to compute the Newton step. Such a method decomposes the KKT matrix as $PLDL^T P^T$, where P is a permutation matrix chosen to reduce fill-in and preserve numerical stability, L is unit lower triangular, and D is diagonal. It is very likely that the permutation found will be good enough to yield an algorithm that solves the KKT system in $O(M)$ flops, just like our method. Ours, however, is completely explicit, and also exploits a small amount of further structure, such as the easy invertibility of the Cholesky factors. Moreover, our method involves no conditionals, which makes it faster (and simpler) than a generic sparse $PLDL^T P^T$ method. And finally, since our method uses an explicit elimination ordering, it expends no effort in discovering a good elimination ordering.

A. First Elimination Step

Our first step is to eliminate the variables df_i from (22). From

$$H_i \, df_i + q_i \, dF + A_i^T \, d\nu = -r_i$$

we obtain

$$df_i = -H_i^{-1}(r_i + q_i \, dF + A_i^T \, d\nu). \tag{26}$$

Combining this with

$$\sum_{i=1}^{M} q_i^T \, df_i + H_F \, dF = -r_F$$

we obtain

$$-\sum_{i=1}^{M} q_i^T H_i^{-1}(r_i + q_i \, dF + A_i^T \, d\nu) + H_F \, dF = -r_F$$

which can be written as

$$-\left(H_F - \sum_{i=1}^{M} q_i^T H_i^{-1} q_i\right) dF + \left(\sum_{i=1}^{M} q_i^T H_i^{-1} A_i^T\right) d\nu = r_F - \sum_{i=1}^{M} q_i^T H_i^{-1} r_i. \tag{27}$$

Now, we use

$$\sum_{i=1}^{M} A_i \, df_i = -r_p$$

and (26) to get

$$\sum_{i=1}^{M} A_i H_i^{-1}(r_i + q_i \, dF + A_i^T \, d\nu) = r_p$$

which we write as

$$\left(\sum_{i=1}^{M} A_i H_i^{-1} q_i\right) dF + \left(\sum_{i=1}^{M} A_i H_i^{-1} A_i^T\right) d\nu = r_p - \sum_{i=1}^{M} A_i H_i^{-1} r_i. \quad (28)$$

Now, we write (27) and (28) as

$$\begin{bmatrix} E_{11} & E_{21}^T \\ E_{21} & E_{22} \end{bmatrix} \begin{bmatrix} dF \\ d\nu \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} \quad (29)$$

where

$$E_{11} = -\left(H_F - \sum_{i=1}^{M} q_i^T H_i^{-1} q_i\right)$$

$$E_{21} = \sum_{i=1}^{M} A_i H_i^{-1} q_i$$

$$E_{22} = \sum_{i=1}^{M} A_i^T H_i^{-1} A_i^T$$

$$e_1 = r_F - \sum_{i=1}^{M} q_i^T H_i^{-1} r_i$$

$$e_2 = r_p - \sum_{i=1}^{M} A_i H_i^{-1} r_i.$$

The reduced system (29) is a set of seven linear equations in seven variables, which is easily solved to get dF and $d\nu$. Once these have been computed, we can find df_i from (26).

B. Second Elimination Step

The reduced system (29) also has a structure that can be exploited, even though it is fully dense. As we will show, the matrix E is not positive definite (or negative definite), so we cannot use a Cholesky factorization of E (or $-E$) to solve the system. We must use an LDL^T factorization, or a factorization such as LDU that ignores the symmetry. Generic algorithms for each of these require numerical pivoting, which complicates (and slows) the algorithm.

The $1,1$ entry E_{11} is negative. This follows from the convexity of ϕ, which implies that the Hessian is positive definite; $-E_{11}$ is a Schur complement of $\nabla^2\phi$, and therefore, positive. Its 2,2 entry E_{22} is, however, positive definite. (It is clearly positive semidefinite; it is positive definite since A is full rank.) Thus, the matrix E is indefinite; it has one negative eigenvalue and six positive eigenvalues. We can solve the reduced system (29) by eliminating the $1,1$ block, which will leave a 6×6 system of equations that is positive definite, and can therefore, be solved by a Cholesky factorization.

This approach does not reduce the flop count, when compared to a generic LDL^T factorization method; but it removes all conditionals and row/column permutations (i.e., pivoting), and therefore, yields a simpler (and faster) algorithm. Since the final system is solved by Cholesky factorization, the lack of pivoting comes at no cost in numerical stability.

To solve the reduced system (29) by this method, we proceed as follows. We first note that

$$dF = (1/E_{11})(e_1 - E_{21}^T\, d\nu)$$

and substituting this into $E_{21}\, dF + E_{22}\, d\nu = e_2$, we get

$$\left(E_{22} - (1/E_{11}) E_{21} E_{21}^T\right) d\nu = e_2 - (1/E_{11}) E_{21} e_1.$$

Since $E_{11} < 0$, the matrix on the left is positive definite. Therefore, we can solve this set of equations using Cholesky factorization; we then find dF from the earlier equation.

