Proceedings of Machine Learning Research vol 331:1–16, 2026 8th Annual Conference on Learning for Dynamics and Control

Chebyshev polynomials meet Nevanlinna-Pick interpolation: An automated procedure for algorithm synthesis

Ibrahim K. Ozaslan OZASLAN@USC.EDU
Tryphon T. Georgiou TRYPHON@UCI.EDU
Mihailo R. Jovanovic MIHAILO@USC.EDU

Editors: G. Sukhatme, L. Lindemann, S. Tu, A. Wierman, N. Atanasov

Abstract

The synthesis of optimization algorithms typically follows a design-first-analyze-later paradigm, a practice that often obscures fundamental performance limits and hampers the systematic design of algorithms operating near these limits. In this paper, we build on a recently proposed frequency-domain control framework that enables a principled approach to algorithm design, integrating the analysis and synthesis stages and thereby elucidating the fundamental performance boundaries. Specifically, we remove restrictive assumptions from recent prior work and extend the framework to encompass a broad class of strongly convex problems with equality constraints. This leads to a new family of algorithms that achieves a sharp trade-off between the number of matrix–vector multiplications per iteration and the convergence rate. Notably, one of the resulting algorithms attains the optimal lower bound on the total number of matrix–vector multiplications required to reach a prescribed accuracy.

Keywords: Automated algorithm design, Chebyshev acceleration, circle criterion, constrained optimization, multi-agent systems, Nevanlinna-Pick interpolation, stability of Lur'e systems

1. Introduction

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

Viewing optimization algorithms through the lens of dynamical systems and applying tools from stability theory facilitates a system-theoretic understanding of their convergence behavior. Although Lyapunov-based analysis is generally regarded as conservative, convergence rates matching known lower bounds can be achieved using Popov-type Lyapunov functions (Muehlebach and Jordan, 2019; Ozaslan and Jovanović, 2025). 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 (Qu and Li, 2019a; Ding and Jovanović, 2020; Zheng et al., 2023; Ozaslan et al., 2024), but their design often relies on deep problem-specific insight. A complementary approach fixes the structure of both the Lyapunov function and the algorithm and uses computational tools to search for valid certificates and parameters (Lessard et al., 2016; Van Scoy et al., 2017; Michalowsky et al., 2021; Scherer et al., 2023; Boyd et al., 2024). While this approach can yield sharp results (Van Scoy et al., 2017; Scherer and Ebenbauer, 2021), its reliance on fixed functional forms limits its ability to reveal fundamental performance bounds.

Recently, a frequency-domain approach has emerged as a powerful tool for automating algorithm design and analysis (Ugrinovskii et al., 2023; Zhang et al., 2024; Wu et al., 2024). In this setting, optimization algorithms are modeled as transfer functions within a feedback system, analytical conditions are derived to guarantee desired algorithmic properties, and Nevanlinna–Pick interpolation is used to construct transfer functions satisfying these conditions. When the specifications are expressed through necessary and sufficient conditions, the approach enables identification of fundamental performance limits. For example, while Nesterov's lower bound on the complexity of gradient-based methods was originally proved in infinite dimensions using adversarial examples, a constructive proof has since been established in finite-dimensional settings using this framework—first time in Ugrinovskii et al. (2023) and independently in Wu et al. (2024), building on Zhang et al. (2024).

In this paper, we extend this control-theoretic framework to address algorithms for convex optimization problems with linear equality constraints,

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

where f is an m-strongly convex function with an L-Lipschitz continuous gradient, $E \in \mathbb{R}^{d\times n}$ is a constraint matrix, and $q \in \mathbb{R}^d$ is a vector. We also define condition numbers $\kappa_f := L/m$ and $\kappa_E := \sigma_1/\sigma_r$, where σ_1 and σ_r denote the largest and smallest (nonzero) singular values of $E^\top E$.

Such problems arise in a variety of applications including federated learning (Li et al., 2020), imaging (Chambolle and Pock, 2016), optimal transport (Peyre and Cuturi, 2019), and fluid dynamics (Taha et al., 2023). A particularly important case arises in decentralized optimization, where E is a gossip (or Laplacian) matrix of the network, $q = 0$, and the objective function is a sum of local costs, $f(x) = \sum_{i=1}^N f_i(x_i)$ (Scaman et al., 2017).

