Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

A prediction-correction inexact alternating direction method for convex nonlinear second-order cone programming with linear constraints

Yaling Zhang[1,2*] and Hongwei Liu[1]

*Correspondence:
zyldella@126.com
[1]School of Mathematics and Statistics, Xidian University, Xi'an, China
[2]School of Computer Science, Xi'an Science and Technology University, Xi'an, China

Abstract
The convex nonlinear second-order cone programming with linear constraints is equivalent to a separate structure convex programming. A prediction-correction inexact alternating direction method is proposed for the separate structure convex programming. In the proposed method, the convex objective function is not required to be Lipschitz continuous and only needs satisfy an inequality. The global convergence result is given. Numerical results demonstrate that our method is efficient for some random second-order cone programming problems in lower accuracy. In addition, our method can be extended to the convex nonlinear circular cone programming with linear constraints. We also give the simulation results of the three-fingered grasping force optimization problems.

Keywords: Convex nonlinear second-order cone programming; Separate structure convex programming; Inexact alternating direction method; Three-fingered grasping force optimization

1 Introduction

In this paper, we consider the convex nonlinear second-order cone programming (CNSOCP) with linear constraints

$$\begin{aligned} &\min f(x) \\ &\quad \text{s.t. } Ax = b, \quad x \in K, \end{aligned} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a nonlinear continuously differentiable convex function, $A \in \mathbb{R}^{m\times n}$ is a full row rank matrix, $b \in \mathbb{R}^m$ is a vector, $x = [x_1, \dots, x_N] \in \mathbb{R}^{n_1} \times \cdots \times \mathbb{R}^{n_N}$ is viewed as a column vector in $\mathbb{R}^{n_1+\cdots+n_N}$, and $\sum_{i=1}^N n_i = n$. In addition, K is a Cartesian product of second-order cones

$$K = K^{n_1} \times K^{n_2} \times \cdots \times K^{n_N},$$

$x_i \in K^{n_i}$, and K^{n_i} is the n_i-dimensional second-order cone

$$K^{n_i} := \left\{ x_i = \begin{bmatrix} x_{i_1} \\ x_{i_0} \end{bmatrix} \in \mathbb{R}^{n_i-1} \times \mathbb{R} : \|x_{i_1}\| \le x_{i_0} \right\}, \tag{2}$$

where $\|\cdot\|$ denotes the Euclid norm.

The projection $P_{K^{n_i}}(x_i)$ on the second-order cone K^{n_i} is [1, 2]

$$P_{K^{n_i}}(x_i) = \bigl(\lambda_1(x_i)\bigr)_+ c_1(x_i) + \bigl(\lambda_2(x_i)\bigr)_+ c_2(x_i), \quad i = 1, 2, \dots, N, \tag{3}$$

where $s_+ := \max(0, s)$, $\lambda_1(x_i) = x_{i_0} - \|x_{i_1}\|$, $\lambda_2(x_i) = x_{i_0} + \|x_{i_1}\|$, $c_1(x_i) = \frac{1}{2}\bigl[\begin{smallmatrix} -w \\ 1 \end{smallmatrix}\bigr]$, $c_2(x_i) = \frac{1}{2}\bigl[\begin{smallmatrix} w \\ 1 \end{smallmatrix}\bigr]$ with $w = \frac{-x_{i_1}}{\|x_{i_1}\|}$ if $x_{i_1} \neq 0$, and any vector in $\mathbb{R}^{n_i-1}$ satisfying $\|w\| = 1$ if $x_{i_1} = 0$. Then the projection $P_K(x)$ on the cone K is

$$P_K(x) = \bigl[P_{K^{n_1}}(x_1), \dots, P_{K^{n_N}}(x_N)\bigr]. \tag{4}$$

The second-order cone programming has wide applications in engineering problems, such as FIR filter design, antenna array weight design, and truss design [3, 4]. There have been many methods proposed for solving linear second-order cone programming (LSOCP) [5–8]. However, the study of nonlinear second-order cone programming (NSOCP) is much more recent and still in its preliminary phase. In paper [9], a primal-dual interior point method was proposed for solving nonlinear second-order cone programming. In paper [10], theoretical properties of an augmented Lagrangian method for solving nonlinear second-order cone optimization problems were considered. The SQP-type method and trust region SQP-filter method were developed for NSOCP in papers [11] and [12]. In paper [13], a Sl_1QP based algorithm with trust region technique was proposed for solving nonlinear second-order cone programming problems. A homotopy method was presented for nonlinear second-order cone programming in paper [14], and global convergence was proven under mild conditions.

The alternating direction method has been an effective first-order approach for solving optimization problems such as variational inequality problems [15], linear programming [16], and semidefinite programming [17]. In paper [18], a modified alternating direction method was presented for convex nonlinear semidefinite programming problem. The method gives a prototype for nonlinear semidefinite programming. The algorithm needs compute the projection on the semidefinite cone and the convex constraints set. The projection on the convex constraints set is very difficult except for some easy constraints, such as linear constraints.

Inspired by paper [18], a prediction-correction inexact alternating direction method is proposed for convex second-order cone programming problems with linear constraints. In the algorithm, the problem is equivalent to a separate structure convex nonlinear programming. Different from the direct extension method in paper [18], we do not need compute the projection on the linear constraints set, and we only compute the projection on the second-order cone. Moreover, the convex objective function in the proposed method is not required to be Lipschitz continuous and only needs satisfy an inequality. We prove the global convergence. Some random second-order cone programming examples are used to test the performance of the efficiency of our proposed approach, which shows our method is efficient for the examples in low accuracy. In addition, we extend our proposed method to the convex nonlinear circular cone programming with linear constraints. The simulation results of the three-fingered grasping force optimization problems are given.

2 A prediction-correction inexact alternating direction method for CNSOCP problems with linear constraints

Firstly, we give an equivalent separate structure convex programming problem to problem (1):

$$\begin{aligned} &\min f(x) \\ &\quad \text{s.t. } Ax = b \\ &\qquad x = y, \quad y \in K. \end{aligned} \tag{5}$$

Under Slater's condition, strong duality holds for problem (5). Hence, x^* is an optimal solution of (5) if and only if there exists $w^* = (x^*, y^*, \lambda^*, \mu^*) \in \Omega = \mathbb{R}^n \times K \times \mathbb{R}^m \times \mathbb{R}^n$ satisfying the following KKT system in variational inequality form:

$$\begin{cases} \langle x - x^*, \nabla f(x^*) - A^T\lambda^* - \mu^* \rangle \geq 0, & \forall x \in \mathbb{R}^n, \\ \langle y - y^*, \mu^* \rangle \geq 0, & \forall y \in K, \\ Ax^* = b, \\ x^* = y^*, \end{cases} \tag{6}$$

where $\langle \cdot \rangle$ denotes the inner product of two vectors.

