Proportional-Integral Projected Gradient Method for Conic Optimization

Yue Yu, Purnanand Elango, Ufuk Topcu, Behçet Açıkmese,

Oden Institute for Computational Engineering and Sciences, The University of Texas, Austin, TX, 78712
Department of Aeronautics and Astronautics, University of Washington, Seattle, WA, 98195

Abstract

Conic optimization is the minimization of a differentiable convex objective function subject to conic constraints. We propose a novel primal-dual first-order method for conic optimization, named proportional-integral projected gradient method (PIPG). PIPG ensures that both the primal-dual gap and the constraint violation converge to zero at the rate of $O(1/k^q)$, where $k$ is the number of iterations. If the objective function is strongly convex, PIPG improves the convergence rate of the primal-dual gap to $O(1/k^2)$. Further, unlike any existing first-order methods, PIPG also improves the convergence rate of the constraint violation to $O(1/k^3)$. We demonstrate the application of PIPG in constrained optimal control problems.

Key words: Convex optimization, first-order methods, optimal control

1 Introduction

Conic optimization is the minimization of a differentiable convex objective function subject to conic constraints:

$$\begin{align*}
\text{minimize} & \quad f(z) \\
\text{subject to} & \quad Hz - g \in K, \quad z \in D,
\end{align*}$$

where $z \in \mathbb{R}^n$ is the solution variable, $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable and convex objective function, $K \subset \mathbb{R}^m$ is a closed convex cone and $D \subset \mathbb{R}^n$ is a closed convex set, $H \in \mathbb{R}^{m \times n}$ and $g \in \mathbb{R}^m$ are constraint parameters. By proper choice of cone $K$, conic optimization (1) generalizes linear programming, quadratic programming, second-order cone programming, and semi-definite programming [Ben-Tal and Nemirovski, 2001, Boyd and Vandenberghe, 2004]. Conic optimization has found applications in various areas, including signal processing [Luo and Yu, 2006], machine learning [Andersen et al., 2011], robotics [Majumdar et al., 2020], and aerospace engineering [Liu et al., 2017, Eren et al., 2017, Malyuta et al., 2021].

The goal of numerically solving optimization (1) is to compute a solution $z^*$ that achieves, up to a given numerical tolerance, zero violation of the constraints in (1) and zero primal-dual gap; the latter implies that $z^*$ is an optimal solution of optimization (1) [Boyd et al., 2011, He and Yuan, 2012, Chambolle and Pock, 2011, Chambolle and Pock, 2016b]. To this end, numerical methods iteratively compute a solution whose constraint violation and primal-dual gap are nonzero at first but converge to zero as the number of iteration $k$ increases.

Due to their low computational cost, first-order methods have attracted increasing attention in conic optimization [Lan et al., 2011, Boyd et al., 2011, O’Donoghue et al., 2016, Chambolle and Pock, 2016a, Yu et al., 2020b]. Unlike second-order methods, such as interior point methods [Nesterov and Nemirovskii, 1994, Andersen et al., 2003], first-order methods do not rely on computing matrix inverses. They consequently are suitable for implementation with limited computational resources.

The existing first-order methods solve optimization (1) by solving two different equivalent problems. The first equivalent problem is the following optimization with equality constraints [Boyd et al., 2011, O’Donoghue...
et al., 2016, Stellato et al., 2020, Yu et al., 2020b):

\[
\begin{align*}
\min_{z, y} & \quad f(z) \\
\text{subject to} & \quad Hz - y = g, \quad y \in K, \quad z \in D.
\end{align*}
\]

In particular, the alternating direction method of multipliers (ADMM) solves optimization (1) by computing one projection onto cone $K$ and multiple projections onto set $D$ in each iteration. ADMM ensures that both the constraint violation and the primal-dual gap converge to zero at rate of $O(1/k)$, where $k$ is the number of iterations [Gabay and Mercier, 1976, Eckstein, 1989, Fortin and Glowinski, 2000, Boyd et al., 2011, He and Yuan, 2012, Wang and Banerjee, 2014]. The proportional-integral projected gradient method for equality constrained optimization (PIPG) ensures the same convergence rates as ADMM, while computing one projection onto cone $K$ and only one projection onto set $D$ in each iteration [Yu et al., 2020b]. Although variants of ADMM [Goldstein et al., 2014, Kadkhodaei et al., 2015, Ouyang et al., 2015, Xu, 2017] and PIPG [Xu, 2017, Yu et al., 2020b] can achieve accelerated convergence rates for strongly convex objective functions, such accelerations are not possible for optimization (2) because the objective function in (2) is independent of variable $y$ and, as a result, not strongly convex.

Another problem equivalent to optimization (1) is the following saddle-point problem, where $K^\circ$ is the polar cone of $K$ [Chambolle and Pock, 2011, Chambolle and Pock, 2016b]:

\[
\begin{align*}
\min_{z \in D} \max_{w \in K^\circ} & \quad f(z) + \langle Hz - g, w \rangle.
\end{align*}
\]

In particular, the primal-dual hybrid-gradient method (PDHG) solves saddle-point problem (3) by computing one projection onto cone $K^\circ$ and one projection onto set $D$ in each iteration. PDHG ensures that the primal-dual gap converges to zero at the rate of $O(1/k)$ when for convex $f$, and at an accelerated rate of $O(1/k^3)$ for strongly convex $f$ [Chambolle and Pock, 2016a, Chambolle and Pock, 2016b]. However, since the constraint $Hz - g \in K$ is not explicitly considered in (3), the existing convergence results on PDHG do not provide any convergence rates of the violation of this constraint [Chambolle and Pock, 2016a, Chambolle and Pock, 2016b].

We compare the per-iteration computation and the convergence rates of ADMM, PIPG and PDHG in Tab. 1. None of these methods simultaneously has accelerated convergence rates (i.e., better than $O(1/k)$) for strongly convex $f$ and guaranteed convergence rates on the constraint violation. To our best knowledge, whether there exists a first-order method that achieves both convergence results remains an open question.

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|}
\hline
\text{Algorithms} & \# \text{of proj. per iter.} & \text{convergence rates} & \# \text{of proj. per iter.} & \text{convergence rates} \\
\hline
\text{ADMM} & O(1/\sqrt{\epsilon}) & 1 & O(1/k) & O(1/k) & O(\ln(1/\epsilon)) & 1 & O(1/k) & O(1/k) \\
\text{PIPG} & 1 & 1 & O(1/k) & O(1/k) & 1 & 1 & O(1/k^3) & N/A \\
\text{PDHG} & 1 & 1 & O(1/k) & N/A & 1 & 1 & O(1/k^3) & O(1/k^3) \\
\hline
\end{array}
\]

$\epsilon > 0$ is a tunable accuracy tolerance in ADMM, $K^\circ$ denotes the polar cone of $K$.

We answer this question affirmatively by proposing a novel primal-dual first-order method for conic optimization, named proportional-integral projected gradient method (PIPG). By combining the idea of proportional-integral feedback control and projected gradient method, PIPG ensures the following convergence results.

1. For convex $f$, both the primal-dual gap and the constraint violation converge to zero at the rate of $O(1/k)$.
2. For strongly convex $f$, the convergence rate can be improved to $O(1/k^3)$ for the primal-dual gap and $O(1/k^3)$ for the constraint violation.

PIPG generalizes both PDHG with constant step sizes [Chambolle and Pock, 2016b, Alg. 1] and PIPG [Yu et al., 2020b]. Compared with the existing methods, PIPG has the following advantages; see Tab. 1 for an overview. In terms of per-iteration cost, it computes one projection onto cone $K^\circ$ and one projection onto set $D$, which is the same as PIPG and PDHG and fewer times of projections than ADMM. In terms of its convergence rates, to our best knowledge, the $O(1/k^3)$ convergence rate of constraint violation has never been achieved before for general conic optimization. We numerically demonstrate these advantages of PIPG on several constrained optimal control problems.

