Quadratically Convergent Algorithms for Optimal Dextrous Hand Grasping

Uwe Helmke, *Senior Member, IEEE*, Knut Hüper, and John B. Moore, *Fellow, IEEE*

***Abstract*—There is a robotic balancing task, namely real-time dextrous-hand grasping, for which linearly constrained, positive definite programming gives a quite satisfactory solution from an engineering point of view. We here propose refinements of this approach to reduce the computational effort. The refinements include elimination of structural constraints in the positive definite matrices, orthogonalization of the grasp maps, and giving a precise Newton step size selection rule.**

***Index Terms*—Dextrous hand, gradient flow, Newton algorithm, optimal grasping, positive definite programming, Riemannian geometry, robotic hand.**

I. Introduction

In ROBOTICS, a key issue is the coordination of independent actuators to achieve a common goal. Thus, for multiple robots lifting an object, walking robots, or a robotic hand grasping and manipulating an object, there must be some balance and optimization of forces. The optimization, which is in essence a mathematical task, must achieve useful grasp plans for implementation in real time. For online dextrous hand grasping in robotics, a requirement is to develop real-time schemes which result in minimal and balanced contact forces satisfying friction cone constraints.

The earlier context for this research starts with [7] and [8] where linear programming techniques are used, but with ill-conditioning problems. Nonlinear programming techniques, as applied in [9], lead to an essentially off-line approach, which is not practical for real-time implementation.

In [1] and [2], linearly constrained positive definite programming methods are developed for an online grasping optimization task. The algorithms appear at times to be quadratically convergent, although this was not guaranteed by any theory, and the selection of the step size involved in the algorithms requires an *ad hoc* line search. Nevertheless, these algorithms are one or two orders of magnitude faster than earlier schemes proposed in the literature, and the optimal solutions calculated appear to be relatively more acceptable in engineering terms. There remains a challenge to achieve guaranteed quadratic convergence, and even faster algorithms if possible. In addition, in the event of changing external forces, or nonfeasible initial conditions, there is a challenge to achieve robust online convergence to the optimal solution. The cost index from [1], [2] appears to be an appropriate one, so there is no real need to refine this aspect in advancing the methods.

The online optimization schemes in [1] and [2] are based on the observation that the friction inequality constraints at the finger contacts can be viewed as a positive definiteness constraint of a matrix, denoted P, which is linear in the contact forces. The balancing of internal and external forces imposes additional linear constraints. The cost function is linear in both P and either P^{-1}, or $-\log(\det(P))$. The penalty term involving P^{-1} or $\det(P)$ ensures that, with an initial positive definite P, a gradient algorithm achieves an optimal P which is positive definite. Slippage at the finger contacts occurs if $\det(P)$ is zero, and there is loss of contact if P becomes indefinite. The linear cost on P ensures that the totality of finger forces is minimal.

Initial insights into the optimization, outlined in [1], arose from the study of gradient flow methods for balancing problems as in [4], and mild generalizations of these. Subsequently, discrete-time versions of these gradient flows with guaranteed global convergence properties have been developed using a Dikin step size familiar to linear and quadratic programming [2]. The approaches in [1] and [2], however, did not lead to precise step-size selection with guaranteed convergence properties, but were based on line-search arguments.

In more recent work [5], the cost index of [1] and [2] is optimized using a generic linear matrix inequality (LMI) semi-definite programming approach [3], [6], [10], [12]. This is claimed in [5] to achieve convergence with less computational effort. Actually, key differences in the LMI approach to that of [1] and [2] turn out to be the step-size selection in a Newton-based scheme, and the handling of the linear constraints. There is a factor of four or so improvement claimed for one example. This relative success underlines the question as to whether or not there is room to surpass the LMI algorithm performance with a more specialized algorithm.

We introduce a number of enhancements and generalizations of the methods of [1] and [2], some of which also apply to enhance the Newton-type LMI approach of [5]. In this work, global convergence is shown involving precise step size selection, with guaranteed *local quadratic convergence* in the neighborhood of the unique global optimum. Thus, convergence occurs to the accuracy of the computer, typically in less than 10

Manuscript received March 22, 2001; revised September 29, 2001. This paper wa recommended for publication by Associate Editor J. Ponce and Editor I. Walker upon evaluation of the reviewers' comments. This work was supported in part by the German–Israeli Foundation for Scientific Research and Development under Grant GIF-I-526-034.06/97 and by the Hong Kong Competitive Earmarked Research Grant.

U. Helmke and K. Hüper are with the Department of Mathematics, University of Würzburg, D-97074 Würzburg, Germany (e-mail: helmke@mathematik.uni-wuerzburg.de; hueper@mathematik.uni-wuerzburg.de).

J. B. Moore is with The Department of Information Engineering, The Chinese University of Hong Kong, Sha Tin, NT, Hong Kong, on leave from the Department of Systems Engineering, Research School of Information Sciences, Australian National University, Canberra, ACT 0200, Australia (e-mail: john.moore@syseng.anu.edu.au).

Publisher Item Identifier S 1042-296X(02)04300-8.

iterations for generic cases, so that online systems can be implemented with confidence, rather than merely relying on the experience that they usually work well. Our main convergence result is reminiscent of similar results in convex programming [10]–[12]. In such a convex programming approach, the step size is selected as unity in the vicinity of an optimum, and otherwise according to a line search. The criteria for determination of which step-size selection to take depends on whether or not the estimate is inside a Dikin ellipsoid. There is inherent discontinuity in the algorithm. Our approach of deriving approximate step sizes is different, in so far as quadratic convergence is achieved by a continuous step size selection scheme. The continuity property enables us to develop a convergence theory using only relatively straightforward ideas from Calculus. We believe that this technique is new in this application and may be of independent interest.

There is additional computational effort reduction resulting from a number of specific contributions:

- Formulation of the finger force inequality constraints in terms of a 2×2 positive definite matrix in the point contact case, rather than in terms of a 3×3 matrix as in [1], [2], and in terms of a 2×2 positive definite complex Hermitian matrix, for the soft contact case, rather than as one 4×4 real symmetric matrix. This circumvents the need to maintain structural linear constraints, which are clearly artifacts, as well as achieving "dramatic" computational effort reduction for the Newton-type algorithm.
- *A priori* orthogonalization of the linear grasping force constraints, which simplifies the computations for the linear constraints for the Euclidean gradient algorithms.
- Calculation of a step size which is guaranteed to give a reduced cost, achieving local quadratic convergence. This can also initialize a proposed quadratically convergent line search algorithm. Asymptotically, the step size is unity for quadratic convergence.
- The optimum step size is observed to be the smallest real zero of a polynomial constructed from quadratic polynomials associated with each finger, and a maximum step size for remaining within the cone is given in terms of the solution of quadratic equations associated with each finger.

In Section II, the robotic dextrous hand grasping constraints are reformulated to simplify positive definite programming. In Section III, the cost function to optimize grasp forces is given and its relevant properties. In Section IV, relatively simple-to-calculate Newton-type algorithms, based on Riemannian gradients are studied. Novel, explicit step size selections for our algorithms appear in Section V, together with the main quadratic convergence results. Conclusions are drawn in Section VI.

