**Proportional-integral Projected Gradient Method for Model Predictive Control**

Yue Yu, Purnanand Elango and Behçet Açıkmese

**Abstract**—Recently there has been an increasing interest in primal-dual methods for model predictive control (MPC), which require minimizing the (augmented) Lagrangian at each iteration. We propose a novel first order primal-dual method, termed proportional-integral projected gradient method, for MPC where the underlying finite horizon optimal control problem has both state and input constraints. Instead of minimizing (the augmented) Lagrangian, each iteration of our method only computes a single projection onto the state and input constraint set. We prove that our method ensures convergence to optimal solutions at $O(1/k)$ and $O(1/k^2)$ rate if the objective function is convex and, respectively, strongly convex. We demonstrate our method via a trajectory-planning example with convexified keep-out-zone constraints.

I. INTRODUCTION

Model predictive control (MPC) provides a systematic approach for automatic control with physical and operational constraints [1], [2], [3], [4], [5]. The key ingredient in MPC is solving a finite horizon discrete-time convex optimal control problem that can be expressed in the following form

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2}z^T H z + h^T z \\
\text{subject to} & \quad G z = g, \quad z \in \mathbb{Z},
\end{align*}
\]

(1)

where the trajectory variable $z$ aims to minimize a convex quadratic cost function $\frac{1}{2}z^T H z + h^T z$ subject to linear dynamics constraints $G z = g$ together with convex state and input constraint $z \in \mathbb{Z}$. Throughout we assume $\mathbb{Z}$ is the Cartesian product of convex sets whose Euclidean projection can be evaluated at low computational cost. Such assumption applies to many popular state and input constraints used in MPC; see Tab. I for some examples and [6] for a detailed discussion. In addition to convex MPC problems, solution to problem (1) is also an integral part of problems with nonlinear dynamics and non-convex constraints. In these cases, a sequence of convex sub-problems modeled by (1) are solved to obtain the solution of the original non-convex problem, as done in sequential convex programming [7], [8] and successive convexification methods [9], [10], [11].

Recently there has been an increasing interest in first order primal-dual methods for MPC. Such methods solve problem (1) together with its dual problem by updating both primal and dual variables at each iteration. For example, the dual fast gradient method first updates the primal variables by optimizing the Lagrangian, then the dual variables using Nesterov’s method [12], [13], [14]. Similarly, the Chambolle & Pock method first updates the primal variables by optimizing the augmented Lagrangian, then the dual variables using extrapolation [15]. The alternating directional method of multipliers (ADMM) first updates two copies of the primal variables by optimizing the augmented Lagrangian: one subject to $G z = g$, the other subject to $z \in \mathbb{Z}$. The dual variables then simply integrate the difference between the two copies [16], [17], [18], [19]. Compared with second order methods [20] and first order primal methods [21], first order primal-dual methods allow both efficient per-iteration computation and general state and input constraints.

The common challenge in implementing the aforementioned primal-dual methods is to optimize the (augmented) Lagrangian during each iteration. In general, such optimization requires either inner loop iterations that costs multiple projections onto set $\mathbb{Z}$ [22], or solving linear equation systems. The latter requires either Ricatti recursion [23], [18] or pre-computing matrix inverse/decomposition [14], [16], [17].

Recently, [24] tried to address this challenge by proposing a primal-dual projected gradient method. Instead of minimizing the (augmented) Lagrangian, each iteration of this method only computes a single projection onto set $\mathbb{Z}$. Unfortunately, [24] only considers strongly convex objective functions, and does not provide any explicit convergence rate.

We propose a novel first order primal-dual method, termed proportional-integral projected gradient method, for MPC. Our method strictly improves the one in [24] because 1) it uses the same per-iteration computation, namely computing a single projection onto set $\mathbb{Z}$ rather than minimizing the (augmented) Lagrangian, 2) achieves explicit convergence rate of $O(1/k)$ and $O(1/k^2)$ when the objective function is convex and, respectively, strongly convex, and 3) significantly outperforms existing methods in numerical experiments.

The rest of the paper is organized as follows. Section II & Pock method first updates the primal variables by optimizing the augmented Lagrangian, then the dual variables using extrapolation [15]. The alternating directional method of multipliers (ADMM) first updates two copies of the primal variables by optimizing the augmented Lagrangian: one subject to $G z = g$, the other subject to $z \in \mathbb{Z}$. The dual variables then simply integrate the difference between the two copies [16], [17], [18], [19]. Compared with second order methods [20] and first order primal methods [21], first order primal-dual methods allow both efficient per-iteration computation and general state and input constraints.

The common challenge in implementing the aforementioned primal-dual methods is to optimize the (augmented) Lagrangian during each iteration. In general, such optimization requires either inner loop iterations that costs multiple projections onto set $\mathbb{Z}$ [22], or solving linear equation systems. The latter requires either Ricatti recursion [23], [18] or pre-computing matrix inverse/decomposition [14], [16], [17].

Recently, [24] tried to address this challenge by proposing a primal-dual projected gradient method. Instead of minimizing the (augmented) Lagrangian, each iteration of this method only computes a single projection onto set $\mathbb{Z}$. Unfortunately, [24] only considers strongly convex objective functions, and does not provide any explicit convergence rate.