The rest of the paper is organized as follows. After some preliminary results on convex analysis, Section 2 re-
Let $\mathbb{K} \subset \mathbb{R}^m$ be a closed convex cone, *i.e.*, $\mathbb{K}$ is a closed convex set and $\gamma w \in \mathbb{K}$ for any $w \in \mathbb{K}$ and $\gamma \in \mathbb{R}_+$. The polar cone of $\mathbb{K}$ is also a closed convex cone given by

$$K^* := \{ w \in \mathbb{R}^m : \langle w, y \rangle \leq 0, \forall y \in \mathbb{K} \}. \quad (8)$$

### 2.2 Related work

We briefly review three existing first-order primal-dual conic optimization methods: ADMM, PIPGeq, and PDHG. In the following, let $\alpha, \beta, \gamma$ denote positive scalar step sizes, and $\{\alpha^j\}_{j \in \mathbb{N}}, \{\beta^j\}_{j \in \mathbb{N}}, \{\gamma^j\}_{j \in \mathbb{N}}$ denote sequences of positive scalar step sizes. For simplicity, we assume all methods are terminated after a fixed number of iterations, denoted by $k \in \mathbb{N}$.

#### 2.2.1 Alternating direction method of multipliers

As a special case of Douglas-Rachford splitting method [Eckstein, 1989, Fortin and Glowinski, 2000], alternating direction method of multipliers (ADMM) solves optimization (1) by solving the equivalent optimization (2) using Algorithm 1 [Gabay and Mercier, 1976, Boyd et al., 2011, He and Yuan, 2012].

**Algorithm 1 ADMM**

**Input:** $k, \alpha, z^1 \in \mathbb{D}, y^1 \in \mathbb{K}, w^1 \in \mathbb{R}^m$

**Output:** $z^k$

1: for $j = 1, 2, \ldots, k - 1$ do
2: \hspace{1em} $z^{j+1} = \text{argmin}_{z \in \mathbb{D}} \left[ f(z) + \frac{\alpha}{2} \| Hz - y^j - g + w^j \|^2 \right]$
3: \hspace{1em} $y^{j+1} = \pi_\mathbb{K} [ Hz^{j+1} - g + w^j ]$
4: \hspace{1em} $w^{j+1} = w^j + Hz^{j+1} - y^{j+1} - g$
5: end for

Generally, the minimization in the line 2 of Algorithm 1 can only be solved approximately up to a numerical tolerance $\epsilon > 0$ using iterative methods. Such methods need to compute at least $O(1/\sqrt{\epsilon})$ projections onto set $\mathbb{D}$ if $f$ is merely convex, and $O(\ln \frac{1}{\epsilon})$ projections if function $f$ is strongly convex; see [Nesterov, 2018, Chp. 2] for a detailed discussion.

There has been many variants of ADMM developed in the literature. However, none of them lead to any significant benefits for optimization in (2). For example, [Ouyang et al., 2015] and [Xu, 2017, Alg. 1] simplified the minimization in the line (2) of Algorithm 1 by approximating function $f$ using its linearization. However, solving the resulting approximate minimization still requires multiple projections onto set $\mathbb{D}$. On the other hand, although the convergence of ADMM can be accelerated when the objective function is strongly convex [Goldstein et al., 2014, Kadkhodaie et al., 2015, Ouyang et al., 2015, Xu, 2017], such acceleration does not apply.

views existing first-order conic optimization methods. Section 3 introduces PIPG along with its convergence results. Section 4 demonstrate PIPG via numerical examples on constrained optimal control. Finally, Section 5 concludes and comments on future work.
to the optimization (2). The reason is because the objective function in (2) is not strongly convex with respect to (in fact, does not depend on) variable $y$.

2.2.2 Proportional-integral projected gradient method for equality constrained optimization

Motivated by applications in model predictive control, the proportional-integral projected gradient method for equality constrained optimization (PIPGeq) solves optimization (1) by solving the equivalent optimization (2) using Algorithm 2.

Algorithm 2 PIPGeq

**Input:** $k, \alpha, \beta, z^1 \in \mathbb{D}, y^1 \in \mathbb{K}, w^1 \in \mathbb{R}^m$.

**Output:** $z^k$.

1: for $j = 1, 2, \ldots, k - 1$
2: \hspace{1em} $v^j + 1 = w^j + \beta (H (z^j - y^j) - g)$
3: \hspace{1em} $z^j + 1 = \pi_E [z^j - \alpha \nabla f (z^j) + H^T \nu^j + 1)]$
4: \hspace{1em} $y^j + 1 = \pi_E [y^j + \nu^j + 1]$
5: \hspace{1em} $w^j + 1 = \beta (H z^j + 1 - y^j + 1 - g)$
6: end for

Unlike line 2 in Algorithm 1, line 3 in Algorithm 2 computes only one projection onto set $\mathbb{D}$ instead of multiple times. As a result, PIPGeq can achieve the same convergence rates as those of ADMM while lowering the iteration computation cost [Xu, 2017, Yu et al., 2020b].

2.2.3 Primal-dual hybrid gradient method

Motivated by applications in computational imaging, the primal-dual hybrid gradient method (PDHG) was first introduced in [Chambolle and Pock, 2011] and later shown to be equivalent to Douglas-Rachford splitting method [O’Connor and Vandenberghe, 2020]. Later, another variant of PDHG was introduced in [Chambolle and Pock, 2016b], which is an instance of three-operator splitting methods [Vu, 2013, Condat, 2013, Chen et al., 2016, Davis and Yin, 2017, Yan, 2018].

To solve optimization (1), PDHG solves the equivalent convex-concave saddle point problem (3) instead. If function $f$ is merely convex, PDHG uses Algorithm 3. If function $f$ is $\mu$-strongly convex for some $\mu > 0$, then PDHG uses Algorithm 4 instead.

Algorithm 3 PDHG with constant step sizes

**Input:** $k, \alpha, \beta, z^1 \in \mathbb{D}, w^1 \in \mathbb{K}^\circ$.

**Output:** $z^k$.

1: for $j = 1, 2, \ldots, k - 1$
2: \hspace{1em} $z^j + 1 = \pi_E [z^j - \alpha \nabla f (z^j) + H^T w^j + 1)]$
3: \hspace{1em} $w^j + 1 = \pi_E [w^j + \beta \nabla f (z^j + 1 + 1 - g)]$
4: end for

The primal-dual gap converges to zero at the rate of $O(1/k)$ and $O(1/k^2)$ for Algorithm 3 and Algorithm 4, respectively [Chambolle and Pock, 2016b]. However, to our best knowledge, there is no convergence result on

Algorithm 4 PDHG with varying step sizes

**Input:** $k, \alpha_j, \beta_j, \gamma_j \in \mathbb{D}$, $\mu, z^1 \in \mathbb{K}^\circ$.

**Output:** $z^k$.

1: for $j = 1, 2, \ldots, k - 1$
2: \hspace{1em} $w^j + 1 = \pi_E [w^j + \beta \nabla f (z^j + 1 + 1) - g)]$
3: \hspace{1em} $z^j + 1 = \pi_E [z^j - \alpha \nabla f (z^j) + H^T w^j + 1)]$
4: end for

the constraint violation for either Algorithm 3 or Algorithm 4.

3 Proportional-integral projected gradient method

We introduce a novel first-order primal-dual method, namely proportional-integral projected gradient method (PIPG), for conic optimization (1), and discuss its convergence rates in terms of the constraint violation and the primal-dual gap.