II. Grasping Constraints

Consider the simplest of all grasping problems, namely that of a statically balanced grasp using point or soft finger contacts. See [1] for a more complete context of robotic grasping and formulation of optimization tasks.

A. Grasping Constraints: Background

1) Constraint Equalities and Inequalities: Consider N fingers with the *point contact* forces at the i-th finger denoted $c_{i,1} \in \mathbb{R}$, the normal force component, and $c_{i,2}, c_{i,3} \in \mathbb{R}$, the tangential components. Coulomb's law for a point contact friction model (with no slippage) is that for each i

$$c_{i,2}^2 + c_{i,3}^2 < \mu_i^2 c_{i,1}^2, \quad c_{i,1} > 0 \tag{1}$$

where $\mu_i > 0$ denotes the Coulomb friction at the point contact of the ith finger. Denoting c as the vector

$$\begin{aligned} c &= [c_1' \quad c_2' \quad \cdots \quad c_N']' \in \mathbb{R}^{3N} \\ c_i &= [c_{i,1} \quad c_{i,2} \quad c_{i,3}]' \in \mathbb{R}^3 \end{aligned} \tag{2}$$

then the balance of external forces can be written as a linear equation

$$Wc = f_{\text{ext}} \in \mathbb{R}^6. \tag{3}$$

The grasp map $W \in \mathbb{R}^{6 \times 3N}$ is necessarily full rank for so-called force closure [5]. It contains the $3N$ contact wrench directions in its columns and maps forces from the contact frames to the coordinate frame of the grasped object center of mass.

For the case of *soft finger contact* forces, the inequality constraints in an elliptic approximation are

$$\alpha_i^2 \left(c_{i,2}^2 + c_{i,3}^2\right) + \beta_i^2 c_{i,4}^2 < c_{i,1}^2, \quad c_{i,1} > 0 \tag{4}$$

where $\alpha_i = \sqrt{(1/\mu_i)}, \beta_i = \sqrt{(1/\mu_{t,i})}$, $\mu_{t,i} > 0$ model the relation between torsion and shear limits, and $c_{i,4} \in \mathbb{R}$ is the component of moment about the contact normal.

There are also *joint effort* constraint inequalities, discussed in [5], but these are omitted from consideration for simplicity of presentation. They present no particular difficulties to include within the subsequent theory.

2) Constraints as Linearly Constrained Cones: Recall that a key observation of [1] is that the inequalities (1) for the point contact case are equivalent to the positive definiteness condition

$$P = P' = \text{Blockdiag}(P_1, \ldots, P_N) > 0 \tag{5}$$

where the P_i are given in terms of 3×3 matrices, linear in $c_{i,j}$. There are also structural constraints in P_i that the diagonal elements be identical and that two elements are zero. There are thus $3N$ such constraints, augmenting the constraint (3), of the form

$$\text{tr}(A_j P) = b_j, \quad \text{for } j = 1, \ldots, m = 3N + 6 \tag{6}$$

where A_j have the same block diagonal structure as P

$$A_j = \text{Blockdiag}(A_{j,1}, \ldots, A_{j,N}). \tag{7}$$

The $A_{j,i}$ are 3×3 real matrices.

For the soft finger contact case, the contact forces are characterized by (5) where now the P_i are 4×4. Again P_i is linear in the contact forces $c_{i,j}$ and has $6N$ linear structural constraints, in that its diagonal terms are identical and some off-diagonal terms are zero.

3) Simplification of Cone Constraints: Point Contact Case: A first observation, important for computational effort reduction, is that the inequalities (1) are equivalent to the positive definiteness of (5), but now with the $P_i > 0$ given in terms of 2×2 matrices, rather than 3×3 matrices, as

$$P_i = \begin{bmatrix} p_{i,1} & p_{i,2} \\ p_{i,2} & p_{i,3} \end{bmatrix} = \begin{bmatrix} \mu_i c_{i,1} + c_{i,2} & c_{i,3} \\ c_{i,3} & \mu_i c_{i,1} - c_{i,2} \end{bmatrix} > 0. \tag{8}$$

This constraint is equivalent to (1), since the trace and determinant of P_i are both positive. The number of linear constraints m is reduced by $3N$ to $m = 6$. Similarly, the matrices A_j are now block diagonal with 2×2 symmetric subblocks $A_{j,i}$.

4) Cone Constraints Simplification: Soft Finger Case: Computational savings can be made as well for this case, and robustness achieved, by working with the complex Hermitian block diagonal matrix

$$P = P^H = \text{Blockdiag}(P_1, \ldots, P_N), \tag{9}$$

$$P_i = \begin{bmatrix} c_{i,1} + \alpha_i c_{i,2} & \alpha_i c_{i,3} - j\beta_i c_{i,4} \\ \alpha_i c_{i,3} + j\beta_i c_{i,4} & c_{i,1} - \alpha_i c_{i,2} \end{bmatrix} > 0. \tag{10}$$

There is a corresponding block diagonal structure for complex Hermitian A_j, with 2×2 submatrices $A_{j,i}$. Note that the diagonal elements of P are real, and that when $c_{i,4} = 0$, the point contact case is recovered. These soft finger cone constraints (10) are identical to (4), since the trace and determinant of P_i are both positive. Again there is a reduction of the dimensions of A_j and the number of constraints m is reduced by $6N$ to $m = 6$.

5) Computational Effort and Robustness Implications: There is a factor of two reduction in effort for block multiplication. The main computational effort in the Riemannian gradient P update equations in [2] [see also (31), (32), and (38)] is in calculating an $m \times m$ matrix and its inverse. Thus reducing m, for example from $3N + 6$ to 6, amounts to considerable computational savings. Also, any potential numerical difficulties staying on the constraint submanifold associated with the structural constraints are removed.

6) Orthogonalizing the Grasp Maps: We assume throughout the paper that the grasp map is full rank, that is $A_1, \ldots, A_6$ are linearly independent. An observation which leads to computational effort reduction for calculating Euclidean gradients, but not for the Riemannian metric gradients, is to organize the constraints (6) so that the $A_1, \ldots, A_6$ are orthogonal, i.e.,

$$\text{tr}(A_i A_j') = \delta_{ij} \tag{11}$$

where δ_{ij} is 0 if $i \neq j$ and unity otherwise.

