A POLYNOMIAL-TIME INTERIOR-POINT METHOD FOR
CIRCULAR CONE PROGRAMMING BASED ON
KERNEL FUNCTIONS

YANQIN BAI∗, PENGFEI MA AND JING ZHANG

Department of Mathematics, Shanghai University, Shanghai, 200444, China

(Communicated by Duan Li)

Abstract. We present an interior-point method based on kernel functions for
circular cone optimization problems, which has been found useful for de-
scribing optimal design problems of optimal grasping manipulation for multi-
fingered robots. Since the well-known second order cone is a particular circular
cone, we first establish an invertible linear mapping between a circular cone and
its corresponding second order cone. Then we develop a kernel function based
interior-point method to solve circular cone optimization in terms of the
corresponding second order cone optimization problem. We derive the
complexity bound of the interior-point method and conclude that circular cone
optimization is polynomial-time solvable. Finally we illustrate the performance
of interior-point method by a real-world quadruped robot example of optimal
contact forces taken from the literature [10].

1. Introduction. Linear conic programming extends linear programming models
from linearity to non-linearity and from convexity to non-convexity. Nemirovskii
[21] points out that conic programming allows us to reveal rich structure which
is usually possessed by a convex program and to exploit this structure in order to
process the program efficiently. Up to year of 2000, linear conic optimization mainly
focuses on optimization problems over symmetric cones. Güler [16] summarizes the
classification of symmetric cones encompassing only five cones out of which three are
of interests to optimization research. Renegar [25] points out that these cones are the
nonnegative orthant of the Euclidean space \( \mathbb{R}^n \) which leads to linear programming,
the Lorentz cone in \( \mathbb{R}^n \), which leads to second order cone programming and the cone
of positive semi-definite matrices that leads to semidefinite programming. These
three types of conic programming are first unified into a framework of symmetric
conic programming by Nesterov and Todd and studied extensively. One of the
powerful tools for solving symmetric conic programming problems is well-known
interior-point method (IPM). The reader is referred to [9, 21, 25] for more details
of IPM. Recently an increasing amount attention has been paid to a class of kernel
function based interior-point algorithm, which is useful for solving symmetric conic

∗ Corresponding author.

2010 Mathematics Subject Classification. Primary: 90C25, 90C51; Secondary: 90C60.
Key words and phrases. Linear conic programming problem, interior-point methods,
polynomial-time algorithm.

This research was supported by a grant from the National Natural Science Foundation of China
(No.11371242). This paper is dedicated to the memory of Professor Xiaoling Sun (1963-2014).

∗ Corresponding author.
programming problems [2, 4]. Kernel function is a class of barrier functions which are used to solve constrained problems [27, 28].

Recently, a lot of attentions have been paid to nonsymmetric cone programming problems. Likewise, nonsymmetric cone optimization has a linear objective function and the convex cone constraints. However, the cone constraint is non-self-dual or non-homogeneous cone and is distinctively different with the symmetric cone. Bomze et al [6] point out that nonsymmetric cone optimization is a powerful tool for modeling nonconvex quadratic problems and combinatorial NP-hard problems. Moreover, Burer [11] shows any problem with a quadratic objective, linear constraints and possibly binary variables can equivalently be formulated as a copositive optimization problem. This representation results in a real boost to the field and has opened a whole new approach to many quadratic and combinatorial problems, yielding both a better theoretical understanding and often stronger bounds on the problems. Furthermore, nonsymmetric conic optimization has wide range of applications for the real-world problems. For example, it has been found that circular cone constraints are involved in real engineering problems [8, 10, 17, 18, 19]. Moreover, Li et al [19] considered three- or four-leg support phases shown in following Figure 1, which illustrates that a quadruped robot is in a 3-D workspace

![Figure 1. Quadruped robot](image)

with \(i\) point contacts between the ground and the legs, fixed with a right-handed coordinate frame. Each leg contacts the ground with Coulomb friction, \(n_i, o_i\) and \(t_i\) are the unit inward normal and two unit tangent vectors at the contact \(i\), where \(n_i = o_i \times t_i\). The contact force \(f_i\) can be expressed in the local coordinate frame \(n_i, o_i, t_i\) by \(f_i = (f_{n_i}, f_{o_i}, f_{t_i})^T\), where \(f_{n_i}\), \(f_{o_i}\) and \(f_{t_i}\) are the components of \(f_i\) along \(n_i, o_i\) and \(t_i\), respectively. To ensure no slipping at a contact point, the components of the contact force \(f_i\) must satisfy the contact constraint

\[
\|(f_{o_i}, f_{t_i})\| \leq \mu f_{n_i},
\]

where \(\mu\) is the static friction coefficient of the substrate. The second example is an optimal grasping manipulation for multi-fingered robots [8, 17, 18]. The grasping force of the \(i\)-th finger is subject to the inequality (1), geometrically, which represents a circular cone, in which the set of all forces is constrained and belong to a circular cone centred about the surface normal (see Figure 2) with half-angle \(\beta = \tan^{-1} \mu\).
Although nonsymmetric cone programming problems are useful to discrete non-convex quadratic problems and combinatorial NP-hard problems, solving nonsymmetric cone programming efficiently remains challenge. For nonsymmetric cone optimization, there are two difficulties that are encountered by the properties of non-self-dual in the constraint cone. The first one is that verification of an element belongs a copositive cone is an NP-complete problem. The second one is developing interior-point algorithm with polynomial-time computational complexity. Unlike symmetric conic optimization, there are less IPMs for solving nonsymmetric conic optimization. Recently, Nesterov [22] proposed a new theoretical IPM that is based on an extension of the ideas of self-scaled optimization to the non-symmetric conic optimization. He developed a $4n$-self-concordant barrier for an $n$-dimensional $p$-cone, which is an example of nonsymmetric cone. Matsukawa and Yoshise [20] proposed a primal barrier function Phase I algorithm for solving conic optimization problems over the closed convex cone $K$ called doubly nonnegative cone. Skajaa and Ye [26] designed a homogeneous IPM for nonsymmetric convex conic optimization. All of these IPMs are developed based on self-concordant barrier functions for its corresponding cone.

We are motivated by both of the application of circular cone constraints for quadruped robot and multi-fingered robots and the challenge for designing IPMs for a particular nonsymmetric conic optimization problem. In this paper, we present an interior-point method based on kernel functions for circular cone optimization problems, which has been found useful for describing optimal design problems of optimal grasping manipulation for multi-fingered robots. Since the well-known second order cone is a particular circular cone, we first establish an invertible linear mapping between a circular cone and its corresponding second order cone. Then we develop a kernel function based interior-point method to solve circular cone optimization in terms of the corresponding second order cone optimization problem. We derive the complexity bound of the interior-point method and conclude that
circular cone optimization is polynomial-time solvable. Finally we illustrate the performance of interior-point method by a real-world quadruped robot example of optimal contact forces taken from the literature [10].