C. Summary and Flop Count

We summarize our efficient method for computing the Newton step. We compute the primal and dual residuals, as well as the Hessian, i.e., H_i, H_F, and q_i, using the formulas given in Section IV-D. We compute Cholesky factors L_i of H_i, i.e., lower triangular L_i with $L_i L_i^T = H_i$. We then compute the inverses of the Cholesky factors, i.e., L_i^{-1}. (The more standard method would be to use forward substitution to compute $L^{-1}y$, but here, there is a slight advantage to simply inverting these matrices.) The total flop count for this step is less than around 50 M flops.

Next, we compute $L_i^{-1} A_i^T$, $L_i^{-1} q_i$, and $L_i^{-1} r_i$, at a total cost of 73 M flops. (Since the cost of multiplying a lower triangular 3×3 matrix by a vector is nine flops.) From these, we compute the data in the reduced system, i.e., E_{11}, E_{21}, E_{22}, e_1, and e_2. For example, to compute E_{22}, we express it as

$$E_{22} = \sum_{i=1}^{M} (L_i^{-1} A_i^T)^T (L_i^{-1} A_i^T).$$

This costs 126 M flops. There are 21 entries in the upper triangle of the matrices in the sum (their lower halves are the same as their upper halves); each of these is computed as an inner product of two vectors in $\mathbf{R}^3$, which costs five flops. This gives $5 \times 21 = 105$ flops for each term in the sum, plus 21 M flops to add the matrices.

Forming E_{22} is the dominant cost in forming the reduced system: forming E_{11} and e_1 each costs 6 M flops; forming E_{21} and e_2 each costs 31 M flops. Thus, the total flop count to form the reduced system is around 200 M flops.

The reduced system is solved using the method described in Section V-B, which has a small cost, independent of M, of around $(1/3)7^3 + 2 \times 7^2 \approx 200$ flops. Finally, df_i's are computed from (26), at a cost around 31 M flops. All together, then, we can compute the Newton step in around 350 M plus a modest constant around 200 flops.

VI. Barrier Method

In a classic primal barrier method, the barrier subproblem (20) is solved for an increasing sequence of values of t. The Newton method for each subproblem is initialized at the optimal solution of the previous one. A typical method for increasing the parameter t is to multiply it by a factor on the order of 10

(see, e.g., [29, Sec. 10.3]). This is repeated until $4M/t$ is smaller than the required tolerance. This basic barrier method is reliable, and can solve the FOP using on the order of 50 or so Newton steps, provided the problem is feasible, and the starting point is reasonably close to feasible. Since each Newton step can be carried out very quickly, this is already an attractive algorithm. We will show, however, that a variation on the basic barrier method can reliably solve the problem to a good enough guaranteed relative accuracy (say, 1%) using almost an order of magnitude fewer Newton steps, i.e., between five and ten steps.

When the FOP (8) is infeasible, the basic primal barrier method fails during the first barrier subproblem, which never finds a feasible point (since no such point exists). Even when the FOP is feasible, but nearly infeasible, the number of iterations required can be much larger than the typical number. These problems can be handled by preceding the basic barrier method with a special phase I method, designed to find a feasible point, or establish the infeasibility of the FOP. We will address this issue in Section VI-B. For now, we assume that a feasible, or nearly feasible, starting point is available.

A. Custom Primal Barrier Method

Our basic method consists of solving the barrier subproblem, for a fixed value of t, using Newton's method. Once f is feasible, we evaluate the current maximum force $F^{\max} = \max\{\|f^{(1)}\|, \ldots, \|f^{(M)}\|\}$ using the current value of f, and the current dual bound $F^{\rm dual}$ [from (16)] using the current estimate of ν, i.e., $\bar{\nu}/t$, where $\bar{\nu}$ is the current value of the dual variable in the Newton subproblem. We terminate if the current maximum force $F^{\max}$ is within a given tolerance of the current dual lower bound.

given
starting point (f, F, ν) that satisfies (21)
required tolerance $\epsilon_{\rm abs} > 0$
parameter $\eta \in (0, 1)$
Set $t := 4M/(\eta\epsilon_{\rm abs})$.
repeat
1. Compute Newton step $(df, dF, d\nu)$ using the methods described in Section V.
2. *Line search and update.*
   Choose a step size γ using backtracking (see Section IV-C).
3. *Update.* $(f, F, \nu) := (f, F, \nu) + \gamma(df, dF, d\nu)$.
4. **if** $Af + \omega^{\rm ext} = 0$, evaluate $F^{\max}$ and the dual lower bound $F^{\rm dual}$ (16).

until $F^{\max} - F^{\rm dual} \leq \epsilon_{\rm abs}$.

We make some observations about this algorithm. First, the algorithm exits with a guarantee (via the dual bound) that the maximum force is no more than $\epsilon_{\rm abs}$-suboptimal, even though Newton method may not have converged. Second, the exit condition must hold eventually; if it did not, Newton method converges to the optimal point for the barrier subproblem, at which we have

$$F^{\max} - F^{\rm dual} \leq F - F^{\rm dual} = \frac{4M}{t} = \eta\epsilon_{\rm abs} < \epsilon_{\rm abs}$$

a contradiction. The parameter η gives a "margin" that guarantees that if Newton's method were to converge, the gap between $F^{\max}$ and the lower bound $F^{\rm dual}$ is smaller than our tolerance. We have found good results with $\eta = 0.8$.

