A Dynamical Systems Perspective to Convergence Rate Analysis of Proximal Algorithms

Mahyar Fazlyab, Alejandro Ribeiro, Manfred Morari, and Victor M. Preciado

Abstract—In this paper, we develop a semidefinite programming (SDP) framework for convergence rate analysis of proximal algorithms designed to solve convex composite optimization problems. We first represent these algorithms as linear dynamical systems interconnected with a nonlinear component. We then propose a family of time-varying nonquadratic Lyapunov functions that are particularly useful for establishing arbitrary (exponential or subexponential) convergence rates. Using Integral Quadratic Constraints (IQCs) to describe the class of allowable nonlinearities in the interconnection, we derive sufficient conditions for Lyapunov stability of proximal algorithms in terms of Linear Matrix Inequalities (LMIs). We show how the developed LMI-based framework can be used to establish convergence rates by specializing it to the proximal gradient method and its accelerated variant for both convex and strongly convex problems.

Index Terms—Composite Convex Optimization, Proximal Algorithms, Nesterov’s Accelerated Method, Semidefinite Programming, Integral Quadratic Constraints.

I. INTRODUCTION

Optimization problems in which the objective function can be decomposed into smooth and nonsmooth components arise in several applications, such as signal and image processing [1], statistical inference [2], and feedback control [3], [4]. For this class of optimization problems, proximal algorithms—including Douglas-Rachford Splitting and ADMM as special cases [5], [6]—are commonly used. Intuitively, proximal algorithms can be viewed as a generalization of gradient methods to constrained and nonsmooth problems. However, the convergence analysis of gradient methods are not easily transferable to their proximal variants unless a careful correspondence is made between the properties of their gradient mappings [7]. Further, the convergence analysis of these algorithms are often pursued on a case-by-case basis. The goal of the present work is to develop a semidefinite programming (SDP) framework to unify the convergence analysis of proximal algorithms.

Recently, there has been a growing interest in analyzing the performance of optimization algorithms by semidefinite programming [8]–[10]. This idea was first introduced by Drori and Teboulle [8], in which the authors formulated the worst-case performance of an optimization algorithm as an SDP, which they call performance estimation problem. Specifically, the authors focus on the gradient method and its accelerated variant for smooth unconstrained convex minimization of convex objectives. This approach has been extended to other algorithms and problem classes, such as strongly convex problems [9] and proximal algorithms [11]. The formulation of performance estimation problem, despite being able to yield new performance bounds, is highly algorithm dependent.

In an attempt to depart from classical algorithmic view, Lessard et. al [12] have recently adapted the notion of Integral Quadratic Constraints from robust control theory to develop an SDP framework for stability analysis of first-order optimization methods in an algorithm-independent manner. They make this connection by viewing first-order methods as linear dynamical systems in feedback connection with a nonlinear component that can be modeled by IQCs. More specifically, they formulate a small SDP to certify exponential convergence rates when the objective function is strongly convex. Variants of this idea for nonstrongly convex problems have been reported in [13], [14], which, respectively, make use of dissipativity theory and nonquadratic Lyapunov functions to capture subexponential convergence rates. Lyapunov functions generalize the notion of energy in dynamical systems and have recently gained more attention in the context of optimization algorithms [15]–[18].

In this paper, we develop an LMI framework able to certify both exponential and subexponential convergence rates for proximal algorithms designed to solve convex composite optimization problems. Our starting point is to represent these algorithms as linear dynamical systems interconnected with a nonlinear component (Section II). For stability analysis, we construct a nonquadratic Lyapunov function that allows us to establish arbitrary convergence rates (Section III-B). By bounding the drift of the Lyapunov function from one iteration to the next, we derive an LMI whose feasibility guarantees convergence at a specified rate (Theorem 1). Therefore, the task of verifying a convergence rate reduces to solving an LMI problem. While the framework in [12] focuses on numerical verification of exponential convergence rates for proximal algorithms, our work puts more emphasis on analytical convergence rates for both exponential and subexponential convergence rates. This requires developing new IQCs for the gradient mapping of proximal algorithms. We illustrate the utility of the developed framework by analyzing the proximal gradient method and its accelerated variant, for both convex and strongly convex objective functions (Section IV).