The paper is organized as follows. In Section 2, we establish the geometric and algebraic relationship between the circular cone and the second order cone. Furthermore, based on the spectral factorization of second order cone, we extend it to the circular cone. In Section 3, we consider circular cone programming problems and establish its dual theorem. Then we introduce the optimality condition and central path for circular cone programming problems in Section 4. In Section 5, based on the invertible linear map established between the circular cone and the second order cone, we present an interior-point algorithm for circular cone programming in terms of that of second order cone programming. Section 6 implements our interior-point method by a real-world quadruped robot example of optimal contact forces taken from the literature [10]. Finally, we conclude and give future research in Section 7.

Some notations used throughout the paper are as follows. \( \mathbb{R}^n \) and \( \mathbb{R}^n_+ \) denote the space of \( n \)-dimensional real column vectors and the space of \( n \)-dimensional nonnegative real column vectors, respectively. Thus, \( x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n \) is viewed as a column vector in \( \mathbb{R}^n \). For any \( x, y \in \mathbb{R}^n \), the Euclidean inner product and norm are denoted \( \langle x, y \rangle = x^Ty \) and \( \| x \| = \sqrt{x^Tx} \). Given a set \( S \), we denote \( S^\circ \) by the interior of \( S \). Finally, \( I_n \) denotes the \( n \times n \) identity matrix.

2. Second order cone and circular cone. The second order cone in \( \mathbb{R}^n \), also called the Lorentz cone, is defined to be

\[
L^n := \{(x_1, x_{2:n})^T \in \mathbb{R} \times \mathbb{R}^{n-1} | \| x_{2:n} \|_2 \leq x_1 \},
\]

where \( x_{2:n} = (x_2, \ldots, x_n)^T \) and \( \| \cdot \|_2 \) denotes the Euclidean norm. The second order cone programming (SOCP) has received considerable attentions from researchers because of its wide range of applications, the particular structure and algebraic properties. IPMs exploit the special structure of SOCP problems for polynomial-time efficiency [1].

Given a rotation angle \( \theta \in (0, \frac{\pi}{2}) \), we can define an \( n \)-dimensional circular cone that associates with a second order cone in \( \mathbb{R}^n \).

\[
C^n_{\theta} := \{(x_1, x_{2:n})^T \in \mathbb{R} \times \mathbb{R}^{n-1} | \cos \theta \| x \|_2 \leq x_1 \}.
\]

Obviously, if the rotation angle \( \theta = \frac{\pi}{4} \), then the circular cone is the second order cone. In other words, the second order cone is a special circular cone. Throughout the paper, we call \( L^n \) an association cone with \( C^n_{\theta} \). Like the second order cone, a circular cone is a pointed closed convex cone having hyper-spherical sections orthogonal to its axis of revolution around which the cone is invariant to rotation.

Recall the definition of dual cone of a given cone \( K \subseteq \mathbb{R}^n \),

\[
K^* := \{ y | \langle y, x \rangle \geq 0, \forall x \in K \},
\]

where \( \langle \cdot, \cdot \rangle : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \) denotes an Euclidean inner product on \( \mathbb{R}^n \). Then, we can verify that

\[
(C^n_{\theta})^* = C^n_{\frac{\pi}{2} - \theta} = \{(x_1, x_{2:n})^T \in \mathbb{R} \times \mathbb{R}^{n-1} | \sin \theta \| x \|_2 \leq x_1 \}.
\]

We state the definition of symmetric cone as below.

**Definition 2.1.** Let \( K \subseteq \mathbb{R}^n \) be a proper cone, that is, a pointed closed convex cone with nonempty interior, \( K \) is called a symmetric cone,

- if \( K \) is self-dual, that is, \( K = K^* \)
• $\mathcal{K}$ is homogeneous, i.e., for $\forall x, y \in \mathcal{K}^\circ$, there exists a invertible mapping $G$ such that $G(\mathcal{K}) = \mathcal{K}$ and $y = G(x)$.

It is straightforward to verify that the second order cone is a symmetric cone. However, a circular cone is not self-dual, i.e., $\mathcal{K} \neq \mathcal{K}^*$ unless $\theta = \pi/4$.

The geometric interpretation of a circular cone, its dual cone and associated second order cone are shown in Figure 3.

![Figure 3. The graph of circular cone and second order in $\mathbb{R}^3$](image)

2.1. Algebraic representation between second order cone and circular cone. In this subsection, we study the algebraic relationship between a circular cone and its associated second order cone. We first recall the algebraic representation developed by Chen et al [12] between the second order cone $\mathcal{L}^n$ and the circular cone $\mathcal{C}^n_\theta$. Then we recall the classical spectral factorization for the second order cone $\mathcal{L}^n$ summarized by Alizadeh in [1]. Finally, we conclude with a spectral factorization for circular cone $\mathcal{C}^n_\theta$. For the detail of algebra of circular cone, please refer to [12, 13, 29].

Algebraic relationship between the second order cone and the circular cone is established in terms of an invertible linear mapping which plays a crucial rule in designing interior-point algorithm for circular cone programming.

Given a rotation angle $\theta$, let

$$A_\theta := \begin{bmatrix} \tan \theta & 0 \\ 0 & I_{n-1} \end{bmatrix},$$

where $I_{n-1}$ is the $n-1$ dimensional unit matrix. It is straightforward to verify that

$$A_\theta^{-1} = \begin{bmatrix} \cotan \theta & 0 \\ 0 & I_{n-1} \end{bmatrix}. \quad (6)$$

The following theorem states the basic algebraic relationship between the second order cone and the circular cone.
Theorem 2.2 ([20]). Let $\mathcal{L}^n$ and $\mathcal{C}_\theta^n$ be defined as in (2) and (3), respectively. Then, we have

1. $\mathcal{C}_\theta^n = A_\theta^{-1} \mathcal{L}^n$ and $\mathcal{L}^n = A_\theta \mathcal{C}_\theta^n$.
2. $A_\theta \mathcal{L}^n = \mathcal{C}_\frac{\pi}{2} - \theta^n$ and $\mathcal{C}_\frac{\pi}{2} - \theta^n = A_\theta^2 \mathcal{C}_\theta^n$.
3. $(\mathcal{C}_\theta^n)^* = \mathcal{C}_\frac{\pi}{2} - \theta^n$ and $((\mathcal{C}_\theta^n)^*)^* = \mathcal{C}_\theta^n$.

Obviously, we have the following corollary.

Corollary 1. If $x \in (\mathcal{C}_\theta^n)^\circ$ and $s \in ((\mathcal{C}_\theta^n)^*)^\circ$, then one has $A_\theta x \in (\mathcal{L}^n)^\circ$ and $A_\theta^{-1} s \in (\mathcal{L}^n)^\circ$. 