For the point contact case, define

$$p = [p_1' \quad p_2' \quad \cdots \quad p_N']', \quad p_i = [p_{i,1} \quad p_{i,2} \quad p_{i,3}]'$$
$$p = Jc, \quad J = \text{Blockdiag}(J_1, \ldots, J_N) \tag{12}$$

where

$$J_i = \begin{bmatrix} \mu_i & 1 & 0 \\ 0 & 0 & 1 \\ \mu_i & -1 & 0 \end{bmatrix}, \quad J_i^{-1} = \begin{bmatrix} \frac{1}{2\mu_i} & 0 & \frac{1}{2\mu_i} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & 1 & 0 \end{bmatrix}. \tag{13}$$

The constraint $Wc = f_{\text{ext}}$ can then be rewritten in terms of p and and a $6 \times 3N$ matrix Θ with orthonormal rows as

$$\Theta p = b \tag{14}$$

where $\Theta\Theta' = I$, and

$$b = Sf_{\text{ext}}, \quad S = ((WJ^{-1})(WJ^{-1})')^{-\frac{1}{2}}, \quad \Theta = SWJ^{-1}. \tag{15}$$

Now denote the elements of Θ as $\Theta_{j,i}$ and the jth row of (14) as $\Theta_j p = b_j$. Then, by working with one row at a time, (14) can be written as (6) where

$$\begin{aligned} &\text{tr}(A_j P) = b_j \quad \text{for } j = 1, 2, \ldots, 6, \\ &A_j = \text{Blockdiag}(A_{j,1}, \ldots, A_{j,N}), \\ &A_{j,1} = \begin{bmatrix} \Theta_{j,1} & \Theta_{j,2} \\ 0 & \Theta_{j,3} \end{bmatrix}, \quad A_{j,2} = \begin{bmatrix} \Theta_{j,4} & \Theta_{j,5} \\ 0 & \Theta_{j,6} \end{bmatrix}, \ldots \end{aligned} \tag{16}$$

The orthogonality of Θ ensures the orthogonality of the A_j. That is, (11) holds. Notice that, in order to derive orthogonality, we have chosen not to work with symmetric A_j. However, replacing A_j by the symmetric matrix $(A_j + A_j')/2$, and noting that $\text{tr}(A_j P) = \text{tr}(A_j' P)$, we can assume without loss of generality that A_j is symmetric. Therefore, we assume this subsequently.

The corresponding soft finger results follow likewise.

7) Computational Effort Reduction From Orthogonalization: One implication of orthogonalization is that working with the six constraints (17), there is a computational reduction in calculating the Euclidean gradient (25) and (27). There is not any reduction for the Riemannian metric gradient calculation (31), (32), and (38). The Riemannian metric gradient turns out, as we show below, to be a Newton direction.

It is known that Newton algorithms, although quadratically convergent in the neighborhood of the optimum, are usually not faster than the linearly convergent gradient algorithms outside this neighborhood. The computational savings from orthogonalization of the grasp map, by a factor of 3 to 5 on typical grasping examples, are an incentive to use a Euclidean gradient scheme initially, for say three or four iterations at the cost of one Newton iteration. Then, it is best to switch to the more expensive Newton algorithm for the last few iterations.

III. Grasping Cost Function and Properties

For simplicity, we focus on the point contact case. The analysis for the soft finger case follows along similar lines. Let $\mathcal{P}_B(n)$ denote the set of block diagonal, real or complex Hermitian positive definite $n \times n$ matrices $P = P^H > 0$, consisting of N 2×2 blocks P_i. Of course, $n = 2N$. Denote the affine constraints as $\mathcal{L}(n) := \{P \,|\, \mathrm{tr}(A_j P) = b_j,\ j = 1, 2, \ldots, 6\}$, and the constraints on P as $\mathcal{P}_{B\mathcal{L}} := \mathcal{P}_B(n) \cap \mathcal{L}(n)$.

Consider the cost index $\Phi : \mathcal{P}_{B\mathcal{L}}(n) \to \mathbb{R}$

$$\Phi(P) = \mathrm{tr}(P) - \log(\det(P)). \tag{17}$$

More general indices with positive definite weighting matrices on P in each of the terms of (17) can be considered as well along the lines of the following theory. However, we will not do so here.

We now consider, in turn, some features of the cost function which lead to an optimization with guaranteed convergence.

A. Convexity of the Cost Function

It is known from [1] and [2] that such cost functions as in (17) have compact sublevel sets on $\mathcal{P}_{B\mathcal{L}}$, ensuring the existence of global minima. Moreover, the cost function on $\mathcal{P}_{B\mathcal{L}}$ is strictly convex. This implies that there are no critical points other than a unique global minimum, denoted P^*. A proof of this result is included for completeness and to set up some notation.

The tangent space of $\mathcal{P}_{B\mathcal{L}}$ is the $(3N-6)$-dimensional space $T_P := T_P(\mathcal{P}_{B\mathcal{L}})$. That is,

$$T_P = \{\Omega \in \mathcal{S}_B(n) \,|\, \mathrm{tr}(A_j \Omega) = 0 \text{ for all } j = 1, 2, \ldots, 6\} \tag{18}$$

where $\mathcal{S}_B(n)$ denotes the set of block diagonal, real or complex Hermitian $n \times n$ matrices Ω, consisting of N 2×2 blocks Ω_i. For any $\Delta \in T_P$, consider the cost function $\phi(\alpha)$ and its derivatives $\phi'(\alpha), \phi''(\alpha)$ with respect to α

$$\begin{aligned}\phi(\alpha) &= \mathrm{tr}(P + \alpha\Delta) - \log(\det(P + \alpha\Delta))\\ \phi'(\alpha) &= \mathrm{tr}(\Delta) - \mathrm{tr}((P + \alpha\Delta)^{-1}\Delta)\\ \phi''(\alpha) &= \mathrm{tr}(((P + \alpha\Delta)^{-1}\Delta)^2).\end{aligned} \tag{19}$$

Clearly, $\phi''(0) > 0$, implying strict convexity of $\Phi(P)$ at any P.

We show next that the optimization task is well posed.

Theorem III.1: The function $\Phi : \mathcal{P}_{B\mathcal{L}} \to \mathbb{R}$ is strictly convex with compact sublevel sets and

$$\lim_{P \to \partial\mathcal{P}_{B\mathcal{L}}} \Phi(P) = \infty.$$

The Hessian of Φ at any point $P \in \mathcal{P}_{B\mathcal{L}}$ is

$$\mathcal{H}_\Phi(P)(\Omega_1, \Omega_2) = \mathrm{tr}(P^{-1}\Omega_1 P^{-1}\Omega_2) \tag{20}$$

and is positive definite. In particular, there is a unique local and global minimum

$$P^*(A_j, b_j) \in \mathcal{P}_{B\mathcal{L}}$$

of Φ. Moreover, P^* depends smoothly on A_j and b_j, $j = 1, \ldots, 6$.

Proof: By the above argument, Φ is a sum of the convex function $\mathrm{tr}(P)$ and the strictly convex function $-\log(\det(P))$ and is therefore strictly convex.

The Hessian of Φ in $\mathcal{P}_B$ coincides with that of $-\log\det P$ and is thus given as in the theorem. The formula for the Hessian on $\mathcal{P}_{B\mathcal{L}}$ follows, since the restriction of a Hessian to a linear subspace is the Hessian of the restriction to the subspace.