Algorithm 5 summarizes the proposed method, where $k \in \mathbb{N}$ is the maximum number of iterations, and $\alpha_j^k, \beta_j^k$ are sequences of positive scalar step sizes that will be specified later. We note that, instead of maximum number of iterations, one can use alternative stopping criteria, such as the distance between $H z^j - g$ and $K$ reaching a given tolerance.

Algorithm 5 PIPG

**Input:** $k, \alpha_j, \beta_j, \gamma_j \in \mathbb{D}$, $v^1 \in \mathbb{K}^\circ$.

**Output:** $z^k$.

1: for $j = 1, 2, \ldots, k - 1$
2: \hspace{1em} $w^j + 1 = \pi_E [w^j + \beta \nabla f (z^j + 1 + 1) - g)]$
3: \hspace{1em} $z^j + 1 = \pi_E [z^j - \alpha \nabla f (z^j) + H^T w^j + 1)]$
4: \hspace{1em} $v^j + 1 = \beta \nabla f (z^j + 1 + 1 - g)$
5: end for

The name PIPG is due to the following observations. First, if $K = \{0\}$, then $K^\circ = \mathbb{R}^m$ and line 2 and line 4 in Algorithm 5 become the following:

$$w^j + 1 = v^j + \beta (H z^j - g), \quad (9a)$$
$$v^j + 1 = v^j + \beta \nabla f (z^j + 1 + 1 - g). \quad (9b)$$

Using (9b) one can show that

$$v^1 = v^1 + \sum_{i=2}^j \beta_i - 1 (H z^i - g).$$

Hence $v^k$ is a weighted summation, or numerical integration, of $H z^j - g$ from $i = 2$ to $i = j$. Further, (9a) states that $w^j$ adds a proportional term of $H z^j - g$ to $v^j$, hence $w^j$ in (9a) is a proportional-integral term of
$Hz^j - g$. Second, if $H$ is a zero matrix, then line 3 in Algorithm 5 becomes a projected gradient method that minimizes $f$ over set $D$ [Nesterov, 2018, Sec. 2.2.5]. Therefore Algorithm 5 can be interpreted as a combination of proportional-integral feedback control and the projected gradient method. Similar idea has also been popular in equality constrained optimization [Wang and Elia, 2010, Yu et al., 2020a, Yu and Açıkmeşe, 2020, Yu et al., 2020b].

Remark 1 Notice that the $w^{j+1}$ in (9a) is otherwise identical to the $v^{j+1}$ in (9b) except that (9a) uses $z^j$ whereas (9b) uses $z^{j+1}$. Such scheme is also known as a prediction-correction step, which has been popular in many first-order primal-dual methods, including the extra-gradient and mirror-prox method [Korpelevich, 1977, Nemirovski, 2004, Nesterov, 2007], the accelerated linearized ADMM [Ouyang et al., 2015, Xu, 2017], the primal-dual fixed point methods [Krol et al., 2012, Chen et al., 2013, Chen et al., 2016, Yan, 2018] and the accelerated mirror descent method [Cohen et al., 2018].

Remark 2 One can verify that if $\alpha^j \equiv \alpha$ and $\beta^j \equiv \beta$ for $j = 1, 2, \ldots, k$, then Algorithm 5 is equivalent to Algorithm 3, the latter was first introduced in [Chambolle and Pock, 2016b, Alg. 1].

Next, we will show the convergence results of Algorithm 5. To this end, we will frequently use the following quadratic distance function to closed convex cone $\mathbb{K}$:

$$d_k(w) := \min_{v \in \mathbb{K}} \frac{1}{2} \|w - v\|^2,$$

which is continuously differentiable and convex [Nesterov, 2018, Lem. 2.2.9]. We will also use the following Lagrangian function:

$$L(z, w) := f(z) + \langle Hz - g, w \rangle.$$

We make the following assumptions on optimization (1).

**Assumption 1** (1) Function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable. There exists $\mu, \lambda \in \mathbb{R}_+$ with $\mu \leq \lambda$ such that $f$ is $\mu$-strongly convex and $\lambda$-smooth, i.e.,

$$\frac{\mu}{2} \|z - z'\|^2 \leq B_f(z, z') \leq \frac{\lambda}{2} \|z - z'\|^2$$

for all $z, z' \in \mathbb{R}^n$.

(2) Set $D \subset \mathbb{R}^n$ and cone $\mathbb{K} \subset \mathbb{R}^m$ are closed and convex.

(3) There exists $z^* \in D$ and $w^* \in \mathbb{K}^c$ such that

$$L(z^*, w^*) \leq L(z^*, w^*) \leq L(z, w^*)$$

for all $z \in D$ and $w \in \mathbb{K}^c$.

Under the above assumptions, the quantity $L(z, w) - L(z^*, w)$, also known as the primal-dual gap evaluated at $(z, w)$, is non-negative [Boyd et al., 2011, He and Yuan, 2012, Chambolle and Pock, 2011, Chambolle and Pock, 2016b]. The following proposition provides a sufficient condition on $z$ and $w$ under which the primal-dual gap $L(z, w) - L(z^*, w)$ equals zero and $z$ is an optimal solution of optimization (1).

**Proposition 1** If there exists $z \in D$ and $w \in \mathbb{K}^c$ such that

$$L(z, w) - L(z^*, w) \leq 0,$$

for all $z \in D$ and $w \in \mathbb{K}^c$, then $z$ is an optimal solution of optimization (1), i.e., $Hz - g \in \mathbb{K}$ and $f(z) \leq f(z^*)$ for any $z \in D$ such that $Hz - g \in \mathbb{K}$.

**Proof** See Appendix A.

As our first step, the following lemma proves a key inequality for our later discussions.

**Lemma 1** Suppose that Assumption 1 holds with $(w^j, z^j, v^j)_{j=1}^k$ is computed using Algorithm 5 where $\alpha^j, \beta^j > 0$ and $\alpha^j (\lambda + \sigma \beta^j) = 1$ for some $\sigma \geq \|H\|^2$ and all $j = 1, 2, \ldots, k$. Then

$$\beta^j d_k(Hz^j - g) + L(z^{j+1}, w) - L(z, w^{j+1})$$

$$\leq \left( \frac{1}{2\alpha^j} - \frac{\mu}{2} \right) \|z^j - z\|^2 + \frac{1}{2\beta^j} \|v^j - w\|^2$$

$$- \frac{1}{2\alpha^j} \|z^{j+1} - z\|^2 - \frac{1}{2\beta^j} \|v^{j+1} - w\|^2,$$

for all $z \in D$, $w \in \mathbb{K}^c$, and $j = 1, 2, \ldots, k$.

**Proof** See Appendix B.

Equipped with Lemma 1, we are ready to prove the convergence results of Algorithm 5. The idea is to first summing up the inequality in Lemma 1 corresponding to different value of $j$, then using the Jensen’s inequality.

We start with the case where $\mu = 0$, i.e., function $f$ is merely convex. The following theorem shows the convergence results of Algorithm 5 in this case.

**Theorem 1** Suppose that Assumption 1 holds with $\mu = 0$, and $(w^j, z^j, v^j)_{j=1}^k$ is computed using Algorithm 5 with $\alpha^j = \frac{1}{\lambda + \sigma \beta^j}$ and $\beta^j = \beta$ and all $j = 1, 2, \ldots, k$, where $\beta > 0$ and $\sigma \geq \|H\|^2$. Let

$$z^k := \frac{1}{k} \sum_{j=1}^k z^j, \quad v^k := \frac{1}{k} \sum_{j=1}^k v^j,$$

$$\bar{\sigma} := \sum_{j=1}^k (\alpha^j)^{-1}.$$
and $V^1(z, w) := \frac{1}{2\sigma} \|z - z\|^2 + \frac{1}{2\sigma} \|v^1 - w\|^2$ for all $z \in \mathbb{D}$ and $w \in \mathbb{K}^\circ$. Then $\tilde{z}^k, \tilde{w}^k \in \mathbb{D}$ and $\tilde{w}^k \in \mathbb{K}^\circ$, and
\[
d_k(H\tilde{z}^k - g) \leq \frac{V^1(z^*, w^*)}{\beta k},
\]
\[
L(\tilde{z}^k, w) - L(z, \tilde{w}^k) \leq \frac{V^1(z, w)}{k},
\]
for all $z \in \mathbb{D}$, $w \in \mathbb{K}^\circ$.