---

Let $\mathcal{L}^n$ and $\mathcal{C}_\theta^n$ be defined as in (2) and (3), respectively. Then, we have

1. $\mathcal{C}_\theta^n = A_\theta^{-1} \mathcal{L}^n$ and $\mathcal{L}^n = A_\theta \mathcal{C}_\theta^n$.
2. $A_\theta \mathcal{L}^n = \mathcal{C}_\frac{\pi}{2} - \theta^n$ and $\mathcal{C}_\frac{\pi}{2} - \theta^n = A_\theta^2 \mathcal{C}_\theta^n$.
3. $(\mathcal{C}_\theta^n)^* = \mathcal{C}_\frac{\pi}{2} - \theta^n$ and $((\mathcal{C}_\theta^n)^*)^* = \mathcal{C}_\theta^n$.

Obviously, we have the following corollary.

Corollary 1. If $x \in (\mathcal{C}_\theta^n)^\circ$ and $s \in ((\mathcal{C}_\theta^n)^*)^\circ$, then one has $A_\theta x \in (\mathcal{L}^n)^\circ$ and $A_\theta^{-1} s \in (\mathcal{L}^n)^\circ$.
2.2. Spectral factorization associated with second order cone and circular cone. In this subsection we recall the spectral factorization of a vector associated with the second order cone $\mathcal{L}^n$. Followed by the spectral factorization associated with the circular cone $\mathcal{C}^n_\theta$ [29], we specialize the spectral factorizations for circular cone $\mathcal{C}^n_\theta$ and its dual cone $(\mathcal{C}^n_\theta)^*$. Using these spectral factorizations, we construct the corresponding barrier functions for $\mathcal{C}^n_\theta$ and $(\mathcal{C}^n_\theta)^*$, which are used to design the search direction for a primal dual interior-point algorithm in Section 3.

Alizadeh et al [1] pointed out that the spectral factorization is the special case of the so-called Euclidean Jordan algebra.

Given $x = (x_1, x_{2:n})^T \in \mathcal{R} \times \mathcal{R}^{n-1}$. Then $x$ can be decomposed as
\[ x = \lambda_1 u^{(1)} + \lambda_2 u^{(2)}, \] (9)
where \( \lambda_1, \lambda_2 \) and \( u^{(1)}, u^{(2)} \) are the spectral values and associated spectral vectors of $x$ given by
\[ \lambda_i = x_1 + (-1)^i \|x_{2:n}\|, \] (10)
\[ u^{(i)} = \begin{cases} \frac{1}{2}(1, (-1)^i \frac{x_{2:n}}{\|x_{2:n}\|}) & \text{if } x_{2:n} \neq 0, \\ \frac{1}{2}(1, (-1)^i w) & \text{if } x_{2:n} = 0, \end{cases} \] (11)
for $i = 1, 2$, with $w$ being any vector in $\mathcal{R}^{n-1}$ satisfying $\|w\| = 1$. If $x_{2:n} \neq 0$, the decomposition (9) is unique.

Obviously, $\lambda_i \geq 0, i = 1, 2$ if $x \in \mathcal{L}^n$.

The following theorem is the spectral factorization for a vector $x$ associated with circular cone $\mathcal{C}^n_\theta$.

**Theorem 2.4** ([29]). Given a circular $\mathcal{C}^n_\theta$ with $\theta \in (0, \frac{\pi}{2})$, for any $x \in \mathcal{R}^n$, one has
\[ x = \lambda_1 z^{(1)} + \lambda_2 z^{(2)} \] (12)
where
\[ \lambda_1 = x_1 - \|x_{2:n}\| \cot \theta, \quad \lambda_2 = x_1 + \|x_{2:n}\| \tan \theta \]
and
\[ z^{(1)} = \begin{bmatrix} \sin^2 \theta \\ -\sin \theta \cos \theta \end{bmatrix}, \quad z^{(2)} = \begin{bmatrix} \cos^2 \theta \\ \sin \theta \cos \theta \end{bmatrix}, \]
\[ \text{if } x_{2:n} \neq 0, \text{ then } w = \frac{x_{2:n}}{\|x_{2:n}\|} \text{ and otherwise, then } w \text{ can be any vector in } \mathcal{R}^{n-1} \text{ satisfying } \|w\| = 1. \]

Note that the above spectral factorization is hold for all $x \in \mathcal{R}^n$. Throughout the paper, we focus on the $\mathcal{C}^n_\theta$ and $(\mathcal{C}^n_\theta)^*$. It is easy to have the next corollary.

**Corollary 2.**
1. For any $x = (x_1, x_{2:n})^T \in \mathcal{C}^n_\theta$, we have
\[ x_1 - \|x_{2:n}\| \cot \theta \geq 0 \text{ and } x_1 + \|x_{2:n}\| \tan \theta \geq 0. \]
2. For any $x = (x_1, x_{2:n})^T \in (\mathcal{C}^n_\theta)^*$, we have
\[ x_1 - \|x_{2:n}\| \tan \theta \geq 0 \text{ and } x_1 + \|x_{2:n}\| \cot \theta \geq 0. \]
3. Circular cone programming and duality theorem. In this paper, we consider the following circular cone programming as follows:

\[(\text{CCP}) \min_x \{ c^T x : Ax = b, \ x \in K \}. \quad (13)\]

Its dual problem becomes

\[(\text{CCD}) \max_{y,s} \{ b^T y : A^T y + s = c, \ s \in K^* \}, \quad (14)\]

where \(K \subseteq \mathbb{R}^n\) and \(K^* \subseteq \mathbb{R}^n\) are the Cartesian product of circular cones, i.e.,

\[K = K^1 \times K^2 \times \cdots \times K^N\]

\[K^* = (K^1)^* \times (K^2)^* \times \cdots \times (K^N)^*, \quad (15)\]

with \(K^j = C^j_\theta\) for each \(j\), and \(n = \sum_{j=1}^N n_j\). Furthermore, we partition the vectors \(x, s, c\), and the matrix \(A\) accordingly as

\[x = (x^1; x^2; \ldots; x^N), \ s = (s^1; s^2; \ldots; s^N)\]

with \(x^j \in K^j, \ s^j \in (K^j)^*\), \(c = (c^1; c^2; \ldots; c^N)\) with \(c^j \in \mathbb{R}^{n_j}\), and \(A = (A^1, \ldots, A^N)\) is a circular cone and \((K^j)^*\) is its dual cone.

Assume that \(m \leq n\) and \(\text{rank}(A) = m\). The simplest case for (CCP) is that \(N = 1\), that is the case we consider in Section 5 where we take that \(K = C^n_\theta\). As a matter of fact, it is easy to verify that (14) is the Lagrangian dual

\[\max_y \left\{ \min_{x \in K} \{ c^T x - (Ax - b)^T y \} \right\}\]

of (13), using the fact that \(\min_{x \in K} \{ u^T x : x \in K \} = 0 \) if \(u \in K^*\) and \(u \to -\infty\), otherwise. We can also check the following weak duality theorem.

**Theorem 3.1 (Weak Duality Theorem).** If \(x\) is feasible in (13) and \((y,s)\) in (14), then

\[c^T x \geq b^T y,\]

with equality holds if and only if \(s^T x = 0\).