Notation and Preliminaries. We denote the set of real numbers by $\mathbb{R}$, the set of real $n$-dimensional vectors by $\mathbb{R}^n$, the set of $m \times n$-dimensional matrices by $\mathbb{R}^{m \times n}$, and the
n-dimensional identity matrix by $I_n$. We denote by $\mathbb{S}_+^n$, $\mathbb{S}_+^{m+n}$, and $\mathbb{S}_+^{n+m}$ the sets of $n$-by-$n$ symmetric, positive semidefinite, and positive definite matrices, respectively. For $P \in \mathbb{S}_+^{n+m}$, we denote the condition number of $P$ by $\text{cond}(P)$, which is defined as the ratio between the minimum and maximum eigenvalue of $P$. For $M \in \mathbb{R}^{n \times n}$ and $x \in \mathbb{R}^n$, we have that $x^\top M x = \frac{1}{2} x^\top (M + M^\top) x$. The Euclidean norm is displayed by $\| \cdot \|_2 : \mathbb{R}^n \to \mathbb{R}_+$. Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a closed proper function. The effective domain of $f$ is denoted by $\text{dom}(f) = \{ x \in \mathbb{R}^n : f(x) < \infty \}$. The indicator function $\mathbb{I}_X : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ of a closed nonempty convex set $X \subset \mathbb{R}^n$ is defined as $\mathbb{I}_X(x) = 0$ if $x \in X$, and $\mathbb{I}_X(x) = +\infty$ otherwise. A differentiable function is (strongly) convex if and only if it satisfies

$$\nabla f(x)^\top(x - y) + \frac{m_f}{2} \|y - x\|^2_2 + f(x) \leq f(y),$$

(1)

for all $x, y \in \text{dom}(f)$. The parameter $m_f \geq 0$ quantifies the minimum curvature of $f$. A differentiable $f$ whose gradient is Lipschitz continuous with parameter $0 \leq L_f < \infty$ satisfies

$$f(y) \leq f(x) + \nabla f(x)^\top(y - x) + \frac{L_f}{2} \|y - x\|^2_2,$$

(2)

for all $x, y \in \text{dom}(f)$. The parameter $L_f$ quantifies the maximum curvature of $f$. We denote by $\mathcal{F}(m_f, L_f)$ the class of functionals satisfying both (1) and (2). Note that in this class, it must hold that $m_f \leq L_f$. We define the condition number of $f \in \mathcal{F}(m_f, L_f)$ as $\kappa_f = L_f / m_f$ whenever $k_f < \infty$. The case $L_f = \infty$ corresponds to nondifferentiable convex functions. For $g \in \mathcal{F}(0, \infty)$, we denote $\partial g$ as the subdifferential of $g$, which is defined as

$$\partial g(x) = \{ \gamma \in \text{dom}(g) : g(x) + \gamma^\top(y - x) \leq g(y) \}. \quad (3)$$

II. Proximal Algorithms as Dynamical Systems

Proximal algorithms are concerned with solving optimization problems of the form

$$X_* = \arg \min_{x \in \mathbb{R}^d} \{ F(x) = f(x) + g(x) \}, \quad (4)$$

where $x \in \mathbb{R}^d$ is the vector of decision variables, $f : \mathbb{R}^d \to \mathbb{R}$ is closed, proper, convex and differentiable, while $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is closed, proper, convex, and nondifferentiable. We assume that the optimal solution set $X_*$ is nonempty and closed, and the optimal value $F(x_*)$ is finite. Under these assumptions, the optimality condition for (4) is given by

$$x_* \in \text{dom}(F) \subseteq \text{dom}(f) \cap \text{dom}(g). \quad (5)$$

where $\text{dom}(F) = \text{dom}(f) \cap \text{dom}(g)$. Depending on the choice of $f$ and $g$, (4) describes various specialized optimization problems. For instance, the case $g(x) \equiv 0$ corresponds to unconstrained smooth programming; the case $g(x) = \mathbb{I}_X(x)$, where $\mathbb{I}_X(x)$ is the indicator function of a nonempty, closed, convex set $X \subseteq \mathbb{R}^d$, is equivalent to constrained smooth programming; finally, the case $f(x) \equiv 0$ describes nonsmooth programming.

Formally, the objective function in (4) is nonsmooth and subgradient methods exhibit extremely slow convergence [19]. Splitting methods such as proximal algorithms circumvent this issue by exploiting the special structure of the objective function to achieve convergence rates comparable to their counterparts in smooth programming. In what follows, we introduce proximal algorithms by adopting a dynamical systems perspective.