The last claim follows from a simple application of the Implicit Function Theorem. To this end, let

$$\begin{aligned}(A, b) &:= ((A_1, b_1), \ldots, (A_6, b_6)) \in (\mathcal{S}_B(n) \times \mathbb{R}^n)^6\\ \Gamma &:= \{(A, b) \,|\, \mathrm{tr}(A_i A_j) = \delta_{ij}\}.\end{aligned}$$

Clearly, $\mathcal{S}_B(n) = T_P \oplus T_P^\perp$ where $T_P^\perp = \mathrm{span}\{A_1, \ldots, A_6\}$. Consider the smooth function $\Psi : \mathcal{P}_B(n) \times \Gamma \to T_P \oplus T_P^\perp$ defined as

$$\Psi(P, (A, b)) := \left(\nabla\Phi(P), \sum_j ((\mathrm{tr}(A_j P) - b_j) A_j)\right) \tag{21}$$

where the Euclidean gradient $\nabla\Phi(P)$ is defined subsequently in (25). Thus, $\Psi(P, (A, b)) = 0$ if and only if $P = P^*(A, b) \in \mathcal{P}_{B\mathcal{L}}$. The claim follows from the Implicit Function Theorem, once it is verified that the partial derivative of Ψ with respect to P induces a linear isomorphism from $\mathcal{S}_B(n)$ onto $T_P \oplus T_P^\perp$. To see this we decompose any tangent vector $\Omega \in \mathcal{S}_B(n)$ as $\Omega = \Omega_t + \Omega_n$ where $\Omega_t \in T_P$ and $\Omega_n \in T_P^\perp$. Obviously, $\mathrm{tr}(A_j \Omega_t) = 0$ for all j, and the restriction

$$T_P^\perp \to T_P^\perp, \quad \Omega_n \mapsto \sum_j \mathrm{tr}(A_j \Omega_n) A_j \tag{22}$$

is a linear isomorphism. In fact, $\Omega_n = \sum_j \mu_j A_j$ for unique μ_j and

$$(\mu_1, \ldots, \mu_6) \mapsto \sum_i \mathrm{tr}\left(A_i \left(\sum_j \mu_j A_j\right)\right) A_i = \sum_i \mu_i A_i \tag{23}$$

is a linear isomorphism.

Finally, the linearization of $\nabla\Phi(P)$ in the direction Ω_t is seen from (25) as the linear map $D(\nabla\Phi(P)) : T_P \to T_P$, defined by

$$D(\nabla\Phi(P))\Omega_t = P^{-1}\Omega_t P^{-1} - \sum_{i=1}^{6} \gamma_i' A_i \tag{24}$$

where $\gamma_i' = \mathrm{tr}(A_i P \Omega_t P),\ i = 1, \ldots, 6$. Suppose Ω_t is in the kernel of $D(\nabla\Phi(P)) : T_P \to T_P$. Then

$$\Omega_t = \sum_{i=1}^{6} \gamma_i' P A_i P$$

and thus for $j = 1, \ldots, 6$

$$0 = \mathrm{tr}(A_j \Omega_t) = \sum_{i=1}^{6} \gamma_i' \,\mathrm{tr}(A_j P A_i P).$$

By positive definiteness of the 6×6 matrix with (ij)th entry equal to $\mathrm{tr}(A_j P A_i P)$, this implies $\gamma_1' = \cdots = \gamma_6' = 0$, and therefore $\Omega_t = 0$. This shows that the linearization is injective, and hence is invertible at any P. The result follows. ■

B. Euclidean Gradient Algorithm

The Euclidean gradient is

$$\nabla\Phi(P) = I - P^{-1} - \sum_{i=1}^{6}\gamma_i A_i, \quad \gamma_i = \mathrm{tr}(A_i(I - P^{-1})). \tag{25}$$

Here we have assumed that the A_i are orthogonalized as in (17). Both gradients are in the tangent space T_P of (18). To verify this, first observe that $\mathrm{tr}(A_j\nabla\Phi(P)) = 0$ for all j. Moreover, $\nabla\Phi(P) \in T_P(\mathcal{P}_{B\mathcal{L}})$. Also, the directional derivative $D\Phi(P)\cdot(\xi)$ satisfies

$$D\Phi(P)\cdot(\xi) = \mathrm{tr}(\nabla\Phi(P)\xi) \quad \text{for all } \xi \in \mathcal{P}_{B\mathcal{L}}. \tag{26}$$

The standard Euclidean gradient algorithm for convex $\Phi(P)$ is

$$P_{k+1} = P_k - \alpha_k\nabla\Phi(P_k) =: \mathcal{F}(P_k). \tag{27}$$

This clearly goes in a "downhill" direction, if $\alpha_k > 0$. For $P_k \neq P^*$ and $\alpha_k > 0$ sufficiently small, this step achieves a reduced cost. The step size is chosen small enough to preserve positive definiteness of P. More precisely, and referring to [2, Theorems 4–6], it is chosen so that the mapping $\mathcal{F} : \mathcal{P}_{B\mathcal{L}} \to \mathcal{P}_{B\mathcal{L}}$ is a continuous map with the property

$$\Phi(P_{k+1}) < \Phi(P_k), \qquad \text{for all } P_k \neq P^*. \tag{28}$$

In [1] and [2], an explicit choice of a step size α_k guaranteeing convergence is not given, and the line search arguments have been implicit, rather than explicit. We will not consider this algorithm any further, as our step size selections do not lead to a quadratically convergent algorithm.

C. Newton Algorithm

Quadratic convergence rates for optimizing the strictly convex function $\Phi(P)$ can be achieved by working with the Hessian matrix $\mathcal{H}_\Phi(P)$ and a Newton algorithm, as

$$P_{k+1} = P_k - \alpha_k\mathcal{H}_\Phi(P_k)^{-1}\nabla\Phi(P_k). \tag{29}$$

For suitable step-size selection α_k, we prove global and local quadratic convergence to the optimal solution P^*.

In applying the Newton algorithm, the computation of the inverse of the Hessian requires arithmetic operations of order $(3N)^3$ for the point contact case, and $(4N)^3$ for the soft finger case. To see this, rewrite the algorithm in terms of vectors rather than matrices, and note the vector dimensions are $3N$ or $4N$, respectively. We revisit this algorithm below, showing that the Newton step can be effectively calculated as a Riemannian gradient step using only order 6^3 multiplications.

D. Riemannian Metric and Gradients

Let us endow $\mathcal{P}_{B\mathcal{L}}$ with the Riemannian metric

$$g(P;\xi,\eta) = \mathrm{tr}(P^{-1}\xi P^{-1}\eta) \tag{30}$$

where ξ,η are block diagonal matrices with the same structure as P and A_j, with 2×2 sub blocks ξ_i,η_i. The explicit gradient with respect to this metric [11], being in the tangent space of $\mathcal{P}_{B\mathcal{L}}$ is

$$\mathrm{grad}\,\Phi(P) = P\left(I - P^{-1} - \sum_{i=1}^{6}\beta_i A_i\right)P \tag{31}$$

where the β_i come from the solution of

$$\begin{bmatrix}\beta_1\\ \vdots\\ \beta_6\end{bmatrix} = G^{-1}\begin{bmatrix}\mathrm{tr}(A_1(P^2 - P))\\ \vdots\\ \mathrm{tr}(A_6(P^2 - P))\end{bmatrix}. \tag{32}$$

Here the matrix G with (ij)th entry equal to $\mathrm{tr}(A_iPA_jP)$ is necessarily full rank for $P > 0$ and the A_j are linearly independent for all j. Note that

$$\mathrm{tr}(A_i\,\mathrm{grad}\,\Phi(P)) = 0, \qquad \text{for all } i. \tag{33}$$

Note also that there is no computational simplification due to the *a priori* orthogonalization of the grasp map, as for the Euclidean gradient. We would need to "orthogonalize" at every step the products $P_k^{1/2}A_jP_k^{1/2}$, in the same way as we "orthogonalized" A, in order to achieve the simplification $G = I$. For the Euclidean gradient, the corresponding G is simply $G = (\mathrm{tr}(A_iA_j))_{ij} = I$.