Proof See Appendix D.

Remark 3 Unlike the results in [Chambolle and Pock, 2016b], Theorem 1 and Theorem 2 prove not only the convergence of the primal-dual gap, but also the convergence of the constraints violation. In addition, if $\alpha^j \equiv \alpha$ and $\beta^j \equiv \beta$ for $j = 1, 2, \ldots, k$, then one can show that Algorithm 5 is equivalent to Algorithm 3; in other words, the results in Theorem 1 also apply to Algorithm 3.

Remark 4 When using varying step sizes, Algorithm 5 differs from Algorithm 4 in the relation between step sizes and the iteration number: the one in Algorithm 5 is explicit, whereas the one in Algorithm 4 is implicitly defined by a recursive formula [Chambolle and Pock, 2016b, Sec. 5.2]. Furthermore, we can prove the convergence rate of the constraint violation for Algorithm 5, whereas similar rate for Algorithm 4 is, to our best knowledge, does not exist in the literature.

4 Applications to constrained optimal control

We demonstrate the application of PIPG to constrained optimal control problems. In Section 4.1, we show how to formulate a typical constrained optimal control problem as an instance of conic optimization (1), and provide examples from mechanical engineering and robotics. In Section 4.2, we demonstrate the performance of PIPG via said examples, and compare it against the existing methods reviewed in Section 2.2. Throughout we let $n_x, n_u, p_s, p_v \in \mathbb{N}$ denote the dimension of different vector spaces, $\Delta \in \mathbb{R}_+$ denote a positive sampling time period, and $t \in \mathbb{N}$ denote a discrete time index.

4.1 Constrained optimal control

We consider the following linear time invariant system
\[
\frac{d}{ds} x(s) = A_c x(s) + B_c u(s) + h_c
\]
where $x : \mathbb{R}_+ \to \mathbb{R}^{n_x}$ and $u : \mathbb{R}_+ \to \mathbb{R}^{n_u}$ denote the state and input function, respectively. Matrix $A_c \in \mathbb{R}^{n_x \times n_x}$, $B_c \in \mathbb{R}^{n_x \times n_u}$, and vector $h_c \in \mathbb{R}^{n_x}$ are known parameters.

If the input changes value only at discrete time instants, then we can simplify dynamics (13) as follows. Let $\Delta \in \mathbb{N}_+$ and $x_t := x(t\Delta)$, $u_t := u(t\Delta)$ for all $t \in \mathbb{N}$. Suppose that
\[
u(s) = u(t\Delta), \quad t\Delta \leq s < (t + 1)\Delta,
\]
for all $t \in \mathbb{N}$. Then dynamics equation (13) is equivalent to the following
\[
x_{t+1} = Ax_t + Bu_t + h,
\]
for all $t \in \mathbb{N}$, where
\[
A = \exp(A_c\Delta), \quad B = \left(\int_0^\Delta \exp(A_c s)ds\right) B_c,
\]
\[
h = \left(\int_0^\Delta \exp(A_c s)ds\right) h_c.
\]
For further details on the above equivalence, we refer the interested readers to [Chen, 1999, Sec. 4.2.1].

Let \( \{x_{t+1}, u_t\}_{t=0}^{\tau-1} \) denote a length-\( \tau \) input-state trajectory of system (14) for some \( \tau \in \mathbb{N} \), and \( \{\hat{x}_{t+1}, \hat{u}_t\}_{t=0}^{\tau-1} \) denote a desired length-\( \tau \) reference input-state trajectory. A typical optimal control problem is the minimization of the difference between \( \{x_{t+1}, u_t\}_{t=0}^{\tau-1} \) and \( \{\hat{x}_{t+1}, \hat{u}_t\}_{t=0}^{\tau-1} \) subject to various constraints:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \sum_{t=0}^{\tau-1} \left( \|x_{t+1} - \hat{x}_{t+1}\|_Q^2 + \|u_t - \hat{u}_t\|_R^2 \right) \\
\text{subject to} & \quad x_{t+1} = Ax_t + Bu_t + h, \quad 0 \leq t \leq \tau - 1, \quad (16a) \\
& \quad \|u_{t+1} - u_t\|_\infty \leq \gamma, \quad 0 \leq t \leq \tau - 2, \quad (16b) \\
& \quad C_x x_t - a_t \geq 0, \quad x_t \in \mathbb{R}^n, \quad 1 \leq t \leq \tau, \quad (16c) \\
& \quad D_t u_t - b_t \geq 0, \quad u_t \in \mathbb{R}^m, \quad 0 \leq t \leq \tau - 1. \quad (16d) \\
& \quad \text{In particular, the objective function in (16a) is a quadratic distance between \( \{x_{t+1}, u_t\}_{t=0}^{\tau-1} \) and \( \{\hat{x}_{t+1}, \hat{u}_t\}_{t=0}^{\tau-1} \), where \( Q \in \mathbb{R}^{n \times n} \) and \( R \in \mathbb{R}^{m \times m} \) are given symmetric and positive definite weighting matrices. The constraints in (16b) ensure that input-state trajectory \( \{0, \ldots, x_{\tau}\} \) agree with the dynamics (14), where \( x_0 \in \mathbb{R}^n \) is the given initial state. The constraints in (16c) upper bound the elementwise difference between two consecutive inputs by \( \gamma \in \mathbb{R}_+ \), which prevents frequent and large input variations [Betts, 2010, Sec. 4.10]. The constraints in (16d) and (16e) describe possible physical and operational constraints on states and inputs, where \( C_t \in \mathbb{R}^{p_x \times n_x}, D_t \in \mathbb{R}^{p_u \times n_x}, a_t \in \mathbb{R}^{p_x}, b_t \in \mathbb{R}^{p_u}, x_t \in \mathbb{R}^n \), and \( u_t \in \mathbb{R}^m \) are closed convex sets.}

One can transform optimization (16) into a special case of optimization (1) using particular choices of the parameters. See Appendix E for the detailed transformation.

In the following, we will provide two illustrating examples of optimization (16) from mechanical engineering and robotics applications. For simplicity, all problem parameters will be unitless.

4.1.1 Oscillating masses control

We consider the problem of controlling a one-dimensional oscillating masses system using external forcing [Wang and Boyd, 2009, Kögel and Findeisen, 2011, Jerez et al., 2014]. The system consists of a sequence of \( N \) masses connected by springs to each other, and to walls on either side. Each mass has value 1, and each spring has a spring constant of 1. See Fig. 1 for an illustration.