A. State-space Formulation

Consider an iterative first-order algorithm that generates a sequence of points $\{ x_k \}_{k \geq 0}$ satisfying $\lim_{k \to \infty} F(x_k) = F(x_*)$. We then consider the following state-space representation for the iterations of the algorithm:

$$\begin{align*}
\xi_{k+1} &= A_k \xi_k + B_k u_k, \\
y_k &= C_k \xi_k, \\
u_k &= \phi(y_k), \\
x_k &= E \xi_k,
\end{align*}$$

(6)

where at each iteration index $k \geq 0$, $\xi_k \in \mathbb{R}^n$ is the state vector ($n \geq d$), $u_k \in \mathbb{R}^d$ is the input vector, and $y_k \in \mathbb{R}^d$ is the output vector that is transformed by the nonlinear map $\phi : \mathbb{R}^d \to \mathbb{R}^d$ to generate the feedback input $u_k$; see Fig. 1 for a block diagram representation. Note that the matrices $(A_k, B_k, C_k)$ can change with $k$, representing an algorithm with possibly iteration-dependent parameters.

First-order algorithms can often be represented in the state-space form (6), where the matrices $(A_k, B_k, C_k, E)$ differ for each algorithm. The nonlinear map $\phi$ consists in the first-order oracle of the objective function $F$. For instance, for unconstrained smooth programming (i.e., when $g(x) \equiv 0$ in (4)), we have that $\phi = \nabla f$ [12]. In this paper, we are interested in proximal algorithms for which the nonlinearity $\phi$ is the generalized gradient mapping. We introduce this mapping next.

B. Generalized Gradient Mapping

Let $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ be a closed proper convex function. The proximal operator $\Pi_{g, \alpha} : \mathbb{R}^d \to \mathbb{R}^d$ of $g$ with parameter $\alpha > 0$ is defined as

$$\Pi_{g, \alpha}(x) = \text{argmin}_{y \in \mathbb{R}^d} \{ g(y) + \frac{1}{2\alpha} \|y - x\|^2_2 \}. \quad (7)$$

When $g(x) \equiv 0$, we have that $\Pi_{g, \alpha}(x) = x$. Further, when $g(x) = \mathbb{I}_X(x)$, we obtain $\Pi_{g, \alpha}(x) = \text{argmin}_{y \in X} \|y - x\|_2$, i.e., $\Pi_{g, \alpha}(x)$ is the projection of $x$ onto $X$. In general, the proximal operator is practically useful only when the strongly convex problem in the right-hand side of (7) is easy to solve. Examples of such cases are $g(x) = \|x\|_1$ and $g(x) = \mathbb{I}_X(x)$.
where $\mathcal{X}$ is a box constraint; see [20] or [21] for more information about proximal operators.

For the composite function $F$ in (4), the generalized gradient mapping $\phi: \mathbb{R}^d \to \mathbb{R}^d$ is defined as
\[
\phi(x) = \frac{1}{\alpha} (x - \Pi_{g,\alpha}(x - \alpha \nabla f(x))), \quad \alpha > 0, \tag{8}
\]
with $\text{dom}(\phi) = \text{dom}(F)$. Notice that when $g(x) \equiv 0$ (so that $\Pi_{g,\alpha}(x) = x$), the generalized gradient mapping simplifies to the gradient function $\nabla f$. In the following, we represent few proximal algorithms in the state-space form (6).

**Example 1 (Proximal Gradient Method)** The proximal gradient method is amongst the simplest proximal algorithms, which is defined by the recursion
\[
x_{k+1} = \Pi_{g,\alpha k}(x_k - \alpha_k \nabla f(x_k)). \tag{9}
\]
Using the definition (8) of the generalized gradient mapping, we can equivalently write this recursion as
\[
x_{k+1} = x_k - \alpha_k \phi(y_k), \tag{10}
\]
where $\phi(x) = g(x) - x$ for all $x \in \mathbb{R}$. For this algorithm, the matrices $(A_k, B_k, C_k)$ are given by
\[
\begin{bmatrix}
A_k & B_k \\
C_k & 0
\end{bmatrix} = \begin{bmatrix}
I_d & -\alpha_k I_d \\
-I_d & 0
\end{bmatrix}. \tag{11}
\]