The augmented Lagrangian function for the separate structure convex nonlinear programming problem is

$$L(x, y, \lambda, \mu) = f(x) - \lambda^T(Ax - b) - \mu^T(x - y) + \frac{1}{2\beta_1}\|Ax - b\|^2 + \frac{1}{2\beta_2}\|x - y\|^2, \tag{7}$$

where $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^n$, $\beta_1, \beta_2 > 0$.

The exact alternating direction method for (7) is given as follows.

The exact alternating direction method

Given $w^0 = (x^0, y^0, \lambda^0, \mu^0) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$, and $\beta_1, \beta_2 > 0$. For $k = 0, 1, 2, \ldots$, then:

Step 1. Find $x^{k+1} \in \mathbb{R}^n$, which satisfies

$$\begin{aligned} &\biggl\langle x - x^{k+1}, \nabla f\bigl(x^{k+1}\bigr) - A^T\lambda^k - \mu^k + \frac{1}{\beta_1}A^T\bigl(Ax^{k+1} - b\bigr) \\ &\quad + \frac{1}{\beta_2}\bigl(x^{k+1} - y^k\bigr) \biggr\rangle \geq 0, \quad \forall x \in \mathbb{R}^n. \end{aligned} \tag{8}$$

Step 2. Find $y^{k+1} \in K$, which satisfies

$$\biggl\langle y - y^{k+1}, \mu^k - \frac{1}{\beta_2}\bigl(x^{k+1} - y^{k+1}\bigr) \biggr\rangle \geq 0, \quad \forall y \in K. \tag{9}$$

Step 3. Update the Lagrange multiplier by

$$\lambda^{k+1} = \lambda^k - \frac{1}{\beta_1}\left(Ax^{k+1} - b\right). \tag{10}$$

Step 4. Update the Lagrange multiplier by

$$\mu^{k+1} = \mu^k - \frac{1}{\beta_2}\left(x^{k+1} - y^{k+1}\right). \tag{11}$$

It is time consuming to solve problem (8) exactly. Therefore, a prediction-correction inexact alternating direction method is presented to solve problem (5). In the proposed method, we will convert Step 1 and Step 2 to simple projection operations. For this purpose, all we need is the following fact from convex geometry.

Lemma 1 ([19]) *Let Θ be a closed convex set in a Hilbert space and $P_\Theta(x)$ be the projection of x on Θ. Then*

$$\langle z - y, y - x\rangle \geq 0, \quad \forall z \in \Theta \quad \Longleftrightarrow \quad y = P_\Theta(x). \tag{12}$$

Taking $x = \hat{x}^k - \alpha_1(\nabla f(\hat{x}^k) - A^T\lambda^k - \mu^k + \frac{1}{\beta_1}A^T(A\hat{x}^k - b) + \frac{1}{\beta_2}(\hat{x}^k - y^k))$ and $y = \hat{x}^k$ in (12), then (8) is equivalent to the following nonlinear equation:

$$\hat{x}^k = P_{R^n}\left[\hat{x}^k - \alpha_1\left(\nabla f\left(\hat{x}^k\right) - A^T\lambda^k - \mu^k + \frac{1}{\beta_1}A^T\left(A\hat{x}^k - b\right) + \frac{1}{\beta_2}\left(\hat{x}^k - y^k\right)\right)\right], \tag{13}$$

where α_1 can be any positive number.

Taking $x = \hat{y}^k - \alpha_2(\mu^k - \frac{1}{\beta_2}(\hat{x}^k - \hat{y}^k))$ and $y = \hat{y}^k$ in (12), then (9) is equivalent to the following nonlinear equation:

$$\hat{y}^k = P_K\left[\hat{y}^k - \alpha_2\left(\mu^k - \frac{1}{\beta_2}\left(\hat{x}^k - \hat{y}^k\right)\right)\right], \tag{14}$$

where α_2 can be any positive number.

It is time consuming to solve problem (13) directly due to the existence of the term $\nabla f(\hat{x}^k)$ and $A^TA\hat{x}^k$ in (13). The inexact approach is used to solve problem (13), which is similar to the method in paper [18]. Let

$$R_1\left(x^k, \hat{x}^k\right) = \left(1 - \frac{\alpha_1}{\beta_2} - \gamma_1\frac{\alpha_1}{\beta_1}\right)\left(x^k - \hat{x}^k\right) - \alpha_1\left(\nabla f\left(x^k\right) - \nabla f\left(\hat{x}^k\right)\right)$$

and

$$R_2\left(x^k, \hat{x}^k\right) = \frac{\alpha_1}{\beta_1}\left(\left(\gamma_1 I_n - A^TA\right)\left(x^k - \hat{x}^k\right)\right),$$

where $\gamma_1 > \lambda_{\max}(A^TA)$, and $\lambda_{\max}(A^TA)$ is the largest eigenvalue of A^TA.

Instead of computing (13), we compute

$$\begin{aligned}\hat{x}^k &= P_{R^n}\left[\hat{x}^k - \alpha_1\left(\nabla f\left(\hat{x}^k\right) - A^T\lambda^k - \mu^k + \frac{1}{\beta_1}A^T\left(A\hat{x}^k - b\right)\right.\right.\\ &\quad \left.\left. + \frac{1}{\beta_2}\left(\hat{x}^k - y^k\right)\right) + R_1\left(x^k,\hat{x}^k\right) + R_2\left(x^k,\hat{x}^k\right)\right]\\ &= x^k - \alpha_1\left(\nabla f\left(x^k\right) - A^T\lambda^k - \mu^k + \frac{1}{\beta_1}A^T\left(Ax^k - b\right) + \frac{1}{\beta_2}\left(x^k - y^k\right)\right). \end{aligned} \tag{15}$$

Obviously, problem (15) only needs compute the projection on K, whose solution is used as an approximation to the solution of variational inequality (8).

Let $\alpha_2 = \beta_2$ in (14). Then we have

$$\hat{y}^k = P_K\left[-\alpha_2\left(\mu^k - \frac{1}{\beta_2}\hat{x}^k\right)\right]. \tag{16}$$

Now we present the prediction-correction inexact alternating direction method. To simplify the following analysis, we denote

$$G = \begin{pmatrix} I_n & 0 & 0 & 0\\ 0 & \frac{\alpha_1}{\beta_2}I_n & 0 & 0\\ 0 & 0 & \alpha_1\beta_1 I_m & 0\\ 0 & 0 & 0 & \alpha_1\beta_2 I_n \end{pmatrix}.$$

The prediction-correction inexact alternating direction method

Step 0. Given $w^0 = (x^0, y^0, \lambda^0, \mu^0) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^n$, $\beta_1, \beta_2 > 0$, $\eta \in (0,1)$, and $\alpha_0 = \frac{\eta}{1+\eta(\frac{1}{\beta_2}+\frac{\gamma_1}{\beta_1})}$. Set $k = 0$.