We propose a novel first order primal-dual method, termed proportional-integral projected gradient method, for MPC. Our method strictly improves the one in [24] because 1) it uses the same per-iteration computation, namely computing a single projection onto set $\mathbb{Z}$ rather than minimizing the (augmented) Lagrangian, 2) achieves explicit convergence rate of $O(1/k)$ and $O(1/k^2)$ when the objective function is convex and, respectively, strongly convex, and 3) significantly outperforms existing methods in numerical experiments.

The rest of the paper is organized as follows. Section II

---

**TABLE I**: Examples of simple convex sets and projections

| Set $X$                                      | Projection of $x$ onto $X$ if $x \notin X$                                      |
|----------------------------------------------|---------------------------------------------------------------------------------|
| $\{x | ||x||_2 \leq \alpha \}$                | $\frac{1}{||x||_2^2} x$                                                        |
| $\{x | \alpha \leq x \leq u \}$              | $\min(\max|\{x, \alpha, u\})$                                                 |
| $\{x | (a, x) \leq \alpha, a \neq 0 \}$      | $x - ((a, x) - \alpha) \frac{a}{\alpha}$                                      |
| $\{x = (y, \alpha) | ||y||_2 \leq \alpha \}$ | $(0, 0)$ if $||y||_2 \leq -\alpha$;                                           |
| $\{x = (y, \alpha) | ||y||_2 \leq \alpha \}$ | $\frac{1}{2\alpha} (y, \alpha)$ otherwise                                    |
| $\{x, f(x) \} | f(y) \leq \alpha \}$                   | $x$ solves $y \in x + (f(x) - \alpha)\partial f(x)$                           |
| $\{x | f(x) \leq \alpha \}$                  | $(I + \mu f'(x)^-)^{-1}(x)$ where                                               |
|                                              | $\mu$ solves $f((I + \mu f'(x)^-)^{-1}(x)) = \alpha$                          |

Here function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous, convex and finite valued; $\partial f$ denotes the subdifferential of $f$, max/min is evaluated element-wise.
reviews several existing first order primal-dual methods for MPC. Section III introduces our method together with its convergence guarantee. Section IV discusses the implementation details of our method, which is demonstrated via a trajectory-planning example in Section V. Finally, Section VI concludes and comments on future directions.

Notation: Let \( z, y \in \mathbb{R}^n \) and \( H, P \in \mathbb{R}^{n \times n} \) with \( H = H^\top, P = P^\top \). We denote \( \langle z, y \rangle = z^\top y, \|z\|_2 = \sqrt{z^\top z} \), \( |z|_H = \sqrt{z^\top H z} \), and \( H < (z) P \) if and only if \( P - H \) is positive (semi)-definite. The Euclidean projection onto a closed convex set \( Z \subseteq \mathbb{R}^n \) is denoted by \( \pi_Z : \mathbb{R}^n \to Z \) where \( \pi_Z[z] = \arg\min_{z \in Z} \|z' - z\|_2 \).

II. RELATED WORK

In this section, we briefly review some existing first order primal-dual methods for MPC. In the following, we let \( k \) and \( j \) denote iteration counters and \( \alpha \) denote the step size.

A. Dual fast gradient method

Assuming matrix \( H \) is positive definite, dual fast gradient method [12], [13], [14] solves problem (1) as follows

\[
\begin{align*}
    z^{k+1} & = \arg\min_{z \in \mathbb{R}^n} \frac{1}{2} z^\top H z + h^\top z + \langle v^k, G z \rangle, \\
    w^{k+1} & = v^k + \alpha (G z^{k+1} - g), \\
    v^{k+1} & = w^{k+1} + \frac{\alpha}{\mu + \alpha} (w^{k+1} - w^k), \tag{2a-c}
\end{align*}
\]

The idea is to apply Nesterov’s method [25, Sec. 2.2] to the dual problem of (1). In general, the minimization step in (2a) can only be solved approximately using another inner loop of Nesterov’s method [22], which iterates as follows [25, Sec. 2.2] (\( j \) denotes the inner loop iteration counter)

\[
\begin{align*}
    z^{j+1} & = \pi_Z[y^j - \alpha (H y^j + h + G^\top v^j)], \\
    y^{j+1} & = z^{j+1} + \frac{\alpha}{\mu + \alpha} (z^{j+1} - z^j), \tag{3}
\end{align*}
\]

where \( 0 < \mu I \preceq H \preceq \lambda I \).

B. Chambolle & Pock method

The Chambolle & Pock method [26] solves problem (1) using the following iterations [15]

\[
\begin{align*}
    z^{k+1} & = \arg\min_{z \in \mathbb{R}^n} \frac{1}{2} z^\top H z + h^\top z + \frac{1}{2\alpha} \|z + \alpha G^\top w^k - z^k\|_2^2, \tag{4a} \\
    w^{k+1} & = w^k + \alpha (2z^{k+1} - z^k - g). \tag{4b}
\end{align*}
\]

Unlike method (2), this method does not require matrix \( H \) to be positive definite. However, the minimization step in (4a) is just as challenging as the one in (2a), and can only be solved approximately via Nesterov’s method as follows (again, \( j \) denotes the inner loop iteration counter)

\[
\begin{align*}
    z^{j+1} & = \pi_Z[y^j - \frac{1}{\lambda} (H y^j + h + G^\top v^j)], \\
    y^{j+1} & = z^{j+1} + \frac{\sqrt{\lambda} - \sqrt{\mu}}{\sqrt{\lambda} + \sqrt{\mu}} (z^{j+1} - z^j), \tag{5}
\end{align*}
\]

where \( 0 < \mu I \preceq H \preceq \frac{1}{\lambda} I \leq \lambda I \).