IV. Newton Algorithm via Riemannian Gradient

A. The Hessian

For $\Phi(P)$ on $\mathcal{P}_B(n)$, the Hessian is the quadratic form, given from (20), as $\mathcal{H}_\Phi(P)(\Omega,\Omega) = \mathrm{tr}(P^{-1}\Omega P^{-1}\Omega)$. The restriction of the Hessian on $\mathcal{L}(n)$ satisfies, noting (33)

$$\begin{aligned}\mathcal{H}_\Phi(P)(\mathrm{grad}\,\Phi(P),\Omega) &= \mathrm{tr}(P^{-1}\mathrm{grad}\,\Phi(P)P^{-1}\Omega)\\ &= \mathrm{tr}((I - P^{-1})\Omega)\\ &= \mathrm{tr}(\nabla\Phi(P)\Omega)\end{aligned} \tag{34}$$

for all tangent vectors $\Omega \in T_P(\mathcal{P}_B(n) \cap \mathcal{L}(n))$.

As a consequence of (34), we obtain

$$\begin{aligned}\nabla\Phi(P) &= \mathcal{H}_\Phi(P)\,\mathrm{grad}\,\Phi(P)\\ \mathrm{grad}\,\Phi(P) &= \mathcal{H}_\Phi(P)^{-1}\nabla\Phi(P)\end{aligned} \tag{35}$$

where $\mathcal{H}_\Phi(P)^{-1} : T_P \to T_P$ is the linear isomorphism between tangent spaces, defined by

$$u \mapsto \mathcal{H}_\Phi(P)^{-1}u = u^* \tag{36}$$

with

$$\mathcal{H}_\Phi(P)(u^*,\Omega) = \mathrm{tr}(u\Omega), \qquad \text{for all } \Omega. \tag{37}$$

Note that the linear map $\mathcal{H}_\Phi(P)$ is a well-defined linear isomorphism, as the Hessian $\mathcal{H}_\Phi(P)(\cdot,\cdot)$ is nondegenerate.

B. Newton Algorithm Revisited

Now a Newton algorithm, seeking to minimize $\Phi(P)$ on $\mathcal{P}_{B\mathcal{L}}$, is simply, via (35), a gradient algorithm with respect to the Riemannian metric (30)

$$\begin{aligned}P_{k+1} &= P_k - \alpha_k\mathcal{H}_\Phi(P_k)^{-1}\nabla\Phi(P_k)\\ &= P_k - \alpha_k\,\mathrm{grad}\,\Phi(P_k).\end{aligned} \tag{38}$$

The computations for the Riemannian metric gradient version of the Newton algorithm are considerable simpler than for the standard Euclidean version, being of order 6^3 arithmetic operations compared to $(3N)^3$ for the point contact case.

V. Explicit Step-Size Selection for Quadratic Convergence

In order to numerically implement the Newton algorithm, the step-size factor α_k has to be appropriately chosen. To this end, we consider at each time instant the "downhill" gradient direction $\Delta = -\mathrm{grad}\,\Phi(P)$ in the tangent space T_P.

Consider the cost function $\phi(\alpha)$ and its derivatives with respect to α, as in (20). Now Δ inherits the same block structure of P, so that

$$\phi(\alpha) = \sum_{i=1}^{N} \mathrm{tr}(P_i + \alpha\Delta_i) - \log\left(\det \prod_{i=1}^{N}(P_i + \alpha\Delta_i)\right)$$
$$\phi'(\alpha) = \sum_{i=1}^{N}\left(\mathrm{tr}(\Delta_i) - \frac{p_i'(\alpha)}{p_i(\alpha)}\right) = \mathrm{tr}(\Delta) - \frac{p'(\alpha)}{p(\alpha)} \qquad (39)$$

where $p_i(\alpha) = \det(P_i + \alpha\Delta_i)$. Convexity of $\Phi(P)$ ensures that the line search is a convex minimization task, at least for step size $\alpha \in [0, \alpha_{\max})$, where $\alpha_{\max}$ indicates the step size leading to the cone boundary.

The critical points of $\phi(\alpha)$ on $\alpha \in \mathbb{R}$ are given as the real roots of the polynomial equation

$$p(\alpha)\,\mathrm{tr}(\Delta) - p'(\alpha) = 0, \quad p(\alpha) := \prod_{i=1}^{N} p_i(\alpha). \qquad (40)$$

A. Optimum Step Size α^*

A preliminary observation is that, since $\phi'(\alpha)|_{\alpha=0} < 0$ and $\phi(\alpha)$ is convex for all $(P + \alpha\Delta) \in \mathcal{P}_{B\mathcal{L}}$, then the desired line search minimum for $\alpha \in [0, \alpha_{\max})$ occurs at the smallest positive real root α^* of the polynomial equation (40), with $\Delta = -\mathrm{grad}\,\Phi(P)$. This characterization does not yield an explicit formula for α^*, with guaranteed regularity properties at the optimal solution. We therefore must search for a useful approximation of α^* that is simple to calculate.

Another preliminary observation is that the maximum step size $\alpha_{\max}$, which keeps the step within $\mathcal{P}_B(n)$, is the smallest positive real root of

$$\det(P + \alpha\Delta) = \prod_{i=1}^{N} \det(P_i + \alpha\Delta_i) = 0. \qquad (41)$$

This root is found analytically by searching for the smallest real root of the second-order polynomial equations $\det(P_i + \alpha\Delta_i) = 0$ for $i = 1, 2, \ldots, N$. With any step size selection such that $P_k \to P^*$, as $k \to \infty$, it follows that $\Delta_i \to 0$ and $\alpha_{\max} \to \infty$. This may be compared to the Dikin step-size selection used in [1].

B. Explicit Step-Size Selection and Convergence Result

We now derive an explicit step-size selection that leads to quadratic convergence of the Newton algorithm.