Step 1. The prediction step: for a given $w^k = (x^k, y^k, \lambda^k, \mu^k)$, set

$$\begin{cases} \hat{x}^k = x^k - \alpha_1(\nabla f(x^k) - A^T\lambda^k - \mu^k + \frac{1}{\beta_1}A^T(Ax^k - b) + \frac{1}{\beta_2}(x^k - y^k)),\\ \hat{y}^k = P_K[\hat{x}^k - \alpha_2\mu^k],\\ \hat{\lambda}^k = \lambda^k - \frac{1}{\beta_1}(A\hat{x}^k - b),\\ \hat{\mu}^k = \mu^k - \frac{1}{\beta_2}(\hat{x}^k - \hat{y}^k), \end{cases}$$

where $\alpha_1 = (0.1)^i\alpha_0$ satisfies the following inequations by the Armijo line search strategy for $i \in N$ (the set of natural numbers):

$$\begin{aligned}&\left\|\alpha_1\left(\nabla f\left(x^k\right) - \nabla f\left(\hat{x}^k\right)\right) + \frac{\alpha_1}{\beta_1}A^TA\left(x^k - \hat{x}^k\right)\right\|\\ &\quad \le \left(\eta\left(1 - \frac{\alpha_1}{\beta_2}\right) + (1-\eta)\frac{\alpha_1\gamma_1}{\beta_1}\right)\left\|x^k - \hat{x}^k\right\|. \end{aligned} \tag{17}$$

Step 2. The correction step: compute the next iteration point $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1}, \mu^{k+1})$ by the following equation:

$$w^{k+1} = P_{\Omega}\left(w^k - \rho_k d\left(w^k, \hat{w}^k\right)\right), \tag{18}$$

where ρ_k is stepsize determined by

$$\rho_k = \nu\rho_k^* = \nu\frac{\langle w^k - \hat{w}^k, d^k\rangle_G}{\|d(w^k, \hat{w}^k)\|_G}, \qquad \nu \in (0,2),$$

and

$$d(w^k, \hat{w}^k) = \begin{pmatrix} R_1(x^k, \hat{x}^k) + R_2(x^k, \hat{x}^k) \\ y^k - \hat{y}^k \\ \lambda^k - \hat{\lambda}^k \\ \mu^k - \hat{\mu}^k \end{pmatrix}.$$

Here, $\langle \xi_1 \cdot \xi_2\rangle_G = \xi_1^T G \xi_2$ for any vectors $\xi_1, \xi_2 \in \mathbb{R}^n$ and $\|\xi\|_G = \sqrt{\xi^T G \xi}$ for any vector $\xi \in \mathbb{R}^n$.

Remark If the constant L satisfies the following inequality:

$$\|\nabla f(\hat{x}^k) - \nabla f(x^k)\| \le L\|\hat{x}^k - x^k\|, \tag{19}$$

we choose α_1 so that

$$\alpha_1 \le \frac{\eta}{L + \eta(\frac{1}{\beta_2} + \frac{\gamma_1}{\beta_1})} \tag{20}$$

with certain $0 < \eta < 1$. It is easily proven that inequality (17) holds if inequality (20) holds. But different from (20), $\nabla f(x)$ does not require to be Lipschitz continuous in inequality (17). We take the Armijo line search strategy which uses a geometric decreasing series to get α_1 until condition (17) holds.

3 The convergence result

In this section, we extend and modify the convergence results of the alternating direction methods for convex nonlinear semidefinite programs in paper [18] and study convergence of our proposed method.

Lemma 2 *The sequence $\hat{w}^k = (\hat{x}^k, \hat{y}^k, \hat{\lambda}^k, \hat{\mu}^k)$ generated by the prediction-correction inexact alternating direction method satisfies*

$$\begin{aligned} &\langle \hat{w}^k - w^*, d(w^k, \hat{w}^k)\rangle_G \\ &\quad = \langle \hat{x}^k - x^*, R_1(x^k, \hat{x}^k) + R_2(x^k, \hat{x}^k)\rangle + \frac{\alpha_1}{\beta_2}\langle \hat{y}^k - y^*, y^k - \hat{y}^k\rangle \\ &\qquad + \alpha_1\beta_1\langle \hat{\lambda}^k - \lambda^*, \lambda^k - \hat{\lambda}^k\rangle + \alpha_1\beta_2\langle \hat{\mu}^k - \mu^*, \mu^k - \hat{\mu}^k\rangle \ge 0, \end{aligned} \tag{21}$$

where $w^* = (x^*, y^*, \lambda^*, \mu^*)$ is a KKT point of system (6).

Proof Taking $y = \hat{y}^k$ in the second inequality in system (6), we obtain

$$\langle \hat{y}^k - y^*, \mu^*\rangle \ge 0. \tag{22}$$

Coupled with (11) and taking $y = y^*$ in (9), we have

$$\langle y^* - \hat{y}^k, \hat{\mu}^k\rangle \geq 0. \tag{23}$$

Combining with (22) and (23), we have

$$\langle \hat{y}^k - y^*, \mu^* - \hat{\mu}^k\rangle \geq 0. \tag{24}$$

In addition, from (9) and (11), we have

$$\langle y^k - \hat{y}^k, \hat{\mu}^k\rangle \geq 0, \langle \hat{y}^k - y^k, \mu^k\rangle \geq 0.$$

Combining with the two inequalities above, we have

$$\langle \hat{y}^k - y^k, \mu^k - \hat{\mu}^k\rangle \geq 0. \tag{25}$$

Note that (15) can be written equivalently as

$$\begin{aligned}&\left\langle x - \hat{x}^k, \alpha_1\left(\nabla f\left(\hat{x}^k\right) - A^T\hat{\lambda}^k - \hat{\mu}^k + \frac{1}{\beta_2}\left(\hat{y^k - y^k}\right)\right)\right.\\ &\quad\left. - R_1\left(x^k, \hat{x}^k\right) - R_2\left(x^k, \hat{x}^k\right)\right\rangle \geq 0, \quad \forall x \in R^n.\end{aligned}$$

Setting $x = x^*$, we have

$$\begin{aligned}&\left\langle x^* - \hat{x}^k, \alpha_1\left(\nabla f\left(\hat{x}^k\right) - A^T\hat{\lambda}^k - \hat{\mu}^k + \frac{1}{\beta_2}\left(\hat{y^k - y^k}\right)\right)\right.\\ &\quad\left. - R_1\left(x^k, \hat{x}^k\right) - R_2\left(x^k, \hat{x}^k\right)\right\rangle \geq 0, \quad \forall x \in R^n.\end{aligned} \tag{26}$$

Taking $x = \hat{x}^k$ in the first inequality in system (6), we have

$$\alpha_1\langle \hat{x}^k - x^*, \nabla f\left(x^*\right) - A^T\lambda^* - \mu^*\rangle \geq 0. \tag{27}$$