C. ADMM

One of the most popular methods for problem (1) is ADMM [16], [17], [18], [19], which iterates as follows

\[
\begin{align*}
    y^{k+1} & = \arg\min_{z:Hz=g} \frac{1}{2} z^\top H z + h^\top z + \frac{1}{2\alpha} \|z + w^{k} - y^{k}\|_2^2, \tag{6a} \\
    z^{k+1} & = \pi_Z[y^{k+1} + w^k], \\
    w^{k+1} & = w^k + z^{k+1} - y^{k+1}. \tag{6b-c}
\end{align*}
\]

Notice that ADMM solves two subproblems for primal variables: minimization of a quadratic function over a hyperplane in (6a) and the projection in (6b). The minimization in (6a) is equivalent to solving the following system of linear equations for variable \( z \)

\[
\begin{pmatrix}
    H + \frac{1}{\alpha} I & G^\top \\
    G & 0
\end{pmatrix}
\begin{pmatrix}
    z \\
    v
\end{pmatrix}
= \begin{pmatrix}
    -h - \frac{1}{\alpha} (w^k - y^k)
\end{pmatrix}, \tag{7}
\]

which requires pre-computing either matrix inverse [17] or LDL decomposition [16]. If both matrix \( H \) and \( G \) are time invariant, such pre-computation only needs to be executed once. However, for time varying applications, e.g., those from nonlinear MPC [27], such precomputation needs to be executed every time matrix \( H \) or \( G \) is updated.

D. Matrix splitting method

Inspired by matrix splitting methods, Blanchard & Adegbe [24] proposed the following primal-dual projected gradient method

\[
\begin{align*}
    z^{k+1} & = \pi_Z[z^k - \alpha(H z^k + h + G^\top w^{k})], \\
    w^{k+1} & = w^k + \alpha (G z^{k+1} - g). \tag{8a-b}
\end{align*}
\]

Unlike method (2), (4) and (6), each iteration in (8), termed proportional-integral method, computes the projection onto set \( Z \) only once, rather than multiple times as in (3) or (5), and does not require solving any linear equations as in (7). Unfortunately, the results in [24] only shows asymptotic convergence when matrix \( H \) is positive definite, without any explicit convergence rate guarantee.

III. PROPORTIONAL-INTEGRAL PROJECTED GRADIENT METHOD

In this section, we introduce our main contribution, an improvement to method (5), termed proportional-integral projected gradient method. We will show that our method achieves explicit \( O(1/k) \) and \( O(1/k^2) \) convergence rate when matrix \( H \) is positive semi-definite and, respectively, positive definite.

The method we propose iterates as follows

\[
\begin{align*}
    v^{k} & = w^k + \beta k (G z^k - g), \tag{9a} \\
    z^{k+1} & = \pi_Z[z^k - \alpha k (H z^k + h + G^\top v^{k})], \\
    w^{k+1} & = w^k + \beta (G z^{k+1} - g). \tag{9b-c}
\end{align*}
\]

Compared with method (5), the main difference here is an additional step in (9a). This seemingly minor modification, however, leads to explicit \( O(1/k) \) and \( O(1/k^2) \) convergence rate.
rate when matrix $H$ is positive semi-definite and, respectively, positive definite. The latter is optimal for solving problem $\text{(1)}$ using first order methods [28, Thm.1.1].

**Remark 1.** Notice that (9a) and (9c) compute a proportional and, respectively, integral feedback of the affine constraints violation. Hence an intuitive interpretation of (9) is applying projected gradient method to variable $z$, where the gradient is corrected by a proportional-integral (PI) feedback. Similar PI feedback was also used in distributed optimization algorithms [29], [30].

We now prove the convergence properties of method (9). First, we group our assumptions as follows.

**Assumption 1.** Suppose
1) set $Z \subset \mathbb{R}^n$ is closed and convex; matrix $H \in \mathbb{R}^{n \times n}$ is symmetric, matrix $G \in \mathbb{R}^{m \times n}$ has full row rank, there exists $0 \leq \mu \leq \lambda$ and $\sigma \geq 0$ such that $\mu I \leq H \leq \lambda I$ and $G^T G \leq \sigma I$.
2) there exists $z^* \in \mathbb{R}^n$ and $w^* \in \mathbb{R}^m$ such that
$$G z^* = g, \quad z^* \in Z,$$
$$\langle H z^* + h + G^T w^*, z - z^* \rangle \geq 0, \quad \forall z \in Z.$$ (10a)

**Remark 2.** Equation (10a) gives the KarushKuhnTucker conditions of problem $\text{(1)}$. Under the Slater condition for equalities, equation (10a) holds if and only if $z^*$ is an optimal solution for problem $\text{(1)}$; see [25, Thm.3.1.27].

We will use the following result on Euclidean projection.

**Lemma 1.** [25, Lemma. 2.2.7] If set $Z \subset \mathbb{R}^n$ is closed and convex, then
$$\langle \pi_Z[z] - z, z' - \pi_Z[z] \rangle \geq 0, \quad \forall z \in \mathbb{R}^n, z' \in Z.$$ (10b)

The following lemma shows the key property of two consecutive iterations generated by method (9).

