

Automated algorithm design for convex optimization problems with linear equality constraints

Ibrahim K. Ozaslan, Wuwei Wu, Jie Chen, Tryphon T. Georgiou, and Mihailo R. Jovanović

Abstract—Synthesis of optimization algorithms typically follows a *design-then-analyze* approach, which can obscure fundamental performance limits and hinder the systematic development of algorithms that operate near these limits. Recently, a framework grounded in robust control theory has emerged as a powerful tool for automating algorithm synthesis. By integrating design and analysis stages, fundamental performance bounds are revealed and synthesis of algorithms that achieve them is enabled. In this paper, we apply this framework to design algorithms for solving strongly convex optimization problems with linear equality constraints. Our approach yields a single-loop, gradient-based algorithm whose convergence rate is independent of the condition number of the constraint matrix. This improves upon the best known rate within the same algorithm class, which depends on the product of the condition numbers of the objective function and the constraint matrix.

Index Terms—Automated algorithm synthesis, circle criterion, internal model principle, Nevanlinna-Pick interpolation, optimization, primal-dual methods, robust control.

I. INTRODUCTION

The design and analysis of optimization algorithms typically begins with selecting an appropriate algorithmic structure based on the problem’s optimality conditions. This is followed by a convergence analysis, which often relies on creative and problem-specific reasoning. Such a *design-then-analyze* approach typically requires deriving tight analytical inequalities—an inherently challenging and non-systematic task. This difficulty contributes, in part, to the limited understanding of fundamental performance limits within specific algorithm classes.

In his seminal work [1], Nesterov established lower bounds on the worst-case convergence rates of gradient-based algorithms in the unconstrained setting and proposed algorithms that attain these bounds. Although the lower bounds are derived using an infinite-dimensional setup, the design and analysis of the corresponding algorithms rely on the estimate-sequence technique, which does not easily generalize to broader settings. Given the importance of understanding fundamental performance limits, substantial effort has been directed toward the study of accelerated optimization algorithms [2]–[5]. However, a closer examination reveals that many of these approaches start with a predefined algorithmic structure and use reverse engineering, offering

limited insight into the underlying principles that govern optimal performance.

Viewing optimization algorithms through the lens of dynamical systems and applying Lyapunov stability theory enables a system-theoretic understanding of their behavior. Although Lyapunov-based analysis is generally regarded as conservative, convergence rates matching known lower bounds can be achieved using Popov-type Lyapunov functions [6], [7]. Extending such functions to descent-ascent algorithms for constrained optimization or minimax problems, however, remains a significant challenge. In these settings, alternative Lyapunov functions have been proposed [8]–[10], but their design often relies on problem-specific insight. A complementary approach involves fixing the structure of a Lyapunov function and using computational tools to search for valid certificates or associated algorithms [11], [12]. While this method offers practical advantages, it introduces inherent conservatism and, as a result, falls short of revealing fundamental performance limits.

Recently, tools and concepts from robust control theory have been employed to demonstrate that Nesterov’s lower complexity bound also holds in finite-dimensional settings [13]. Building on this insight, a general framework has been developed for the design and analysis of centralized and distributed unconstrained optimization algorithms [14]–[16]. This framework models optimization algorithms as transfer functions within a feedback system, derives analytic conditions to guarantee desirable algorithmic properties, and uses Nevanlinna-Pick interpolation to construct transfer functions that meet these conditions. In addition to enabling automated algorithm synthesis, this approach provides a means for identifying fundamental performance limits when algorithm specifications are expressed through necessary and sufficient conditions.

In this paper, we build on the framework introduced in [14]–[16] to design algorithms for smooth, strongly convex optimization problems subject to linear equality constraints. While such constraints may appear restrictive, they naturally arise in a range of applications, including consensus problems [17] and incompressible fluid dynamics [18].

Among single-loop gradient-based algorithms for convex problems with linear equality constraints, the best known linear convergence rate is achieved by Gradient Descent-Ascent (GDA) [19]. The rate depends on the product of the condition numbers of the strongly convex objective function and the constraint matrix; see Section II-C for details. Interestingly, with an appropriate stepsize selection,

I. K. Ozaslan and M. R. Jovanović are with the Ming Hsieh Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90089. W. Wu and J. Chen are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China. T. T. Georgiou is with the Department of Mechanical and Aerospace Engineering University of California, Irvine, CA 92697. E-mails: ozaslan@usc.edu, w.wu@my.cityu.edu.hk, jichen@cityu.edu.hk, tryphon@uci.edu, mihailo@usc.edu.

the continuous-time counterpart of GDA can attain a convergence rate that is independent of the condition number of the constraint matrix [20]. Although continuous-time guarantees do not directly translate to discrete-time settings, this observation motivates the question of whether the worst-case convergence rate of a discrete-time gradient-based algorithm can be determined solely by the larger of the two condition numbers. In this paper, we take a step toward addressing this question. Leveraging the aforementioned framework, we design a single-loop gradient-based algorithm whose convergence rate depends only on the condition number of the objective function, provided this condition number is larger than that of the constraint matrix.

The rest of the paper is organized as follows. In Section II, we define the problem and introduce a control-theoretic formulation for algorithm design. We also outline desired specifications and discuss the relevant literature. In Section III, we present the framework for automated algorithm synthesis within considered algorithm class and obtain the main result. In Section IV, we use computational experiments to demonstrate the validity of our theoretical findings and, in Section V, we offer concluding remarks.