For $\Delta = -\mathrm{grad}\,\Phi(P_k)$, $P := P_k$, $\alpha \geq 0$, and $L := P^{-1/2}\Delta P^{-1/2}$, consider

$$\begin{aligned}\phi(\alpha) &= \mathrm{tr}(P + \alpha\Delta) - \log\det(P + \alpha\Delta)\\ &= \phi(0) + \alpha\,\mathrm{tr}\,\Delta - \log\det(I + \alpha L).\end{aligned}$$

The first and the second derivative of ϕ are

$$\begin{aligned}\phi'(\alpha) &= \mathrm{tr}\,\Delta - \mathrm{tr}((I + \alpha L)^{-1}L)\\ \phi''(\alpha) &= \mathrm{tr}((I + \alpha L)^{-1}L)^2 = \|(I + \alpha L)^{-1}L\|^2 \geq 0.\end{aligned}$$

The *Newton Decrement*, $\lambda_0(P)$, is given as

$$\begin{aligned}\lambda_0(P) &:= \sqrt{\mathrm{tr}(\nabla\Phi(P)\mathcal{H}_\Phi(P)^{-1}\nabla\Phi(P))}\\ &= \sqrt{\mathrm{tr}(\nabla\Phi(P)\,\mathrm{grad}\,\Phi(P))}.\end{aligned} \qquad (42)$$

Since $\Delta = -\mathrm{grad}\,\Phi(P)$, and recalling (35), we obtain

$$\begin{aligned}-\phi'(0) &= \mathrm{tr}((I - P^{-1})\,\mathrm{grad}\,\Phi(P))\\ &= \mathrm{tr}(\nabla\Phi(P)\,\mathrm{grad}\,\Phi(P))\\ &= \lambda_0(P)^2.\end{aligned}$$

Moreover, since $P^{-1}\mathrm{grad}\,\Phi(P)P^{-1} = \nabla\Phi(P) + \sum_j c_j A_j$, for suitable c_j, then recalling (33), we have

$$\begin{aligned}-\phi'(0) &= \mathrm{tr}(P^{-1}\mathrm{grad}\,\Phi(P)P^{-1}\,\mathrm{grad}\,\Phi(P))\\ &= \phi''(0) = \|L\|^2.\end{aligned}$$

Therefore, the Newton decrement is $\lambda_0(P) = \|L\|$.

Let $\lambda := \|L\|_2$ denote the 2-norm, that is the largest singular value of L. For $t \geq 0$, then $I + tL \geq (1 - t\|L\|_2)I$ implies $(I + tL)^{-1}L^2(I + tL)^{-1} \leq (1 - t\lambda)^{-2}L^2$. Therefore

$$\phi''(t) \leq \frac{\|L\|^2}{(1 - t\|L\|_2)^2}$$

and thus by monotonicity

$$\sup_{0\leq t\leq\alpha} \phi''(t) \leq \frac{\|L\|^2}{(1 - \alpha\|L\|_2)^2}.$$

By the Mean Value Theorem, this implies

$$|\phi'(\alpha) - \phi'(0)| \leq \left(\sup_{0\leq t\leq\alpha} \phi''(t)\right)\alpha \leq \frac{\alpha\|L\|^2}{(1 - \alpha\|L\|_2)^2} \leq -\phi'(0) \qquad (43)$$

where the desired last inequality holds only if α is chosen such that

$$\alpha\|L\|^2 + (1 - \alpha\|L\|_2)^2\phi'(0) \leq 0. \qquad (44)$$

The smallest positive root of this quadratic polynomial is

$$\alpha_0^{**} = \frac{|\phi'(0)|}{\phi''(0)} \cdot \frac{1 + 2\lambda^{**} - \sqrt{1 + 4\lambda^{**}}}{2(\lambda^{**})^2} \qquad (45)$$

where $\lambda^{**} = \lambda(P)\cdot(|\phi'(0)|)/(\phi''(0)) > 0$. Observe that, in this case, since $-\phi'(0) = \|L\|^2 = \phi''(0)$, we have the simplified formula

$$\alpha_0^{**}(P) = \frac{1 + 2\lambda(P) - \sqrt{1 + 4\lambda(P)}}{2\lambda(P)^2}. \qquad (46)$$

Lemma V.1: The function $f : [0, \infty) \to \mathbb{R}$

$$f(x) = \frac{1 + 2x - \sqrt{1 + 4x}}{2x^2} \qquad (47)$$

is strictly monotonically decreasing with

$$f(0)=1, \quad \lim_{x\to\infty} f(x)=0.$$

Proof: Consider the Taylor series expansion

$$\sqrt{1+4x}=1+2x-2x^2+4x^3+O(x^4).$$

Therefore, we obtain $f(x)=1-2x+O(x^2)$, which proves $f(0)=1$ and $f'(0)=-2$. Simple manipulations show that the derivative $f'(x)$ is zero if and only if $(1+x)\sqrt{1+4x}=1+3x$. By squaring up and cancellations, this is seen to be equivalent to $x=0$. But $f'(0)<0$ and therefore $f'(x)$ must be negative for all $x\geq 0$, and the result follows. ∎

Since $\lambda(P)\leq\lambda_0(P)$, we conclude that $\alpha_0^{**}(P)\geq\alpha_0^*(P)$, where $\alpha_0^*(P)$ is defined by (48). Note that the lemma ensures that always

$$0<\alpha_0^*(P)\leq\alpha_0^{**}(P)\leq 1.$$

Moreover

$$\lim_{P\to P^*}\alpha_0^*(P)=1$$

holds if P converges to a critical point P^* of Φ. Furthermore, the function $P\mapsto\alpha_0^*(P)$ on $\mathcal{P}_{B\mathcal{L}}$ is continuous.

For any $0<\alpha<\alpha^*$, we have $\phi'(\alpha)<0$ and hence $\phi(\alpha)<\phi(0)$. Standard Lyapunov-type arguments as, for example, in [2], shows that the recursion (49) converges to the unique global minimum P^*.

To show quadratic convergence, we need a lemma.

Note that the result of the following lemma is well known from Calculus for the case of C^2-maps. Since this assumption is not satisfied in our application, we need to prove the result under the weaker regularity assumption.

Lemma V.2: Let U be an open subset of $\mathbb{R}^d$, and $\mathcal{F}:U\to U$ be a C^1-map such that the derivative $D\mathcal{F}(x)$ is Lipschitz continuous at any $x\in U$. Let $x^*\in U$ denote a fixed point of $\mathcal{F}$ with $D\mathcal{F}(x^*)=0$. Then the recursion $x_{k+1}=\mathcal{F}(x_k)$ in U is locally quadratically convergent to x^*.

Proof: Choose $r>0$ and $L>0$ such that, using the operator norm and Euclidean norm, respectively,

$$\|D\mathcal{F}(x)\|\leq L|x-x^*|$$

holds for all $x\in\mathbb{R}^d$ with $|x-x^*|\leq r$. Let $g(t):=\mathcal{F}(x^*+t(x-x^*))$, $0\leq t\leq 1$. Thus, g is a C^1 function and by the Fundamental Theorem of Calculus

$$\begin{aligned}|\mathcal{F}(x)-\mathcal{F}(x^*)|&=|g(1)-g(0)|\\&=\left|\int_0^1 g'(t)\,dt\right|\\&=\left|\int_0^1 D\mathcal{F}(x^*+t(x-x^*))(x-x^*)\,dt\right|\\&\leq\int_0^1\|D\mathcal{F}(x^*+t(x-x^*))\|\cdot|x-x^*|\,dt\\&\leq\frac{1}{2}L|x-x^*|^2\end{aligned}$$

using the Lipschitz bound. The result follows. ∎

We now state and prove the main convergence result of the paper.