Existing algorithms. A lower bound on the iteration complexity of gradient-based algorithms for problem (1), along with an algorithm achieving this bound, was established in Salim et al. (2022a). This algorithm attains ϵ-accuracy in $O(\sqrt{\kappa_f} \log \frac{1}{\epsilon})$ iterations but requires $O(\sqrt{\kappa_E})$ matrix–vector multiplications per iteration. When matrix–vector multiplications dominate the computational cost—as in distributed optimization with limited communication budgets (Qu and Li, 2019b; Lan et al., 2020; Xu et al., 2021)—inner-loop iterations become undesirable. Among single-loop methods, Gradient Descent–Ascent (GDA) achieves $O(\kappa_f \kappa_E \log \frac{1}{\epsilon})$ iteration complexity, while a variant of the Condat–Vu algorithm (Condat, 2013; Vu, 2013), known as PAPC (Salim et al., 2022b), achieves $O(\max(\kappa_f, \kappa_E) \log \frac{1}{\epsilon})$. In the special case where E is the gossip matrix and $q = 0$, the accelerated PAPC algorithm attains $O(\max(\sqrt{\kappa_f \kappa_E}, \kappa_E) \log \frac{1}{\epsilon})$ complexity (Kovalev et al., 2020), while accelerated EXTRA achieves $O(\sqrt{\kappa_f \kappa_E} \log(\kappa_f \kappa_E) \log \frac{1}{\epsilon})$ (Li and Lin, 2020).

Our contribution. We develop a unified framework that enables automated algorithm design across diverse problem settings. Using this framework, we synthesize a family of methods—referred to as *Interpolated Gradient Methods* (I-GM)—for solving problem (1). Each member of this family is characterized by a polynomial whose maximal value determines the convergence rate and whose degree determines the number of matrix–vector multiplications per iteration. This parametrization provides flexibility to accommodate different performance requirements, while the polynomial selection remains fully principled and integrated into the synthesis procedure.

We demonstrate this flexibility through two concrete instances:

(i) The first method establishes a sharp trade-off between the number of matrix–vector multiplications per iteration ℓ and the worst-case convergence rate, achieving an iteration complexity of $O(\max(\kappa_f, \kappa_E/\ell) \log \frac{1}{\epsilon})$. Adjusting ℓ allows balancing local information propagation and convergence rate in decentralized settings with limited communication/computation budgets.

(ii) The second method employs a family of polynomials that provably minimizes the convergence rate among all degree-ℓ polynomials. This method achieves $O(\kappa_f \log \frac{1}{\epsilon})$ iteration complexity with $O(\sqrt{\kappa_E/\kappa_f})$ matrix–vector multiplications per iteration, attaining the optimal communication complexity of $O(\sqrt{\kappa_f \kappa_E} \log \frac{1}{\epsilon})$ (Scaman et al., 2017).

Importantly, our guarantees arise directly from the synthesis procedure without requiring a separate convergence analysis or the prior construction of a Lyapunov function.

Related work. Our framework overcomes key limitations of recent interpolation-based methods (Zhang et al., 2024; Ozaslan et al., 2025a,b) by: (i) eliminating structural restrictions on E through an additional polynomial parameterization of the transfer functions; (ii) establishing guarantees for general convex functions; and (iii) requiring only one-sided bounds on the step sizes. In contrast, prior approaches rely on exact parameter selection, require restrictive assumptions on the singular values of E (e.g., lying in $(2/3, 1)$ (Zhang et al., 2024) or $(2/\kappa_f, 1]$ (Ozaslan et al., 2025b)), and are limited to specific nonlinearities, such as $f(x) = \|x\|^4$ (Zhang et al., 2024).

2. Automated algorithm synthesis

Our framework proceeds through a sequence of structured steps. We first specify the desired algorithmic properties (Sec. 2.1) and employ the $\mathcal{Z}$-transform to represent the class of candidate algorithms (Sec. 2.2). The algorithm design is formulated as a controller synthesis problem (Sec. 2.3), and the resulting design specifications admit analytical characterization in terms of the associated transfer functions (Sec. 2.4). These characterizations allow the control synthesis task to be recast as an algebraic problem, which is subsequently expressed as an interpolation problem via suitably chosen conformal mappings (Sec. 2.5). Application of the Nevanlinna–Pick interpolation technique yields all admissible solutions (Sec. 2.6), giving rise to a family of algorithms, each parameterized by a polynomial (Sec. 2.7). Finally, the optimal polynomial family that optimizes the convergence rate for a fixed number of matrix-vector operations is obtained by solving a Chebyshev minimax problem, whose unique solution is given by the Chebyshev polynomials (Sec. 2.8).

For clarity of exposition, we simplify the synthesis procedure by replacing the constraint with $E^\top E x = E^\top q$ and by assuming $q = 0$. The first modification incurs no loss of generality, since multiplying the equality constraint by $E^\top$ leaves the feasible set unchanged. Crucially, the resulting algorithms never explicitly form $W := E^\top E$; they require only matrix–vector products with $E^\top$ and E. The second assumption, $q = 0$, is made purely for clarity—extending the framework to arbitrary q requires only a straightforward modification, as detailed in Sec. 2.7, without affecting any theoretical guarantees of the synthesized algorithms.