II. PROBLEM FORMULATION AND A CLASS OF ALGORITHMS

We consider a constrained convex optimization problem,

$$\begin{aligned} & \underset{x}{\text{minimize}} \quad f(x) \\ & \text{subject to} \quad Ex - q = 0 \end{aligned} \quad (1)$$

where $x \in \mathbb{R}^n$ is the optimization variable, $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function, and $E \in \mathbb{R}^{d \times n}$, $q \in \mathbb{R}^d$ are parameters in the linear constraint. Without loss of generality, we assume that E is a symmetric positive semidefinite matrix. This assumption is not restrictive because multiplying the equality constraint in (1) by E^\top does not change the feasible set. Hence, in our analysis the constraint can be replaced with $E^\top Ex = E^\top q$ without the need to explicitly construct $E^\top E$ in our algorithm (only matrix-vector multiplication involving E^\top and E is required). We also note that the feasibility of equality constraint implies that $E^\top E$ is a singular matrix.

Assumption 1: The equality constraint is homogeneous, i.e., $q = 0$. The constraint matrix $E \in \mathbb{R}^{n \times n}$ is singular, symmetric, and positive semidefinite, i.e., $E = E^\top \succeq 0$.

When $q \neq 0$, the homogeneity assumption can be satisfied by writing the constraint as $E^\top E(x - \bar{x}) = 0$ where \bar{x} is a feasible solution, and then replacing x in (1) with $x + \bar{x}$.

We also make the following assumption on the objective function, which is standard in the design of linearly convergent algorithms.

Assumption 2: The objective function f in (1) is m -strongly convex with an L -Lipschitz continuous gradient ∇f .

In the remainder of this section, we utilize the \mathcal{Z} -transform to represent a class of algorithms that we consider and outline design specifications. We also discuss the existing works on the lower complexity bounds within this class.

A. Algorithm representation

We examine gradient-based optimization algorithms. These can be described by three transfer functions which are (potentially) parameterized by the constraint matrix E :

- $\mathcal{K}_0(z, E)$ determines the mapping between the optimization variable and the input into a gradient;
- $\mathcal{K}_1(z, E)$ determines how the past iterates are used to update the optimization variable;
- $\mathcal{K}_2(z, E)$ determines how the past gradients are used to update the optimization variable.

Let $\hat{x}(z)$ and $\widehat{\nabla f}(z)$ be the \mathcal{Z} -transforms of sequences of iterates $\{x^k\}_{k=0}^\infty$ and gradients $\{\nabla f(x^k)\}_{k=0}^\infty$. Algorithms that we consider take the following form,

$$z\mathcal{K}_0(z, E)\hat{x}(z) = \mathcal{K}_1(z, E)\hat{x}(z) + \mathcal{K}_2(z, E)\widehat{\nabla f}(z). \quad (2)$$

While $\mathcal{K}_0(z, E)$ is the identity matrix in many cases, it plays a crucial role for accelerated methods such as Nesterov's accelerated algorithm [1] or two-step momentum method [21]. In addition to the momentum-based accelerated methods, distributed algorithms such as EXTRA [22], DIGing [23], or ABN [24], can also be brought into the form given by (2); see [14] for additional details.

We next demonstrate that formulation (2) encompasses not only centralized or distributed algorithms for unconstrained minimization but also primal-dual methods for optimization problems with equality constraints.

As an illustration, let us consider the gradient descent-ascent algorithm associated with (1) [19],

$$x^{k+1} = x^k - \alpha_1(\nabla f(x^k) + E^\top y^k) \quad (3a)$$

$$y^{k+1} = y^k + \alpha_2 E x^k \quad (3b)$$

where α_1 and α_2 are the stepsizes. The \mathcal{Z} -transform of (3) can be used to eliminate the dual variable $\hat{y}(z)$,

$$(z^2 I - 2zI + I + \alpha_1 \alpha_2 E^\top E) \hat{x}(z) = \alpha_1 (1 - z) \widehat{\nabla f}(z)$$

and the division of this equation with z leads to (2) with,

$$\mathcal{K}_0(z, E) = I$$

$$\mathcal{K}_1(z, E) = 2I - (I + \alpha_1 \alpha_2 E^\top E)z^{-1}$$

$$\mathcal{K}_2(z, E) = -\alpha_1 (1 - z^{-1})I.$$

We note that the dual update in (3) can be replaced by its incremental variant (also known as the alternating GDA),

$$y^{k+1} = y^k + \alpha_2 E x^{k+1}.$$

This asynchronous version of a primal-dual algorithm can similarly be represented by (2) with the only change,

$$\mathcal{K}_1(z, E) = 2I - \alpha_1 \alpha_2 E^\top E - z^{-1}I.$$

Furthermore, the primal-dual algorithm based on the proximal augmented Lagrangian [25] associated with (1) can be also cast within the same framework. Thus, a broad class of optimization algorithms can be represented using characterization (2).

B. Design specifications

We next discuss desired specifications for algorithms that we aim to design to solve optimization problem (1).

1) *Explicitness*: To design a single-loop algorithm, we need to avoid circular dependence in the evaluation of x^{k+1} . In other words, the computation of x^{k+1} should not exploit $E x^{k+1}$ or $\nabla f(x^{k+1})$. This requirement can be relaxed if one can afford to utilize linear system solvers [26] or evaluation of proximal operators [16] in the iterative scheme.

2) *Optimality*: The algorithm should asymptotically converge to a solution x^* of problem (1). Under Assumption 1, the necessary and sufficient conditions for x^* are given by,

$$\nabla f(x^*) \in \mathcal{R}(E^\top), \quad x^* \in \mathcal{N}(E) \quad (4a)$$

where \mathcal{R} and \mathcal{N} are the range and null spaces of a given matrix. These inclusions can alternatively be characterized by the existence of vectors $\delta_1^* \in \mathbb{R}^r$ and $w_2^* \in \mathbb{R}^{n-r}$ such that

$$\nabla f(x^*) = V_1 \delta_1^*, \quad x^* = V_2 w_2^*. \quad (4b)$$

Here, $r < n$ determines the rank of the matrix E with the singular value decomposition

$$E = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix} \quad (5)$$

where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$,

$$\bar{\sigma} := \sigma_1 \geq \dots \geq \sigma_r =: \underline{\sigma} > 0,$$

and the columns of matrices $V_1 \in \mathbb{R}^{n \times r}$ and $V_2 \in \mathbb{R}^{n \times n-r}$ form the respective orthonormal bases of $\mathcal{R}(E^\top)$ and $\mathcal{N}(E)$. We use $\kappa_E := \bar{\sigma}/\underline{\sigma}$ to denote the (effective) condition number of E and $\kappa_f := L/m$ to denote the condition number of the strongly convex objective function f .