**Example 2 (Accelerated Proximal Gradient Method)** The accelerated proximal gradient method is defined by
\[
x_{k+1} = \Pi_{g,\alpha_k}(y_k - \alpha_k \nabla f(y_k)), \tag{12}
\]
yielding $y_k = x_k + \beta_k (x_k - x_{k-1})$, where the parameter $\beta_k$ obeys the following update rule [22],
\[
\beta_k = \frac{(t_{k-1} - 1)}{t_k}, \quad t_k = \frac{1}{2} \left( 1 + \sqrt{1 + 4 t_{k-1}^2} \right), \tag{13}
\]
with $t_{-1} = 0$. For future reference, note that $t_k$ satisfies the relationships $t_k^2 - \frac{1}{4} t_{k-1}^2 = 0$ and $t_{k-1} \geq (k+1)/2$ for $k \geq 1$. Using (8), we can rewrite (12) as
\[
x_{k+1} = x_k + \beta_k (x_k - x_{k-1}) - \alpha_k \phi(y_k), \tag{14}
\]
yielding $y_k = x_k + \beta_k (x_k - x_{k-1})$. By defining the state vector $\xi_k = [x_k^T, x_{k+1}^T]^T \in \mathbb{R}^{2d}$, we can represent (14) in state-space form as follows,
\[
\begin{bmatrix}
0 & I_d \\
-\beta_k I_d & (\beta_k + 1) I_d
\end{bmatrix} \xi_k + \begin{bmatrix}
0 \\
-\alpha_k I_d
\end{bmatrix} u_k, \tag{15}
\]
\[
y_k = -\beta_k I_d \begin{bmatrix}
\beta_k + 1 \\
I_d
\end{bmatrix} \xi_k,
\]
\[
u_k = \phi(y_k).
\]
Therefore, the matrices $(A_k, B_k, C_k)$ are given by
\[
\begin{bmatrix}
A_k & B_k \\
C_k & 0
\end{bmatrix} = \begin{bmatrix}
0 & I_d \\
-\beta_k I_d & (\beta_k + 1) I_d
\end{bmatrix} \begin{bmatrix}
-\alpha_k I_d \\
0
\end{bmatrix}. \tag{16}
\]

Notice that when $\beta_k \equiv 0$, the accelerated algorithm in Example 2 reduces to the classical proximal gradient method as in Example 1. In the rest of the paper, we consider the general representation (6).

Since the dynamical system (6) solves the optimization problem asymptotically, its fixed points must coincide with the optimal solutions. Moreover, it can be easily verified that the first-order optimality condition in (3) can be equivalently represented as $\phi(x_*) = 0$. Therefore, the fixed points of (6) satisfy
\[
\xi_* = A_k \xi_*, \quad x_* = y_* = C_k \xi_* \text{ for all } k \geq 0. \tag{17}
\]

**III. LMI-based Convergence Analysis**

In control theory, absolute stability theory guarantees stability of linear dynamical systems in feedback interconnection with a memoryless and possibly time-varying nonlinearity [23]. There are various criteria and approaches to perform stability analysis of these interconnections, such as the circle criterion—which includes the positivity and small gain theorems as special cases—and the Popov criterion. A distinguishing feature of these approaches is the allowable class of nonlinearities they focus on. In this regard, Integral Quadratic Constraints, originally proposed by Megretski and Rantzer [24], is a powerful tool for describing various classes of nonlinearities, and are particularly useful for LMI-based stability analysis. In what follows, we briefly discuss pointwise IQCs, which was introduced by Lessard et al. [12] in the context of optimization theory.

**A. Pointwise IQCs**

Consider a mapping $\phi: \mathbb{R}^d \to \mathbb{R}^d$, and a “reference” input-output pair $(x_*, \phi(x_*))$. We say that $\phi$ satisfies the pointwise IQC defined by $(Q_\phi, x_*, \phi(x_*))$ on $\mathcal{S} \subseteq \mathbb{R}^d$ if the following inequality holds:
\[
\begin{bmatrix}
x - x_* \\
\phi(x) - \phi(x_*)
\end{bmatrix}^T Q_\phi \begin{bmatrix}
x - x_* \\
\phi(x) - \phi(x_*)
\end{bmatrix} \geq 0, \tag{18}
\]
where $Q_\phi \in \mathbb{S}^{2d}$ is a symmetric, indefinite matrix. The functional properties of $\phi$ can often be represented as constraints of the form (18). For instance, suppose $\phi(x)$ is $L_\phi$-Lipschitz continuous on $\mathcal{S} \subseteq \mathbb{R}^d$ for some positive and finite $L_\phi$, i.e., $\|\phi(x) - \phi(x_*)\|_2 \leq L_\phi \|x - x_*\|_2$, for all $(x, x_*) \in \mathcal{S} \times \mathcal{S}$. By squaring both sides and rearranging the terms, we obtain the following quadratic constraint,
\[
\begin{bmatrix}
x - x_* \\
\phi(x) - \phi(x_*)
\end{bmatrix}^T \begin{bmatrix}
L_\phi^2 I_d & 0 \\
0 & -I_d
\end{bmatrix} \begin{bmatrix}
x - x_* \\
\phi(x) - \phi(x_*)
\end{bmatrix} \geq 0, \tag{19}
\]
which represents Lipschitz continuity. Consider the feedback interconnection in Fig. 1. The basic idea behind IQC is to remove the nonlinear component $\phi$ but still enforce the quadratic constraint (18) on its input-output pairs. Lessard et al. [12] adapt the theory of IQCs for use in optimization algorithms. Specifically, they translate the first-order definitive properties of convex functions into various forms of IQCs for their gradient mappings. For proximal algorithms, the nonlinearity of interest is the generalized gradient mapping, defined as in (8). In the following proposition, we characterize this map using IQCs.
Proposition 1 (IQC for generalized gradient mapping)
Consider the composite function $F = f + g$, where $f \in \mathcal{F}(m_f, L_f)$ and $g \in \mathcal{F}(0, \infty)$. Then, the generalized gradient mapping, defined in (8), satisfies the pointwise IQC defined by $(Q_\phi, x, \phi(x))$, where $Q_\phi$ is given by