We model the dynamics of the oscillating masses system as follows. At time \( t \Delta \), we let \( x_t = \begin{bmatrix} r_t^T & s_t^T \end{bmatrix}^T \) denote the state of the system, where the \( i \)-th element of vector \( r_t \in \mathbb{R}^N \) and \( s_t \in \mathbb{R}^N \) is the displacement and velocity of the \( i \)-th mass, respectively. Further, we let \( u_t \in \mathbb{R}^N \) denote the input to the system at time \( t \), whose \( i \)-th element is the external force exerted to the \( i \)-th mass. We let \( x_0 = 0_{2N} \) be the state of the system at time 0. Let \( L_N \in \mathbb{R}^{N \times N} \) is a symmetric tri-diagonal matrix whose diagonal entries are 2, and its sub-diagonal and super-diagonal entries are \(-1\). The discrete time dynamics of this system with sampling time period \( \Delta \) is given by (14) and (15) where

\[
A_c = \begin{bmatrix} 0_{N \times N} & I_N \\ -L_N & 0_{N \times N} \end{bmatrix}, \quad B_c = \begin{bmatrix} 0_{N \times N} \\ I_N \end{bmatrix}, \quad h_c = 0_{2N}.
\]

Fig. 1. The oscillating masses system

We consider the following constraints at each discrete time \( t \). The displacement, velocity and external force on each mass cannot exceed \([-\delta_1, \delta_1], [-\delta_2, \delta_2] \) and \([-\rho, \rho]\), respectively, where \( \delta_1, \delta_2, \rho \in \mathbb{R}_+ \). Further, for each external force, the maximum change in its magnitude within a sampling period \( \Delta \) is \( \gamma \). The aforementioned constraints are given by (16c), (16d) and (16e) where

\[
\mathbb{X} = \{r \in \mathbb{R}^N \mid \|r\|_\infty \leq \delta_1\} \times \{s \in \mathbb{R}^N \mid \|s\|_\infty \leq \delta_2\}, \quad (17)
\]

\[
\mathbb{U} = \{u \in \mathbb{R}^N \mid \|u\|_\infty \leq \rho\}.
\]

Here the conic constraints in (16d) and (16e) (i.e., \( C_t x_t - a_t \geq 0 \) and \( D_t x_t - b_t \geq 0 \)) are not considered.

4.1.2 Quadrotor path planning

We consider the problem of flying a quadrotor from its initial position to a target position while avoiding collision with cylindrical obstacles, see Fig. 3 for an illustration. For the quadrotor dynamics, we consider the 3DoF model of the Autonomous Control Laboratory (ACL) custom quadrotor [Szymk, 2019, Ch.3]; see Fig. 5 and Fig. 5 for an illustration.

Fig. 2. Autonomous Control Laboratory custom quadrotor

We model the dynamics of the quadrotor as follows. At time \( t \Delta \), the state of the quadrotor is given by \( x_t = \begin{bmatrix} \hat{r}_t^T & \hat{s}_t^T \end{bmatrix}^T \) denote
of the quadrotor's center of mass, respectively. We let $x_0$ be the state of the system at time 0. The input of the quadrotor at time $t$ is the thrust vector generated by its propellers, denoted by $u_t \in \mathbb{R}^3$. Let $m_0 = 0.35$ be the mass of the quadrotor and $g_0 = 9.8$ be the gravitational constant. The discrete time quadrotor dynamics with sampling time period $\Delta$ is given by (14) and (15) where

$$A_c = \begin{bmatrix} 0_{3 \times 3} & I_3 \\ 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix}, \quad B_c = \frac{1}{m_0} \begin{bmatrix} 0_{3 \times 3} \\ I_3 \end{bmatrix}, \quad h_c = \begin{bmatrix} 0_3 \\ -g_0 \end{bmatrix}.$$
Therefore, projections onto sets that are Cartesian products of sets with simple projection formulas, such as the set $X$ in (17) and (21), also admit simple formulas.

4.2.2 Numerical experiments

We demonstrate the numerical performance of PIPG using the two examples of optimization (16), namely the oscillating masses problem and the quadrotor path planning problem discussed in Section 4.1. We summarize the values of different problem parameters of these two examples in Appendix F.

We compare the performance of PIPG, ADMM, PIPGeq and PDHG using optimization (16) as follows. We initialize all methods using vectors whose entries are sampled from the standard normal distribution. We compare the performance of different methods using the convergence of the following two quantities:

$$\text{error}_{\text{opt}}^j := \frac{\|z^j - z^*\|^2}{\|z^*\|^2}, \quad \text{error}_{\text{fea}}^j := \frac{d_g(Hz^j - g)}{\|z^*\|^2},$$

where $z^j \in \mathbb{D}$ is the candidate solution computed of optimization (16) at the $j$-th iteration for $j = 2, 3, \ldots, k$, and $z^*$ be the ground truth optimal solution of optimization (16) computed using commercial software Mosek [MOSEK ApS, 2019]. In addition, we also consider a restarting variant of PIPG where the iteration counter $j$ is periodically reset to 1. Such restarting scheme is a popular heuristics for improving practical convergence performance of primal-dual methods [Su et al., 2016, Xu, 2017].

The convergence results of different methods in terms of $e^j_{\text{opt}}$ and $e^j_{\text{fea}}$ using 100 independent random initializations are illustrated in Fig. 5. From these results we can see that PIPG clearly outperforms existing methods, especially when combined with the restarting heuristics. Note that, although the performance of ADMM is close to PIPG in the oscillating masses example, the iteration cost of ADMM is much higher than PIPG, as shown in Tab. 1. Therefore, PIPG still has clear advantage against ADMM.

5 Conclusions

We propose a novel primal-dual first-order method for conic optimization, named PIPG. We prove the convergence rates of PIPG in terms of the constraint violation and the primal-dual gap. We demonstrate the application of PIPG using examples in constrained optimal control. However, several questions still remain open. For example, it is unclear whether our method allow real-time implementation more efficient than interior point methods, or whether there are other restarting heuristics better than the periodic one in Section 4. We aim to answer these open questions in our future work.

A Proof of Proposition 1

We will use the following results.

Lemma 2 [Rockafellar, 2015, Thm. 27.4] Let set $\mathbb{D} \subset \mathbb{R}^n$ be closed and convex and function $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and convex. If $f(z^*) \leq f(z)$ for all $z \in \mathbb{D}$, then $\langle \nabla f(z^*), z - z^* \rangle \geq 0$ for any $z \in \mathbb{D}$.

Lemma 3 [Rockafellar and Wets, 2009, Cor. 6.21] If $K \subset \mathbb{R}^m$ is a closed convex cone, then $K^o$ is a closed convex cone and $(K^o)^o = K$.

We are now ready to prove Proposition 1.

Proof First, if (12) holds, then we immediately have

$$L(z, w) \leq L(z, \overline{w}) \leq L(z, \overline{w})$$

(A.1)

for all $z \in \mathbb{D}$ and $w \in K^o$. The fist inequality above states that $-L(z, \overline{w}) \leq -L(z, w)$ for all $w \in K^o$, which, due to Lemma 2, implies that

$$\langle Hz - g, w - \overline{w} \rangle \leq 0$$

(A.2)
for all \( w \in \mathbb{K}^n \). By letting \( w = 0 \) and \( w = 2\pi \) in (A.2), we conclude that
\[
\langle H\pi - g, \overline{w} \rangle = 0. \tag{A.3}
\]
Combining (A.2) and (A.3) gives \( \langle H\pi - g, w \rangle \leq 0 \) for all \( w \in \mathbb{K}^n \). Hence \( H\pi - g \in (\mathbb{K}^n)^\circ = \mathbb{K} \), where the last step is due to Lemma 3.

Second, let \( z \) be such that \( z \in \mathbb{D} \) and \( H\pi - g \in \mathbb{K} \). Since \( \overline{w} \in \mathbb{K}^n \), using (8) we can show
\[
L(z, \overline{w}) = f(z) + \langle Hz - g, \overline{w} \rangle \leq f(z). \tag{A.4}
\]
Further, using (A.1) and (A.3) we can show
\[
f(\pi) = L(\pi, \overline{w}) \leq L(z, \overline{w}). \tag{A.5}
\]
By combining (A.4) and (A.5) we have \( f(\pi) \leq f(z) \). Since \( z \) is otherwise arbitrary except that \( z \in \mathbb{D} \) and \( H\pi - g \in \mathbb{K} \), the proof is completed.