**Proof.** Notice that

\[c^T x - b^T y = (A^T y + s)^T x - (Ax)^T y = s^T x \geq 0,\]

with inequality holds. \(\square\)

**Remark 1.** In (13) and (14), if one replaces \(C^n_\theta\) with the nonnegative orthant, (13) and (14) becomes the standard form of linear programming problems. In that case, we have strong duality theorem. However, for circular cone programming, such property no longer holds.

In order to develop an interior-point algorithm, we need the following theorem [5].

**Theorem 3.2 (Dual Theorem).** Consider a circular cone programming (13) problem and its dual problem (14). The following results hold.

(a) The dual problem is conic, and the dual problem of dual is the primal.

(b) The value of the dual objective value at each dual feasible solution \((y,s)\) is less than the primal objective value at each primal feasible solution \(x\), and the duality gap

\[c^T x - b^T y\]

is nonnegative at each “primal-dual feasible pair” \((x,y,s)\).

(c.1) If the primal (13) is bounded below and strictly feasible (i.e., there exists \(x \in K^c\) such that \(Ax = b\)), then the dual (14) achieves its optimal solution and there is no duality gap.
If the dual (14) is bounded above and strictly feasible (i.e., \( A^T y + s = c \) and \( s \in (K^\ast)^\circ \) for some \((y, s)\) ), then the primal (14) achieves optimal solution and there is no duality gap.

Assume that at least one of the problems (13), (14) is bounded and strictly feasible. Then a primal-dual feasible pair \((x, y, s)\) is a pair of optimal solutions to the respective problems if and only if

\[
c^T x = b^T y \quad \text{(zero duality gap)} \tag{d.1}
\]

or

\[
x^T s = 0 \quad \text{(complementary slackness).} \tag{d.2}
\]

A useful consequence is the following Corollary 3 (Strong Duality Theorem). Assume that both (13) and (14) are strictly feasible. Then both problems achieve optimality with no duality gap and (d.1), (d.2) are necessary and sufficient for the optimality of a primal-dual feasible pair.

4. Optimality conditions and central path. Based on the strong duality theorem of circular cone programming problems, given a rotation angle \( \theta \), we consider the simplest case of circular cone programming problems and discuss the optimality conditions as follows.

\[
\text{(CCP)} \quad \min_x \{c^T x : Ax = b, \ x \in C_\theta^n\}. \tag{16}
\]

Its dual problem is

\[
\text{(CCD)} \quad \max_{y, s} \{b^T y : A^T y + s = c, \ s \in (C_\theta^n)^\ast\}, \tag{17}
\]

Solving the primal dual pair of circular cone programming problems (16) and (17) is equivalent to solve the following optimality conditions

\[
Ax = b, \ x \in C_\theta^n,
A^T y + s = c, \ s \in (C_\theta^n)^\ast,
x^T s = 0. \tag{18}
\]

We assume that both (CCP) and (CCD) satisfy the interior-point condition, i.e., there exists \((x^0, y^0, s^0)\) such that \(Ax^0 = b, \ x^0 \in (C_\theta^n)^\circ, \ A^T y^0 + s^0 = c, \ s^0 \in ((C_\theta^n)^\ast)^\circ\).

Now we use both the invertible linear mapping and its inverse mapping

\[
A_\theta = \begin{bmatrix} \tan \theta & 0 \\ 0 & I_{n-1} \end{bmatrix}, \quad A_\theta^{-1} = \begin{bmatrix} \cot \theta & 0 \\ 0 & I_{n-1} \end{bmatrix} \tag{19}
\]

to transform the feasible sets of (CCP) and (CCD), respectively. \(x \in \{x \in R^n \mid Ax = b, \ x \in C_\theta^n\}\) and \(s \in \{x \in R^n \mid A^T y + s = c, \ s \in (C_\theta^n)^\ast\}\), respectively, let \(\tilde{x} = A_\theta x\) and \(\tilde{s} = A_\theta^{-1} s\). Moreover, let \(\tilde{A} = AA_\theta^{-1}\) and \(\tilde{c} = A_\theta^{-1} c\). We have the following associated primal dual pair of second order cone programming problems as follows.

\[
\text{(SOCP)} \quad \min_{\tilde{x}} \{\tilde{c}^T \tilde{x} : \tilde{A} \tilde{x} = b, \tilde{x} \in L^n\}, \tag{20}
\]

and

\[
\text{(SOCD)} \quad \max_{\tilde{y}, \tilde{s}} \{b^T \tilde{y} : A^T \tilde{y} + \tilde{s} = \tilde{c}, \tilde{s} \in L^n\}. \tag{21}
\]
The optimality conditions for (20) and (21) are as follows.
\[ \tilde{A} \tilde{x} = b, \quad \tilde{x} \in \mathcal{L}^n, \]
\[ \tilde{A}^T \tilde{y} + \tilde{s} = \tilde{c}, \quad \tilde{s} \in \mathcal{L}^n, \tag{22} \]
\[ \tilde{x}^T \tilde{s} = 0. \]

By the interior-point condition holds for (CCP) and (CCD) and Corollary 1, (20) and (21) satisfy the interior-point condition too, i.e., there exists \((x_0, y_0, s_0)\) such that
\[ A \tilde{x}_0 = b, \quad \tilde{x}_0 \in (\mathcal{L}^n)^\circ, \]
\[ \tilde{A}^T \tilde{y}_0 + \tilde{s}_0 = \tilde{c}, \quad \tilde{s}_0 \in (\mathcal{L}^n)^\circ. \]

Under the interior-point condition assumption, finding the optimal solutions of (20) and (21) is therefore equivalent to solve the system (22). In fact, by [15, Corollary 2.1], the system (22) is equivalent to
\[ \tilde{A} \tilde{x} = b, \quad \tilde{x} \in \mathcal{L}^n, \]
\[ \tilde{A}^T \tilde{y} + \tilde{s} = \tilde{c}, \quad \tilde{s} \in \mathcal{L}^n, \]
\[ \tilde{x}^T \tilde{s} = 0, \tag{23} \]
where \(\tilde{x} \circ \tilde{s} := (\tilde{x}^T \tilde{s}, \tilde{x}^1 \tilde{s}_2 + \tilde{s}_1 \tilde{x}_2, \ldots, \tilde{x}^1 \tilde{s}_n + \tilde{s}_1 \tilde{x}_n) \in \mathcal{R}^n.\) We replace the third equation in (23) by a parameteric equation \(\tilde{x} \circ \tilde{s} = \mu e,\) with \(\mu > 0.\) Thus we consider the system
\[ \tilde{A} \tilde{x} = b, \quad \tilde{x} \in \mathcal{L}^n, \]
\[ \tilde{A}^T \tilde{y} + \tilde{s} = \tilde{c}, \quad \tilde{s} \in \mathcal{L}^n, \]
\[ \tilde{x} \circ \tilde{s} = \mu e \tag{24} \]
where \(e = (1, 0, \ldots, 0)^T.\) For each \(\mu > 0,\) (24) has a unique solution \((\tilde{x}(\mu), \tilde{y}(\mu), \tilde{s}(\mu)).\)

We call \(\tilde{x}(\mu)\) the \(\mu\)-center of (20) and \((\tilde{y}(\mu), \tilde{s}(\mu))\) the \(\mu\)-center of (21). Note that at the \(\mu\)-center we have
\[ \tilde{x}(\mu)^T \tilde{s}(\mu) = \frac{1}{2} \text{tr}(\tilde{x} \circ \tilde{s}) = \frac{1}{2} \text{tr}(\mu e) = \mu, \]
where \(\text{tr}(x) = 2x_1.\) If \(\mu \to 0\) then the limit of the central path yields an optimal solution \((\tilde{x}^*, \tilde{y}^*, \tilde{s}^*)\) for (20) and (21). From Theorem 2.3, an optimal solution \((x^*, y^*, s^*)\) to (13) and (14) is \((A_0^{1/2} \tilde{x}, \tilde{y}, A_0 \tilde{s}).\) It is easy to present an an interior-point algorithm for nonsymmetric circular conic programming algorithm in terms of solving the optimality condition (23).