$$Q_\phi = \left[ \begin{array}{cc} \frac{1}{2\alpha} (\gamma_f^2 - 1) & \frac{1}{2} \\ \frac{1}{2} & -\alpha \end{array} \right] \otimes I_d, \quad (20)$$

with $\gamma_f = \max \{ \| 1 - \alpha L_f \|, \| 1 - \alpha m_f \| \}$.

The proof of Proposition 1 can be found in [13]. Notice that when $g(x) \equiv 0$, we have that $\phi = \nabla f$, according to (8). In this case, the IQC of Proposition 1 simply describes the gradient function $\nabla f$.

We remark that in [12], the authors use a different block diagonal representation of proximal algorithms, in which the linear component is in parallel feedback connections with diagonal representation of proximal algorithms, in which the quantifies the suboptimality of $g$ as a possibly time-varying symmetric matrix. The first term in (18) is important to remark that there is no restriction on the second term.

Consider the composite function $F = f + g$, where $f \in \mathcal{F}(m_f, L_f)$ and $g \in \mathcal{F}(0, \infty)$. Correspondingly, define the generalized mapping $\phi$ as in (8). Then, the following inequality holds for all $x, y \in \text{dom}(F)$,

$$F(y - \alpha \phi(y)) - F(x) \leq \phi(y)^T (y - x) - \frac{m_f}{2} \| y - x \|^2_2 + \frac{1}{2} L_f \alpha^2 - \alpha \| \phi(y) \|^2_2. \quad (24)$$

Lemma 1 bounds the difference between the objective values at arbitrary points $x$ and $y - \alpha \phi(y)$. The bound is obtained by combining the assumptions about the objective components, namely, convexity of $g$, (strong) convexity of $f$, and Lipschitz continuity of $\nabla f$. The proof is referred to the Appendix. With the technical setting clarified, we state our main result in the following Theorem.

Theorem 1 (Main Result) Consider the dynamical system (6) for solving the optimization problem in (4), where $\phi$ is defined as in (8), $f \in \mathcal{F}(m_f, L_f)$ and $g \in \mathcal{F}(0, \infty)$. Assume there exists a nonnegative and nonincreasing sequence $\{a_k\}_{k \geq 0}$ satisfying the following LMI,

$$(a_{k+1} - a_k) M_k^{(0)} + a_k M_k^{(1)} + M_k^{(2)} + \sigma_k M_k^{(3)} \leq 0, \quad (25a)$$

with

$$M_k^{(0)} = \frac{1}{2} \begin{bmatrix} -m_f C_k^T C_k & C_k^T \sigma f \alpha^2 - 2\alpha I_d \end{bmatrix}, \quad (25b)$$
$$M_k^{(1)} = \frac{1}{2} \begin{bmatrix} -m_f (C_k - E)^T (C_k - E) & (C_k - E)^T (L_f \alpha^2 - 2\alpha I_d) \end{bmatrix}, \quad (25c)$$
$$M_k^{(2)} = \begin{bmatrix} A_k^T P_{k+1} A_k & A_k^T P_{k+1} B_k \\
B_k^T P_{k+1} A_k & B_k^T P_{k+1} B_k \end{bmatrix}, \quad (25d)$$
$$M_k^{(3)} = \begin{bmatrix} C_k & 0 \\
0 & I_d \end{bmatrix} \otimes Q_\phi \begin{bmatrix} C_k & 0 \\
0 & I_d \end{bmatrix}, \quad (25e)$$

where $\sigma_k \geq 0$, $P_k \in S^d_+$ for all $k \geq 0$. Then, the sequence $\{x_k\}_{k \geq 0}$ satisfies (23).