Because of Assumption 2, problem (1) has a unique solution, implying uniqueness of δ_1^* and w_2^* as well.

3) *Linear convergence*: For a class of optimization problems (1) under Assumptions 1 and 2, we require algorithms to linearly converge at a rate of, at least, ρ . The linear convergence implies the existence of $\rho \in (0, 1)$ and $M > 0$ that are independent from the initial condition x^0 such that,

$$\|x^k - x^*\| \leq M \|x^0 - x^*\| \rho^k. \quad (6)$$

In contrast to the existing approaches, such as [17], the above framework provides not only asymptotic convergence guarantees but also a, possibly non-conservative, lower bound on the worst-case convergence rate for algorithms that satisfy all design specifications.

C. Discussion of the existing literature

Any gradient-based algorithm for solving problem (1) under Assumption 2 has to satisfy a lower bound $\|x^k - x^*\| \geq c/k^2$ for some constant c if the problem is infinite dimensional or the number of iterations is smaller than half the problem dimension [27]. However, this sublinear lower bound does not preclude the possibility of achieving linear convergence characterized in (6) due to scaling factor M . An

algorithm with a convergence rate matching the lower bound in unconstrained minimization setting is proposed in [28], but it utilizes a linear solver along the iterations and, therefore, is not a single-loop method. To our knowledge, the best known rate among the algorithms satisfying desired specifications is achieved by the (non)incremental GDA [19], [29],

$$\rho_{\text{gda}} = \sqrt{1 - \frac{1}{\kappa_f} \left(\frac{1}{\kappa_E} - \frac{1}{\kappa_E^2} \right)}. \quad (7)$$

III. AUTOMATED ALGORITHM SYNTHESIS

Herein, we formulate the algorithm design as a controller synthesis problem and provide analytical characterization of design specifications in terms of the underlying transfer functions in (2). By recasting control synthesis as an interpolation problem we provide a solution that leads to an implementable optimization algorithm with a guaranteed convergence rate.

A. From algorithm to controller design

The class of optimization algorithms given by (2) can be viewed as a Lur'e system, i.e., a feedback interconnection of an LTI system and a static nonlinear block. The LTI system in Fig. 1 can be represented by $\hat{x}(z) = \mathcal{H}(z, E) \widehat{\nabla f}(z)$, where

$$\mathcal{H}(z, E) := (z\mathcal{K}_0(z, E) - \mathcal{K}_1(z, E))^{-1} \mathcal{K}_2(z, E) \quad (8)$$

and the static nonlinear block takes the form,

$$\Delta(e^k) := \nabla f(e^k + x^*) - \nabla f(x^*) \quad (9)$$

with the error defined as $e^k := x^k - x^*$. Under Assumption 2, the nonlinear map (9) satisfies,

$$m\|e\|^2 \leq \langle \Delta(e), e \rangle \leq L\|e\|^2, \quad \forall e \in \mathbb{R}^n$$

thereby implying sector-boundedness of Δ in $[m, L]$.

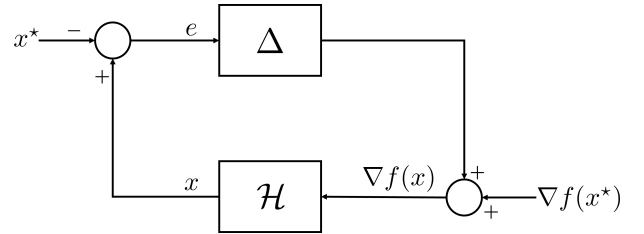


Fig. 1. A class of optimization algorithms given by (2) can be viewed as a feedback interconnection of an LTI system with a static nonlinear block.

In contrast to a conventional interpretation, the plant in this control system is represented by a static nonlinear block Δ . Our goal is to design a linear controller $\mathcal{H}(z, E)$, which represents the optimization algorithm, to track step inputs $\frac{z}{z-1} x^*$ with unknown magnitudes x^* while rejecting the step disturbances $\frac{z}{z-1} \nabla f(x^*)$ with unknown magnitudes $\nabla f(x^*)$ for all possible sector-bounded plants Δ in $[m, L]$. The controller is parameterized by a constraint matrix E of a rank $r < n$ with singular values $\sigma_i \in [\underline{\sigma}, \bar{\sigma}]$, $i = 1, \dots, r$.

Before discussing design specifications, we employ a standard technique to transform the Lur'e system into a form that is convenient for analysis/synthesis.