Theorem V.1: For any $P\in\mathcal{P}_{B\mathcal{L}}$, let $\lambda_0(P)$ denote the Newton Decrement and let

$$\alpha_0^*(P)=\frac{1+2\lambda_0(P)-\sqrt{1+4\lambda_0(P)}}{2\lambda_0(P)^2}. \tag{48}$$

For any initial condition $P_0\in\mathcal{P}_{B\mathcal{L}}$ the algorithm

$$P_{k+1}=P_k-\alpha_0^*(P_k)\,\mathrm{grad}\,\Phi(P_k) \tag{49}$$

converges quadratically fast to the unique global minimum $P^*\in\mathcal{P}_{B\mathcal{L}}$ of Φ.

Moreover, the function $P\mapsto\alpha_0^*(P)$ is continuous on $\mathcal{P}_{B\mathcal{L}}$ and satisfies $\lim_{P\to P^*}\alpha_0^*(P)=1$.

Proof: The proof goes by verifying that the map $\mathcal{F}:\mathcal{P}_{B\mathcal{L}}\to\mathcal{P}_{B\mathcal{L}}$, $\mathcal{F}(P)=P-\alpha_0^*(P)\,\mathrm{grad}\,\Phi(P)$ satisfies the assumptions of the previous lemma. First note that $\alpha_0^*(P)$ is smooth for any $P\in\mathcal{P}_{B\mathcal{L}}$ with $P\neq P^*$. Thus, $\mathcal{F}$ is smooth and hence $D\mathcal{F}$ is Lipschitz continuous at any $P\neq P^*$. If we could prove that $\mathcal{F}$ is even C^2 at P^*, we could finish off with a simple Taylor series argument. Unfortunately, this is not true and therefore we require a more complicated argument.

A first step is to show that the derivative of $\mathrm{grad}\,\Phi(P)$ at P^* is the identity transformation. For arbitrary tangent vectors $\Omega,\Omega_1\in T_{P^*}$, we have

$$\begin{aligned}&\mathcal{H}_\Phi(P^*)(D\,\mathrm{grad}\,\Phi(P^*)\Omega,\Omega_1)\\&\quad=\mathrm{tr}\left(P^{*-1}\left(\Omega-\sum_i(D\beta_i(P^*)\Omega)P^*A_iP^*\right)P^{*-1}\Omega_1\right.\\&\quad=\mathrm{tr}\left(\left(\Omega-\sum_i(D\beta_i(P^*)\Omega)A_i\right)\Omega_1\right.\\&\quad=\mathrm{tr}(P^{*-1}\Omega P^{*-1}\Omega_1)\\&\quad=\mathcal{H}_\Phi(P^*)(\Omega,\Omega_1).\end{aligned}$$

Thus, $D\,\mathrm{grad}\,\Phi(P^*)\Omega=\Omega$ for all $\Omega\in T_{P^*}$ which proves the claim.

Note that the Newton decrement $\lambda_0(P)$ is the norm of a smooth function and therefore is Lipschitz continuous. Moreover, α_0^* is the composition of the smooth function $f(x)$ of (47) with the Newton decrement and therefore is Lipschitz continuous as well. Furthermore, the derivative of the Newton decrement, $\lambda_0(P)=\|\mathcal{G}(P)\|$ where $\mathcal{G}(P):=\mathcal{H}_\Phi(P)^{-1/2}\nabla\Phi(P)$ is bounded as

$$\|D\lambda_0(P)\|\leq\|D\mathcal{G}(P)\|$$

for all $P\neq P^*$. Since $D\mathcal{G}(P)$ is a smooth function of P, the derivative of λ_0 is locally bounded around P^*. Applying the chain rule to $\alpha_0^*(P)=f(\lambda_0(P))$, we conclude that the same assertion holds for the derivative of α_0^*. The derivative of $\mathcal{F}$ at any $P\neq P^*$ is

$$I-\alpha_0^*(P)D\,\mathrm{grad}\,\Phi(P)-D\alpha_0^*(P)\,\mathrm{grad}\,\Phi(P).$$

The second summand is the product of a Lipschitz continuous function with a smooth function. Therefore, the second summand is Lipschitz continuous. The third summand is a product of a locally bounded function and a smooth function vanishing at P^*. Therefore, the third summand is also Lipschitz continuous at P^*. This shows the local Lipschitz continuity of the derivative of $\mathcal{F}$. Moreover

$$\lim_{P \to P^*} D\mathcal{F}(P) = I - \alpha_0^*(P^*) D \operatorname{grad} \Phi(P^*) = 0$$

since $\alpha_0^*(P^*) = 1$ and $D \operatorname{grad} \Phi(P^*) = I$. In particular, $\mathcal{F}$ is a C^1-function with vanishing derivative at the optimum. The desired result follows from the previous lemma. ■

C. Iterative Step-Size Selection

An improved estimate $\alpha_\ell^*(P_k)$ for the desired line-search minimum α^* of $\Phi(P_k + \alpha\Delta)$ on $[0, \alpha_{\max})$ can be found by iterating the construction of the previous section.

Proceeding inductively from $\alpha_{-1}^* := 0, \alpha_0^*, \ldots, \alpha_\ell^*$, with the previous construction, replacing P, L, ϕ by $P_\ell := P + \alpha_\ell^* \Delta, L_\ell := P_\ell^{-1/2} \Delta P_\ell^{-1/2}, \phi_\ell(\alpha) := \phi(\alpha + \alpha_\ell^*)$, we obtain iterative step-size selections as

$$\alpha_{\ell+1}^* = \alpha_\ell^* - s_\ell \phi''(\alpha_\ell^*)^{-1} \phi'(\alpha_\ell^*), \quad \alpha_{-1}^* = 0 \tag{50}$$

$$s_\ell = \frac{1 + 2\lambda_\ell - \sqrt{1 + 4\lambda_\ell}}{2\lambda_\ell^2}, \quad \lambda_\ell = -\phi'(\alpha_\ell^*)/\sqrt{\phi''(\alpha_\ell^*)}. \tag{51}$$

Here $\phi'(\alpha_\ell^*)$ is found by working with second-order polynomials, as in (39). The second derivative $\phi''(\alpha_\ell^*)$ requires very little extra effort since

$$\phi''(\alpha) = \sum_{i=1}^{N} \left(\left(\frac{p_i'(\alpha)}{p_i(\alpha)} \right)^2 - \frac{p_i''(0)}{p_i(\alpha)} \right). \tag{52}$$

Notice that $p_i''(\alpha) = p_i''(0)$ is a constant independent of α. In general, $\phi'(\alpha) \neq \phi''(\alpha)$ except when $\alpha = 0$.

Theorem V.2: The sequence of step sizes α_ℓ^* defined by (52) and (51) is monotonically increasing and converges quadratically fast to the optimal step size α^*.

Proof: By construction, $\phi'(\alpha_\ell^*) < 0$ for all ℓ and also $\alpha_\ell^* \le \alpha^*$. Moreover, $0 \le s_\ell < 1$. Therefore $\alpha_\ell^* \le \alpha_{\ell+1}^*$. By monotonicity $\lim_{\ell \to \infty} \alpha_\ell^* = \alpha^{**}$ exists. Thus, α^{**} is a fixed point of the algorithm, and therefore $\phi'(\alpha^{**}) = 0$. Since α^* is the smallest positive root of ϕ', we conclude $\alpha^{**} = \alpha^*$. In particular, $\lim_{\ell \to \infty} \lambda_\ell = 0$ and thus $s_\ell \to 1$.