5. **Interior-point algorithm for circular cone programming.** In this section, we present an interior-point algorithm for solving (13) and (14) and derive the complexity bound.

We first recall an existing interior-point algorithm for SOCP presented by Bai et al in [3] and [4]. Then we develop a corresponding interior-point algorithm for nonsymmetric circular cone programming.

5.1. **Kernel function based interior-point algorithm for SOCP.** Before presenting our algorithm, we recall a univariate \(\psi : (0, \infty) \to [0, \infty)\) a kernel function, which satisfies the following three conditions.
\[ \psi'(1) = \psi(1) = 0 \]
\[ \psi''(t) > 0, \]
\[ \lim_{t \to 0} \psi(t) = \lim_{t \to \infty} \psi(t) = \infty. \tag{25} \]
We associate a given kernel function $\psi(t)$ to a real-valued barrier function $\Psi(x)$ on $\mathcal{L}^n$, based on the spectral factorization associated with the second order cone $\mathcal{L}^n$ as follows.

$$\Psi(x) := \text{tr}(\psi(x)) = 2(\psi(x))_1 = \psi(\lambda_{\text{max}}(x)) + \psi(\lambda_{\text{min}}(x)), \quad \forall x \in \mathcal{L}_+^n, \quad (26)$$

where $(\psi(x))_1$ denotes the first coordinate of the vector $\psi(x)$.

Moreover, we define a barrier function depending on the primal variable $x$ and the dual variable $s$ and the barrier parameter $\mu$,

$$\Phi(x, s, \mu) := \Psi(v).$$

Based on $\Phi(x, s, \mu)$, an interior-point algorithm for SOCP is summarized as follows.

---

**Primal-dual interior-point algorithm for second order cone programming**

**Input:**
- A threshold parameter $\tau > 1$;
- an accuracy parameter $\varepsilon > 0$;
- a fixed barrier update parameter $\eta \in (0, 1)$.

**begin**
- $x := e$; $s := e$; $\mu := 1$;
- **while** $2N\mu \geq \varepsilon$ **do**
  - $\mu := (1 - \eta)\mu$;
  - **while** $\Phi(x, s, \mu) > \tau$ **do**
    - **begin**
      - $x := x + \alpha \Delta x$;
      - $s := s + \alpha \Delta s$;
      - $y := y + \alpha \Delta y$;
    - **end**
  - **end**
**end**

---

**Figure 4.** Primal-dual interior-point algorithm for SOCP

5.2. **Kernel function based interior-point algorithm for CCP.** Now let us present a primal-dual interior-point algorithm for nonsymmetric circular cone programming problems. Given a rotation angle $\theta$, we use the same initial parameters such as $\tau$, $\rho$, etc, and we use the same barrier function $\Phi(x, s, \mu)$. The framework of primal-dual interior-point algorithm are as follows.

5.3. **Analysis of the interior-point algorithm for circular cone programming.** Because the core of primal-dual interior-point algorithm for CCP-CCD (5.2) is the same as that of primal-dual interior-point algorithm for SOCP (5.1), the analysis of the computational complexity of (5.2) is similar to that of (5.1). Here we briefly recall the analysis of primal-dual interior-point algorithm for SOCP.
Primal-dual interior-point algorithm for nonsymmetric circular cone programming

Input:
A rotation angle $\theta$, $0 < \theta < \frac{\pi}{2}$;  
a threshold parameter $\tau > 1$;  
an accuracy parameter $\varepsilon > 0$;  
a fixed barrier update parameter $\eta \in (0, 1)$;  
a matrix $A_\theta$ and its inverse $A_\theta^{-1}$.

\begin{align*}
&\text{begin}\  
x := \frac{e}{\tan \theta}; \ s := \frac{e}{\cotan \theta}; \ \mu := 1; \\
&\text{Compute } \tilde{x} := A_\theta x; \ \tilde{s} := A_\theta^{-1} s; \ \tilde{A} = AA_\theta^{-1}; \ \tilde{c} = A_\theta^{-1} c; \\
&\text{begin} \\
&\text{Solve an associated SOCP by primal-dual algorithm in Fig. 4} \\
&\text{When } N\mu \leq \varepsilon, \text{ obtain } \varepsilon\text{-approximate solution } (\tilde{x}, \tilde{y}, \tilde{s}) \\
&\text{Compute an approximate solution solution } x = A_\theta^{-1} \tilde{x}, \ y = \tilde{y}, s = A_\theta \tilde{s}. \\
&\text{end}
\end{align*}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{algorithm.png}
\caption{Primal-dual interior-point algorithm for CCP}
\end{figure}

Lemma 5.1 (Lemma 5.1 in [2]). Choose a suitable kernel function $\psi(t)$. If $K$ denotes the number of inner iterations between two successive $\mu$-updates in the inner iteration of the algorithm for SOCP, then

$$K \leq \frac{L^\gamma}{\kappa \gamma},$$

where both $\kappa > 0$ and $\gamma$ are constants and $L = 2N\psi \left( \frac{\varphi(t)}{\sqrt{1-\eta}} \right)$, $\varphi$ is the inverse function of kernel function $\psi(t)$($t \geq 1$).

Furthermore, the iteration bound for the interior-point algorithm for SOCP states as follows:

Theorem 5.2 ([3]). Choose a suitable kernel function $\psi(t)$. The iteration bound of the algorithm 5.2 is given by

$$\frac{1}{\eta \kappa \gamma} \left[ 2N\psi \left( \frac{\varphi(t)}{\sqrt{1-\eta}} \right) \right]^{\gamma} \log \frac{2N}{\epsilon}. \quad (27)$$

Note that this iteration bound is completely determined by the (eligible) kernel function $\psi(t)$, the parameters $\eta$ and $\tau$ in the algorithm. Also note that when $2N$ is replaced by $n$, Therefore, we conclude that the complexity bound for nonsymmetric circular cone programming is as below.