**Lemma 2.** Suppose Assumption 1 holds and sequence $\{z^k, w^k\}$ is generated by (9). If $\lambda + \sigma \beta^k = \frac{1}{\alpha^k}$ for all $k \geq 1$, then
$$\frac{\beta^k}{2} \|G z^k - g\|_2^2 + \frac{1}{2} \|z^{k+1} - z^*\|_H^2 \leq \frac{1}{2} (\frac{\lambda}{\alpha^k} - \mu) \|z^k - z^*\|_2^2 + \frac{1}{2 \sigma} \|w^k - w^*\|_2^2$$
$$- \frac{1}{2 \alpha^k} \|z^{k+1} - z^*\|_2^2 - \frac{1}{2 \sigma} \|w^{k+1} - w^*\|_2^2. (11)$$

**Proof.** First, applying Lemma 1 to (9b) gives
$$0 \leq \frac{\beta^k}{2} \|z^k - z^*\|_2^2 + \frac{1}{2} \|z^{k+1} - z^*\|_H^2$$
$$+ \langle H z^k + h, z^k - z^{k+1} \rangle + \langle w^k, G(z^k - z^{k+1}) \rangle, (11)$$
where we also used (9a) and (10a). Next, (10b) implies that
$$0 \leq - \langle H z^k + h, z^k - z^{k+1} \rangle - \langle w^k, G(z^k - z^{k+1}) \rangle. (12)$$
In addition, one can directly verify the following four identities, which can be interpreted as instances of the law of cosines; see Fig. 1 for an illustration.
$$\langle z^{k+1} - z^k, z^* - z^{k+1} \rangle$$
$$- \frac{1}{2} \|z^k - z^*\|_2^2 - \frac{1}{2} \|z^{k+1} - z^*\|_2^2 - \frac{1}{2} \|z^{k+1} - z^{k+1}\|_2^2, (13)$$
$$\|G z^k - g, g - G z^{k+1}\|$$
$$= \frac{1}{2} \|G(z^{k+1} - z^k)\|_2^2 - \frac{1}{2} \|G z^k - g\|_2^2 - \frac{1}{2} \|G z^{k+1} - g\|_2^2, (14)$$
$$\langle H \hat{z} \hat{z}^T (z^k - z^*) \rangle, H \hat{z} \hat{z}^T (z^* - z^{k+1}) \rangle$$
$$= \frac{1}{2} \|z^{k+1} - z^k\|_H^2 - \frac{1}{2} \|z^k - z^*\|_H^2 - \frac{1}{2} \|z^{k+1} - z^*\|_H^2, (15)$$
$$\frac{1}{2} \|w^{k+1} - w^*\|_2^2 - \frac{1}{2} \|w^k - w^*\|_2^2$$
$$= \langle w^k - w^*, w^{k+1} - w^* \rangle + \frac{1}{2} \|w^{k+1} - w^k\|_2^2$$
$$= \beta^k \langle w^k - w^*, G(z^{k+1} - z^*) \rangle + \frac{(\sigma \beta^k)^2}{2} \|G z^{k+1} - g\|_2^2, (16)$$
where matrix $H \hat{z}$ in (15) is the positive semi-definite square root of $H$, and the last step in (16) is due to (9c) and (10a).

Further, the assumption that $0 \leq \mu I \leq H \leq \lambda I$ and $G^T G \leq \sigma I$ gives
$$\frac{\beta^k}{2} \|z^k - z^*\|_2^2 \leq \frac{1}{2} \|z^k - z^*\|_H^2, (17a)$$
$$\frac{1}{2} \|z^{k+1} - z^k\|_H^2 \leq \frac{1}{2} \|z^{k+1} - z^*\|_2^2, (17b)$$
$$\frac{1}{2} \|G(z^{k+1} - z^k)\|_2^2 \leq \frac{1}{2} \|z^{k+1} - z^*\|_2^2. (17c)$$
Finally, summing up together (11), (12), $\frac{1}{\alpha^k} \times 13$, $\beta^k \times 14$, \[13\], $\frac{1}{\alpha^k} \times 16$, (17a), (17b) and $\beta^k \times (17c)$, then using the assumption that $\lambda + \sigma \beta^k = \frac{1}{\alpha^k}$, we obtain the desired result.

We start with the case where $\mu = 0$, i.e., matrix $H$ is only positive semi-definite and the objective function in problem $\text{(1)}$ is only convex. The following theorem shows that, using constant step sizes, the iterations in (9) converge to optimum at the rate of $O(1/k)$.

**Theorem 1.** Suppose Assumption 1 hold with $\mu = 0$, and sequence $\{v^k, z^k, w^k\}$ is generated by (9) with $\alpha^k = \frac{1}{\beta^k + \lambda}$ and $\beta^k = \beta$ for some $\beta > 0$ and all $k \geq 1$. Let $V^1 = \frac{1}{\lambda^2} \|z^1 - z^*\|_2^2 + \frac{1}{\sigma^2} \|w^1 - w^*\|_2^2$, then
$$\frac{1}{2} \|G z^k - g\|_2^2 \leq \frac{1}{\lambda^2} \|V^1\|, \quad \frac{1}{2} \|z^k - z^*\|_H^2 \leq \frac{1}{\sigma^2} \|V^1\|, (18)$$
where $z^1 = \frac{1}{k} \sum_{j=1}^k z^j$ and $z^k = \frac{1}{k} \sum_{j=1}^k z^j + 1$.