**B Proof of Lemma 1**

We start with some basic results that are necessary for the proof later. First, using (4), one can verify the following identity:
\[
\langle \nabla f(z) - \nabla f(z'), z'' - z \rangle = B_f(z'', z') - B_f(z'', z) - B_f(z, z'), \quad \forall z, z', z'' \in \mathbb{R}^n. \tag{B.1}
\]
If \( f = ||||^2 \), the above identity becomes the following:
\[
2\langle z - z', z'' - z \rangle = |||z'' - z'|||^2 - |||z'' - z|||^2 - |||z - z'|||^2. \tag{B.2}
\]
Second, we will use Lemma 3, together with the following existing results.

**Lemma 4** \cite[2.2.7]{Nesterov2018} If set \( \mathbb{D} \subset \mathbb{R}^n \) is closed and convex, then \( \langle \pi_\mathbb{D}[z] - z, z' - \pi_\mathbb{D}[z] \rangle \geq 0 \) for any \( z \in \mathbb{R}^n \) and \( z' \in \mathbb{D} \).

**Lemma 5** \cite[Thm. 6.30]{Bauschke2017} If \( \mathbb{K} \subset \mathbb{R}^m \) is a closed convex cone, then \( \pi_\mathbb{K}[w] + \pi_\mathbb{K}^{-1}[w] = w \) for all \( w \in \mathbb{K}^m \).

We are now ready to prove Lemma 1.

**Proof** Let \( z, w, j \) be an arbitrary element in set \( \mathbb{D} \), cone \( \mathbb{K}^n \), and set \( \{1, 2, \ldots, k\} \), respectively. We start with constructing an upper bound for \( L(z^{j+1}, w) - L(z, w^{j+1}) \). To this end, first we use (11) and (4) to show the following identities
\[
L(z^{j+1}, w) - L(z, w) = B_f(z^{j+1}, z) + \langle \nabla f(z) + H^\top w, z^{j+1} - z \rangle, \tag{B.3}
\]
and
\[
L(z, w) - L(z, w^{j+1}) = \langle Hz - g, w - w^{j+1} \rangle. \tag{B.4}
\]
Second, by applying Lemma 3 to the two projections in line 2 and 3 in Algorithm 5 we can show the following two inequalities
\[
0 \leq \langle w^{j+1} - w - \beta^j(Hz^j - g), z - z^{j+1} \rangle, \tag{B.5a}
0 \leq \langle z^{j+1} - z^j + \alpha^j(\nabla f(z^j) + H^\top w^{j+1}), z - z^{j+1} \rangle. \tag{B.5b}
\]
Third, line 4 in Algorithm 5 implies the following
\[
0 = \langle v^{j+1} - w^{j+1} - \beta^j H(z^{j+1} - z^j), w - v^{j+1} \rangle. \tag{B.6}
\]
Summing up (B.3), (B.4), \( \frac{1}{\beta^j} \times (B.5a) \), \( \frac{1}{\alpha^j} \times (B.5b) \) and \( \frac{1}{\beta^j} \times (B.6) \) gives the following inequality
\[
L(z^{j+1}, w) - L(z, w^{j+1}) \\
\leq B_f(z^{j+1}, z) + \langle \nabla f(z) - \nabla f(z'), z^{j+1} - z \rangle + \frac{1}{\alpha^j} \langle z^{j+1} - z^j, z - z^{j+1} \rangle + \frac{1}{\alpha^j} \langle w^{j+1} - w^j, w - w^{j+1} \rangle + \langle v^{j+1} - w^{j+1}, H(z^{j+1} - z^j) \rangle. \tag{B.7}
\]
Our next step is to bound the inner product terms in (B.7). To this end, first we use (B.1) and (B.2) to show the following identities
\[
\langle \nabla f(z) - \nabla f(z'), z^{j+1} - z \rangle = B_f(z^{j+1}, z') - B_f(z^{j+1}, z) - B_f(z, z'), \tag{B.8}
\]
and
\[
2\langle z^{j+1} - z^j, z - z^{j+1} \rangle = |||z^{j+1} - z|||^2 - |||z^{j+1} - z^j|||^2 - |||z^j - z|||^2, \tag{B.9}
\]
and
\[
2\langle w^{j+1} - w^j, w - w^{j+1} \rangle = |||w^{j+1} - w|||^2 - |||w^{j+1} - w^j|||^2 - |||w^j - w|||^2. \tag{B.10}
\]
Second, by completing the square we can show
\[
2\beta^j \langle v^{j+1} - w^{j+1}, H(z^{j+1} - z^j) \rangle \\
\leq |||v^{j+1} - w^{j+1}|||^2 + (\beta^j)^2 \|H(z^{j+1} - z^j)\|^2. \tag{B.12}
\]
Notice that now all inner product terms in (B.7) can be upper bounded. Finally, we further simplify these upper bounds. To this end, first we use the item 1 in Assumption 1 and the fact that \( \|H\|^2 \leq \sigma \) to show the following
\[
B_f(z^{j+1}, z^j) \leq \frac{\alpha^j}{2} \|z^{j+1} - z^j\|^2, \tag{B.13a}
\]
and
\[
-B_f(z, z^j) \leq -\frac{\beta^j}{2} \|z^j - z\|^2. \tag{B.13b}
\]
\[ \|H(z^{j+1} - z^j)\| \leq \sigma \|z^{j+1} - z^j\|^2. \]  
(B.13c)

Second, we let \( y^j := \frac{1}{\beta^j} (\nu^j + \beta^j (H z^j - g) - w^{j+1}) \). Applying Lemma 3 and Lemma 5 to the projection in line 2 of Algorithm 5 we can show that \( \beta^j y^j \in (K^\circ)^\circ = K \). Since \( K \) is a cone and \( \beta^j > 0 \), we know \( y^j \in K \). Therefore, using (10) and definition of \( y^j \), we can show

\[ d_K(H z^j - g) \leq \frac{1}{\mu} \|H z^j - g - y^j\|^2 - \frac{1}{\beta^j (\sigma+\mu)} \|w^{j+1} - v^j\|^2. \]  
(B.14)

Finally, summing up (B.7), (B.8), \( \frac{1}{\alpha^2} \times (B.9), \frac{1}{\beta^j} \times (B.10), \frac{1}{\beta^j} \times (B.11), \frac{1}{\beta^j} \times (B.12), (B.13a), (B.13b), \frac{\beta^j}{\alpha^2} \times (B.13c), \) and \( \beta^j \times (B.14) \), and using the assumption that \( \alpha^j (\lambda + \sigma \beta^j) = 1 \) we obtain the desired results.

C Proof of Theorem 1

We will use the following result.

**Lemma 6** [Nesterov, 2018, Lem. 3.1.1] If function \( f : \mathbb{R}^n \to \mathbb{R} \) is convex, then

\[ f \left( \sum_{j=1}^k \gamma_j z^j \right) \leq \sum_{j=1}^k \gamma_j f(z^j) \]  
(C.1)

for any \( z^1, z^2, \ldots, z^k \in \mathbb{R}^n \) and \( \gamma^1, \gamma^2, \ldots, \gamma^k \in \mathbb{R}_+ \).

We are now ready to prove Theorem 1.