Theorem 5.3. Algorithm 5.2 has the iteration bounds $O(\sqrt{n} \log n) \log \frac{n}{\tau}$ for large-update methods and $O(\sqrt{n} \log \frac{n}{\tau})$ for small-update methods for solving nonsymmetric circular cone programming (13) and (14), respectively.
In [3], it says that a suitable kernel function is chosen and (20) and (21) are solved by algorithm 5.1, the algorithm has an iteration bound \( O(\sqrt{n} \log n) \log \frac{1}{\epsilon} \) for large-update method and \( O(\sqrt{n}) \log \frac{1}{\epsilon} \) for small-update method, respectively. Therefore, for nonsymmetric circular cone programming (13) and (14), it also has same iteration bounds as SOCP (20) and (21).

**Remark 2.** If we transform circular cone programming (13) and (14) into SOCP (20) and (21), the interior-point algorithms for solving SOCP can be used to solve (20) and (21).

6. **Numerical tests.** In this section, we use a practical numerical example to illustrate the performance of our interior-point method for solving CCP problems described in Section 5. The numerical testing is carried out on a Lenovo PC (2.6GHz, 4.00GB of RAM) with the use of Matlab R2012a. The parameters are taken as \( \epsilon = 10^{-5}, \tau = 1, \eta = 0.7 \). The kernel functions \( \psi(t) \) are taken as \( \psi_1(t) = \frac{t^2-1}{2} - \log t, t > 0 \) and \( \psi_2(t) = \frac{t^2-1}{2} + \frac{t^2-1}{2}, t > 0 \), respectively. Our example is for quadruped robot appeared in [10].

**Example 1.** We consider a force optimization problem for quadruped robot in Figure 1. Assume that each leg contacts one point with the ground. The quadruped robot has four contact points, which have positions \( p_i \in \mathbb{R}^3 \) (in the global coordinate system), for \( i = 1, 2, 3, 4 \). The force applied at a contact point \( p_i \) is denoted by \( f_i \in \mathbb{R}^3 \), and is given in a local coordinate system. We denote its components as \( f_i = (f_{n_i}, f_{o_i}, f_{t_i})^T \). Thus \( f_{n_i} \) is the normal (inward) force applied at contact point \( p_i \), and \( (f_{o_i}, f_{t_i}) \) is the tangential force applied at contact point \( p_i \). The contact force \( f_i \) satisfy the following circular cone constraints

\[
\| (f_{n_i}, f_{t_i}) \| \leq \mu_i f_{n_i}, \, i = 1, 2, 3, 4.
\]

where \( \mu_i = \tan \theta_i \) is the friction coefficient at contact point \( p_i \). The above inequalities are equivalent to \( f_i \in C_{\mu_i}^3, \, i = 1, 2, 3, 4 \).

The quadruped robot satisfies the force equilibrium condition and torque equilibrium condition. Let \( Q_i \) be the \( 3 \times 3 \) matrix that transforms forces in the local coordinate system at \( p_i \) into the global coordinate system. Thus, the force at contact point \( p_i \) is \( Q_i f_i \) (in the global coordinate system). The force equilibrium condition is

\[
Q_1 f_1 + Q_2 f_2 + Q_3 f_3 + Q_4 f_4 + f_{ext} = 0 \tag{28}
\]

where \( f_{ext} \in \mathbb{R}^3 \) is the total external force that acts on the quadruped robot (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

\[
p_1 \otimes Q_1 f_1 + p_2 \otimes Q_2 f_2 + p_3 \otimes Q_3 f_3 + p_4 \otimes Q_4 f_4 + \tau_{ext} = 0 \tag{29}
\]

where \( \tau_{ext} \in \mathbb{R}^3 \) is the total external torque that acts on the object (in the global coordinate system). We define the contact force vector \( f \in \mathbb{R}^{12} \) as \( f = (f_1, f_2, f_3, f_4) \), contact matrices \( A_i \in \mathbb{R}^{6 \times 3} \) as

\[
A_i = \begin{bmatrix}
Q_i \\
p_i \otimes Q_i
\end{bmatrix}, \, i = 1, 2, 3, 4,
\]
and overall contact matrix \( A = [A_1, A_2, A_3, A_4] \in \mathbb{R}^{6 \times 12} \). We collect the external force and torque into a single external wrench \( \omega_{\text{ext}} = (-f_{\text{ext}}, -\tau_{\text{ext}}) \in \mathbb{R}^6 \). Thus, we can write the equilibrium conditions (28) and (29) as

\[
Af = \omega_{\text{ext}}.
\]  

(30)

At each contact point \( i \), a direction is \( c_i \in \mathbb{R}^3 \). A set of contact force vectors \( f_i \in C_{\theta_i}^3 \) can be found as a work does as small as possible in \( c_i \), while satisfying the torque equilibrium conditions (30). We define the direction vector \( c \in \mathbb{R}^{12} \) as

\[
c = (c_1, c_2, c_3, c_4).
\]

The force optimization problem for quadruped robot can be expressed as the following circular cone programming

\[
\min_{f} \left\{ c^T f : Af = \omega_{\text{ext}}, f_i \in C_{\theta_i}^3, i = 1, 2, 3, 4 \right\}.
\]  

(31)

For problem (31), we have the following data: \( A = [A_1, A_2, A_3, A_4] \) where

\[
A_1 = \begin{bmatrix}
5 & 1 & 1 \\
1 & 1 & 1 \\
4 & 6 & 3 \\
3 & 3 & 3 \\
\end{bmatrix},
A_2 = \begin{bmatrix}
3 & 6 & 6 \\
1 & 6 & 2 \\
5 & 4 & 1 \\
4 & 5 & 1 \\
\end{bmatrix},
A_3 = \begin{bmatrix}
4 & 3 & 6 \\
3 & 2 & 6 \\
1 & 5 & 2 \\
5 & 6 & 5 \\
\end{bmatrix},
A_4 = \begin{bmatrix}
3 & 3 & 1 \\
6 & 1 & 2 \\
6 & 2 & 6 \\
4 & 4 & 5 \\
\end{bmatrix},
\]

\[
b = \begin{bmatrix}
43 \\
32 \\
39 \\
54 \\
\end{bmatrix},
c_i = \begin{bmatrix}
2 \\
1 \\
0 \\
0 \\
\end{bmatrix}, i = 1, 2, 3, 4.
\]

and \( K = C_{\theta_1}^3 \times C_{\theta_2}^3 \times C_{\theta_3}^3 \times C_{\theta_4}^3 \). The initial points are

\[
x_i = \begin{bmatrix}
2\cot \theta \\
1.0000 \\
0 \\
\end{bmatrix},
s_i = \begin{bmatrix}
2\tan \theta \\
1.0000 \\
0 \\
\end{bmatrix}, i = 1, 2, 3, 4.
\]

and \( y = (0, 0, 0, 0, 0) \).