1) *Coordinate transformation*: The change of variables using the singular vectors of E ,

$$w := V^\top x = \begin{bmatrix} V_1^\top x \\ V_2^\top x \end{bmatrix} =: \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \quad (10)$$

together with the definitions

$$\begin{aligned} \epsilon &= V^\top e = \begin{bmatrix} w_1 \\ w_2 - w_2^* \end{bmatrix}, \quad w^* = V^\top x^* = \begin{bmatrix} 0 \\ w_2^* \end{bmatrix} \\ \delta &= V^\top \Delta(V\epsilon) = \begin{bmatrix} \delta_1 \\ \delta_2 \end{bmatrix}, \quad \delta^* = V^\top \nabla f(x^*) = \begin{bmatrix} \delta_1^* \\ 0 \end{bmatrix} \end{aligned}$$

can be used to cast the transfer function in Fig. 1 as,

$$\mathcal{H}(z, E) = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \begin{bmatrix} \mathcal{H}(z, \Sigma) & 0 \\ 0 & \mathcal{H}(z, 0) \end{bmatrix} \begin{bmatrix} V_1^\top \\ V_2^\top \end{bmatrix}.$$

The resulting two linear systems are coupled by the nonlinear block, as shown in Fig. 2. Since these subsystems are in the diagonal form, they can be further split into the scalar systems, each of which shares the same scalar transfer function parameterized by either singular values or zero,

$$\mathcal{H}(z, \Sigma) = \text{diag}(\{h(z, \sigma_i)\}_{i=1}^r), \quad \mathcal{H}(z, 0) = h(z, 0)I_{n-r}.$$

Our task is to design a scalar transfer function $h(z, \sigma)$ with parameter σ taking an arbitrary value within the interval $[\underline{\sigma}, \bar{\sigma}]$ or being equal to zero.

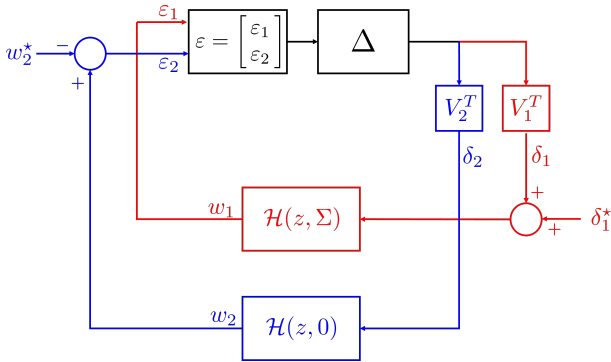


Fig. 2. Decomposition of an LTI system in Fig. 1 into two subsystems coupled by the static nonlinear element Δ .

2) *Loop transformation*: To make nonlinear component Δ sector-bounded in $[0, \infty)$, we apply loop transformation [30, Sec. 6.5] to the feedback interconnection in Fig. 1. This transformation enables the use of standard passivity-based stability criteria without introducing additional conservatism. Specifically, the transformation takes the pair (\mathcal{H}, Δ) with sector-bounded Δ in $[m, L]$ as input and gives the pair $(\bar{\mathcal{H}}, \bar{\Delta})$ with transformed nonlinearity $\bar{\Delta}$ being sector-bounded in $[0, \infty)$ as output. The one-to-one correspondence between the two transfer functions is given by

$$\mathcal{H}(z, E) = (m\bar{\mathcal{H}}(z, E) - LI)^{-1}(\bar{\mathcal{H}}(z, E) - I). \quad (11)$$

In contrast to \mathcal{H} , $\bar{\mathcal{H}}$ has to be a stable transfer function to guarantee stability of the transformed Lur'e system; this is

because $\bar{\Delta}$ is sector-bounded in $[0, \infty)$.

Using coordinate transformation (10), we can also diagonalize the transformed linear system,

$$\bar{\mathcal{H}}(z, \Sigma) = \text{diag}(\{\bar{h}(z, \sigma_i)\}_{i=1}^r), \quad \bar{\mathcal{H}}(z, 0) = \bar{h}(z, 0)I_{n-r}$$

with the following one-to-one relation between h and \bar{h} ,

$$h(z, \sigma) = \frac{\bar{h}(z, \sigma) - 1}{m\bar{h}(z, \sigma) - L}. \quad (12)$$

B. Analytical characterization of design specifications

We next obtain conditions for transfer function $\bar{h}(z, \sigma)$ to satisfy design specifications outlined in Section II-B.

1) *Explicitness as causality*: A necessary condition for explicitness is the strict causality of the linear controller in the Lur'e system. The controller is strictly causal when transfer function $\mathcal{H}(z, E)$ is strictly proper, i.e., $\mathcal{H}(\infty, E) = 0$. Using the correspondence in (11), this is equivalent to $\bar{\mathcal{H}}(\infty, E) = I$, and the scalar transfer function should satisfy

$$\bar{h}(\infty, \sigma) = 1, \quad \forall \sigma \in [\underline{\sigma}, \bar{\sigma}] \cup \{0\}. \quad (13a)$$

2) *Optimality as input tracking and disturbance rejection*: The asymptotic convergence of the optimization algorithm to the optimal solution is equivalent to the asymptotic tracking of the step input $\frac{z}{z-1}x^*$ and rejection of the step disturbance $\frac{z}{z-1}\nabla f(x^*)$ in the Lur'e system. To achieve this goal, the subsystems $\mathcal{H}(z, \Sigma)$ and $\mathcal{H}(z, 0)$ must have a blocking zero and a pole at $z = 1$, respectively. This follows from internal model principle [31]. For exogenous signals (4), Proposition 1 establishes that these conditions are not only sufficient but also necessary for input tracking and disturbance rejection.

Proposition 1: Let the feedback interconnection in Fig. 1 be stable for any sector bounded nonlinearity Δ in $[m, L]$. The error asymptotically converges to zero, i.e., $\lim_{k \rightarrow \infty} e^k = 0$, if and only if the transformed scalar transfer function satisfies

$$\bar{h}(1, \sigma) = \begin{cases} 1, & \sigma \in [\underline{\sigma}, \bar{\sigma}] \\ L/m, & \sigma = 0. \end{cases} \quad (13b)$$

Proof: See Appendix. \blacksquare

3) *Linear convergence as ρ -stability*: A necessary and sufficient condition for linear convergence of the synthesized algorithm is the ρ -stability of the Lur'e system in Fig. 1. We use the circle criterion [30, Sec. 7.1] to certify the ρ -stability.