Choosing a *fixed* value of t (as we do in the algorithm described before) is a poor choice for solving general SOCPs, especially when high accuracy is required, and the number of constraints is large. But for the FOP, a fixed choice of t works very well, at least for modest M (say, smaller than 100). If high accuracy is needed, or M is large (say, over 100), a more conventional schedule for t (as in a primal barrier method) would likely perform better.

We can modify the algorithm to stop with a guaranteed *relative* tolerance instead of a guaranteed absolute tolerance by replacing the stopping criterion with

$$\frac{F^{\max} - F^{\rm dual}}{F^{\rm dual}} \leq \epsilon_{\rm rel}. \tag{30}$$

In this case, we must ensure that the value of t used is large enough to drive $F^{\max} - F^{\rm dual}$ small enough so that (30) will eventually hold. This requires

$$t > \frac{4M}{\epsilon_{\rm rel} F^{\rm dual}}.$$

To do this, we can adjust t during the algorithm as follows. We start with an absolute tolerance, as in the algorithm before, chosen so that it is likely to be sufficiently small to work for the required relative tolerance. We terminate if the relative tolerance is reached (in which case we are done), or when the absolute tolerance is reached. In the latter case, we simply update t to be

$$t = \frac{4M}{\eta\epsilon_{\rm rel} F^{\rm dual}}$$

where $F^{\rm dual}$ is the current value of the dual bound, and continue until the relative stopping criterion (30) is satisfied (which must occur).

Finally, we describe two modes of early termination for the algorithm that we will employ in Section VII. We are given two target values $L_{\rm tar}$ and $U_{\rm tar}$. We terminate the algorithm once we have achieved either $F^{\max} \leq U_{\rm tar}$ (i.e., the maximum force is less than the upper target value) or $F^{\rm dual} \geq L_{\rm tar}$ (i.e., the dual bound shows that the optimal maximum force is more than lower target value).

B. Phase I Problem

The custom primal barrier method described in the previous section will work when started from any infeasible point that strictly satisfies the cone constraints, provided the original FOP is strictly feasible. If the starting point is far from feasibility, however, the algorithm can require a large number of iterations. When a good starting point is not available, it is more efficient to break the optimization up into two phases. In phase I, we determine a feasible point, i.e., an f that satisfies the constraints (17), or verify that the problem is infeasible, by exhibiting a feasible solution of the alternative inequalities (18). In phase II, we use the custom barrier method to compute the optimal solution of the FOP (7), starting from the feasible point found in phase I.

One choice for the phase I problem for the FOP is

$$\begin{array}{ll} \text{minimize} & s \\ \text{subject to} & f^{(i)} + se_3 \in K_i, \quad i = 1, \ldots, M \\ & Af + \omega^{\text{ext}} = 0 \\ & s \geq -1 \end{array} \tag{31}$$

where the optimization variables are $s \in \mathbf{R}$ and $f \in \mathbf{R}^{3M}$, and $e_3 = (0, 0, 1)$. This phase I problem is also in conic form, i.e., an SOCP. The variable s (which is also the objective in the phase I problem) can be interpreted as a fictitious force added to each of the normal forces $f_z^{(i)}$ for the purpose of satisfying the friction cone constraint. But these fictitious forces do not appear in the equilibrium condition $Af + \omega^{\text{ext}} = 0$. The goal is to drive this fictitious force to become zero (or negative); if we succeed, the corresponding f is evidently feasible for the FOP (7). On the other hand, if the optimal value of the phase I problem is positive, it means that the original FOP (7) is infeasible. The inequality $s \geq -1$ is really not needed; it keeps the algorithm from finding a point that has very negative s, at the cost of very large forces.

For the phase I problem, we can easily construct a feasible point from which to start the barrier method. First, we choose any f that satisfies the equality constraints $Af + \omega^{\text{ext}}$, for example, the least-norm solution

$$f = -A^T(AA^T)^{-1}\omega^{\text{ext}}. \tag{32}$$

As an initial value of s, we can choose any value that satisfies

$$s > \max\left\{ (1/\mu_i)\sqrt{f_x^{(i)2} + f_y^{(i)2}} - f_z^{(i)} \;\middle|\; i = 1, \ldots, M \right\}.$$

The pair (f, s) is then feasible for the phase I problem. Of course, the force vector f in (32) need not satisfy the cone constraints; for example, the normal forces $f_z^{(i)}$ can be negative. We also note that computing f from (32) can be done very efficiently, because the associated normal equations have exactly the same block sparsity pattern that allows us to use the methods of Section V.

We use a barrier method to solve the phase I problem (31), starting from this feasible point, and terminating as soon as $s < 0$, since the corresponding f is then strictly feasible for (7). In each outer iteration, we solve the problem

$$\begin{array}{ll} \text{minimize} & ts + \phi(f, s) \\ \text{subject to} & Af + \omega^{\text{ext}} = 0 \end{array} \tag{33}$$

where $t > 0$ is a parameter and ϕ is the log barrier for the cone constraints in the phase I problem:

$$\phi(f, s) = \sum_{i=1}^{M} -\log(\mu_i^2(f_z^{(i)} + s)^2 - f_x^{(i)2} - f_y^{(i)2})$$

when $f^{(i)} + se_3 \in \mathbf{int}K_i$ and $\phi(f, s) = \infty$ otherwise. The solution of this subproblem is at most $2M/t$ suboptimal for the phase I problem (31), since here we have M cones, each with θ-value 2.