**Proof.** With this choice of $\alpha^k$ and $\beta^k$, the inequality in Lemma 2 becomes the following: for all $j \geq 1$,
$$\frac{\beta^k}{2} \|G z^j - g\|_2^2 + \frac{1}{2} \|z^j - z^*\|_H^2 \leq V^j - V^{j+1}, \quad \text{where} \quad V^j = \frac{1}{\lambda^2} \|z^j - z^*\|_2^2 + \frac{1}{\sigma^2} \|w^j - w^*\|_2^2.$$

Summing up this inequality for $j = 1, \ldots, k$ gives
$$\sum_{j=1}^k \left( \frac{\beta^k}{2} \|G z^j - g\|_2^2 + \frac{1}{2} \|z^j - z^*\|_H^2 \right) \leq V^1 - V^{k+1} \leq V^1.$$
where the last step is because $V^{k+1} \geq 0$. Hence
\[
\frac{\beta}{\alpha} \sum_{j=1}^{k} \left\| Gz^j - g^j \right\|_2^2 \leq V^1, \quad \frac{1}{2} \sum_{j=1}^{k} \left\| z^{j+1} - z^* \right\|_H^2 \leq V^1,
\]
Finally, applying Jensen’s inequality to the above two inequalities gives the desired result.

If $\mu > 0$, i.e., matrix $H$ is positive definite and the objective function in problem (11) is strongly convex, the dual fast gradient method in (12) achieves convergence rate of $O(1/k^2)$ for problem (11), which is optimal [28, Thm.1.1]. This optimal rate can be matched by method (9) using varying step sizes, as shown by the following theorem.

**Theorem 2.** Suppose Assumption 1 hold with $\mu > 0$, and sequence $\{v^k, z^k, w^k\}$ is generated by (9) with $\alpha_k = \frac{(k+1)\mu}{2(\mu+\lambda)}$, $\beta_k = \frac{(k+1)\mu}{2\mu}$ for all $k \geq 1$. Let $V^1 = \frac{1}{2(\mu+\lambda)} \left\| z^1 - z^* \right\|_2^2 + \frac{\mu}{2\mu} \left\| w^1 - w^* \right\|_2^2$, then
\[
\frac{\beta}{\alpha} \sum_{j=1}^{k} \left\| Gz^j - g^j \right\|_2^2 \leq \frac{12\lambda \sigma}{\mu(k^2+6k+11)} V^1, \quad \frac{1}{2} \left\| \tilde{z} - z^* \right\|_H^2 \leq \frac{4\lambda}{\mu(k^5+3)} V^1,
\]
where $\tilde{z}^k = \frac{3}{k(k+\lambda)} \sum_{j=1}^{k} (j+1)(j+2)z^j$ and $\tilde{z} = \frac{2}{k(k+\lambda)} \sum_{j=1}^{k} (j+2)z^j$.

**Proof.** With this choice of $\alpha_k$ and $\beta_k$, the inequality in Lemma 2 becomes: for all $j \geq 1$,
\[
\frac{(j+1)\mu}{4\sigma} \left\| Gz^j - g^j \right\|_2^2 + \frac{\mu}{2\sigma} \left\| z^{j+1} - z^* \right\|_2^2 \leq \frac{1}{(\frac{1}{\sigma} - \mu)} \left\| z^j - z^* \right\|_2^2 + \frac{1}{2\sigma} \left\| w^j - w^* \right\|_2^2 + V^2,
\]
where $V^2 = \frac{1}{2\sigma} \left\| z^j - z^* \right\|_2^2 + \frac{1}{2\sigma} \left\| w^j - w^* \right\|_2^2$. Let $\kappa = \frac{\lambda}{\mu} \geq 1$, then it is straightforward to verify the following
\[
\frac{(j+1)\mu}{4\sigma} \left\| Gz^j - g^j \right\|_2^2 + \frac{\mu}{2\sigma} \left\| z^{j+1} - z^* \right\|_2^2 \leq (j + 2\kappa - 1)V^2 - (j + 2\kappa)V^2.
\]
Summing up this inequality for $j = 1, 2, \ldots, k$ gives
\[
\sum_{j=1}^{k} \left\{ \frac{(j+1)(j+2)\mu}{4\sigma} \left\| Gz^j - g^j \right\|_2^2 + \frac{\mu}{2\sigma} \left\| z^{j+1} - z^* \right\|_2^2 \right\} \leq 2\kappa V^1 - (k + 2\kappa)V^1.
\]
where the last inequality is because $V^{k+1} \geq 0$. Since $\kappa \geq 1$, the above inequality implies the following
\[
\sum_{j=1}^{k} \left\{ \frac{(j+1)(j+2)\mu}{4\sigma} \left\| Gz^j - g^j \right\|_2^2 + \frac{\mu}{2\sigma} \left\| z^{j+1} - z^* \right\|_2^2 \right\} \leq 2\kappa V^1, \quad \sum_{j=1}^{k} \frac{\mu}{2\sigma} \left\| z^{j+1} - z^* \right\|_2^2 \leq 2\kappa V^1.
\]
Finally, applying Jensen’s inequality to the above two inequalities and using $\kappa = \frac{\lambda}{\mu}$ gives the desired result.