The circle criterion implies ρ -stability of the transformed Lur'e system $(\bar{\mathcal{H}}, \bar{\Delta})$, with sector-bounded nonlinear block $\bar{\Delta}$ in $[0, \infty)$, if the scalar transfer function $h(\gamma z, \sigma)$ is *strictly positive real* for all $\gamma \in (\rho, 1]$. This can be equivalently expressed as,

$$\begin{aligned} \text{i) } & \bar{h}(\gamma z, \sigma) \in \mathcal{RH}_\infty \\ \text{ii) } & \text{Re}(\bar{h}(\gamma z, \sigma)) > 0, \quad \forall z \in \mathbb{D}^c, \quad \sigma \in [\underline{\sigma}, \bar{\sigma}] \cup \{0\}. \end{aligned} \quad (13c)$$

for all $\gamma \in (\rho, 1]$, where \mathcal{RH}_∞ denotes set of real, rational, stable transfer functions, $\text{Re}(\cdot)$ is the real part of a complex number, and $\mathbb{D} := \{z \in \mathbb{C}, |z| < 1\}$.

C. From controller design to interpolation

Design of a scalar transfer function \bar{h} that satisfies causality (13a), tracking (13b), and ρ -stability (13c) conditions can be cast as the following interpolation problem:

Find a $\rho \in (0, 1)$ and design $\bar{h}(z, \sigma)$ such that conditions in (13) hold for every $\gamma \in (\rho, 1]$. (P1)

Direct application of the Nevanlinna-Pick interpolation technique [32, Ch. 9] to problem (P1) is not possible because of conditional statement (13b) for different values of σ . Hence, following [14], we adopt a greedy approach to obtain a solution as outlined below.

Since interpolation condition (13b) includes two cases ($\sigma \in [\underline{\sigma}, \bar{\sigma}]$ or $\sigma = 0$), we first construct a candidate solution ($\rho_1, \bar{h}_1(z, \sigma)$) to (P1) with conditions (13) applied only to $\sigma \in [\underline{\sigma}, \bar{\sigma}]$. Similarly, we obtain another candidate solution ($\rho_2, \bar{h}_2(z)$), ensuring that conditions (13) hold for $\sigma = 0$. We then combine these to obtain a solution to problem (P1) by setting $\rho = \max(\rho_1, \rho_2)$ and defining $\bar{h}(z, \sigma) = \bar{h}_1(z, \sigma)$ while enforcing $\bar{h}(z, 0) = \bar{h}_2(z)$. The last step leverages the additional degrees of freedom offered by the Nevanlinna-Pick interpolation. Lemmas 1 and 2 provide a solution to problem (P1) for $\sigma \in [\underline{\sigma}, \bar{\sigma}]$ and $\sigma = 0$, respectively.

Lemma 1: All transfer functions satisfying

$$\begin{aligned} \bar{h}(\gamma z, \sigma) &\in \mathcal{RH}_\infty \\ \bar{h}(\infty, \sigma) &= \bar{h}(1, \sigma) = 1 \\ \operatorname{Re}(\bar{h}(\gamma z, \sigma)) &> 0, \quad z \in \mathbb{D}^c \end{aligned}$$

for all $\gamma \in (0, 1]$ and $\sigma \in [\underline{\sigma}, \bar{\sigma}]$ are given by

$$\bar{h}_1(\gamma z, \sigma) = \frac{z(z - \gamma) + \bar{g}(z, \sigma)(\gamma z - 1)}{z(z - \gamma) - \bar{g}(z, \sigma)(\gamma z - 1)} \quad (14)$$

where $\bar{g}(z, \sigma) \in \mathcal{RH}_\infty$ and $\|\bar{g}(z, \sigma)\|_\infty < 1$, $\forall \sigma \in [\underline{\sigma}, \bar{\sigma}]$.

Proof: The proof is a direct application of Nevanlinna-Pick interpolation; see [14, Prop. 1] for details. ■

Lemma 2: A transfer function satisfying

$$\begin{aligned} \bar{h}(\gamma z) &\in \mathcal{RH}_\infty \\ \bar{h}(\infty) &= 1 \\ \bar{h}(1) &= L/m \\ \operatorname{Re}(\bar{h}(\gamma z)) &> 0, \quad z \in \mathbb{D}^c \end{aligned}$$

for $\gamma \in (\rho, 1]$ exists if and only if $\rho \geq (L - m)/(L + m)$. Furthermore, for $\rho = (L - m)/(L + m)$, the transfer function that satisfies all four conditions is unique and is given by,

$$\bar{h}_2(\gamma z) = \frac{\gamma z + \rho}{\gamma z - \rho}. \quad (15)$$

Proof: See [14, Theorem 5]. ■

We combine two candidate solutions (14) and (15) as described above: we set $\bar{h}(z, \sigma) = \bar{h}_1(z, \sigma)$ and determine $\bar{g}(z, \sigma)$ in Lemma 1 to ensure $\bar{h}(z, 0) = \bar{h}_2(z)$. This yields,

$$\bar{g}(z, 0) = -\frac{1}{\rho} \frac{z - \rho}{z - 1/\rho}$$

where

$$\rho = \frac{L - m}{L + m} = 1 - \frac{2}{\kappa_f + 1}. \quad (16)$$

This brings interpolation problem (P1) to:

Design $g(z, \sigma)$ such that

$$\begin{cases} \bar{g}(z, \sigma) \in \mathcal{H}_\infty \text{ and } \|\bar{g}(z, \sigma)\|_\infty < 1, & \sigma > 0 \\ \bar{g}(z, 0) = -\frac{1}{\rho} \frac{z - \rho}{z - 1/\rho}, & \sigma = 0. \end{cases} \quad (P2)$$

A solution to problem (P2) is given by

$$\bar{g}(z, \sigma) = -\frac{1 - \sigma}{\rho} \frac{z - \frac{\rho}{1 - \sigma}}{z - \frac{1 - \sigma}{\rho}} \quad (17)$$

with additional restrictions on the singular values,

$$\frac{2m}{m + L} < \sigma \leq 1. \quad (18)$$

This solution was obtained by moving the unstable pole at $1/\rho$ of $\bar{g}(z, 0)$ inside the unit circle by ensuring that the \mathcal{H}_∞ norm of $\bar{g}(z, \sigma)$ is less than one.