**Proof** Let \( z, w, j \) be an arbitrary element in set \( \mathbb{D}, K^\circ \) and \( \{1, 2, \ldots, k\} \), respectively. Let \( V^j(z, w) = \frac{1}{\alpha^j} \|z^j - z\|^2 + \frac{1}{\beta^j} \|w^j - w\|^2 \). Since \( \alpha^j = \frac{1}{\beta^j + \lambda} \) and \( \beta^j = \beta \), the inequality in Lemma 1 implies the following:

\[ L(z^{j+1}, w) - L(z, w^{j+1}) + \beta d_K(H z^j - g) \leq V^j(z, w) - V^{j+1}(z, w), \]

for all \( z \in \mathbb{D} \), \( w \in K^\circ \), and \( j = 1, 2, \ldots, k \). Summing up this inequality for \( j = 1, \ldots, k \) gives

\[ \sum_{j=1}^k \left( L(z^{j+1}, w) - L(z, w^{j+1}) + \beta d_K(H z^j - g) \right) \leq V^1(z, w) - V^{k+1}(z, w), \]  
(C.2)

for all \( z \in \mathbb{D} \) and \( w \in K^\circ \), where the last step is because \( V^{k+1}(z, w) \geq 0 \). From (10) and item 3 Assumption 1 we know that \( d_K(H z^j - g) \) and \( L(z^{j+1}, w^*) - L(z^j, w^{j+1}) \) are non-negative for all \( j \). Hence (C.2) implies the following

\[ \sum_{j=1}^k \left( L(z^{j+1}, w) - L(z, w^{j+1}) \right) \leq V^1(z, w), \]

\[ \beta \sum_{j=1}^k \beta d_K(H z^j - g) \leq V^1(z^*, w^*), \]

for all \( z \in \mathbb{D}, w \in K^\circ \), where the second inequality is obtained by letting \( z = z^* \) and \( w = w^* \) in (C.2).

Finally, applying the Jensen’s inequality in (6) to convex function \( L(\cdot, w), -L(z, \cdot) \), and \( d_K(\cdot) \) in the above two inequalities, respectively, we obtain the desired results.

D Proof of Theorem 2

We will use Lemma 6 in the following proof.

**Proof** Let \( z, w, j \) be an arbitrary element in set \( \mathbb{D}, K^\circ \) and \( \{1, 2, \ldots, k\} \), respectively. Let \( V^j(z, w) = \frac{1}{\alpha^j} \|z^j - z\|^2 + \frac{1}{\beta^j} \|w^j - w\|^2 \). Since \( \alpha^j = \frac{2}{(j+1) \mu + 2 \lambda} \) and \( \beta^j = \frac{(j+1) \mu}{2 \sigma} \), the inequality in Lemma 1 implies the following:

\[ L(z^{j+1}, w) - L(z, w^{j+1}) + \frac{(j+1) \mu}{2 \sigma} d_K(H z^j - g) \leq \frac{1}{(\alpha^j - \mu)} \|z^j - z\|^2 + \frac{1}{\beta^j} \|w^j - w\|^2 - V^{j+1}(z, w), \]  
(D.1)

for all \( z \in \mathbb{D}, w \in K^\circ \), and \( j = 1, 2, \ldots, k \). Let \( \kappa = \frac{\lambda}{\mu} \geq 1 \), then one can verify the following

\[ \frac{(1/\alpha^j - \mu)}{(j+2\kappa)} (j+2\kappa) = \frac{1}{\alpha^j} (j+2\kappa - 1), \]

\[ \frac{1}{\beta^j} (j+2\kappa) \leq \frac{1}{\beta^j} (j+2\kappa - 1). \]  
(D.2)

Hence multiplying (D.1) with \( (j+2\kappa) \) then substituting in (D.2) we can show

\[ (j+2\kappa) \left( L(z^{j+1}, w) - L(z, w^{j+1}) \right) + \frac{(j+1)(j+2\kappa)}{2 \sigma} d_K(H z^j - g) \leq (j+2\kappa - 1)V^j(z, w) - (j+2\kappa)V^{j+1}(z, w), \]

for all \( z \in \mathbb{D}, w \in K^\circ \), and \( j = 1, 2, \ldots, k \). Summing up this inequality for \( j = 1, 2, \ldots, k \) gives

\[ \sum_{j=1}^k (j+2\kappa) \left( L(z^{j+1}, w) - L(z, w^{j+1}) \right) + \frac{(j+1)(j+2\kappa)}{2 \sigma} d_K(H z^j - g) \leq 2\kappa V^1(z, w) - (k+2\kappa) V^{k+1}(z, w) \leq 2\kappa V^1(z, w), \]  
(D.3)

for all \( z \in \mathbb{D} \) and \( w \in K^\circ \), where the last step is because \( V^{k+1}(z, w) \geq 0 \). From (10) and item 3 Assumption 1 we know that \( d_K(H z^j - g) \) and \( L(z^{j+1}, w^*) - L(z^j, w^{j+1}) \) are non-negative for all \( j \). Hence the above inequality implies the following

\[ \sum_{j=1}^k (j+2) \left( L(z^{j+1}, w) - L(z, w^{j+1}) \right) \leq 2\kappa V^1(z, w), \]

\[ \sum_{j=1}^k \frac{(j+1)(j+2\kappa)}{2 \sigma} d_K(H z^j - g) \leq 2\kappa V^1(z^*, w^*), \]

for all \( z \in \mathbb{D} \) and \( w \in K^\circ \), where we used the fact that \( \kappa \geq 1 \), and the second inequality is obtained by letting \( z = z^* \) and \( w = w^* \) in (D.3).
Finally, applying the Jensen’s inequality in Lemma 6 to convex function \( L(\cdot, w), -L(z, \cdot), \) and \( d_{\xi}(\cdot) \) in the above two inequalities, respectively, we obtain the desired results.

### E Transformation from an optimal control problem to a conic optimization

We will use the following notation. We let \( \otimes \) denotes the Kronecker product, and \((\mathbb{D})^\tau \) denotes the Cartesian product of \( \tau \) copies of set \( \mathbb{D} \).

The optimization in (16) is a special case of (1) by letting

\[
\begin{align*}
  z &= \left[ x_1^T \ x_2^T \ \cdots \ x_\tau^T \ u_0^T \ u_1^T \ \cdots \ u_{\tau-1}^T \right]^T, \\
  f(z) &= \frac{1}{2} z^T P z + \langle p, z \rangle, \\
  P &= \begin{bmatrix} I_\tau \otimes \tau & 0 \\ 0 & I_\tau \otimes \tau \end{bmatrix}, \\
  p &= \begin{bmatrix} \tilde{x}_1^T \ \tilde{x}_2^T \ \cdots \ \tilde{x}_\tau^T \ u_0^T \ u_1^T \ \cdots \ u_{\tau-1}^T \end{bmatrix}^T, \\
  H &= \begin{bmatrix} \mathbb{A} & \mathbb{B} \\ 0 & \mathbb{E} \end{bmatrix}, \\
  K &= 0_{n_\tau} \times \mathbb{R}_{+}^{2n_{\tau}(\tau-1)+\tau(p_{\tau}+p_{n_{\tau}})}, \mathbb{D} = \mathbb{X} \times \mathbb{U},
\end{align*}
\]

where

\[
\begin{align*}
  \mathbb{A} &= I_{n_\tau} - \begin{bmatrix} 0 & 0 \\ I_{\tau-1} \otimes A & 0 \end{bmatrix}, \\
  \mathbb{B} &= -I_\tau \otimes B, \\
  \mathbb{E} &= \begin{bmatrix} C_1 & 0 & \cdots & 0 \\ 0 & C_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & C_\tau \end{bmatrix}, \\
  \mathbb{C} &= \begin{bmatrix} D_1 & 0 & \cdots & 0 \\ 0 & D_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & D_\tau \end{bmatrix}, \\
  \pi &= \begin{bmatrix} a_1^T & a_2^T & \cdots & a_\tau^T \end{bmatrix}^T, \\
  \tilde{h} &= \begin{bmatrix} b_1^T & b_2^T & \cdots & b_\tau^T \end{bmatrix}^T, \\
  \tilde{\mathbb{X}} &= (\mathbb{X})^\tau, \ \tilde{\mathbb{U}} = (\mathbb{U})^\tau.
\end{align*}
\]

### F Parameters of the optimal control problems in Section 4.2

#### Oscillating masses

We let the number of masses to be \( N = 4 \). In (16), we let \( \tau = 30 \), \( Q = I_{2N}, R = I_N \), \( \tilde{u}_t = 0_N \) and \( \tilde{x}_{t+1} = \begin{bmatrix} 1_N \ 0 \end{bmatrix}^T \) for all \( t = 0, \ldots, \tau - 1 \). We also let \( \gamma = 0.5 \) in (16c), \( \Delta = 0.25 \) in (15), and \( \delta_1 = \delta_2 = 2 = 2 \) in (17).