**Remark 3.** Compared with Theorem 7, Theorem 2 used an averaged sequence with increasing weights, similar to those in subgradient method [31] and accelerated ADMM [32]. Notice that Theorem 2 shows that the constraint violation converges at a $O(1/k^3)$ rate, even faster than $O(1/k^2)$, which is highly desirable in practice as constraint violation is often used as the stopping criterion.

**IV. EFFICIENT IMPLEMENTATION FOR MPC**

In this section, we provide the pseudocode implementation of method (9) for the following tracking problem
\[
\begin{align*}
&\text{minimize} & & \frac{1}{2} \sum_{t=1}^{T} \left\| x_t - y_t \right\|_2^2 + \frac{1}{2} \sum_{t=0}^{T-1} \left\| u_t \right\|_H^2, \\
&\text{subject to} & & x_t = A_t x_{t-1} + B_t u_{t-1}, \\
& & & u_{t-1} \in U_{t-1}, \quad x_t \in X_t, \quad 1 \leq t \leq T.
\end{align*}
\]
where, for all $1 \leq t \leq T$: closed convex sets $X_t \subset \mathbb{R}^{n_x}$ and $U_{t-1} \subset \mathbb{R}^{n_u}$ describe feasible sets for state variable $x_t$ and, respectively, input variable $u_{t-1}$; $A_t \in \mathbb{R}^{n_x \times n_x}$, $B_t \in \mathbb{R}^{n_u \times n_u}$ describe the linear dynamics of the plant; $y_t$ gives the reference value for $x_t$.

We first rewrite problem (20) as a special case of problem (1) by defining the following
\[
\begin{align*}
&z = [u_0^T, x_1^T, \ldots, u_{T-1}^T, x_T^T], & Z = \prod_{t=1}^{T} (U_{t-1} \times X_t), \\
&H = \text{blkdiag} (R_0, Q_1, \ldots, R_{T-1}, Q_T), & H = \begin{bmatrix}
\begin{array}{cccc}
-Br_0 & I \\
& -A_1 & B_1 & I \\
& & & \ddots & \ddots \\
& & & & -A_{T-1} & -B_{T-1} & I
\end{array}
\end{bmatrix},
\end{align*}
\]
\[
g = [x_0^T, A_0^T, 0^T, \ldots, 0^T]^T.
\]

We are now ready to implement (9) for problem (20). We partition variables $w$ and $v$ as follows
\[
v = [v_1^T, v_2^T, \ldots, v_T^T], & w = [w_1^T, w_2^T, \ldots, w_T^T],
\]
where $v_t, w_t \in \mathbb{R}^{n_x}$ corresponds to constraint $x_t = A_t x_{t-1} + B_t u_{t-1}$ for $1 \leq t \leq T$. In addition, the separable structure of set $Z$ defined by (21) allows separable computation of its Euclidean projection. Based on these observations, we implement algorithm (9) for problem (20) in Algorithm 1 where we introduce dummy parameters $A_T v_{T-1} = 0$ to simplify our notation. Notice that updates of variables corresponding to different value of $t$ can be executed in parallel, hence the algorithm run-time can be almost independent of horizon $T$.

**V. NUMERICAL EXAMPLES**

In this section we compare our method against the existing methods reviewed in Section IV over a trajectory-planning problem with keep-out-zone constraints, where all parameters are chosen as unit-less for simplicity.

We consider a trajectory-planning (finite horizon optimal control) problem where a 2D planner is trying to track a beeline trajectory from initial to target position while avoiding collision with a circular keep-out-zone; see Fig. 2 for an illustration. Here this problem is an instant of an MPC problem, which is solved repetitively as new state information becomes available. The dynamics of the planner
Algorithm 1 Pl projected gradient method

Input: $x_0, X_t, U_{t-1}, Q_t, y_t, R_{t-1}, A_{t-1}, B_{t-1}$ for all $1 \leq t \leq T$. Initialize $k = 1, u_{t-1}, x_t, w_t$ for all $1 \leq t \leq T$; let $A_{TV}T+1 := 0$.

while $k \leq k_{\text{max}}$ do

    $k \leftarrow k + 1$

    For all $1 \leq t \leq T$:
    
    $v_t \leftarrow w_t + \beta^k (x_t - A_{t-1} x_{t-1} - B_{t-1} u_{t-1})$
    $u_{t-1} \leftarrow \pi_{U_{t-1}} [u_{t-1} - \alpha^k (R_{t-1} u_{t-1} - B_{t-1} v_t)]$
    $x_t \leftarrow \pi_{X_t} [x_t - \alpha^k (Q_t (x_t - y_t) + v_t - A_{t}^T v_{t+1})]$
    $w_t \leftarrow v_t + \beta^k (x_t - A_{t-1} x_{t-1} - B_{t-1} u_{t-1})$

end while

Output: $\{u_0, x_1, \ldots, u_{T-1}, x_T\}$

---

Fig. 2: Trajectory-planning with rotating halfspace constraint.