Combining with (26) and (27), we have

$$\begin{aligned}&\langle \hat{x}^k - x^*, \alpha_1 A^T\left(\hat{\lambda}^k - \lambda^*\right)\rangle + \langle \hat{x}^k - x^*, \alpha_1\left(\hat{\mu}^k - \mu^*\right)\rangle\\ &\quad + \left\langle \hat{x}^k - x^*, \frac{\alpha_1}{\beta_2}\left(y^k - y^{k+1}\right)\right\rangle + \langle x^{k+1} - x^*, R_1\left(x^k, \hat{x}^k\right) + R_2\left(x^k, \hat{x}^k\right)\rangle\\ &\quad\geq \alpha_1\langle \hat{x}^k - x^*, \nabla f\left(\hat{x}^k\right) - \nabla f\left(x^*\right)\rangle \geq 0.\end{aligned} \tag{28}$$

Based on the first part on the left-hand side of (28) and the third equation in system (6), we have

$$\begin{aligned}\langle \hat{x}^k - x^*, \alpha_1 A^T\left(\hat{\lambda}^k - \lambda^*\right)\rangle &= \alpha_1\langle A\hat{x}^k - Ax^*, \hat{\lambda}^k - \lambda^*\rangle\\ &= \alpha_1\langle A\hat{x}^k - b, \hat{\lambda}^k - \lambda^*\rangle\\ &= \alpha_1\beta_1\langle \hat{\lambda}^k - \lambda^*, \lambda^k - \hat{\lambda}^k\rangle.\end{aligned} \tag{29}$$

By (11), (24), the last equation in system (6), and the second part on the left-hand side of (28), we have

$$\begin{aligned}
&\alpha_1\langle \hat{x}^k - x^*, \hat{\mu}^k - \mu^*\rangle + \alpha_1\langle \hat{y}^k - y^*, \mu^* - \hat{\mu}^k\rangle \\
&\quad = \alpha_1\langle \hat{\mu}^k - \mu^*, \hat{x}^k - x^* - \hat{y}^k + y^*\rangle \\
&\quad = \alpha_1\langle \hat{\mu}^k - \mu^*, \hat{x}^k - \hat{y}^k\rangle \\
&\quad = \alpha_1\beta_2\langle \hat{\mu}^k - \mu^*, \mu^k - \hat{\mu}^k\rangle.
\end{aligned} \tag{30}$$

In addition, from the third part on the left-hand side of (28), we have

$$\begin{aligned}
&\frac{\alpha_1}{\beta_2}\langle \hat{x}^k - x^*, y^k - \hat{y}^k\rangle \\
&\quad = \frac{\alpha_1}{\beta_2}\langle \hat{y}^k - y^*, y^k - \hat{y}^k\rangle + \frac{\alpha_1}{\beta_2}\langle \hat{x}^k - \hat{y}^k, y^k - \hat{y}^k\rangle \\
&\quad = \frac{\alpha_1}{\beta_2}\langle \hat{y}^k - y^*, y^k - \hat{y}^k\rangle - \frac{\alpha_1}{\beta_2}\langle \hat{y}^k - y^k, \mu^k - \hat{\mu}^k\rangle.
\end{aligned} \tag{31}$$

It follows from (25)–(26) and (28)–(31) that

$$\langle \hat{w}^k - w^*, d(w^k, \hat{w}^k)\rangle_G \geq 0. \qquad \square$$

Lemma 3 *If α_1 satisfies (17), then for any k, we have*

$$\langle w^k - \hat{w}^k, d(w^k, \hat{w}^k)\rangle_G \geq \left(1 - \alpha_1\left(\frac{1}{\beta_2} + \frac{\gamma_1}{\beta_1}\right)\right)(1-\eta)\|w^k - \hat{w}^k\|_G^2. \tag{32}$$

Proof Since α_1 satisfies (17), then the following inequality holds:

$$\begin{aligned}
&\langle x^k - \hat{x}^k, R_1 + R_2\rangle \\
&\quad = \left(1 - \frac{\alpha_1}{\beta_2}\right)\|x^k - \hat{x}^k\|_2^2 - \alpha_1(x^k - \hat{x}^k)^T(\nabla f(x^k) - \nabla f(\hat{x}^k)) \\
&\qquad + \frac{\alpha_1}{\beta_1}(x^k - \hat{x}^k)^T A^T A(x^k - \hat{x}^k) \\
&\quad \geq \left(1 - \alpha_1\left(\frac{1}{\beta_2} + \frac{\gamma_1}{\beta_1}\right)\right)(1-\eta)\|x^k - \hat{x}^k\|_2^2.
\end{aligned}$$

By the inequality above, we have

$$\begin{aligned}
&\langle w^k - \hat{w}^k, d(w^k, \hat{w}^k)\rangle_G \\
&\quad = \langle x^k - \hat{x}^k, R_1 + R_2\rangle + \frac{\alpha_1}{\beta_2}\|y^k - \hat{y}^k\|_2^2 + \alpha_1\beta_1\|\lambda^k - \hat{\lambda}^k\|_2^2 + \alpha_1\beta_2\|\mu^k - \hat{\mu}^k\|_2^2 \\
&\quad \geq \left(1 - \alpha_1\left(\frac{1}{\beta_2} + \frac{\gamma_1}{\beta_1}\right)\right)(1-\eta)\|x^k - \hat{x}^k\|_2^2 + \frac{\alpha_1}{\beta_2}\|y^k - \hat{y}^k\|_2^2 \\
&\qquad + \alpha_1\beta_1\|\lambda^k - \hat{\lambda}^k\|_2^2 + \alpha_1\beta_2\|\mu^k - \hat{\mu}^k\|_2^2 \\
&\quad \geq \left(1 - \alpha_1\left(\frac{1}{\beta_2} + \frac{\gamma_1}{\beta_1}\right)\right)(1-\eta)\|w^k - \hat{w}^k\|_G^2. \qquad \square
\end{aligned}$$

Now, we give the convergent conclusion.

Theorem 1 *The sequence $w^k = (x^k, y^k, \lambda^k, \mu^k)$ generated by the predictor-corrector inexact alternating direction method converges to a KKT point $w^* = (x^*, y^*, \lambda^*, \mu^*)$ of problem* (6).