Since $\alpha \mapsto \alpha - s(\alpha)\phi'(\alpha)/\phi''(\alpha)$ is smooth on $[0, \infty)$, having derivative zero at α^*, we conclude local quadratic convergence, as claimed. ■

Inevitably, there are some ad hoc aspects to any line search, weighting the cost of additional iterations against improvement in accuracy. Typically, between one and four steps are used in a line search for grasping problems. There is up to an order of magnitude savings in the line search, because of the explicit formulas involved.

VI. CONCLUSION

A new construction of a quadratically convergent Newton algorithm for dextrous hand grasping has been proposed. The new algorithm is being currently imported into robotic hands, and this work will be reported subsequently by others. Matlab simulations have been done for "verification" of the upper bound on iteration number, this being about 10, which is about the same as for the best algorithm of the earlier paper [2] using the same cost function. The improved efficiency of our algorithm is confirmed by operation counts per iteration. For example, for four fingers, and focusing on the easiest to calculate improvements, we achieve improvement factors of more than 15 for the point contact case and more than 75 for the soft finger contact case, respectively.

REFERENCES

[1] M. Buss, H. Hashimoto, and J. B. Moore, "Dextrous hand grasping force optimization," *IEEE Trans. Robot. Automat.*, vol. 12, pp. 406–418, June 1996.

[2] M. Buss, L. Faybusovich, and J. B. Moore, "Dikin-type algorithms for dextrous grasping force optimization," *Int. J. Robot. Res.*, vol. 17, no. 8, pp. 831–839, Aug. 1998.

[3] D. den Hertog, *Interior Point Approach to Linear, Quadratic and Convex Programming*. Boston, MA: Kluwer, 1994.

[4] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*. New York: Springer, 1993.

[5] L. Han, J. C. Trinkle, and Z. X. Li, "Grasp analysis as linear matrix inequality problems," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Detroit, MI, 1999, pp. 1261–1268.

[6] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA: SIAM, 1994.

[7] F. T. Cheng and D. E. Orin, "Efficient algorithm for optimal force distribution—The compact-the dual LP method," *IEEE Trans. Robot. Automat.*, vol. 6, pp. 178–187, Apr. 1990.

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

[9] Y. Nakamura, K. Nagai, and T. Yoshikawa, "Dynamics and stability in coordination of multiple robotic mechanisms," *Int. J. Robot. Res.*, vol. 8, no. 2, pp. 45–61, 1989.

[10] Y. Nesterov and A. Nemirovsky, *Interior-Point Polynomial Methods in Convex Programming*: SIAM, 1994.

[11] L. Faybusovich, "On a matrix generalizations of affine-scaling vector fields," *SIAM J. Matrix Anal.*, vol. 16, no. 3, pp. 886–897, 1995.

[12] L. Vandenberg, S. Boyd, and S. P. Wu, "Determinant maximization with linear matrix inequality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 19, no. 2, pp. 499–533, Apr. 1998.

Uwe Helmke (SM'96) was born in Bremen, Germany, in 1952. He received thd Diploma and the Ph.D. degrees from the University of Bremen, Germany, in 1979 and 1983, respectively.

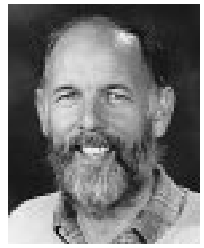
From 1984 until 1995, he held positions first as a Teaching and Research Fellow and then as a Privatdozent (German equivalent of Senior Lecturer) at the Department of Mathematics, University of Regensburg. Since 1995, he has been on the faculty of the University of Würzburg, where he is Full Professor in Mathematics and holder of the Chair in Real Analysis. He had held short visiting academic appointments at Harvard University, Ben Gurion University of the Negev (Israel), Australian National University, and the Chinese University of Hong Kong. His research interests are in dynamical systems and control theory, with special emphasis on algebraic and geometric methods. He is coauthor, with J. Moore, of the textbook *Optimization and Dynamical Systems* (Berlin, Germany: Springer-Verlag, 1994) and co-editor of the Proceedings Volumes *Systems and Networks: Mathematical Theory of Networks and Systems I, II* (Berlin, Germany: Akademie-Verlag, 1994), *Operators, Systems and Linear Algebra* (Berlin, Germany: Teubner-Verlag, 1997), and *Advances in Mathematical Systems Theory* (Cambridge, MA: Birkhäuser 2001)

Dr. Helmke is a member of the Steering Committee of MTNS, has been on the Program and Organizing Committees of various international conferences in Systems and Control Theory and is currently Associate Editor for *Mathematics of Control, Signals, and Systems, Systems and Control Letters*, and *SIAM Journal on Matrix Analysis and Applications.*

Knut Hüper was born in Karlsruhe, Germany, in 1962. He received the Diploma Physicist and the Doctor rer.nat. degrees from the Munich University of Technology, Munich, Germany, in 1990 and 1996, respectively.

From 1990 to 1995, he was a Teaching and Research Assistant at the Institute for Circuit Theory and Signal Processing, Munich University of Technology. In 1995 he was a Teaching and Research Assistant at the Institute for Algorithms and Cognitive Systems, University of Karlsruhe, Germany. In 1996 he spent two months as a Visiting Associate at the Research School of Information Sciences and Engineering in the Institute of Advanced Studies, Australian National University, Canberra. In 2000 he was a Post-Doctoral Researcher at the Department of Mathematics and the Institute for Systems and Robotics, University of Coimbra, Portugal. Since 1996, he has been a Teaching and Research Assistant at the Institute of Mathematics, Würzburg University, Germany. His research interests are in the area of Lie theory and geometric optimization on manifolds. His most recent interests are the application of dynamical systems methods to matrix algorithms, interpolation problems on manifolds, and the Mathematics behind Nuclear-Magnetic-Resonance experiments. He was a Co-guest Editor of a special issue of *Systems & Control Letters*.

Dr. Hüper is member of the AMS, SIAM, and the German GAMM.



John B. Moore (S'66–M'68–SM'77–F'79) was born in China in 1941. He received the bachelor and masters degrees in electrical engineering in 1963 and 1964, respectively, and the Ph.D. degree in electrical engineering from the University of Santa Clara, California, in 1967.

He was appointed Senior Lecturer at the Electrical Engineering Department, University of Newcastle, U.K., in 1967, and promoted to Associate Professor in 1968, and full Professor (personal chair) in 1973. He was Department Head for the period 1975–1979. In 1982, he was appointed as a Professorial Fellow in the Department of Systems Engineering, Research School of Physical Sciences, Australian National University, Canberra, and promoted to Professor in 1990. He has been head of the department since 1992–1999, 2002–present. The department is now located in the Research School of Information Sciences and Engineering.

Dr. Moore is a Fellow of the Australian Academy of Technological Sciences and Engineering and the Australian Academy of Science.