We use Newton's method to solve the smooth subproblem (33). We omit the details and formulas involved since they are very close to the ones encountered earlier. We note, however, that each step of Newton's method for the phase I problem involves the solution of a set of equations with *exactly* the same sparsity structure encountered earlier in (23), i.e., block diagonal, with M 3×3 blocks, bordered by seven dense rows and columns. (The particular coefficient matrices H_i, q_i, and H_F, however, are different.) Therefore, the method described in Section V can be used to efficiently compute the Newton step for the phase I problem.

When we solve the phase I problem with an absolute tolerance ϵ_{abs} (which corresponds to $t = 2M/\epsilon_{\text{abs}}$), three outcomes can occur. We can terminate (early) having found a feasible point; we can also terminate early, once the dual bound for the phase I problem shows the optimal value is positive. The third possibility is an ambiguous outcome, in which we have computed a set of forces that are feasible, when an additional normal contact force no more than ϵ_{abs} is added. This outcome is extremely rare, and in any case, not a problem in practice. When a FOP is just on the boundary between feasible and infeasible, it can be considered infeasible for all practical purposes.

VII. Solving Multiple Force Optimization Problems

In this section, we consider the problem of efficiently solving a family of related FOPs. In the simplest case, we have multiple FOPs to solve to some given accuracy. We can always solve them one by one, using the method described before for each problem instance. If the data for the FOPs differ sufficiently, treating each problem as a brand new one is not a bad approach; but if the problem data are close, we can more efficiently solve them using *warm-start* techniques. Since the computational effort and time required to solve a FOP is measured by the number of Newton steps required, the goal is to solve FOP with fewer Newton steps than would be typically required.

A. Warm-Start

Suppose we have solved a FOP, and then need to solve a new one with problem data that is "close," i.e., the external wrench ω^{ext}, contact geometry matrix A, and coefficients of friction μ_i are not far from those of the previous problem. In a warm-start technique, we simply use the previously computed force vector f as the starting point for Newton's method, in the barrier subproblem associated with the new FOP to be solved, assuming that it satisfies the friction cone constraints for the new problem. (If it does not, we can simply add some extra inward normal force to each contact force that does not satisfy its cone constraint.) This force vector will not, in general, satisfy the equilibrium conditions for the new problem, but, assuming that the new problem is strictly feasible, a feasible point will eventually be found.

If the new FOP problem data are close enough to the previous FOP problem data, the new problem can be solved in just a few Newton steps. On the other hand, it is possible that this approach can require *more* iterations than a *cold-start* method, i.e., solving the new problem as a new one, with a phase I followed by a phase II. As an extreme example, if the new problem is infeasible, then Newton method will simply fail to converge. A very simple method for getting around this is to run the warm-start method for a fixed and small number of

iterations, such as 6; if a feasible point has not been found by then, the warm-start process is abandoned, and the new problem is solved using a cold-start. The hope is that the iterations saved in the successful warm-start attempts more than offset the extra iterations wasted in warm-start failures.

We note that general interior-point methods are *not* well suited to warm-start techniques. Our method handles warm-start well since it is a primal barrier method, with a fixed value of t, so it inherits the (very good) warm-start properties of Newton's method.

As an example application, consider the problem of computing optimal grasping forces for an object, as it moves along a known trajectory. We assume that the contact points on the object remain the same, but along the trajectory, the external wrench changes. The external wrench can include a gravity term, as well as a term associated with accelerating the object. By choosing a global coordinate system tied to the object, the matrix A remains constant along the trajectory, as the object moves, but the external wrench ω^{ext} varies. After computing the optimal grasping forces at one time, we can use a warm-start technique to rapidly compute the optimal grasping forces at the next sampled time. (In this special case with A not changing, we can also use the method described in Section II-F. We compute 12 wrenches that resist $\pm e_i$; thereafter, we can immediately generate feasible, if not optimal, contact force vectors at each point along the trajectory.) Another example is computing the optimal grasping forces at the vertices of a box in wrench space, i.e., all 64 combinations of high and low values of the six components of ω^{ext}:

$$\omega^{\mathrm{ext}}{}_i \in \{l_i, u_i\}, \qquad i = 1, \ldots, 6.$$

We can solve the FOP for the average or nominal value of ω^{ext},

$$\omega^{\mathrm{ext}} = \frac{1}{2}(l + u)$$

and then attempt to compute the optimal contact forces for each vertex using a warm-start method.

B. Short-Circuiting

Short-circuiting refers to early termination of the FOP algorithm, i.e., termination before the required relative or absolute tolerance has been reached. This can be done in two interesting cases. In the first case, the goal is compute the *worst case* or maximum value of the minimum grasping force over a set of FOPs. In the second case, the goal is compute the *best case* or minimum value of the minimum grasping force over a set of FOPs. Of course, both of these problems can be solved by solving each FOP (possibly using warm-start techniques), and then, simply computing the maximum (or minimum) of the optimal grasping forces.