Proof It is easy to prove that solving the optimal condition (6) for problem (5) is equivalent to finding a zero point of the residual function

$$e(w) = \left\| \begin{matrix} x - P_{R^n}(x - \alpha_1(\nabla f(x) - A^T\lambda - \mu)) \\ y - P_K(y - \alpha_2\mu) \\ Ax - b \\ x - y \end{matrix} \right\|_G . \tag{33}$$

By (15), we have

$$\begin{aligned} \hat{x}^k = P_{R^n}\Big[& \hat{x}^k - \alpha_1\Big(\nabla f(\hat{x}^k) - A^T\hat{\lambda}^k - \hat{\mu}^k + \frac{1}{\beta_2}(\hat{y}^k - y^k)\Big) \\ & + R_1(x^k, \hat{x}^k) + R_2(x^k, \hat{x}^k)\Big]. \end{aligned} \tag{34}$$

From (14), we have

$$\hat{y}^k = P_K[\hat{y}^k - \alpha_2\hat{\mu}^k]. \tag{35}$$

Based on (33)–(35) and the nonexpansion property of the projection operator, we have

$$\begin{aligned} \|e(\hat{w}^k)\|_G &\le \left\| \begin{matrix} \frac{\alpha_1}{\beta_2}(y^k - \hat{y}^k) + R_1(x^k, \hat{x}^k) + R_2(x^k, \hat{x}^k) \\ 0 \\ \beta_1(\lambda^k - \hat{\lambda}^k) \\ \beta_2(\mu^k - \hat{\mu}^k) \end{matrix} \right\|_G \\ &\le \left\| \begin{matrix} R_1(x^k, \hat{x}^k) + R_2(x^k, \hat{x}^k) \\ 0 \\ \beta_1(\lambda^k - \hat{\lambda}^k) \\ \beta_2(\mu^k - \hat{\mu}^k) \end{matrix} \right\|_G + \left\| \begin{matrix} \frac{\alpha_1}{\beta_2}(y^k - \hat{y}^k) \\ 0 \\ 0 \\ 0 \end{matrix} \right\|_G \\ &\le \delta\|w^k - \hat{w}^k\|_G, \end{aligned} \tag{36}$$

where δ is a positive constant depending on parameters α_1, β_1, β_2. By inequality (17), the value of δ is set as

$$\delta = \sqrt{\max\left\{\beta_1, \beta_2, \frac{\alpha_1}{\beta_2}, 2\left(1 - \frac{\alpha_1}{\beta_2}\right)\right\}}. \tag{37}$$

Thus, based on Lemmas 2 and 3, we have

$$\begin{aligned}
&\|w^{k+1}-w^*\|_G^2\\
&\quad\le \|w^k-\rho_k d(w^k,\hat{w}^k)-w^*\|_G^2\\
&\quad=\|w^k-w^*\|_G^2-2\rho_k\langle w^k-w^*,d(w^k,\hat{w}^k)\rangle_G+\rho_k^2\|d(w^k,\hat{w}^k)\|_G^2\\
&\quad=\|w^k-w^*\|_G^2-2\rho_k\langle w^k-\hat{w}^k+\hat{w}^k-w^*,d(w^k,\hat{w}^k)\rangle_G+\rho_k^2\|w^k-\hat{w}^k\|_G^2\\
&\quad\le\|w^k-w^*\|_G^2-2\rho_k\langle w^k-\hat{w}^k,d(w^k,\hat{w}^k)\rangle_G+\rho_k^2\|d(w^k,\hat{w}^k)\|_G^2\\
&\quad=\|w^k-w^*\|_G^2-\nu(2-\nu)\rho_k^*\langle w^k-\hat{w}^k,d(w^k,\hat{w}^k)\rangle_G\\
&\quad\le\|w^k-w^*\|_G^2-\nu(2-\nu)\rho_k^*\left(1-\alpha_1\left(\frac{1}{\beta_2}+\frac{\gamma_1}{\beta_1}\right)\right)(1-\eta)\|w^k-\hat{w}^k\|_G^2\\
&\quad\le\|w^k-w^*\|_G^2-\nu(2-\nu)\rho_k^*\left(1-\alpha_1\left(\frac{1}{\beta_2}+\frac{\gamma_1}{\beta_1}\right)\right)(1-\eta)\frac{1}{\delta^2}\|e(\hat{w}^k)\|_G^2.
\end{aligned}$$

From the above inequality, we have

$$\|w^{k+1}-w^*\|_G^2\le\|w^k-w^*\|_G^2,\quad k=1,2,\ldots. \tag{38}$$

That is, the sequence $\{w^k\}$ is bounded. From the above inequality, we have

$$\sum_{k=0}^{\infty}\nu(2-\nu)\rho_k^*\left(1-\alpha_1\left(\frac{1}{\beta_2}+\frac{\gamma_1}{\beta_1}\right)\right)(1-\eta)\|w^k-\hat{w}^k\|_G^2<+\infty.$$

This implies that $\lim_{k\to\infty}\|w^k-\hat{w}^k\|_G=0$. Thus, the sequence $\{\hat{w}^k\}$ is also bounded. Then there exists at least one cluster point of $\{\hat{w}_k\}$.

From the above inequality, we have

$$\sum_{k=0}^{\infty}\nu(2-\nu)\rho_k^*\left(1-\alpha_1\left(\frac{1}{\beta_2}+\frac{\gamma_1}{\beta_1}\right)\right)(1-\eta)\|e(\hat{w}^k)\|_G^2<+\infty.$$

This implies that $\lim_{k\to\infty}\|e(\hat{w}^k)\|_G=0$.

Let $\bar{w}$ be a cluster point of $\{\hat{w}^k\}$ and the subsequence $\{\hat{w}^{k_j}\}$ converges to $\bar{w}$. We have

$$\|w^{k_j}-\bar{w}\|_G=\lim_{j\to\infty}\|w^{k_j}-\hat{w}^{k_j}\|_G=0$$

and

$$\|e(\bar{w})\|_G=\lim_{j\to\infty}\|e(w^{k_j})\|_G=0.$$

Therefore, $\bar{w}$ satisfies system (6). Setting $w^*=\bar{w}$, we have

$$\|w^{k+1}-\bar{w}\|_G\le\|w^k-\bar{w}\|_G,$$

the sequence $\{w^k\}$ satisfies $\lim_{k\to\infty}w^k=\bar{w}$. □

4 Simulation experiments

In this section we present computational results of the proposed method. All the algorithms are run in MATLAB 7.0 environment on an Inter Core processor 1.80 GHz personal computer with 2.00 GB of Ram.

In the predictor-corrector inexact alternating direction method, we set $\beta_1 = 0.8$, $\beta_2 = 0.8$, $\gamma_1 = \lambda_{\max}(A^TA) + 0.0001$. The initial points x^0, y^0 are randomly generated, and λ^0, μ^0 are zeros.

Example 4.1 In the first test example, we test the following CNSOCP, which was derived from paper [11]:

$$\begin{aligned}
&\min y^TQy + \sum_{i=1}^{n}\left(d_iy_i^4 + f_iy_i\right)\\
&\text{s.t. } By + \begin{pmatrix} b_{n_1}\\ \vdots \\ b_{n_N}\end{pmatrix} \in K,
\end{aligned}$$

where the elements of the matrix B are randomly generated from the interval $[0,2]$, $b_j = e^j$, d_i and f_i are randomly generated from the intervals $[0,1]$ and $[-1,1]$, respectively. In addition, C is given by $Q = C^TC$, where C is an $n \times n$ matrix whose elements are randomly generated from the interval $[0,1]$.