Theorem 1 states that if we can find a triple $(a_k, P_k, \sigma_k)$ that satisfy the LMI in (25), we can certify an $O(1/a_k)$ convergence rate for the algorithm. Therefore, the LMI (25) prescribes a general recipe for convergence rate analysis of all proximal algorithms that can be cast in the form (6). It is important to remark that there is no restriction on the sequence $\{a_k\}_{k \geq 0}$ other than nonnegativity and monotonicity. Hence, we can characterize both exponential convergence rates ($a_k = \rho^{-k}, 0 \leq \rho < 1$), as well as subexponential ($a_k = k^p, p > 0$, for example) convergence rates.

C. Strongly Convex Problems

In the definition of the Lyapunov function in (22), we may have a certain degree of freedom in choosing the parameters $a_k$ and $P_k$, which is further determined by the assumptions we make about the objective function. For example, in strongly convex problems, we expect exponential convergence, both in terms of the function value (i.e., $F(x_k) - F(x_*)$), as well as the distance to the optimal solution (i.e., $\| x_k - x_* \|$). In this case, we can precondition $a_k$ and $P_k$ in (21) to simplify the LMI in (25). Explicitly,
by setting $a_k = \rho^{-2k}a_0$, $a_0 > 0$, $F_k = \rho^{-2k}P_0$, $0 < \rho < 1$, and $P_0 \in S^n_{++}$ in (21), the Lyapunov function reads as
\[ V_k(\xi) = \rho^{-2k} \left( a_0(F(x) - F(x_0)) + (\xi - \xi_*)^T P_0(\xi - \xi_*) \right). \]

With this choice, satisfaction of condition (22) together with strong convexity admit the bound
\[ \frac{m_f}{2} \|x_k - x_\ast\|^2 \leq F(x_k) - F(x_\ast) \leq \frac{V_0(\xi_0)}{a_0} \rho^{2k}, \tag{26a} \]
where the left and right inequalities follows from strong convexity of $F$ and (23), respectively. Therefore, by this particular selection, exponential convergence is established both in terms of function value, as well as distance to the optimal point. On the other hand, by selecting $a_k \equiv 0$, $P_k = \rho^{-2k}P_0$, $0 < \rho < 1$, and $P_0 \in S^n_{++}$, the condition in (22) admits the inequality
\[ \|\xi_k - \xi_\ast\|^2 \leq \text{cond}(P_0)\|\xi_0 - \xi_\ast\|^2 \rho^{2k}, \tag{26b} \]
where \text{cond}(P_0) denotes the condition number of $P_0$. Finally, by selecting $F_k \equiv 0$ and $a_k = \rho^{-2k}a_0$, $a_0 > 0$, the condition in (22) yields the inequality
\[ F(x_k) - F(x_\ast) \leq (F(x_0) - F(x_\ast)) \rho^{2k}. \tag{26c} \]

In summary, depending on the selection of $a_k$ and $P_k$, we may obtain three different performance bounds, as given in (26a)-(26c). In the next section, we will use Theorem 1 to recover several convergence results in the literature.

IV. Symbolic Convergence Rates

A. Classical Proximal Gradient Method

For the classical gradient method, the matrices $(A_k, B_k, C_k)$ are given by (11). By selecting $P_k = p_k I_d$, $p_k \geq 0$, we obtain the following LMI, according to (25),
\[ (a_k + 1 - a_k)M_k^{(1)} + a_k M_k^{(1)} + M_k^{(2)} + \sigma_k M_k^{(3)} \leq 0. \tag{27a} \]

The matrices $M_k^{(i)}$, $i = 0, 1, 2, 3$ are given by
\[ M_k^{(0)} = \frac{1}{2} \begin{bmatrix} -m_f & 1 \\ 1 & (L_f \alpha^2 - 2\alpha) \end{bmatrix} \otimes I_d, \tag{27b} \]
\[ M_k^{(1)} = \frac{1}{2} \begin{bmatrix} 0 & 0 \\ 0 & (L_f \alpha^2 - 2\alpha) \end{bmatrix} \otimes I_d, \tag{27c} \]
\[ M_k^{(2)} = \begin{bmatrix} p_k - p_k - \alpha p_{k+1} \\ -\alpha p_{k+1} \end{bmatrix} \otimes I_d, \tag{27d} \]
\[ M_k^{(3)} = \frac{1}{2\alpha} \begin{bmatrix} \gamma_f^2 - 1 & \frac{1}{2} \alpha \\ \frac{1}{2} \alpha & \frac{1}{2} \end{bmatrix} \otimes I_d, \tag{27e} \]
where $\gamma_f = \max\{|1 - \alpha L_f|, |1 - \alpha m_f|\}$. We remark that the size of the LMI (27) is $2 \times 2$, independent of the problem dimension $d$.