2.1. Design specifications

Linearity. We focus on finite-memory algorithms with linear updates in both the optimization variables and their corresponding gradients.

Explicitness. We consider single-loop algorithms that compute x^{k+1} without evaluating Ex^{k+1} or $\nabla f(x^{k+1})$. This requirement can be relaxed when linear system solvers (Ozaslan et al., 2023) or proximal operators are available along the iterations.

Optimality. We require asymptotic convergence to the unique solution $x^\star$ of problem (1). A necessary and sufficient condition for $x^\star$, when $q = 0$, is the existence of unique vectors $\delta_1^\star \in \mathbb{R}^r$ and $w_2^\star \in \mathbb{R}^{n-r}$ such that $\nabla f(x^\star) = V_1 \delta_1^\star$ and $x^\star = V_2 w_2^\star$. Here, $r < n$ denotes the rank of the matrix $W = E^\top E$, whose singular value decomposition is given by

$$W \;=\; \left[\begin{array}{cc} V_1 & V_2 \end{array}\right] \left[\begin{array}{cc} \Sigma & 0 \\ 0 & 0 \end{array}\right] \left[\begin{array}{c} V_1^\top \\ V_2^\top \end{array}\right], \;\; \Sigma \;:=\; \operatorname{diag}(\sigma_1, \ldots, \sigma_r), \;\; V \;:=\; \left[\begin{array}{cc} V_1 & V_2 \end{array}\right].$$

Linear convergence. We require linear convergence at a rate of at least $\rho \in (0, 1)$, i.e., $\|x^k - x^\star\| \leq M\|x^0 - x^\star\|\rho^k$ for all $x^0 \in \mathbb{R}^n$ and $k \geq 0$, where $M > 0$ is a constant.

2.2. Algorithm representation

Finite-memory algorithms with linear updates in both the optimization variables and their gradients can be represented as

$$z\mathcal{K}_0(z, W)\widehat{x}(z) \;=\; \mathcal{K}_1(z, W)\widehat{x}(z) \;+\; \mathcal{K}_2(z, W)\widehat{\nabla f}(z) \tag{2}$$

where $\widehat{x}(z)$ and $\widehat{\nabla f}(z)$ denote the $\mathcal{Z}$-transforms of the sequences $\{x^k\}_{k=0}^\infty$ and $\{\nabla f(x^k)\}_{k=0}^\infty$. The transfer functions are (potentially) parameterized by the constraint matrix E, or, in the particular setting of (1), by $W := E^\top E$. In this representation, $\mathcal{K}_0(z, W)$ characterizes the mapping from the gradient input to the optimization variable, while $\mathcal{K}_1(z, W)$ and $\mathcal{K}_2(z, W)$ determine how past iterates and past gradients, respectively, contribute to the update of the optimization variable.

The class of algorithms that can be expressed in the form (2) includes—but is not limited to—accelerated methods such as Nesterov's Accelerated Gradient (Nesterov, 2003); distributed algorithms such as DIGing (Nedic et al., 2017), EXTRA (Shi et al., 2015), and $\mathcal{ABN}$ (Xin et al., 2019); as well as descent–ascent type methods such as GDA (Alghunaim and Sayed, 2020) and PAPC (Salim et al., 2022a). An illustrative example is provided in Appendix A.

2.3. From algorithm to controller design

The optimization algorithms given by (2) represent a feedback interconnection between an LTI system defined by the input–output map $\widehat{x}(z) = \mathcal{H}(z, W)\widehat{\nabla f}(z)$ and a static nonlinearity Δ, where

$$\mathcal{H}(z, W) \;:=\; (z\mathcal{K}_0(z, W) \;-\; \mathcal{K}_1(z, W))^{-1}\mathcal{K}_2(z, W), \quad \Delta(e) \;:=\; \nabla f(e + x^\star) \;-\; \nabla f(x^\star) \tag{3}$$

with $e := x - x^\star$. The nonlinear operator Δ is sector bounded in $[m, L]$, i.e., $\langle \Delta(e) - me, \Delta(e) - Le \rangle \leq 0$ for all $e \in \mathbb{R}^n$. The resulting feedback interconnection forms a Lur'e system, whose block diagram is shown in Fig. 1.

The objective is to synthesize a linear controller $\mathcal{H}(z, W)$, representing the optimization algorithm, that satisfies the aforementioned design specifications. To this end, we first apply a coordinate transformation that brings the Lur'e system into a form suitable for analysis and synthesis and then obtain analytical characterizations of the design specifications.