Let

$$z = By + \begin{pmatrix} b_{n_1}\\ \vdots \\ b_{n_N}\end{pmatrix}.$$

Then, as the form in problem (1), we have

$$A = [B - I_n], \qquad x = \begin{bmatrix} y\\ z\end{bmatrix} \in \mathbb{R}^n \times K, \qquad b = -\begin{pmatrix} b_{n_1}\\ \vdots \\ b_{n_N}\end{pmatrix}.$$

Obviously, the problems in Example 4.1 are not Lipschitz continuous, but inequality (17) holds for each k. The detailed test problems are shown in Table 1. In Table 1, the first three test problems are given in [11]. The other 12 test problems are generated based on the method in Example 4.1 and extended the scale of the problem.

For the test problems in Table 1, we compare our proposed method with the SQP-type algorithm in paper [11], and the compared results are listed in Table 1. The SQP-type algorithm is implemented using the SeDuMi solver [20] to solve the subproblems by transforming them into LSOCPs. In SQP-type algorithm, the parameters are set similar to those in paper [11]. The stopping criterion is $\frac{\|\Delta x^k\|}{\|x^k\|} < 10^{-3}$. Let $\Delta f(x^k) = f(x^k) - f(x^{k-1})$. In Example 4.1, our algorithm is terminated when

$$\max\left\{\frac{\|x^k - x^{k-1}\|}{\|x^k\|}, \frac{\|y^k - y^{k-1}\|}{\|y^k\|}, \frac{\|\lambda^k - \lambda^{k-1}\|}{\|\lambda^k\|}, \frac{\|\mu^k - \mu^{k-1}\|}{\|\mu^k\|}, \frac{|\Delta f(x^k))|}{|f(x^k)|}\right\} \le \epsilon$$

for $\epsilon = 10^{-3}$ and $\nu = 0.9$.

Table 1 The compared results for the test problems with medium scale

Problems	m	n	SOC	α_1	PCIADM		SQP	
					Iter.	Time	Iter.	Time
P01	10	20	2×5	$0.1\alpha_0$	37	0.046	6	0.360
P02	30	60	$2 \times 5; 1 \times 20$	$0.01\alpha_0$	50	0.109	7	0.620
P03	50	100	$2 \times 5; 2 \times 20$	$0.01\alpha_0$	74	0.187	7	1.461
P04	70	140	$2 \times 5; 3 \times 20$	$0.01\alpha_0$	69	0.265	8	1.962
P05	90	180	$2 \times 5; 4 \times 20$	$0.01\alpha_0$	26	0.171	8	2.652
P06	110	220	$2 \times 5; 5 \times 20$	$0.01\alpha_0$	30	0.343	9	2.924
P07	130	260	$2 \times 5; 6 \times 20$	$0.01\alpha_0$	87	1.041	9	5.401
P08	150	300	$2 \times 5; 7 \times 20$	$0.01\alpha_0$	24	0.393	10	7.676
P09	170	340	$2 \times 5; 8 \times 20$	$0.01\alpha_0$	25	0.642	11	10.95
P10	190	380	$2 \times 5; 9 \times 20$	$0.01\alpha_0$	34	1.096	11	14.72
P11	210	420	$2 \times 5; 10 \times 20$	$0.01\alpha_0$	53	2.187	12	20.21
P12	230	460	$2 \times 5; 11 \times 20$	$0.01\alpha_0$	33	1.781	13	25.43
P13	250	500	$2 \times 5; 12 \times 20$	$0.01\alpha_0$	34	2.422	14	32.15
P14	270	540	$2 \times 5; 13 \times 20$	$0.01\alpha_0$	51	5.484	14	40.62
P15	290	580	$2 \times 5; 14 \times 20$	$0.01\alpha_0$	104	11.67	15	53.03

In Table 1, an entry of the form "2×5" in the "SOC" column means that there are two 5-dimensional second-order cones, and "α_1" denotes the parameters in the prediction-correction inexact alternating direction method. For the test problems, the iteration number and average CPU time are used to evaluate the performances of the proposed method. The test results are shown in Table 1. In Table 1, "Time" represents the average CPU time (in seconds) and "Iter." denotes the average number of iterations. In addition, "PCIADM" represents the predictor-corrector inexact alternating direction method, and "SQP" represents the SQP-type algorithm in [11].

The results in Table 1 show that prediction-correction inexact alternating direction method costs less CPU time than the SQP-type algorithm in [11]. But the average number of iteration steps of our proposed method is higher than that of the SQP-type algorithm in [11]. Furthermore, the prediction-correction inexact alternating direction method is a first-order algorithm. Therefore, the prediction-correction inexact alternating direction method appears to be beneficial for large scale second-order cone problems with a large number of second-order cones and dense constraint matrices in low accuracy.

Example 4.2 In this example, the grasping force optimization problem for the multi-fingered robotic hand [21, 22] is used to test the performance of the proposed PCIADM. For the robotic hand with m fingers, the optimization problem can be formulated as a convex quadratic circular cone programming problem

$$\begin{aligned} &\min \frac{1}{2} f^T f \\ &\quad \text{s.t. } Gf = -w_{\text{ext}} \\ &\qquad \left\| (f_{i1}, f_{i2})^T \right\| \leq \nu f_{i3} \quad (i = 1, 2, \ldots, m), \end{aligned} \tag{39}$$

where $f = [f_{11}, f_{12}, \cdot, f_{m3}]$ is the grasping force, G is the grasping transformation matrix, w_{ext} is the time-varying external wrench, and μ is the friction coefficient.

In this example, we consider a three-fingered grasping force optimization example [22]. The three-finger robot hand grasps a polyhedral with the grasp points $[0, 1, 0]^T$, $[1, 0.5, 0]^T$, and $[0, -1, 0]^T$, and the robot hand moves along a vertical circular trajectory of radius r

with constant velocity v_1. Let $x = [f_{13}, f_{11}, f_{12}, f_{23}, f_{21}, f_{22}, f_{23}, f_{31}, f_{32}]^T$. Then problem (39) is reformulated as a convex quadratic circular cone programming problem:

$$\begin{aligned} &\min \frac{1}{2} x^T Qx \\ &\quad \text{s.t. } Ax = b \\ &\qquad \|(x_2, x_3)\| \le \tan\theta_1 x_1 \\ &\qquad \|(x_5, x_6)\| \le \tan\theta_2 x_4 \\ &\qquad \|(x_8, x_9)\| \le \tan\theta_3 x_7, \end{aligned} \tag{40}$$

where $Q = \operatorname{diag}(1,1,1,1,1,1,1,1,1)$ is an identity matrix, $\theta_1 = \theta_2 = \theta_3 = \operatorname{actan}(\nu^{-1})$,

$$A = \begin{pmatrix} 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & -0.5 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0.5 & 0 & -1 & 0 & 1 & 0 \end{pmatrix}, \qquad b = \begin{pmatrix} 0 \\ -f_c \sin\theta(t) \\ Mg - f_c \cos\theta(t) \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Here, M is the mass of the polyhedral, $g = 9.8$ m/s^2, $f_c = Mv_1^2/r$ the centripetal force, t is the time, and $\theta(t) = v_1 t/r \in [0, 2\pi]$. In this example, we set the data as follows: $M = 0.1$ kg, $r = 0.2$ m, $n = 0.4\pi$ m/s, and $\mu = 0.6$.