Strongly Convex Case. We first consider the selection $a_k \equiv 0$ for strongly convex settings. Then the LMI (27) simplifies to
\[ \begin{bmatrix} p_k - p_k - \alpha p_{k+1} \\ -\alpha p_{k+1} \end{bmatrix} + \sigma_k \begin{bmatrix} \gamma_f^2 - 1 & \frac{1}{2} \alpha \\ \frac{1}{2} \alpha & \frac{1}{2} \end{bmatrix} \leq 0. \]

It can be verified that the above LMI ensures that $\sigma_k/(2\alpha) \leq p_k/\gamma_f^2$ and $p_{k+1} - p_k \leq \sigma_k/(1 - \gamma_f^2)/(2\alpha)$. These two conditions together imply that $p_{k+1} \leq \rho p_k/\gamma_f^2$. Or, in other words, $p_k = \gamma_f^{-2k}p_0$, $p_0 > 0$. Using the bound (26b) and recalling the expression for $\gamma_f$ in Proposition 1, we can establish that
\[ \|x_k - x_\ast\|^2 \leq (\max\{|1 - \alpha L_f|, |1 - \alpha m_f|\})^{2k} \|x_0 - x_\ast\|^2. \]

Note that when $0 \leq \alpha \leq 2/L_f$, it holds that $2/(m_f + L_f) \leq \gamma_f \leq 1$. In particular, the fastest decay rate is attained at $\alpha = 2/(m_f + L_f)$, and is equal to $\gamma_f^2 = (L_f - m_f)/2(L_f + m_f)^2$.

On the other hand, setting $p_k \equiv 0$ in (27) yields the LMI
\[ \begin{bmatrix} \frac{m_f}{2}(a_{k+1} - a_k) \\ a_{k+1} - a_k \end{bmatrix} \leq 0. \]

Omitting the details, we obtain from the above LMI that $a_{k+1} \leq \rho^2 a_k$ and $0 \leq \alpha \leq 2/L_f$, where $\rho^2 = 1 + m_f/(L_f \alpha^2 - 2\alpha)$. Substituting these expressions in (23) yields the convergence result
\[ F(x_{k+1}) - F(x_\ast) \leq (1 + m_f(L_f \alpha^2 - 2\alpha) \rho^2)(F(x_0) - F(x_\ast)). \]

In particular, the optimal decay rate is attained at $\alpha = 1/L_f$, and is equal to $\rho = 1 - m_f/L_f$. Note that [11] establishes a better bound in terms of the objective value.

Convex Case. When the differentiable component of the objective is convex (i.e., $m_f = 0$), we select $p_k = p_0 > 0$ in (27) to arrive at the LMI
\[ \begin{bmatrix} \frac{m_f}{2}(a_{k+1} - a_k) \\ a_{k+1} - a_k \end{bmatrix} \leq 0. \]

This LMI enforces that $a_{k+1} = a_k + 2p_0 \alpha$ with $a_0 \geq 0$ and $\alpha > 0$ satisfying $(L_f \alpha^2 - 2\alpha)a_0 + 2p_0 \alpha^2 \leq 0$. Substituting these expressions in (23), we obtain
\[ \|x_k - x_\ast\|^2 \leq \frac{m_f}{2} (F(x_0) - F(x_\ast)) + p_0 \|x_0 - x_\ast\|^2. \]

In particular, if $a_0 = 0$, then it must hold that $0 \leq \alpha \leq 1/L_f$, and we recover the convergence result in [1, Theorem 3.1].

We close this section by considering the accelerated variant of the Proximal gradient method for both convex and strongly convex cases.