Coordinate transformation. We utilize the singular vectors of the matrix W to introduce the change of variables $w := V^\top x$ and express the transfer function $\mathcal{H}$ as

$$\mathcal{H}(z, W) \;=\; \left[\begin{array}{cc} V_1 & V_2 \end{array}\right] \left[\begin{array}{cc} \mathcal{H}(z, \Sigma) & 0 \\ 0 & \mathcal{H}(z, 0) \end{array}\right] \left[\begin{array}{c} V_1^\top \\ V_2^\top \end{array}\right]. \tag{4}$$

The resulting subsystems are diagonal in the transformed coordinates and can thus be decomposed into scalar subsystems, each governed by the same scalar transfer function parameterized either by a singular value or by zero; i.e., $\mathcal{H}(z, \Sigma) = \operatorname{diag}\left(\{h(z, \sigma_i)\}_{i=1}^r\right)$ and $\mathcal{H}(z, 0) \;=\; h(z, 0)I_{n-r}$.

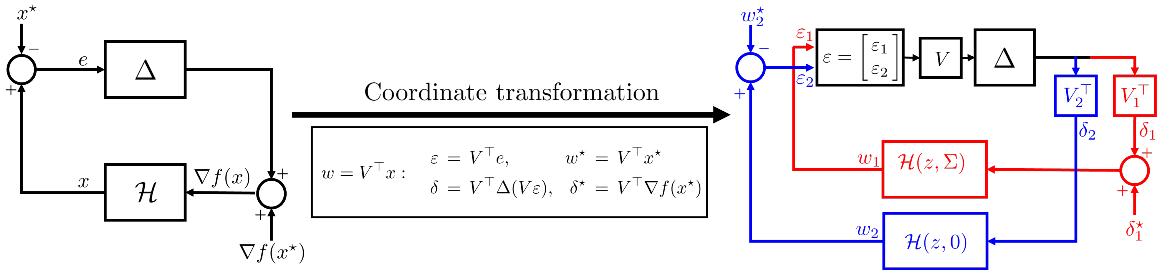

Figure 1: Left: Lur'e system representation of the optimization algorithm. Right: decomposition of the left system into two subsystems coupled through the nonlinear element.

Loop transformation. We apply the loop transformation (Khalil, 2002, Sec. 6.5) to the Lur'e system to enable the use of standard passivity-based stability criteria without introducing additional conservatism. This transformation maps the pair $(\mathcal{H}, \Delta)$, where the nonlinearity Δ is sector-bounded in $[m, L]$, to the pair $(\overline{\mathcal{H}}, \overline{\Delta})$, in which the transformed nonlinearity $\overline{\Delta}$ is sector-bounded in $[0, \infty)$. The bijection between the two transfer functions is given by

$$\mathcal{H}(z, W) = (m\overline{\mathcal{H}}(z, W) - LI)^{-1}(\overline{\mathcal{H}}(z, W) - I). \tag{5}$$

Using the same coordinate transformation as before, the transformed linear system can also be diagonalized, yielding the following bijection between the original and transformed scalar transfer functions, $h(z, \sigma) = (\overline{h}(z, \sigma) - 1)/(m\overline{h}(z, \sigma) - L)$.

The design problem thus reduces to finding a scalar transfer function $\overline{h}(z, \sigma)$ that satisfies all design specifications for every $\sigma \in [\underline{\sigma}, \overline{\sigma}] \cup \{0\}$, where $\overline{\sigma} \geq \sigma_1 \geq \sigma_r \geq \underline{\sigma} > 0$.

2.4. Analytical characterization of design specifications

Explicitness as causality. The strict causality of the linear controller in the Lur'e system provides a necessary and sufficient condition for explicitness with respect to gradient computations. The controller is strictly causal when the transfer function $\mathcal{H}(z, W)$ is strictly proper, i.e., $\mathcal{H}(\infty, W) = 0$. Under the bijection (5), this condition is equivalent to $\overline{\mathcal{H}}(\infty, W) = I$, that is,

$$\overline{h}(\infty, \sigma) = 1, \ \forall \sigma \in [\underline{\sigma}, \overline{\sigma}] \cup \{0\}. \tag{6a}$$

A necessary and sufficient condition for explicitness with respect to matrix multiplications is that the highest-order-z term in the denominator of $h(z, \sigma)$ does not multiply the parameter σ. For illustration, the transfer function $h(z, \sigma) = \frac{1}{\sigma z}$ yields implicit iterations of the form $Ex^{k+1} = x^k$.