is modeled as a double integrator with sampling time 0.5 s. The planner is subject to $\ell_2$ norm constraints on its velocity $q \in \mathbb{R}^2$ and acceleration input $u \in \mathbb{R}^2$. In addition, a rotating half-space constraint is imposed on its position $p \in \mathbb{R}^2$, which convexifies the keep-out-zone; see [33] for a detailed discussion. We model this tracking problem as a special case of problem (20) with the following choice of parameters:

\[
A_{t-1} = \begin{bmatrix}
1 & 0 & 0.5 & 0 \\
0 & 1 & 0 & 0.5 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix},
B_{t-1} = \begin{bmatrix}
0.125 & 0 & 0 \\
0 & 0.125 & 0 \\
0.5 & 0 & 0 \\
0 & 0.5 & 0
\end{bmatrix},
\]

\[
Q_t = \text{diag}(1, 0.5, 1, 0.5),
R_{t-1} = \text{diag}(1, 0.5),
\]

\[
X_t = \left\{ x = \begin{bmatrix} p \\ q \end{bmatrix} \mid \begin{bmatrix} -\cos(\theta t) \\ \sin(\theta t) \end{bmatrix} \right\} \quad p \geq 2, \|q\|_2 \leq 0.25,
\]

\[
U_{t-1} = \{ u \|u\|_2 \leq 0.1 \}, \quad x_0 = [-2.5 \quad 0.6 \quad 0 \quad 0],
\]

for $1 \leq t \leq T$, where $\theta = 0.063$ in (23c) is a constant rotating rate [33]. Note that $Q_t$ and $R_t$ in (23b) are diagonal but not identity, which is common in practice. The reference trajectory $y_t^{T}_{t=1}$ in (20) is chosen as a beeline trajectory from initial position $(-2.5, 0.6)$ to target position $(2.9, 0.3)$ without considering the position constraint on $p$ in (23c).

We compare our method against all the other methods reviewed in Section III In terms of step size: for our method (9), we choose $\alpha^k$ and $\beta^k$ according to Theorem 1 and Theorem 2 for constant and, respectively, varying step sizes; for dual fast gradient method, we choose $\alpha$ according to [12, Thm.1]; for Cahrmbolle & Pock method, we choose $\alpha$ according to [15, Eqn. (9)]; for ADMM, we choose $\alpha = 2$ as suggested in [17]; for matrix splitting method, we choose $\alpha$ according to [24, Prop.2]. In addition, the inner loop iteration used by each iteration of method 2 and 4 are warm-started using results from the last outer iteration and terminated if $\|z^{t+1} - z^t\|_2 / \|z^t\|_2 \leq \epsilon_{\text{inner}}$, where $\epsilon_{\text{inner}}$ is chosen between 0.1% and 0.01%.

We summarize our results as follows. Fig. 3 shows the convergence over iterations of different algorithms with same initialization for $T = 25$, where $z^*$ is computed using ECOS [34] together with JuMP [35]. Fig. 4 shows the computation costed by different algorithms for $T = \{5, 15, 25, 35, 45\}$ to reach the tolerance for constraint violation (we use $\ell_{p^*}$ norm since it measures the maximum pointwise constraint violation along the trajectory), where each data point is averaged over 200 independent experiments using initialization sampled from standard normal distribution. Note that we omitted method (4), (8) and (9) with constant step sizes in Fig. 4 due to their slow convergence.

In these simulations, our method with varying step sizes (var.) significantly outperforms the others, especially for large scale problems with high accuracy requirement. Our method with constant step sizes (const.) converges slower— but still marginally outperforms method (4) and (8)—since it is designed for non-strongly convex objective functions.

VI. CONCLUSION

We introduced a novel first order primal-dual method for MPC, which uses a single projection onto the state and input constraint set per-iteration, and achieves $O(1/k)$ and $O(1/k^2)$ convergence rate when the objective function is convex and, respectively, strongly convex. Our method also outperforms existing methods in numerical experiments. Future directions include real-time implementation and faster empirical convergence using preconditioning.
Fig. 4: Number of projection $\pi_{Z}[\cdot]$ costed to reach condition $\|GZ - g\|_2 \leq \epsilon$. Each data point is averaged over 200 simulations using random initialization.

REFERENCES

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” Automatica, vol. 36, no. 6, pp. 789–814, 2000.

[2] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” Automatica, vol. 50, no. 12, pp. 2967–2986, 2014.

[3] J. S. Qin and T. A. Badgwell, “A survey of industrial model predictive control technology,” Control Eng. Pract., vol. 11, no. 7, pp. 733–764, 2003.

[4] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodríguez, “Model predictive control simple and powerful method to control power converters,” IEEE Trans. Ind. Electron., vol. 56, no. 6, pp. 1826–1838, 2008.

[5] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe, “Model predictive control in aerospace systems: Current state and opportunities,” J. Guid. Control Dyn., vol. 40, no. 7, pp. 1541–1566, 2017.

[6] H. H. Bauschke, “Projection algorithms and monotone operators,” Ph.D. dissertation, Theses (Dept. of Mathematics and Statistics)/Simon Fraser University, 1996.

[7] R. Ghaemi, J. Sun, and I. V. Kolmanovsky, “An integrated perturbation analysis and sequential quadratic programming approach for model predictive control,” Automatica, vol. 45, no. 10, pp. 2412–2418, 2009.

[8] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, “Gusto: Guaranteed sequential trajectory optimization via sequential convex programming,” in Int. Conf. Robot. Autom. IEEE, 2019, pp. 6741–6747.

[9] Y. Mao, M. Szmuk, and B. Açıkmeşe, “Successive convexification of non-convex optimal control problems and its convergence properties,” in IEEE Conf. Decision Control. IEEE, 2016, pp. 3636–3641.