D. Algorithm synthesis

We are now ready to cast the transfer functions obtained in Section III-C in terms of their state-space realizations, thereby leading to an implementable optimization algorithm.

Substitution of \bar{h} given by (14) into (12) yields the following transfer function for h ,

$$h(z, \sigma) = \frac{\frac{2}{L + m} g(z, \sigma)}{z + g(z, \sigma)}$$

where

$$\begin{aligned} g(z, \sigma) &:= \rho \bar{g}(z/\rho, \sigma) \frac{z - 1}{z - \rho^2} \\ &\stackrel{(17)}{=} -\frac{(1 - \sigma)z - \rho^2}{z - (1 - \sigma)} \frac{z - 1}{z - \rho^2}. \end{aligned}$$

Furthermore, the definition of h in (8),

$$h(z, \sigma) = \frac{k_2(z, \sigma)}{zk_0(z, \sigma) - k_1(z, \sigma)}$$

allows us to determine the transfer functions k_0 , k_1 , and k_2 .

In particular, setting $k_0(z, \sigma) = 1$ gives

$$k_1(z, \sigma) = -g(z, \sigma), \quad k_2(z, \sigma) = \frac{2}{L + m} g(z, \sigma)$$

and, as a result, we have

$$\mathcal{K}_0(z, E) = I \quad (19a)$$

$$\mathcal{K}_1(z, E) = \frac{z - 1}{z - \rho^2} (zI - W)^{-1} (Wz - \rho^2 I) \quad (19b)$$

$$\mathcal{K}_2(z, E) = -(1 - \rho) \mathcal{K}_1(z, E) \quad (19c)$$

where $W := I - E$. The remaining task is to obtain a state-space realization for the input-output map (8) based on the transfer functions given in (19).

Since $\mathcal{K}_0 = I$ and \mathcal{K}_2 is a scaled version of \mathcal{K}_1 , we have

$$\hat{x}(z) = -(1 - \rho)(zI - \mathcal{K}_1(z, E))^{-1} \widehat{\nabla} f(z). \quad (20)$$

This equation can be used to obtain an input-output recursion that relates $\nabla f(x^k)$ and x^k ,

$$\sum_{i=0}^3 A_i x^{k+i} = \sum_{i=0}^3 B_i \nabla f(x^{k+i}) \quad (21)$$

and a corresponding state-space realization of (21) using the observer canonical form,

$$\begin{aligned} \psi^{k+1} &= A \psi^k + B \nabla f(x^k) \\ x^k &= C \psi^k. \end{aligned} \quad (22a)$$

Here,

$$A = \begin{bmatrix} -A_2 & I & 0 \\ -A_1 & 0 & I \\ -A_0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} B_2 \\ B_1 \\ B_0 \end{bmatrix}, \quad C^\top = \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix} \quad (22b)$$

and

$$\begin{aligned} A_0 &= -\rho^2 I, & B_0 &= -(1-\rho)\rho^2 I \\ A_1 &= (1+\rho^2)W + \rho^2 I, & B_1 &= (1-\rho)(\rho^2 I + W) \\ A_2 &= -(2W + \rho^2 I), & B_2 &= -(1-\rho)W \\ A_3 &= I & B_3 &= 0. \end{aligned} \quad (22c)$$

Equivalently, state-space model (22) can be cast in terms of the following recursion,

$$u^k = x^k - (1-\rho)\nabla f(x^k) \quad (23a)$$

$$y^k = W(y^{k-1} + u^k) - \rho^2 u^{k-1} \quad (23b)$$

$$x^{k+1} = \rho^2 x^k + y^k - y^{k-1} \quad (23c)$$

with the initial condition $y^{-1} = u^{-1} = 0$.

Theorem 1: Let Assumptions 1 and 2 hold. If the singular values of the constraint matrix E belong to the interval $(2/\kappa_f, 1]$, where $\kappa_f = L/m$, then algorithm (23) converges linearly to the unique solution of (1) with a rate of at least,

$$\rho_{\text{syn}} = 1 - \frac{2}{\kappa_f + 1}. \quad (24)$$

Remark 1: While the upper bound on the singular values can easily be satisfied by scaling the equality constraint with its largest column-sum or its trace, the lower bound requires the condition number of E to be smaller than half the condition number of the objective function, $\kappa_f/2 = L/(2m)$.

Remark 2: For problem (1) with a rectangular constraint matrix $E \in \mathbb{R}^{d \times n}$, both algorithm (23) and incremental GDA (3) perform only one gradient computation and one matrix-vector multiplication with E and E^\top per iteration. However, while (23) has a realization with $3n$ states, the incremental GDA has $n + d$ states.