Examples of the first case include computing the maximum grasping force that is required over a box in external wrench space, or the maximum grasping force that is required to move the object along a given trajectory. The second case arises in optimizing the position and orientation of a manipulator, relative to an object, using the minimum force required to grasp the object as the objective (see, e.g., [6]). This is the (nonconvex) problem of optimizing the contact points at which to grasp a given object. This can be done using an outer search loop that generates candidate manipulator positions and orientations; for each candidate, we find the resulting contact points, and then, solve the associated FOP to determine the minimum force required to grasp the object. Here, our goal is to find the *best* set of contact points, i.e., the contact points that (among those considered) minimize the maximum grasping force.

Worst case short-circuiting works as follows. We maintain a current worst case value of minimum grasping force (which we can set equal to zero number initially). We then solve each FOP in turn, but we can terminate early if the algorithm finds a feasible grasping force vector with $F^{\max}$ smaller than the current worst case value. If the current FOP has minimum grasping force larger than the current worst case value, its minimum force becomes the new worst case value. This is called *primal short-circuiting* since we can terminate the FOP when the primal value $F^{\max}$ is less than a target value (i.e., the current worst case value).

Best case short-circuiting is similar. Here, we maintain a current best (smallest) value of minimum grasping force (which we can set equal to ∞ initially). We solve each FOP in turn, terminating early if the algorithm finds a dual lower bound F^{dual} on minimum grasping force that exceeds the current best case value. If the current FOP has minimum grasping force smaller than the current best case value, its minimum force becomes the new best case value. This is called *dual short-circuiting* since we terminate the FOP when the dual lower bound F^{dual} exceeds a target value (i.e., the current best case value). Note that in dual short-circuiting, it is possible to expend *zero* Newton steps on a FOP. This occurs if the dual variable ν, from the last problem solved, certifies that the minimum grasping force for the current problem will be worse than the best case already found.

VIII. Numerical Examples

In this section, we give some numerical examples to illustrate the algorithm and variations described before. All our examples use $M = 5$ contact points, determined by the position and the orientation of a particular manipulator with four fingers and one palm contact point. (This, of course, does not affect the algorithm; we are just pointing out that the sets of contact points are not chosen arbitrarily.) The external wrench is due to gravity. We used the coefficient of friction $\mu_i = 0.5$ for all contact points.

We used the algorithm described before, with a phase I to determine feasibility and a phase II to minimize the maximum grasping force. The algorithm parameters used are the ones mentioned in the earlier algorithm description. The stopping criterion is 1% relative accuracy, i.e., $\epsilon_{\mathrm{rel}} = 0.01$. (The algorithm performs well for much higher precisions, but this accuracy is more than adequate for any practical use.)

We solved 10^4 FOPs, generated using a collection of objects, including bars, rods, boxes, and barbells, with various assumed weights and centers of gravity, and various grasp contact points. About 10% of the FOPs were infeasible. The average number of Newton steps required to solve the FOP (or determine the infeasibility) is around 8, with a standard deviation around 2. The minimum number of Newton steps (over the 10^4 test cases)

was 3; the maximum was 16. The average number of Newton steps in phase I was around 2. The average number of Newton steps required to certify infeasible FOPs was around 7.

Our implementation was coded in C++, and run on a 3 GHz Pentium IV. Our implementation required around 50 μs per Newton step, so the time to solve a FOP was around 400 μs.

To illustrate warm-start and primal short-circuiting methods, we considered the problem of finding the worst case grasping force over a box of wrenches with each force and torque component varying $\pm 25\%$. We generated 4000 test problems, and compared the average number of Newton steps required, per FOP, when all 64 vertices, plus the center, of the wrench box were solved using four methods.

1) Cold-start without short-circuiting.
2) Cold-start with short-circuiting.
3) Warm-start without short-circuiting.
4) Warm-start with short-circuiting.

Our warm-start scheme ran for at most six steps, and reverted to a cold-start scheme if a feasible point had not been found by then. (We counted the Newton steps wasted in warm-start failures in our averages.) The following table gives the average number of Newton steps required, per FOP.

Method	Newton steps per FOP
Cold-start without S/C	10.6
Cold-start with S/C	7.0
Warm-start without S/C	4.0
Warm-start with S/C	3.9

Both short-circuiting and warm-start techniques reduce the total number of Newton steps required. (Although the combination of both does not give much further reduction over warm-start alone.) With warm-start, we require about 4 Newton steps per FOP; in our implementation, this corresponds to around 200 μs per FOP.

To illustrate dual short-circuiting, we solved grasp contact point optimization problems. At each iteration, we update the position and orientation of the manipulator, compute the contact points, and then, solve the associated FOP. We used a simple hill-climbing algorithm to (locally) optimize the manipulator position and orientation. Each of these optimization runs required around 130 or so iterations, each of which requires the solution of a FOP. Once again, we considered four schemes: cold- and warm-start, with and without dual short-circuiting.

In dual short-circuiting. we stop solving any FOP when it is determined that the set of contact points being considered cannot give a better solution than the best set of contact points found so far. The following table shows the average number of Newton steps required, per FOP solved, for the three different methods.

Method	Newton steps per FOP
Cold-start without S/C	10.6
Cold-start with S/C	9.7
Warm-start without S/C	4.6
Warm-start with S/C	3.0

These examples show that both dual short-circuiting and warm-start techniques significantly reduce the number of Newton steps required. With warm-start and primal short-circuiting, we are expending only 3 Newton steps per FOP. In our implementation, this corresponds to around 150 μs per FOP.