Let its half-aperture angle be $\theta_i \in (0, \frac{\pi}{2})$, $i = 1, 2, \ldots, N$. Then the n_i-dimensional circular cone denoted by L_{θ_i} is

$$L_{\theta_i} = \left\{ x_i = \begin{bmatrix} x_{i_1} \\ x_{i_0} \end{bmatrix} \in R^{n_i - 1} \times R : \|x_{i_1}\| \le \tan\theta_i x_{i_0} \right\}.$$

Our proposed method also can be extended to the convex nonlinear circular cone programming with linear constraints. In the method, the projection on the second-order cone is substituted for the projection on circular cone. The projection computation on the circular cone is shown in paper [23].

Since b is a time-varying variable, we need solve multiple force optimization problems to some given accuracy. Furthermore, the external wrench b of the next problem is not far from that of the previous problem. Based on the data features of the force optimization problems, we simply use the previously computed optimal force vector x as the starting point for the next force optimization problem.

In Example 4.2, our algorithm is terminated when

$$\max\{\|x^k - x^{k-1}\|, \|y^k - y^{k-1}\|, \|\lambda^k - \lambda^{k-1}\|, \|\mu^k - \mu^{k-1}\|, |\Delta f(x^k)|\} \le \epsilon$$

for $\epsilon = 10^{-4}$ and $\nu = 1.6$.

For 4000 force optimization problems with $t = 0 : 1/4000 : 1$, the average iteration number and average CPU time are used to evaluate the performances of the prediction-correction inexact alternating direction method and interior-point method. As is known to all, the primal-dual interior point methods have been proven to be one of the most efficient class of methods for SOCP. Here the Matlab program codes for primal-dual interior point method are designed from the software package by Sedumi [20]. But SeDuMi software cannot solve the grasping force optimization problem (39) directly, so we need transform (39) as a linear second-order cone programming problem [24]:

$$\begin{aligned}
&\min t\\
&\quad \text{s.t. } Af + \omega^{\text{ext}} = 0\\
&\qquad \sqrt{(t-1)^2 + 2\|f\|^2} \le t + 1\\
&\qquad \left\| \left(f_x^{(i)}, f_y^{(i)}\right)^T \right\| \le \tan\theta_i f_z^{(i)}, \quad i = 1, 2, \ldots, M.
\end{aligned}$$

In the SeDuMi software, the stop criterion is pars.eps = 10^{-4}.

Here, during the 4000 force optimization problems, the value of b will be recalculated in each problem. The test results are shown in Table 2. We also give the test results for 2000 force optimization problems with $t = 0 : 1/2000 : 1$ in Table 2.

The results in Table 2 show that the prediction-correction inexact alternating direction method costs more iteration steps than the interior-point method. On the other hand, the prediction-correction inexact alternating direction method costs less CPU time than SeDuMi.

Table 2 The test results for the multiple force optimization problems

Methods	4000		2000	
	Iter.	Time	Iter.	Time
PCIADM	43	0.0521	59	0.0619
SeDuMi	7	0.1967	7	0.1967

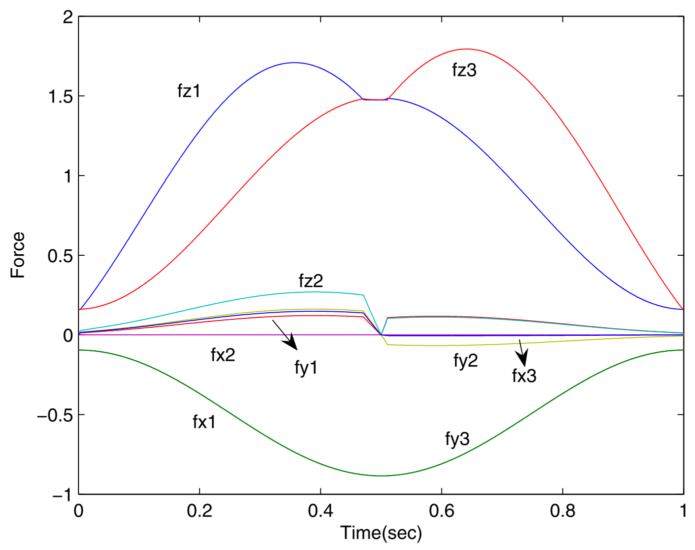

Figure 1 *The trajectories of forces for 4000 FOPs by the PCIADM method.* The optimal forces for the 4000 force optimization problems solved by the prediction-correction inexact alternating direction method are shown in Figure 1