Optimality as input tracking and disturbance rejection. The asymptotic convergence of the optimization algorithm to the optimal solution is equivalent to asymptotic tracking of the step reference signal $\frac{z}{z-1} x^\star$ (of unknown magnitude $x^\star$) and rejection of the step disturbance $\frac{z}{z-1} \nabla f(x^\star)$ (of unknown magnitude $\nabla f(x^\star)$) in the Lur'e system. By the internal model principle (Francis and Wonham, 1976), asymptotic tracking and disturbance rejection can be achieved if the subsystems $\mathcal{H}(z, \Sigma)$ and $\mathcal{H}(z, 0)$ possess a blocking zero and a pole at $z = 1$, respectively. For the transformed scalar systems, this requirement is equivalent to

$$\overline{h}(1, \sigma) = 1, \ \forall \sigma \in [\underline{\sigma}, \overline{\sigma}], \quad \overline{h}(1, 0) = L/m. \tag{6b}$$

Proposition 5 in Appendix B establishes that condition (6b) is both necessary and sufficient for input tracking and disturbance rejection.

Linear convergence as ρ-stability. The ρ-stability of the Lur'e system provides a necessary and sufficient condition for linear convergence of the corresponding algorithm. By the circle criterion (Khalil, 2002, Sec. 7.1), the transformed system $(\overline{\mathcal{H}}, \overline{\Delta})$ is ρ-stable if the transfer function $\overline{h}(\gamma z, \sigma)$ is *strictly positive real* for all $\gamma \in (\rho, 1]$. This can be equivalently expressed as

$$\text{(i) } \overline{h}(\gamma z, \sigma) \in \mathcal{RH}_\infty, \quad \text{(ii) } \operatorname{Re}(\overline{h}(\gamma z, \sigma)) > 0, \ \forall z \in \mathbb{D}^c, \quad \forall \gamma \in [\rho, 1], \ \forall \sigma \in [\underline{\sigma}, \overline{\sigma}] \cup \{0\} \tag{6c}$$

where $\mathcal{RH}_\infty$ denotes the set of all real, rational, stable transfer functions; $\mathbb{C}$ is the complex plane; $\operatorname{Re}(\cdot)$ denotes the real part of a complex number; $\mathbb{D}$ is the open unit disc $\{z \in \mathbb{C} \mid |z| < 1\}$; and $\mathbb{D}^c$ is its complement.

2.5. From controller design to interpolation

With the algorithm representation (2) and the analytical characterization of the design specifications (6), the controller design task reduces to the following algebraic problem:

$$\text{Find a } \rho \in (0,1) \text{ and design a } \overline{h}(z, \sigma) \text{ such that conditions (6) hold for every } \gamma \in (\rho, 1]. \tag{AP}$$

The next step is to apply a transformation to $\overline{h}$ so that conditions (6) are expressed over the unit disc. To this end, we employ the following conformal transformation (Rudin, 1987, Chs. 14–15),

$$\psi(z, \sigma) \;=\; \theta(\overline{h}(\gamma z^{-1}, \sigma))$$

where $\theta(z) = \frac{z-1}{z+1}$ maps the right-half plane to the unit disc and has the inverse map $\theta^{-1}(z) = \frac{1+z}{1-z}$, while the reciprocal z^{-1} maps the unit disc to its complement and vice versa. This transformation allows us to restate (AP) as the following interpolation problem:

$$\text{Given data pairs } \{(z_i, w_i)\}_{i=1}^N, \text{ find a function } \psi \colon \mathbb{D} \to \mathbb{D} \text{ satisfying } \psi(z_i) = w_i. \tag{IP}$$

The interpolation problem is then solved using the Nevanlinna–Pick (NP) method (Doyle et al., 1992, Ch. 9), which provides: (i) a criterion for the existence of a solution—specifically, a lower bound on ρ via the Pick matrix; and (ii) a parametrization of all admissible transfer functions, i.e., all algorithms satisfying the design specifications, through an iterative construction based on Schur's algorithm. In this construction, the transfer functions are represented as compositions of Möbius transformations (Rudin, 1987, Ch. 14) and a *free function* that can be any stable and bounded transfer function with no further restrictions.

2.6. Nevanlinna–Pick interpolation

Direct application of the NP interpolation technique to problem (AP) is not possible because of the conditional statement (6b) for different values of σ. Since the interpolation condition (6b) involves two cases ($\sigma \in [\underline{\sigma}, \overline{\sigma}]$ or $\sigma = 0$), we construct two candidate solutions, $(\rho_1, \overline{h}_1(z))$ and $(\rho_2, \overline{h}_2(z))$, by applying conditions (6) to $\sigma \in [\underline{\sigma}, \overline{\sigma}]$ and to $\sigma = 0$, respectively. The two candidates are then combined to obtain a solution to problem (AP). The following propositions are direct consequence of NP interpolation applied to the individual subproblems; their proofs are thus omitted.