#### Quadrotor path planning

In (16), we let \( \tau = 30 \), \( Q = \begin{bmatrix} I_3 & 0 \\ 0 & 2.5I_3 \end{bmatrix} \), \( R = 0.5I_3, \tilde{u}_t = 0_3 \), and \( \tilde{x}_{t+1} = \begin{bmatrix} \tilde{r}_{t+1}^T \ \tilde{s}_{t+1}^T \end{bmatrix}^T \) where \( \tilde{s}_{t+1} = 0_3 \) and

\[
\tilde{r}_{t+1} = t + 1 \ \begin{bmatrix} 2.5 \\ 1.5 \end{bmatrix} + \left( 1 - \frac{t + 1}{\tau} \right) \begin{bmatrix} -1.5 \\ -0.5 \end{bmatrix},
\]

for all \( t = 0, \ldots, \tau - 1 \). We also let \( \gamma = 0.5 \) in (16c), \( \Delta = 0.25 \) in (15). We let \( \theta = \pi/4, \delta_1 = 3, \delta_2 = 5, \rho_1 = 5, \rho_2 = 2 \) in (18). For all \( t = 1, \ldots, \tau \), we let

\[
c_i^t = 2M_i^T (\tilde{r}_t - a_i^t), \quad a_i^t = \|\tilde{r}_t\|^2 + (q^t)^2 - \|o^t\|^2,
\]

for \( i = 1, 2, 3 \) in (21), where

\[
\begin{align*}
  o^1 &= \begin{bmatrix} -1.5 \\ -1.5 \end{bmatrix}, \\
  o^2 &= \begin{bmatrix} 1.2 \\ -1.2 \end{bmatrix}, \\
  o^3 &= \begin{bmatrix} 1.5 \\ 1.5 \end{bmatrix}, \\
  \rho &= 0.8, \\
  \rho^2 &= 1.2, \quad \rho^3 = 0.8, \quad M = \begin{bmatrix} I_2 & 0_{2 \times 4} \end{bmatrix},
\end{align*}
\]

and \( \tilde{r}_t \) is computed as follows. If \( \|\tilde{r}_t - o^i\| \geq \rho_i \) for all \( i = 1, 2, 3 \), then \( \hat{r}_t = \hat{r}_t \). If there exists \( i \in \{1, 2, 3\} \) such that \( \|\tilde{r}_t - o^i\| < \rho_i \), then \( \hat{r}_t = o^i + \frac{\rho_i - \rho^i}{\|\tilde{r}_t - o^i\|} (\tilde{r}_t - o^i) \).

One can verify that \( \hat{r}_t \neq o^i \) for all \( t = 1, \ldots, \tau \) and \( i = 1, 2, 3 \) and there exists at most one \( i \in \{1, 2, 3\} \) such that \( \|\hat{r}_t - o^i\| < \rho^i \). Hence the \( \tilde{r}_t \) computed in the above manner is well defined and unique.

### References

[Andersen et al., 2003] Andersen, E. D., Roos, C., and Terlaky, T. (2003). On implementing a primal-dual interior-point method for conic quadratic optimization. Math. Program., 95(2):249–277.

[Andersen et al., 2011] Andersen, M., Dahl, J., Liu, Z., Vandenberghe, L., Sra, S., Nowozin, S., and Wright, S. (2011). Interior-point methods for large-scale cone programming. Optim. Math. Learn., 5583.
direction method of multipliers. *SIAM J. Imag. Sci.*, 8(1):644–681.

[O’Connor and Vandenberghe, 2020] O’Connor, D. and Vandenberghe, L. (2020). On the equivalence of the primal-dual hybrid gradient method and douglas-rachford splitting. *Math. Program.*, 179(1):85–108.

[O’Donoghue et al., 2016] O’Donoghue, B., Chu, E., Parikh, N., and Boyd, S. (2016). Conic optimization via operator splitting and homogeneous self-dual embedding. *J. Optim. Theory Appl.*, 169(3):1042–1068.

[Rockafellar, 2015] Rockafellar, R. T. (2015). *Convex Analysis*. Princeton University Press.

[Rockafellar and Wets, 2009] Rockafellar, R. T. and Wets, R. J.-B. (2009). *Variational Analysis*, volume 317. Springer Science & Business Media.

[Stellato et al., 2020] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2020). OSQP: an operator splitting solver for quadratic programs. *Math. Program. Comput.*, 12(4):637–672.

[Su et al., 2016] Su, W., Boyd, S., and Candès, E. J. (2016). A differential equation for modeling nesterov’s accelerated gradient method: Theory and insights. *J. Mach. Learn. Res.*, 17(1):5312–5354.

[Szmuk, 2019] Szmuk, M. (2019). *Successive Convexification & High Performance Feedback Control for Agile Flight*. PhD thesis, Dept. Aeronaut. & Astronaut., Univ. Washington.

[Vũ, 2013] Vũ, B. C. (2013). A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.*, 38(3):667–681.

[Wang and Banerjee, 2014] Wang, H. and Banerjee, A. (2014). Bregman alternating direction method of multipliers. *Proc. Adv. Neural Inf. Process. Syst.*, 4(January):2816–2824.

[Wang and Elia, 2010] Wang, J. and Elia, N. (2010). Control approach to distributed optimization. In *Proc. Allerton Conf. Commun. Control Comput.*, pages 557–561. IEEE.

[Wang and Boyd, 2009] Wang, Y. and Boyd, S. (2009). Fast model predictive control using online optimization. *IEEE Trans. Control Syst. Technol.*, 18(2):267–278.

[Xu, 2017] Xu, Y. (2017). Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. *SIAM J. Optim.*, 27(3):1459–1484.

[Yan, 2018] Yan, M. (2018). A new primal-dual algorithm for minimizing the sum of three functions with a linear operator. *J. Sci. Comput.*, 76(3):1698–1717.

[Yu and Açıkmese, 2020] Yu, Y. and Açıkmese, B. (2020). RLC circuits-based distributed mirror descent method. *IEEE Control Syst. Lett.*, 4(3):548–553.

[Yu et al., 2020a] Yu, Y., Açıkmese, B., and Mesbahi, M. (2020a). Mass–spring–damper networks for distributed optimization in non-Euclidean spaces. *Automatica*, 112:108703.

[Yu et al., 2020b] Yu, Y., Elango, P., and Açıkmese, B. (2020b). Proportional-integral projected gradient method for model predictive control. *IEEE Control Syst. Lett.*

[Zagaris et al., 2018] Zagaris, C., Park, H., Virgili-Llop, J., Zappulla, R., Romanò, M., and Kolmanovsky, I. (2018). Model predictive control of spacecraft relative motion with convexified keep-out-zone constraints. *J. Guid. Control Dyn.*, 41(9):2054–2062.