IX. Variations and Extensions

The methods presented in this paper can be extended in several ways, for example, by using a different objective function or a different contact point friction model.

A. Other Objectives

Instead of the maximum magnitude of the grasping forces, we can take as objective the sum of squares of the force magnitudes

$$F^{\mathrm{sumsq}} = \|f^{(1)}\|^2 + \cdots + \|f^{(M)}\|^2$$

or a sum of the force magnitudes,

$$F^{\mathrm{sum}} = \|f^{(1)}\| + \cdots + \|f^{(M)}\|$$

or the maximum normal force,

$$F^{\mathrm{norm,max}} = \max\{f_z^{(1)}, \ldots, f_z^{(M)}\}.$$

Each of these results in a new conic formulation of the problem, a new dual problem, and a different KKT system to solve in each Newton step. The same structure for the KKT coefficient matrix, however, will occur in all cases (possibly after elimination of some variables), which means that our method can be used to compute the Newton step efficiently, and therefore, to solve the problem efficiently.

Let us give some details for the case with objective F^{sumsq}. We can just as well minimize its square root, i.e., solve the problem

$$\begin{array}{ll} \text{minimize} & F \\ \text{subject to} & (f,F) \in K \\ & f^{(i)} \in K_i, \qquad i = 1, \ldots, M \\ & Af + \omega^{\mathrm{ext}} = 0 \end{array} \tag{34}$$

with variables $F \in \mathbf{R}$ and $f \in \mathbf{R}^{3M}$. Here, K denotes the standard second-order cone in $\mathbf{R}^{3M+1}$:

$$K = \{(z,y) \in \mathbf{R}^{3M} \times \mathbf{R} \mid \|z\| \leq y\}.$$

The only difference between this problem and the FOP (8) is that the cone constraints $(f^{(i)}, F) \in K, i = 1, \ldots, M$, that appear in the FOP are replaced with the single cone constraint $(f, F) \in K$.

Following the methods used in Section III, we arrive at the dual problem

$$\begin{array}{ll} \text{maximize} & \nu^T \omega^{\mathrm{ext}} \\ \text{subject to} & z_i \in K_i^*, \qquad i = 1, \ldots, M \\ & \sum_{i=1}^M \|A_i^T \nu - z_i\|^2 \leq 1 \end{array}$$

the analog of (13). Indeed, the only difference between this dual and the FOP dual (13) is that the sum of norms constraint in the FOP dual becomes a sum of norms squared constraint here. We

can write a compact and explicit dual, the analog of (14), as

$$\begin{array}{ll}\text{maximize} & \nu^T \omega^{\text{ext}} \\ \text{subject to} & \sum_{i=1}^M \mathbf{dist}(A_i^T \nu, K_i^*)^2 \leq 1.\end{array}$$

The barrier function for the minimum sum-of-squares FOP (34) is given by

$$\phi(f,F) = -\log(F^2 - \|f\|_2^2) - \sum_{i=1}^M \log(\mu_i^2 f_z^{(i)2} - f_x^{(i)2} - f_y^{(i)2}).$$

The matrix appearing in the KKT system that defines the Newton step, in fact, is fully dense. However, closer examination reveals that the Hessian term contributed by the second term of the barrier is block diagonal, with 3×3 blocks, and the Hessian term contributed by the first block is actually diagonal plus rank one. Using an un-elimination step (see [Appendix C, 29]), we then obtain a set of linear equations with exactly the form (22).

B. Contact Force Constraints

In this section, we briefly describe, at a high level, how additional (convex) contact force constraints $f \in \mathcal{C}$ can be added to our FOP. The dual objective for the constrained problem will have an additional term (corresponding to the force constraint in the primal) that can be used to increase the dual objective. In the barrier method, we add a log barrier for the force constraints. This new barrier term contributes a term to the gradient and Hessian. If the force constraints are separable, i.e., have the form $f^{(i)} \in \mathcal{C}^{(i)}$, then the corresponding Hessian term is block diagonal, with 3×3 blocks. This implies that the same basic method can be used to efficiently compute the search direction in this case. In particular, the computational effort is still $O(M)$.

C. Other Friction Models

The methods we have described can be extended to other friction models, such as frictionless point contact or soft contact with elliptic or linearized approximation (see, e.g., [26], [32]–[34]).

In this section, we briefly explain how to handle one such extension to soft contact with elliptic model. In this model, the contact points can exert a torque, about the (local) z-axis (i.e., normal axis), as well as a force. So, we introduce normal contact torques, $f_t^{(i)} \in \mathbf{R}, i = 1, \ldots, M$, as a fourth component of the contact point force vector. In this case, the matrices A_i become 6×4, with a fourth column that includes the effect of the torques applied in the normal direction at each contact point; the equilibrium condition is still $Af + \omega^{\text{ext}} = 0$, with these new definitions of f and A. The friction cone, now in $\mathbf{R}^4$, has the form

$$K_i = \left\{ x \in \mathbf{R}^4 \;\middle|\; \sqrt{(x_1^2 + x_2^2)/\mu_i^2 + x_4^2/\sigma_i^2} \leq x_3 \right\} \quad (35)$$

$i = 1, \ldots, M$, where $\sigma_i > 0$ is a parameter, sometimes called the torsional friction coefficient [cf. (2)]. Note that here x_3 represents the normal force, (x_1, x_2) the tangential friction force, and x_4 the friction torque.