Proposition 1 *For any parameter $\sigma \in [\underline{\sigma}, \overline{\sigma}]$ and convergence rate $\rho_1 \in (0,1)$, all transfer functions satisfying the interpolation conditions (6) can be parameterized in terms of a free function $g_1(z) \colon \mathbb{D}^c \to \mathbb{D}$ as*

$$\overline{h}_1(\gamma z) \;=\; \frac{\gamma z(\gamma z - \rho_1^2) \;+\; \rho_1^2\, g_1(\gamma z/\rho_1)(\gamma z - 1)}{\gamma z(\gamma z - \rho_1^2) \;-\; \rho_1^2\, g_1(\gamma z/\rho_1)(\gamma z - 1)}. \tag{7}$$

Proposition 2 *A transfer function satisfying the interpolation conditions (6) for $\sigma = 0$ exists if and only if $\rho_2 \geq \rho_{gd} := (\kappa_f - 1)/(\kappa_f + 1)$. For any convergence rate $\rho_2 \in [\rho_{gd}, 1]$, all such transfer functions can be parameterized by a free function $g_2(z)\colon \mathbb{D}^c \to \mathbb{D}$ as*

$$\overline{h}_2(\gamma z) \;=\; \frac{(\gamma z \rho_{gd} + \rho_2^2)(\gamma z - 1)g_2(\gamma z/\rho_2) \;+\; (\gamma z + \rho_{gd})(\gamma z - \rho_2^2)}{(\gamma z \rho_{gd} - \rho_2^2)(\gamma z - 1)g_2(\gamma z/\rho_2) \;+\; (\gamma z - \rho_{gd})(\gamma z - \rho_2^2)}. \tag{8a}$$

Moreover, for $\rho_2 = \rho_{gd}$, the transfer function satisfying all constraints is unique and given by

$$\overline{h}_2(\gamma z) = (\gamma z + \rho_{gd})/(\gamma z - \rho_{gd}). \tag{8b}$$

When the interpolation conditions in (6) are separated into the two cases $\sigma > 0$ and $\sigma = 0$, the resulting interpolators depend only on this dichotomy and not on the specific value of σ. However, to accommodate both scenarios within a unified design, the transfer function $\overline{h}$ must remain parameterized by σ. Our approach is therefore to set the interpolator $\overline{h}$ equal to $\overline{h}_1$ while modeling the convergence rate ρ_1 within $\overline{h}_1$ as a polynomial in σ, namely,

$$\overline{h}(\gamma z, \sigma) \;=\; \frac{\gamma z(\gamma z - \rho_1^2(\sigma)) \;+\; \rho_1^2(\sigma) g_1(\gamma z/\rho_1(\sigma))(\gamma z - 1)}{\gamma z(\gamma z - \rho_1^2(\sigma)) \;-\; \rho_1^2(\sigma) g_1(\gamma z/\rho_1(\sigma))(\gamma z - 1)} \tag{9}$$

subject to the following constraints on the polynomial:

$$\text{(i) } \rho_1(\sigma) \in [0, 1),\ \sigma > 0;\ \ \text{(ii) } \rho_1(0) \;=\; 1. \tag{10a}$$

The polynomial paramterization ensures that the resulting algorithm avoids matrix decompositions and requires only matrix-vector multiplications. Under constraint (10a)-(i), Proposition 1 guarantees that $\overline{h}$ satisfies conditions (6) for any $\sigma \in [\underline{\sigma}, \overline{\sigma}]$ and $\gamma \in (\bar{\rho}_1, 1]$, where

$$\bar{\rho}_1 \;:=\; \max_{\sigma \in [\underline{\sigma}, \overline{\sigma}]} \; \rho_1(\sigma). \tag{10b}$$

Moreover, under constraint (10a)-(ii), $\overline{h}$ at $\sigma \;=\; 0$ reduces to $\overline{h}(\gamma z, 0) \;=\; (\gamma z + g_1(\gamma z))/(\gamma z - g_1(\gamma z))$. As a result, if the free function is chosen as a constant, $g_1(z) \;=\; \rho_{gd}$, then Proposition 2 guarantees that $\overline{h}$ satisfies (6) also for $\sigma = 0$ and for any $\gamma \in (\rho_{gd}, 1]$.