We choose the rotation angles as \( \theta = \frac{\pi}{12}, \frac{\pi}{10}, \frac{\pi}{8}, \frac{\pi}{7}, \frac{\pi}{6}, \frac{\pi}{5}, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{\pi}{1} \), respectively. The coefficients of friction corresponding with rotation angles are \( \tan \frac{\pi}{12}, \tan \frac{\pi}{10}, \tan \frac{\pi}{8}, \tan \frac{\pi}{7}, \tan \frac{\pi}{6}, \tan \frac{\pi}{5}, \tan \frac{\pi}{4}, \tan \frac{\pi}{3}, \tan \frac{\pi}{2}, \tan \frac{\pi}{1} \), respectively. The numerical results are summarized in Tables 1 and 2. Pobj, Dobj, Itn and DG denote the primal objective function value, the dual objective function value, the iteration number and the duality gap.

Table 1 shows the objective function value, the iteration number and the duality gap. Because the CPU time is almost negligible, we do not report the CPU time. The duality gap implies that desired accuracy is achieved. The simple practical example is shown to justify the efficiency of the proposed algorithm. Observing Table 1, we find that the objective function value gets greater as the angle gets smaller. By (3), the reason for this phenomenon is that the feasible region of the problem (31) gets smaller as the angle gets smaller. On the other hand, the algorithm based on \( \psi_2(t) \) is superior to the algorithm based on \( \psi_1(t) \).

Table 2 shows the duality gaps of each iteration based on \( \psi_2(t) \) for \( \theta = \frac{\pi}{10} \). Observing Table 2, we find that the duality gap between odd iteration and its adjacent odd iteration is reduced with a factor almost 10 after the third iteration.
Table 1. The numerical results for Example 1

| $\theta$ | Pobj | Itn | DG        | Pobj | Itn | DG        |
|----------|------|-----|-----------|------|-----|-----------|
| $\pi/12$ | 20.870090 | 18  | 1.290623e-06 | 20.870088 | 15  | 7.881400e-06 |
| $\pi/10$ | 20.024510 | 18  | 9.554415e-06  | 20.024507 | 16  | 4.043391e-06 |
| $\pi/8$  | 19.092760 | 17  | 1.290726e-06  | 19.092765 | 15  | 1.200469e-06 |
| $\pi/4$  | 17.829635 | 17  | 1.290432e-06  | 17.829635 | 14  | 1.279251e-06 |
| $\pi/2$  | 15.532826 | 16  | 1.290416e-06  | 15.532826 | 18  | 8.382359e-06 |

Table 2. The detailed duality gap based on $\psi_2(t)$ for $\theta = \frac{\pi}{10}$

| Itn | DG       | Itn | DG       |
|-----|----------|-----|----------|
| 1   | 2.770419e+01 | 9   | 1.762983e-02 |
| 2   | 1.898428e+01 | 10  | 5.364676e-03 |
| 3   | 1.047825e+01 | 11  | 1.626194e-03 |
| 4   | 3.735995e+00 | 12  | 4.915940e-04 |
| 5   | 1.854379e+00 | 13  | 1.483108e-04 |
| 6   | 5.997465e+01 | 14  | 1.483108e-05 |
| 7   | 1.869103e+01 | 15  | 1.344574e-05 |
| 8   | 5.762287e-02 | 16  | 4.043391e-06 |

Further, to verify whether iterations required comply with the complexity result in Section 5, we conduct tests in different errors $\epsilon = 10^{-2}$, $\epsilon = 10^{-8}$ and $\epsilon = 10^{-12}$ and do not change the other parameters. The numerical results are summarized in Tables 3, 4 and 5. Observing Tables 3, 4 and 5, we find that the iterations required in practice agree with the complexity result in Section 5.

Example 2. Boyd et al in [10] proposed a force optimization problem for multi-fingered robots, where the grasp force is minimized. Assume that the object is grasped at the $M$ contact points. Then the force optimization problem is formulated as follows:

$$\min_{f} \{ F_{\text{max}} : A f = \omega_{\text{ext}}, f_i \in C_{\delta_i}, i = 1, \ldots, M \}$$

where $F_{\text{max}} = \max \{ \| f_1 \|, \ldots, \| f_M \| \}$, $f_i$ is a contact force at the contact point $p_i$, $f = (f_1, \ldots, f_M) \in \mathbb{R}^{3M}$ and $A \in \mathbb{R}^{6 \times M}$.

We modify the above example slightly to minimize the work caused by force. For the multi-fingered robots that have $M$ contact points with the object, the optimization problem can be expressed into a circular cone optimization problem
Table 3. The numerical results for $\epsilon = 10^{-2}$

| $\theta$   | Pobj | Itn | DG      | $\psi_1(t) = \frac{t^2 - 1}{2} - \log t$ | $\psi_2(t) = \frac{t^2 - 1}{2} + \frac{t^{-2} - 1}{2}$ |
|------------|------|-----|---------|----------------|----------------|
| $\frac{\pi}{12}$ | 20.872661 | 9   | 4.853658e-03 | 20.872289 | 10 | 3.756119e-03 |
| $\frac{\pi}{10}$  | 20.027228 | 11  | 5.341886e-03 | 20.027742 | 10 | 5.364676e-03 |
| $\frac{\pi}{8}$   | 19.096227 | 9   | 6.217684e-03 | 19.097200 | 8  | 8.268439e-03 |
| $\frac{\pi}{6}$   | 17.832871 | 10  | 6.398433e-03 | 17.834034 | 8  | 8.948333e-03 |
| $\frac{\pi}{4}$   | 15.536310 | 8   | 6.524827e-03 | 15.535379 | 9  | 4.831788e-03 |

Table 4. The numerical results for $\epsilon = 10^{-8}$

| $\theta$   | Pobj | Itn | DG      | $\psi_1(t) = \frac{t^2 - 1}{2} - \log t$ | $\psi_2(t) = \frac{t^2 - 1}{2} + \frac{t^{-2} - 1}{2}$ |
|------------|------|-----|---------|----------------|----------------|
| $\frac{\pi}{12}$ | 20.870084 | 21  | 2.533220e-09 | 20.870084 | 21 | 5.470849e-09 |
| $\frac{\pi}{10}$  | 20.024505 | 26  | 6.956931e-09 | 20.024505 | 21 | 9.878296e-09 |
| $\frac{\pi}{8}$   | 19.092974 | 21  | 3.433769e-09 | 19.092974 | 20 | 4.207163e-09 |
| $\frac{\pi}{6}$   | 17.829628 | 25  | 9.406826e-09 | 17.829628 | 20 | 4.546926e-09 |
| $\frac{\pi}{4}$   | 15.532821 | 21  | 3.638380e-09 | 15.532821 | 22 | 8.236674e-09 |

Table 5. The numerical results for $\epsilon = 10^{-12}$

| $\theta$   | Pobj | Itn | DG      | $\psi_1(t) = \frac{t^2 - 1}{2} - \log t$ | $\psi_2(t) = \frac{t^2 - 1}{2} + \frac{t^{-2} - 1}{2}$ |
|------------|------|-----|---------|----------------|----------------|
| $\frac{\pi}{12}$ | 20.870084 | 28  | -1.377735e-09 | 20.870084 | 30 | -3.401532e-09 |
| $\frac{\pi}{10}$  | 20.024505 | 36  | -7.375434e-11 | 20.024505 | 30 | -1.774323e-08 |
| $\frac{\pi}{8}$   | 19.092974 | 29  | -8.984742e-10 | 19.092974 | 28 | -2.375131e-10 |
| $\frac{\pi}{6}$   | 17.829628 | 36  | 3.339551e-13  | 17.829628 | 31 | -3.204846e-09 |
| $\frac{\pi}{4}$   | 15.532821 | 30  | -1.904610e-11 | 15.532821 | 31 | -7.475229e-10 |

as follows.

$$\min_{f} \{ c^T f : Af = \omega_{\text{ext}}, f_i \in C_\theta^3, i = 1, \ldots, M \}$$

where $f = (f_1, \ldots, f_M) \in \mathbb{R}^M$ and $A \in \mathbb{R}^{6 \times M}$.