IV. COMPUTATIONAL EXPERIMENTS

We demonstrate the merits and the effectiveness of our algorithm for the quadratic objective function in (1),

$$f(x) = \frac{1}{2} x^\top Q x + p^\top x.$$

For the problem with $n = 100$ optimization variables and $d = 100$ constraints, we randomly select problem data $Q \in$

$\mathbb{R}^{n \times n}$, $E \in \mathbb{R}^{d \times n}$, and $p \in \mathbb{R}^n$ with $q = 0$ in (1). The eigenvalues of Q are scaled such that $L = \lambda_{\max}(Q) = 2000$ and $m = \lambda_{\min}(Q) = 1$. We generate three instances of constraint matrices E_1 , E_2 , and E_3 with the corresponding singular values scaled such that the largest singular values are the same and equal to $\sigma_1 = 1$, but the smallest singular values take different values $\sigma_r = 0.1$, $\sigma_r = 0.01$, and $\sigma_r = 0.001$, respectively, with $r = 80$. As a result, the (effective) condition numbers of the constraint matrices are determined by $\kappa_{E_1} = 10$, $\kappa_{E_2} = 100$, and $\kappa_{E_3} = 1000$.

We perform computational experiments using our algorithm (23) and the incremental GDA method (3) to compare and contrast the influence of conditions numbers of constraint matrices on the corresponding convergence rates. We set all initial conditions to zero. To optimize the best known convergence rate of the GDA method [19], [29], we set $\alpha_1 = (1 - \kappa_E^{-1})/L$ and $\alpha_2 = m/\sigma_1$ in (3). As illustrated in Fig. 3, the convergence rate of our algorithm is invariant under changes in condition number of the constraint matrix, closely aligning with the theoretical bound. In contrast, the convergence rate of the GDA algorithm significantly deteriorates as the condition number of E increases. Furthermore, as suggested by the comparison of the convergence rates (24) and (7), our algorithm converges substantially faster than the GDA method in all computational experiments.

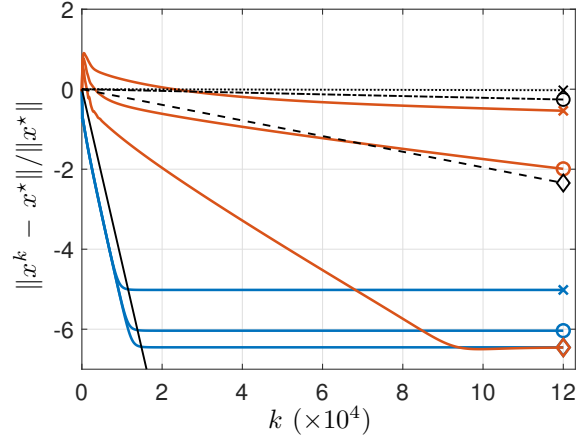


Fig. 3. Convergence of algorithm (23) (blue) and the incremental GDA method (3) (red) for problem (1). The lines with markers show convergence of algorithms in the first (diamond), second (circle), and third (cross) problem instances with the respective constraint matrices E_1 , E_2 , and E_3 . While the condition number of the objective function is set to $\kappa_f = 2000$, the (effective) condition numbers of the constraint matrices are $\kappa_{E_1} = 10$, $\kappa_{E_2} = 100$, and $\kappa_{E_3} = 1000$. The solid black line marks the theoretical convergence rate ρ_{syn} in (24); the black dash, dash-dot, and dot lines show the rate ρ_{gda} in (7) for $i = 1, 2, 3$, respectively.

V. CONCLUDING REMARKS

In this work, we leveraged robust control techniques for the automated synthesis of optimization algorithms. Building on the framework recently proposed in [14], [15], we developed a single-loop, gradient-based algorithm for solving strongly convex optimization problems with linear equality constraints. When the condition number of the constraint matrix is smaller than that of the objective function, we demonstrated that the algorithm's convergence rate depends solely

on the objective function's condition number. Under this assumption, our approach offers a significant improvement over existing methods—such as GDA—whose rates depend on the product of both condition numbers. Promising future directions include relaxing the assumption on the constraint matrix, characterizing the fundamental performance limits of gradient-based methods, and extending the framework to a wider class of constrained optimization problems.

APPENDIX

Proof of Proposition 1: Sufficiency follows from the internal model principle. For necessity, we consider the following equation representing the feedback interconnection in Fig. 1,

$$\hat{e}(z) + \frac{z}{z-1} x^* = \mathcal{H}(z, E)(\hat{\Delta}(z) + \frac{z}{z-1} \nabla f(x^*))$$

where $\hat{e}(z)$ and $\hat{\Delta}(z)$ are the \mathcal{Z} -transforms of the sequences $\{e^k\}_{k=0}^\infty$ and $\{\Delta(e^k)\}_{k=0}^\infty$, respectively. Correspondence (11) yields

$$\begin{aligned} (m\bar{\mathcal{H}}(z, E) - LI)((z-1)\hat{e} + zx^*) \\ = (\bar{\mathcal{H}}(z, E) - I)((z-1)\hat{\Delta}(z) + z\nabla f(x^*)). \end{aligned}$$

By assumption, the feedback interconnection is stable for any sector-bounded nonlinearity Δ in $[m, L]$, which implies that the transfer function $\bar{\mathcal{H}}(z, E)$ does not have any poles on the unit circle, i.e., $\bar{\mathcal{H}}(1, E) \neq \infty$. Therefore, evaluating the above equation at $z = 1$ results in

$$(m\bar{\mathcal{H}}(1, E) - LI)x^* = (\bar{\mathcal{H}}(1, E) - I)\nabla f(x^*).$$

Coordinate transformation (10) in conjunction with optimality conditions (4b) gives

$$V_2(m\bar{\mathcal{H}}(1, 0) - LI)w_2^* = V_1(\bar{\mathcal{H}}(1, \Sigma) - I)\delta_1^*$$

which must hold for any $\delta_1^* \in \mathbb{R}^r$ and $w_2^* \in \mathbb{R}^{n-r}$. Hence,

$$\bar{\mathcal{H}}(1, 0) = (L/m)I_{n-r}, \quad \bar{\mathcal{H}}(1, \Sigma) = I_r.$$