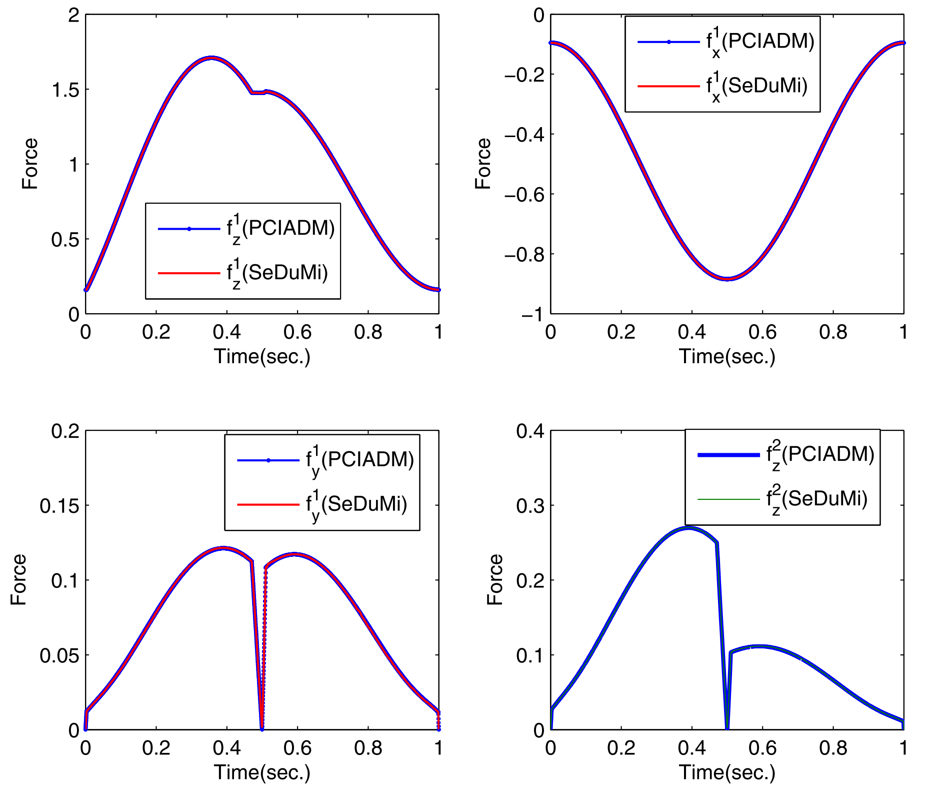

Figure 2 *Comparative results of the two methods*. Figure 2 gives optimal force $f_z^1, f_x^1, f_y^1, f_z^2$ by the two methods

The optimal forces for the 4000 force optimization problems solved by the prediction-correction inexact alternating direction method are shown in Fig. 1. In addition, Fig. 2 gives optimal force $f_z^1, f_x^1, f_y^1, f_z^2$ by the two methods.

The results in Fig. 2 show that the optimal forces solved by the two methods are almost similar. Figures 1–2 and Table 2 demonstrate that our methods are efficient for the grasping force optimization problems.

5 Conclusion

In this paper, the convex nonlinear second-order cone programming problem with linear constraints is equivalent to a separate structure convex programming. A prediction-correction inexact alternating direction method is proposed to solve the separate structure convex programming. In the method, the Lipschitz continuity does not need to be satisfied. In addition, the proposed method does not require to solve sub-variational inequality problems exactly. At each iteration, we only need compute the metric projection on the second-order cone. The proposed predictor-corrector inexact alternating direction method does not require second-order information and it is easy to implement.

Acknowledgements
The authors are very grateful to the editor and two anonymous referees for their valuable comments which led to the improvements of the paper.

Funding
This work was supported by the National Science Basic Research Plan in ShaanXi Province of China (2015JM1031) and the Fundamental Research Funds for the Central Universities (JB150713).

Availability of data and materials
The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
YZ designed the prediction-correction inexact alternating direction method for the convex nonlinear second-order cone programming with linear constraints, gave convergence analysis, and was the major contributor in writing the manuscript. HL performed the experiments of the three-fingered grasping force optimization problems. All authors read and approved the final manuscript.

Publisher's Note

Received: 5 September 2019 Accepted: 13 January 2020 Published online: 15 January 2020

References

1. Outrata, J.V., Sun, D.F.: On the coderivative of the projection operator onto the second-order cone. Set-Valued Var. Anal. **16**, 999–1014 (2008)
2. Kong, L.C., Tuncel, L., Xiu, N.H.: Clarke generalized Jacobian of the projection onto symmetric cones. Set-Valued Var. Anal. **17**, 135–151 (2009)
3. Lu, W.S., Hinamoto, T.: Optimal design of IIR digital filters with robust stability using conic-quadratic-programming updates. IEEE Trans. Signal Process. **51**, 1581–1592 (2003)
4. Luo, Z.Q.: Applications of convex optimization in signal processing and digital communication. Math. Program. **97B**, 177–207 (2003)
5. Alizadeh, F., Goldfarb, D.: Second-order cone programming. Math. Program. **95**, 3–51 (2003)
6. Fukushima, M., Luo, Z.Q., Tseng, P.: Smoothing functions for second-order cone complementarity problems. SIAM J. Optim. **12**, 436–460 (2002)
7. Chen, J.S., Tseng, P.: An unconstrained smooth minimization reformulation of the second-order cone complementarity problems. Math. Program. **104**, 293–327 (2005)
8. Kanzow, C., Ferenczi, I., Fukushima, M.: On the local convergence of semismooth Newton methods for linear and nonlinear second-order cone programs without strict complementarity. SIAM J. Optim. **20**, 297–320 (2009)
9. Yamashita, H., Yabe, H.: A primal-dual interior point method for nonlinear optimization over second-order cones. Optim. Methods Softw. **24**, 407–426 (2009)
10. Liu, Y.J., Zhang, L.W.: Convergence of the augmented Lagrangian method for nonlinear optimization problems over second-order cones. J. Optim. Theory Appl. **139**, 557–575 (2008)
11. Kato, H., Fukushima, M.: An SQP-type algorithm for nonlinear second-order cone programs. Optim. Lett. **1**, 129–144 (2007)
12. Zhang, X., Liu, Z., Liu, S.: A trust region SQP-filter method for nonlinear second-order cone programming. Comput. Math. Appl. **63**, 1569–1576 (2012)
13. Okuno, T., Yasuda, K., Hayashi, S.: Sl_1QP based algorithm with trust region technique for solving nonlinear second-order cone programming problems. Interdiscip. Inf. Sci. **21**, 97–107 (2015)
14. Li, Y., Yu, B., Li, Y.X.: A homotopy method for nonlinear second-order cone programming. Numer. Algorithms **68**, 355–365 (2015)
15. He, B.S., Liao, L.Z., Han, D., Yang, H.: A new inexact alternating directions method for monotone variational inequalities. Math. Program. **92**, 103–118 (2002)
16. Eckstein, J., Bertsekas, D.P.: An alternating direction method for linear programming. LIDS-P, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (1967)
17. Sun, J., Zhang, S.: A modified alternating direction method for convex quadratically constrained quadratic semidefinite programs. Eur. J. Oper. Res. **207**, 1210–1220 (2010)
18. Zhang, S., Ang, J., Sun, J.: An alternating direction method for solving convex nonlinear semidefinite programming problems. Optimization **62**, 527–543 (2013)
19. Kinderlehrer, D., Stampacchia, G.: An Introduction to Variational Inequalities and Their Applications. Academic Press, New York (1980)
20. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optim. Methods Softw. **11–12**, 625–653 (1999)
21. Boyd, S., Wegbreit, B.: Fast computation of optimal contact forces. IEEE Trans. Robot. **23**, 1117–1132 (2007)
22. Ko, C.H., Chen, J.S., Ching, Y.Y.: Recurrent neural networks for solving second-order cone programs. Neurocomputing **74**, 3646–3653 (2011)
23. Zhou, J.C., Chen, J.S.: Properties of circular cone and spectral factorization associated with circular cone. J. Nonlinear Convex Anal. **214**, 807–816 (2013)
24. Zhao, X.Y.: A semismooth Newton-CG augmented Lagrangian method for large scale linear and convex quadratic SDPs. Ph.D. thesis, National University of Singapore (2009)