With these new definitions of $f \in \mathbf{R}^{4M}$, $A \in \mathbf{R}^{6 \times 4M}$, and $K_i \subseteq \mathbf{R}^4$, and assuming that we measure the size of a set of contact forces by the maximum magnitude of the contact forces (and torque), the FOP is unchanged; it has exactly the form (7). The dual is also identical, with the dual of the soft contact friction cone

$$K_i^* = \left\{ y \in \mathbf{R}^4 | \sqrt{\mu_i^2(y_1^2 + y_2^2) + \sigma_i^2 y_4^2} \leq y_3 \right\}.$$

The dual problem is exactly the same; the only change is in the explicit formula for the Euclidean distance to K_i^*. While the Euclidean distance to the aforementioned cone can be computed quickly, there is no longer an explicit formula for it, analogous to (15).

The barrier method is the same, with the obvious modification to the barrier function. The KKT system for the barrier subproblem has the same general form, except that the diagonal blocks are 4×4, instead of 3×3. Otherwise, everything is the same.

X. Conclusion

We have considered the problem of computing an optimal set of contact forces, as measured by the maximum magnitude of the contact forces, subject to the constraint that we hold an object in equilibrium against an external wrench. We have developed a very simple dual problem, that can be interpreted easily, and gives a very cheaply computable lower bound on the optimal grasping force. We have developed a primal barrier algorithm that reliably solves the FOP within ten or so iterations; in contrast, general interior-point methods typically take several or more tens of iterations. Each iteration of our algorithm involves solving a set of linear equations. The special structure of these equations can be exploited to solve these equations with $O(M)$ computational effort, instead of $O(M^3)$ for the naive method. The constant hidden in the $O(M)$ notation is quite modest. For cases when the number of contact forces is small, the advantage here is not huge; but for M large (say, a few tens), the advantage is great. Our nonoptimized implementation can solve a FOP, with five contact forces, in around 400 μs. Our method can handle efficient warm-start, exploiting a good initial guess to solve a set of similar FOPs. This reduces the effort by a factor of 2.

References

[1] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2000, pp. 348–353.

[2] J. Kerr and B. Roth, "Analysis of multifingered hands," *Int. J. Robot. Res.*, vol. 4, no. 4, pp. 3–17, 1986.

[3] Z. Li and S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *IEEE J. Robot. Autom.*, vol. 4, no. 1, pp. 32–44, Feb. 1988.

[4] C. Borst, M. Fischer, and G. Hirzinger, "Grasp planning: How to choose a suitable task wrench space," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, New Orleans, LA, Apr. 2004, pp. 319–325.

[5] D. Kirkpatrick, B. Mishra, and C. Yap, "Quantitative Steinitz's theorems with applications to multifingered grasping," in *Proc. 20th ACM Symp. Theory Comput.*, 1990, pp. 341–351.

[6] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. 1992 IEEE Int. Conf. Robot. Autom.*, Nice, France, pp. 2290–2295.

[7] F. Cheng and D. Orin, "Efficient algorithms for optimal force distribution—The compact dual LP method," *IEEE Trans. Robot. Autom.*, vol. 6, no. 2, pp. 178–187, Apr. 1990.