REFERENCES

- [1] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. New York, NY: Springer, 2003.
- [2] W. Su, S. Boyd, and E. Candes, "A differential equation for modeling Nesterov's accelerated gradient method: theory and insights," *J. Mach. Learn. Res.*, vol. 17, no. 153, pp. 1–43, 2016.
- [3] A. Wibisono, A. C. Wilson, and M. I. Jordan, "A variational perspective on accelerated methods in optimization," *Proc. Natl. Acad. Sci. USA*, vol. 113, no. 47, pp. E7351–E7358, 2016.
- [4] B. Hu and L. Lessard, "Dissipativity theory for Nesterov's accelerated method," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1549–1557.
- [5] B. Shi, S. S. Du, M. I. Jordan, and W. J. Su, "Understanding the acceleration phenomenon via high-resolution differential equations," *Math. Program.*, vol. 195, no. 1, pp. 79–148, 2021.
- [6] M. Muehlebach and M. I. Jordan, "A dynamical systems perspective on Nesterov acceleration," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4656–4662.
- [7] I. K. Ozaslan and M. R. Jovanović, "Accelerated forward-backward and Douglas-Rachford splitting dynamics," *Automatica*, vol. 175, p. 112210, 2025.
- [8] J. Cortes and S. Niederlander, "Distributed coordination for nonsmooth convex optimization via saddle-point dynamics," *J. Nonlinear Sci.*, vol. 29, pp. 1247–1272, 2019.
- [9] G. Qu and N. Li, "On the exponential stability of primal-dual gradient dynamics," *IEEE Contr. Syst. Lett.*, vol. 3, no. 1, pp. 43–48, 2018.

- [10] I. K. Ozaslan, P. Patrinos, and M. R. Jovanović, "Stability of primal-dual gradient flow dynamics for multi-block convex optimization problems," *arXiv:2408.15969*, 2024.
- [11] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM J. Control Optim.*, vol. 26, no. 1, pp. 57–95, 2016.
- [12] S. Michalowsky, C. Scherer, and C. Ebenbauer, "Robust and structure exploiting optimisation algorithms: An integral quadratic constraint approach," *Internat. J. Control*, vol. 94, no. 11, pp. 2956–2979, 2021.
- [13] V. Ugrinovskii, I. R. Petersen, and I. Shames, "A robust control approach to asymptotic optimality of the heavy-ball method for optimization of quadratic functions," *Automatica*, vol. 155, p. 111129, 2023.
- [14] S. Zhang, W. Wu, Z. Li, J. Chen, and T. T. Georgiou, "Frequency-domain analysis of distributed optimization: Fundamental convergence rate and optimal algorithm synthesis," *IEEE Trans. Automat. Control*, vol. 69, no. 12, pp. 8539–8554, 2024.
- [15] W. Wu, S. Zhang, Z. Li, J. Chen, and T. T. Georgiou, "On decomposition and convergence of distributed optimization algorithms," in *Proc. Amer. Control Conf.*, 2024, pp. 3391–3396.
- [16] W. Wu, J. Chen, M. R. Jovanović, and T. T. Georgiou, "Tannenbaum's gain-margin optimization meets Polyak's heavy-ball algorithm," *arXiv preprint arXiv:2409.19882*, 2024.
- [17] S. Boyd, T. Parshakova, E. Ryu, and J. J. Suh, "Optimization algorithm design via electric circuits," *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 68 013–68 081, 2024.
- [18] H. Taha, C. Gonzalez, and M. Shorbagy, "A minimization principle for incompressible fluid mechanics," *Phys. Fluids*, vol. 35, no. 12, 2023.
- [19] S. A. Alghunaim and A. H. Sayed, "Linear convergence of primal-dual gradient methods and their performance in distributed optimization," *Automatica*, vol. 117, p. 109003, Jul. 2020.
- [20] I. K. Ozaslan and M. R. Jovanović, "Tight lower bounds on the convergence rate of primal-dual dynamics for equality constrained convex problems," in *Proc. IEEE Conf. Decis. Control*, 2023, pp. 7318–7323.
- [21] B. Van Scoy, R. A. Freeman, and K. M. Lynch, "The fastest known globally convergent first-order method for minimizing strongly convex functions," *IEEE Contr. Syst. Lett.*, vol. 2, no. 1, pp. 49–54, 2017.
- [22] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.
- [23] A. Nedic, A. Olshevsky, and . Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.
- [24] R. Xin, D. Jakovetic, and U. A. Khan, "Distributed Nesterov gradient methods over arbitrary graphs," *IEEE Signal Process. Lett.*, vol. 26, no. 8, pp. 1247–1251, 2019.
- [25] N. K. Dhirga, S. Z. Khong, and M. R. Jovanović, "The proximal augmented Lagrangian method for nonsmooth composite optimization," *IEEE Trans. Automat. Control*, vol. 64, no. 7, pp. 2861–2868, 2019.
- [26] I. K. Ozaslan, M. Pilanci, and O. Arikan, "M-IHS: An accelerated randomized preconditioning method avoiding costly matrix decompositions," *Linear Algebra Appl.*, vol. 678, pp. 57–91, 2023.
- [27] Y. Ouyang and Y. Xu, "Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems," *Math. Program.*, vol. 185, no. 1, pp. 1–35, 2021.
- [28] A. Salim, L. Condat, D. Kovalev, and P. Richtarik, "An optimal algorithm for strongly convex minimization under affine constraints," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2022, pp. 4482–4498.
- [29] B. Van Scoy, J. W. Simpson-Porco, and L. Lessard, "Automated Lyapunov analysis of primal-dual optimization algorithms: An interpolation approach," in *Proc. IEEE Conf. Decis. Control*, 2023, pp. 1306–1311.
- [30] H. K. Khalil, *Nonlinear Systems*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [31] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.
- [32] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback Control Theory*. New York, NY: Macmillan, 1992.