[10] Y. Mao, D. Duari, M. Szmuk, and B. Açıkmeşe, “Successive convexification of non-convex optimal control problems with state constraints,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 4063–4069, 2017.

[11] Y. Mao, M. Szmuk, X. Xu, and B. Açıkmeşe, “Successive convexification: A superfineenly convergent algorithm for non-convex optimal control problems,” arXiv preprint arXiv:1804.06559[math.OC], 2018.

[12] S. Richter, C. N. Jones, and M. Morari, “Certification aspects of the fast gradient method for solving the dual of parametric convex programs,” Math. Meth. Oper. Res., vol. 77, no. 3, pp. 305–321, 2013.

[13] P. Patrinos and A. Bemporad, “An accelerated dual gradient-projection algorithm for embedded linear model predictive control,” IEEE Trans. Automat. Control, vol. 59, no. 1, pp. 18–33, 2013.

[14] P. Giselsson, “Improved fast dual gradient methods for embedded model predictive control,” IFAC Proc. Vol., vol. 47, no. 3, pp. 2303–2309, 2014.

[15] D. K. M. Kufoalor, S. Richter, L. Insland, T. A. Johansen, M. Morari, and G. O. Eikrem, “Embedded model predictive control on a PLC using a primal-dual first-order method for a subsea separation process,” in Proc. Mediterranean Conf. Control Autom. IEEE, 2014, pp. 368–373.

[16] B. O’Donoghue, G. Stathopoulos, and S. Boyd, “A splitting method for optimal control,” IEEE Trans. Control Syst. Technol., vol. 21, no. 6, pp. 2342–2347, 2013.

[17] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, “Embedded online optimization for model predictive control at megahertz rates,” IEEE Trans. Automat. Control, vol. 59, no. 12, pp. 3238–3251, 2014.

[18] L. E. Sokoler, G. Frison, M. S. Andersen, and J. B. Jørgensen, “Input-constrained model predictive control via the alternating direction method of multipliers,” in Proc. Eur. Control Conf. IEEE, 2014, pp. 115–120.

[19] T. V. Dang, K. V. Ling, and J. M. Maciejowski, “Embedded ADMM-based QP solver for MPC with polytopic constraints,” in Proc. Eur. Control Conf. IEEE, 2015, pp. 3446–3451.

[20] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” IEEE Trans. Control Syst. Technol., vol. 18, no. 2, pp. 267–278, 2009.

[21] S. Richter, C. N. Jones, and M. Morari, “Computational complexity certification for real-time MPC with input constraints based on the fast gradient method,” IEEE Trans. Automat. Control, vol. 57, no. 6, pp. 1391–1403, 2011.

[22] Y. Mao, M. Szmuk, and B. Açıkmeşe, “Successive convexification of non-convex optimal control problems and its convergence properties,” in IEEE Conf. Decision Control and Eur. Control Conf. IEEE, 2011, pp. 501–506.

[23] P. Patrinos, P. Sopasakis, and H. Sarimveis, “A global piecewise smooth Newton method for fast large-scale model predictive control,” Automatica, vol. 47, no. 9, pp. 1906–1922, 2011.

[24] H. A. Blanchard and A. A. Adeckegbe, “An SOR-like method for fast model predictive control,” IFAC-PapersOnLine, vol. 50, no. 1, pp. 14418–14423, 2017.

[25] Y. Nesterov, Lectures on convex optimization. Springer, 2010, vol. 137.

[26] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” J. Math. Imaging Vis., vol. 40, no. 1, pp. 120–145, 2011.

[27] D. Kuzoumpis, G. Frison, A. Zanelli, and M. Diehl, “Recent advances in quadratic programming algorithms for nonlinear model predictive control,” Vietnam J. Math., vol. 46, no. 4, pp. 863–882, 2018.

[28] Y. Ouyang and Y. Xu, “Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems,” Math. Program., pp. 1–35, 2019.

[29] Y. Yu, B. Açıkmeşe, and M. Mesbahi, “Mass–spring–damper networks for distributed optimization in non-Euclidean spaces,” Automatica, vol. 112, pp. 108703, 2020.

[30] Y. Yu and B. Açıkmeşe, “RLC circuits-based distributed mirror descent method,” IEEE Control Syst. Lett., vol. 4, no. 3, pp. 548–553, 2020.

[31] S. Lacoste-Julien, M. Schmidt, and F. Bach, “A simpler approach to obtaining an $O(1/t)$ convergence rate for the projected stochastic subgradient method,” arXiv preprint arXiv:1212.2002[cs.LG], 2012.

[32] Y. Xu, “Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming,” SIAM J. Optim., vol. 27, no. 3, pp. 1459–1484, 2017.

[33] C. Zagaris, H. Park, J. Virgili-Llop, R. Zappulla, M. Romano, and I. Kolmanovsky, “Model predictive control of spacecraft relative motion with convexified keep-out-zone constraints,” Journal of Guidance, Control, and Dynamics, vol. 41, no. 9, pp. 2054–2062, 2018.

[34] A. Domahidi, E. Chu, and S. Boyd, “ECOS: An SOCP solver for embedded systems,” in Proc. Eur. Control Conf. IEEE, 2013, pp. 3071–3076.

[35] I. Dunning, J. Huchette, and M. Lubin, “JuMP: A modeling language for mathematical optimization,” SIAM Review, vol. 59, no. 2, pp. 295–320, 2017.