[8] Y. Liu and M. Wang, "Qualitative test and force optimization of 3D frictional force-closure grasps using linear programming," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1998, pp. 3335–3340.
[9] Y. Liu, "Qualitative test and force optimization of 3-D frictional force-closure grasps using linear programming," *IEEE Trans. Robot. Autom.*, vol. 15, no. 1, pp. 163–173, Feb. 1999.
[10] B. Mishra, "Grasp metrics: Optimality and complexity" in *Algorithmic Foundations of Robotics*, K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, Eds. Wellesley, MA: A. K. Peters, 1995, pp. 137–166.
[11] M. Hanes, S. Ahalt, K. Mirza, and D. Orin, "Neural network control of force distribution for power grasp," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1991, pp. 746–751.
[12] Y. Xia, J. Wang, and L.-M. Fok, "Grasping-force optimization for multifingered robotic hands using a recurrent neural network," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 549–554, Jun. 2004.
[13] M. Teichmann and B. Mishra, "Probabilistic algorithms for efficient grasping and fixturing," *Algorithmica*, vol. 26, no. 3/4, pp. 345–363, 2000.
[14] N. Pollard, "Synthesizing grasps from generalized prototypes," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1996, vol. 3, pp. 2124–2130.
[15] N. Pollard, "Quality-preserving algorithms for grasp synthesis from prototypes," *IEEE Trans. Robot. Autom.*, to be published.
[16] G. Färber and R. Suárez, "Fast on-line determination of quasi-optimal grasp configurations based on off-line analysis and parametrization of wrist position," Institut d/Organitzacio i Control de Sistemes Ind.. Barcelona, Spain, Tech. Rep. IOC-DT-P-2005-1, Jan. 2005.
[17] G. Wöhlke, "Grasp planning & force computation for dextrous object manipulation with multi-finger robot hands," Int. Comput. Sci. Inst. Berkeley, CA, Tech. Rep. TR-94-018, Aug. 1994.
[18] D. Hong Contact force distribution for robotic hand grasping. Blacksburg, Tech. Rep., 2004.
[19] M. Nahon and J. Angeles, "Real-time force optimization in parallel kinematic chains under inequality constraints," *IEEE Trans. Robot. Autom.*, vol. 8, no. 4, pp. 439–450, Aug. 1992.
[20] C. Remond, V. Perdereau, and M. Drouin, "A multi-fingered hand control structure with on-line grasping force optimization," Laboratoire ides Instruments et Systèmes. Paris, France.
[21] X. Chen, K. Watanabe, K. Kiguchi, and K. Izumi, "Optimal force distribution for the legs of a quadruped robot," *Mach. Intell. Robot. Control*, vol. 1, no. 2, pp. 87–94, 1999.
[22] G. Liu and Z. Li, "Real-time grasping-force optimization for multifingered manipulation: Theory and experiments," *IEEE/ASME Trans. Mechatron.*, vol. 9, no. 1, pp. 65–77, Mar. 2004.
[23] R. Murray, S. Sastry, and Z. Li, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC Press, 1994.
[24] M. Buss, H. Hashimoto, and J. Moore, "Dextrous hand grasping force optimization," *IEEE Trans. Robot. Autom.*, vol. 12, no. 3, pp. 406–418, Jun. 1996.
[25] M. Buss, L. Faybusovich, and J. Moore, "Dikin-type algorithms for dexterous grasping force optimization," *Int. J. Robot. Res.*, vol. 17, no. 8, pp. 831–839, 1998.
[26] L. Han, J. Trinkle, and Z. Li, "Grasp analysis as linear matrix inequality problems," *IEEE Trans. Robot. Autom.*, vol. 16, no. 6, pp. 663–674, Dec. 2000.
[27] U. Helmke, K. Hüper, and J. Moore, "Quadratically convergent algorithms for optimal dextrous hand grasping," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 138–146, Apr. 2002.
[28] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra Appl.*, vol. 284, pp. 193–228, Nov. 1998.
[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[30] H. Mittelmann, "An independent benchmarking of SDP and SOCP solvers," *Math. Program. Ser. B*, vol. 95, pp. 407–430, 2003.
[31] Yu. Nesterov and A. Nemirovsky, "Interior-point polynomial methods in convex programming," in *Studies in Applied Mathematics,* vol. 13. Philadelphia, PA: SIAM, 1994.
[32] R. Howe and M. Cutkosky, "Practical force-motion models for sliding manipulation," *Int. J. Robot. Res.*, vol. 15, no. 6, pp. 557–572, 1996.
[33] R. Howe, I. Kao, and M. Cutkosky, "The sliding of robot fingers under combined torsion and shear loading," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1988, pp. 103–105.
[34] N. Xydas and I. Kao, "Modeling of contact mechanics and friction limit surfaces for soft fingers in robotics, with experimental results," *Int. J. Robot. Res.*, vol. 18, no. 9, pp. 941–950, 1999.

Stephen P. Boyd (S'82–M'85–SM'97–F'99) received the A.B. degree (*summa cum laude*) in mathematics from Harvard University, Cambridge, MA, in 1980, and the Ph.D. degree in electrical engineering and computer science from the University of California at Berkeley, Berkeley, in 1985.

He is currently the Samsung Professor of engineering in the Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA. His current research interests include convex programming applications in control, signal processing, and circuit design. He is the author of the books *Linear Controller Design: Limits of Performance* (1991), *Linear Matrix Inequalities in System and Control Theory* (1994), and *Convex Optimization* (2004).

Prof. Boyd is the recipient of the Office of Naval Research (ONR) Young Investigator Award, a Presidential Young Investigator Award, the 1992 American Automatic Control Council (AACC) Donald P. Eckman Award, the Perrin Award for Outstanding Undergraduate Teaching in the School of Engineering, and an Associate Students of Standford University (ASSU) Graduate Teaching Award. In 2003, he received the AACC Ragazzini Education Award. He is a Distinguished Lecturer of the IEEE Control Systems Society, and holds an honorary Ph.D. from the Royal Institute of Technology (KTH), Stockholm, Sweden.

Ben Wegbreit (M'71) received the B.E.S. degree in engineering from Johns Hopkins University, Baltimore, MD, in 1966, and the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA, in 1970.

He is the Chairman and the CEO of Strider Labs, Palo Alto, CA, a robotics company. His current research interests include robotic manipulation, end effector mechanisms, vision, robotic planning, and the broad commercialization of robotics. He taught at Harvard, and then, joined Xerox PARC, where he worked in its halcyon years. He has co-founded several companies, two of which he took public: Convergent Technologies and Epiphany. He spent a decade as a venture capitalist. He has served on the board of directors of more than a dozen companies. Additionally, he has made early-stage private investments in various companies including Genentech, Adobe, Mercury Interactive, and iRobot. He has been a Visiting Scientist at Stanford University, Harvard, and Massachusetts Institute of Technology (MIT). He has served on the editorial board of the *Communications of the Association for Computing Machinery (ACM)*.

Dr. Wegbreit elected as a Fellow of the ACM, in 1993.