B. Accelerated Proximal Gradient Method

Strongly Convex Case. If we select $\beta_k = \sqrt{\gamma_f^{-1}}$, $a_k = a_0 \rho^{-2k}$, and $P_k = \rho^{-2k}P_0$ with
\[ P_0 = \frac{m_f}{2} \begin{bmatrix} 1 - \sqrt{\gamma_f} \\ \sqrt{\gamma_f} \end{bmatrix} \begin{bmatrix} 1 - \sqrt{\gamma_f} & \sqrt{\gamma_f} \end{bmatrix}, \]
then, the LMI condition (25a) is satisfied when $\sigma_k = 0$ and $\rho^2 = 1 - 1/\sqrt{\gamma_f}$. This result is a restatement of [25, Theorem 2.2.3] in view of Theorem 1.
Consider the points $x, y, z$ and the operator, defined in (7), is that

$$\phi$$

Substituting (30) and (31) for all $a_k = t_k$ and $0 < \alpha \leq 1/L_f$. Since $t_k - 1 \geq (k + 1)/2$ for $k \geq 1$, we have that $a_k = t_k^2 \geq (k + 1)^2/4$ for $k \geq 1$. Substituting all these expressions in (23), we can certify the convergence rate

$$F(x_k) - F(x_*) \leq \frac{2\|x_0 - x_*\|_2^2}{\alpha (k + 1)^2}, \quad 0 < \alpha \leq \frac{1}{L_f},$$

which is consistent with the result in [1, Theorem 4.4].

V. CONCLUSIONS

In this paper, we have developed an LMI framework, based on Integral Quadratic Constraints and time-dependent non-quadratic Lyapunov functions, to certify both exponential and subexponential convergence rates for proximal algorithms and under various regularity assumptions. Further, to unify convergence analysis of a wide variety of proximal algorithms and subexponential convergence rates for proximal algorithms.

In the convex setting, it can be verified that

$$0 = \Pi_{g,\alpha}(w) + \frac{1}{\alpha} \Pi_{g,\alpha}(w) - w,$$

or, equivalently,

$$0 = T_g(\Pi_{g,\alpha}(w)) + \frac{1}{\alpha} (\Pi_{g,\alpha}(w) - w), \quad T_g \in \partial g,$$

where $T_g(w)$ denotes a subgradient of $g$ at $w$. On the other hand, by the definition of the generalized gradient mapping in (8), we have that

$$\Pi_{g,\alpha}(y - \alpha \nabla f(y)) = y - \alpha \phi(y),$$

Substituting (30) and $w = y - \alpha \nabla f(y)$ in (29), we can equivalently write $\phi(y)$ as

$$\phi(y) = \nabla f(y) + T_g(y - \alpha \phi(y)).$$

Consider the points $x, y, z \in \text{dom}(f) \cap \text{dom}(g)$. We can write

$$f(z) - f(y) \leq \nabla f(y)^\top (z - y) + \frac{L_f}{2} \|z - y\|^2_2,$$

$$f(y) - f(x) \leq \nabla f(y)^\top (y - x) - \frac{m_f}{2} \|y - x\|^2_2.$$
where, we have made the substitution $\xi_{k+1} = A_k \xi_k + B_k u_k$ from (6) to obtain the first equality. On the other hand, according to Proposition 1, $\phi$ satisfies the IQC

$$\begin{bmatrix} y_k - y_\star - \phi(y_k) \\ \phi(y_k) - \phi(y_\star) \end{bmatrix}^	op Q \phi \begin{bmatrix} y_k - y_\star - \phi(y_k) \\ \phi(y_k) - \phi(y_\star) \end{bmatrix} \geq 0,$$

which, using the identity $y_k - y_\star = C_k (\xi_k - \xi_\star)$, can be rewritten as

$$e_k^\top M_k^{(3)} e_k = e_k^\top \begin{bmatrix} C_k & 0 \\ 0 & I_d \end{bmatrix}^\top Q \phi \begin{bmatrix} C_k & 0 \\ 0 & I_d \end{bmatrix} e_k \geq 0. \quad (39)$$

By substituting (36)-(38) back in (34), we obtain

$$V_{k+1}(\xi_{k+1}) - V_k(\xi_k) \leq e_k^\top \left( (a_{k+1} - a_k) M_k^{(0)} + a_k M_k^{(1)} + M_k^{(2)} \right) e_k. \quad (40)$$

Finally, the LMI (25a) implies that

$$(a_{k+1} - a_k) M_k^{(0)} + a_k M_k^{(1)} + M_k^{(2)} \preceq -\sigma_k M_k^{(3)}. \quad (41)$$

Substituting (41) back in (40) yields

$$V_{k+1}(\xi_{k+1}) - V_k(\xi_k) \leq -\sigma_k e_k^\top M_k^{(3)} e_k \leq 0,$$

where the right inequality follows from (39). Therefore, the Lyapunov function is nonincreasing and the convergence result (23) follows.

\section*{References}