Combining Propositions 1 and 2, the following transfer function provides a solution to problem (AP) with convergence rate $\rho = \max(\bar{\rho}_1, \rho_{gd})$,

$$\overline{h}(\gamma z, \sigma) \;=\; \frac{\gamma z(\gamma z \;-\; \rho_1^2(\sigma)) \;+\; \rho_1^2(\sigma)\,(\gamma z - 1)\rho_{gd}}{\gamma z(\gamma z \;-\; \rho_1^2(\sigma)) \;-\; \rho_1^2(\sigma)\,(\gamma z - 1)\rho_{gd}}. \tag{11}$$

2.7. Algorithm synthesis

Substitution of (11) into (5) yields the original scalar transfer function

$$h(z, \sigma) \;=\; -\tfrac{2}{L+m}\, \rho_1^2(\sigma)(z \;-\; 1)\big/(z^2 \;-\; 2\rho_1^2(\sigma) z \;+\; \rho_1^2(\sigma))$$

The inverse coordinate transformation, illustrated in Fig. 1, recovers the matrix transfer function

$$\mathcal{H}(z, W) \;=\; -\tfrac{2}{L+m}\, \rho_1^2(W)(z \;-\; 1)(z^2 I \;-\; 2\rho_1^2(W) z \;+\; \rho_1^2(W))^{-1}$$

which, together with $\widehat{x}(z) = \mathcal{H}(z, W)\widehat{\nabla f}(z)$, yields the update rule

$$z\widehat{x} \;=\; \rho_1^2(W)\Big((2 \,-\, z^{-1})\widehat{x} \,-\, \tfrac{2}{L+m}(1 \,-\, z^{-1})\widehat{\nabla f}\Big)\,.$$

Finally, taking the inverse $\mathcal{Z}$-transform gives the time-domain recursion

$$x^{k+1} \;=\; \rho_1^2(W)\Big(2x^k \,-\, x^{k-1} - \tfrac{2}{L+m}\big(\nabla f(x^k) \,-\, \nabla f(x^{k-1})\big)\Big) \tag{12}$$

whose convergence rate is upper bounded by $\rho = \max(\bar{\rho}_1, \rho_{gd})$, provided that $\bar{\rho}_1$ satisfies the constraints (10a). A family of methods described by (12), referred to as the *Interpolated Gradient Methods* (I-GM), is obtained for different choices of the polynomial function $\rho_1(\cdot)$; see, e.g., (13).

Step size selection. The requirement of exactly knowing the strong convexity parameter m in (12) can be circumvented by introducing a step size $\alpha_1 := 2/(\hat{L} + m)$, where $\hat{L}$ is an overestimate of the Lipschitz constant. For any $\alpha_1 \in (0, 2/(L+m))$, there exists $\hat{L}$ such that $\rho_{gd} = (\hat{L} - m)/(\hat{L} + m) = 1 - \alpha_1 m$. Hence, without loss of generality and without affecting convergence guarantees, we replace the term $2/(L+m)$ in (12) with a generic step size $\alpha_1 \in (0, 2/(L+m))$ and use $\rho_2 = 1 - \alpha_1 m$ as the corresponding convergence rate.

Nonhomogeneous constraints. When $q \neq 0$, the unique solution to problem (1) decomposes as $x^\star = x^\star_{\shortparallel} + x^\star_\perp$, where $x^\star_{\shortparallel}$ is a particular solution to $E^\top E x^\star_{\shortparallel} = E^\top q$ with $x^\star_{\shortparallel} \in \mathcal{R}(E^\top)$, and $x^\star_\perp$ is the null-space component, i.e., $x^\star_\perp \in \mathcal{N}(E)$. To maintain orthogonality between the two exogenous inputs, we replace $x^\star$ in Fig. 1 with $x^\star_\perp$, and substitute the matrix–vector multiplication $Wx = E^\top E x$ in (12) with the affine transformation $E^\top E(x - x^\star_{\shortparallel})$, which is equivalent to $E^\top(Ex - q)$. This ensures that the output of the LTI system $\mathcal{H}$ is $x - x^\star_{\shortparallel}$, and hence the error takes the correct form $e = x - x^\star_{\shortparallel} - x^\star_\perp = x - x^\star$.

Polynomial design. The final step is to design a polynomial $\rho_1(\sigma)$ that satisfies the constraints (10a). A natural choice is to adopt a monomial form,

$$\rho_1(\sigma) \;=\; (1 \,-\, \alpha_2\sigma)^\ell \tag{13}$$

with a (possibly noninteger) exponent $\ell > 0$ and a step size $\alpha_2 \in (0, , \frac{1}{\bar{\sigma}})$. Substituting (13) into the recursion (12) yields the *Monomial I-GM* algorithm, whose convergence rate is upper bounded by $\rho = \max(1 - \alpha_1 m, (1 - \alpha_2\underline{\sigma})^\ell)$. We refer to Ozaslan et al. (2025a) for further details.