Therefore, it can be solved by our interior-point algorithm efficiently after giving the fixed data of all parameters.
7. Conclusions. In this paper, we have presented an interior-point method based on kernel functions for circular cone optimization problems which can be used to describe optimal design problems of optimal grasping manipulation for multi-fingered robots. Then we have developed a kernel function based interior-point method to solve circular cone optimization in terms of the corresponding second order cone optimization problem. We have concluded that circular cone optimization is polynomial-time solvable. This inspires us to develop an especial interior-point algorithm for solving general nonsymmetric cone optimization problems, where the nonsymmetric cone constraint may be decomposed as an union of several circular cones.

REFERENCES

[1] F. Alizadeh and D. Goldfarb, Second-order cone programming, Mathematical Programming Series B, 95 (2003), 3–51.
[2] Y. Q. Bai, M. Elf. Ghami and C. Roos, A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization, SIAM Journal on Optimization, 15 (2004), 101–128.
[3] Y. Q. Bai, G. Q. Wang and C. Roos, Primal-dual interior-point algorithms for second-order cone optimization based on kernel functions, Nonlinear Analysis: Theory, Methods & Applications, 70 (2009), 3584–3602.
[4] Y. Q. Bai, Kernel Function-Based Interior-point Algorithms for Conic Optimization, Science Press, Beijing, 2010.
[5] A. Ben-Tal and A. Nemirovski, Lectures on Modern Convex Optimization: Analysis, Algorithms and Engineering Applications, MPS/SIAM Series on Optimization, SIAM Publications, Philadelphia, 2001.
[6] I. Bomze, Copositive optimization-Recent developments and applications, European Journal of Operational Research, 216 (2012), 509–520.
[7] I. Bomze, M. Dur and C. P. Tea, Copositive optimization, Mathematical Optimization Society Newsletter, Optima 89 (2012), 2–8.
[8] P. H. Borgstrom, M. A. Batalin, G. Sukhatme, et al., Weighted barrier functions for computation of force distributions with friction cone constraints, Robotics and Automation (ICRA), 2010 IEEE International Conference on, IEEE, (2010), 785–792.
[9] S. Boyd and L. Vandenberghe, Convex Optimization, Cambridge University Press, Cambridge, 2004.
[10] S. Boyd and B. Wegbreit, Fast computation of optimal contact forces, IEEE Transactions on Robotics, 23 (2007), 1117–1132.
[11] S. Burer, On the copositive representation of binary and continuous nonconvex quadratic programs, Mathematical Programming Series A, 120 (2009), 479–495.
[12] Y. L. Chang, C. Y. Yang and J. S. Chen, Smooth and nonsmooth analyses of vector-valued functions associated with circular cones, Nonlinear Analysis: Theory, Methods and Applications, 85 (2013), 160–173.
[13] J. S. Chen, X. Chen and P. Tseng, Analysis of nonsmooth vector-valued functions associated with second order cones, Mathematical Programming Series B, 101 (2004), 95–117.
[14] S. C. Fang and W. X. Xing, Linear Conic Programming: Theory and Applications, Science Press, Beijing, 2013.
[15] M. Fukushima, Z. Q. Luo and P. Tseng, Smoothing functions for second-order-cone complementarity problems, SIAM Journal on Optimization, 12 (2001), 436–460.
[16] O. Güler, Barrier functions in interior point methods, Mathematics of Operations Research, 21 (1996), 860–885.
[17] C. H. Ko and J. S. Chen, Optimal grasping manipulation for multifingered robots using semismooth Newton method, em appear in Mathematical Problems in Engineering, 2013 (http://math.ntnu.edu.tw/~jschen/Publications.html).
[18] B. León, A. Morales and J. Sancho-Bru, Robot Grasping Foundations, In From Robot to Human Grasping Simulation, Springer International Publishing, (2014), 15–31.
[19] Z. J. Li, S. Z. Sam Ge and S. B. Liu, Contact-force distribution optimization and control for quadruped robots using both gradient and adaptive neural networks, *IEEE Transactions on Neural Networks and Learning Systems, 8* (2014), 1460–1473.

[20] Y. Matsukawa and A. Yoshise, A primal barrier function Phase I algorithm for nonsymmetric conic optimization problems, *Japan Journal of Industrial and Applied Mathematics, 29* (2012), 499–517.

[21] A. Nemirovski, Advanced in convex optimization: Conic optimization, In *International Congress of Mathematicians, 1* (2006), 413–444.

[22] Y. Nesterov, Towards nonsymmetric conic optimization, *Optimization Methods and Software, 27* (2012), 893–917.

[23] Y. Nesterov and M.J. Todd, Primal-dual interior-point methods for self-scaled cones, *SIAM Journal on Optimization, 8* (1998), 324–364.

[24] J. Peng, C. Roos and T. Terlaky, New primal-dual algorithms for second-order conic optimization based on self-regular proximities, *SIAM Journal on Optimization, 13* (2002), 179–203.

[25] J. Renegar, *A Mathematical View of Interior-Point Methods in Convex Optimization*, MPS/SIAM Series on Optimization, SIAM Publications, Philadelphia, 2001.

[26] A. Skajaa and Y. Ye, Homogeneous interior-point algorithm for nonsymmetric convex conic optimization, *To appear mathematical programming*, (2014). Available from: http://link.springer.com/journal/10107/onlineFirst/page/1.

[27] C. J. Yu, K. L. Teo, L. S. Zhang and Y. Q. Bai, A new exact penalty function method for continuous inequality constrained optimization problems, *Journal of Industrial Management and Optimization, 6* (2010), 895–910.

[28] C. J. Yu, K. L. Teo, L. S. Zhang and Y. Q. Bai, On a refinement of the convergence analysis for the new exact penalty function method for continuous inequality constrained optimization problem, *Journal of Industrial Management and Optimization, 8* (2012), 485–491.

[29] J. C. Zhou and J. S. Chen, Properties of circular cone and spectral factorization associated with circular cone, *Journal of Nonlinear and Convex Analysis, 14* (2014), 807–816.

Received May 2014; 1st revision December 2014; final revision March 2015.

E-mail address: yqbai@shu.edu.cn
E-mail address: mathpengfeima@126.com
E-mail address: zhangjingsecond@163.com