2.8. Chebyshev polynomials

Despite the elegance of the resulting algorithms, the monomial form (13) is suboptimal with respect to the convergence rate $\bar{\rho}_1$ achieved for a given polynomial degree ℓ, which determines the number of matrix–vector multiplications per iteration. It is therefore natural to seek an optimal polynomial family that minimizes the convergence rate (10b) for a fixed ℓ under the constraints (10a). Chebyshev's Alternation Theorem implies that the Chebyshev polynomials provide the unique solution.

Lemma 3 *(Saad, 2003, Thm 6.25) Let $\mathbb{P}_\ell$ denote the set of all polynomials of degree ℓ. For a nonempty interval $[\underline{\sigma}, \bar{\sigma}]$, let $\zeta = (\sqrt{\bar{\sigma}} - \sqrt{\underline{\sigma}})/(\sqrt{\bar{\sigma}} + \sqrt{\underline{\sigma}})$. The minimax solution to the problem*

$$\min_{p\in\mathbb{P}_\ell,\;\, p(0)=1} \;\; \max_{\sigma\in[\underline{\sigma},\bar{\sigma}]} \;|p(\sigma)| \;=\; 2\zeta^\ell/(1 + \zeta^{2\ell}) \;=\; 1/\mathcal{C}_\ell((\underline{\sigma} + \bar{\sigma})/(\bar{\sigma} - \underline{\sigma})) \tag{14a}$$

is achieved by the shifted Chebyshev polynomial

$$\widehat{\mathcal{C}}_\ell(\sigma) \;:=\; \mathcal{C}_\ell((\bar{\sigma} + \underline{\sigma} - 2\sigma)/(\bar{\sigma} - \underline{\sigma}))/\mathcal{C}_\ell((\bar{\sigma} + \underline{\sigma})/(\bar{\sigma} - \underline{\sigma})) \tag{14b}$$

where $\mathcal{C}_\ell(\sigma)$ denotes the Chebyshev polynomial of the first kind of degree ℓ.

Since the shifted Chebyshev polynomials are not guaranteed to take positive values, we choose ρ_1 as

$$\rho_1(\sigma) \;=\; (1 \;-\; \alpha_2(1 \;-\; \widehat{C}_\ell(\sigma)))^{1/2} \tag{15}$$

where selecting α_2 such that $0 \le \alpha_2 \le \frac{1}{2} \le \max_\sigma \; (1 - \widehat{C}_\ell(\sigma))^{-1}$ ensures constraint (10a)-(i) is satisfied. The square-root in (15) is canceled by the square power in (12), and substitution of (15) into (12) yields Algorithm 2, in which the shifted Chebyshev polynomials are evaluated using Chebyshev iterations given in Algorithm 1 (Barrett et al., 1994, Sec. 2.3.9).

Algorithm 1: Chebyshev iterations

Data: $y^0 = w^k$, E, q, ℓ, $\overline{\sigma}$, $\underline{\sigma}$

$c_1 = \frac{(\overline{\sigma}-\underline{\sigma})^2}{16}, \quad c_2 = -2a^0 = \frac{\overline{\sigma}+\underline{\sigma}}{2}$

$p^0 = -E^\top(Ey^0 - q)/c_2, \quad y^1 = y^0 + p^0$

for $k = 1, 2, \ldots, \ell$ **do**

- $b^k = c_1/a^{k-1}, \quad a^k = -(c_2 + b^k)$
- $p^k = -(E^\top(Ey^k - q) + b^k p^{k-1})/a^k$
- $y^{k+1} = y^k + p^k$

Result: $y^{\ell+1}$

Algorithm 2: Chebyshev I-GM

Data: $v^0 = x^0$, $\alpha_1 \in (0, \frac{1}{L})$, $\alpha_2 \in (0, \frac{1}{2})$, $\ell > 0$, $\overline{\sigma}$, $\underline{\sigma}$

for $k = 1, 2, \ldots$ **do**

- $v^k = x^{k-1} - \alpha_1 \nabla f(x^{k-1})$
- $w^k = x^{k-1} + v^k - v^{k-1}$
- $u^k = \text{Chebyshev}(w^k, E, q, \ell, \overline{\sigma}, \underline{\sigma})$
- $x^{k+1} = (1 - \alpha_2) w^k + \alpha_2 u^k$

Theorem 4 provides a lower bound on the polynomial degree ℓ required for Algorithm 2 to achieve a worst-case convergence rate $\rho = 1 - \alpha_1 m$, which is independent of the estimated condition number $\overline{\kappa}_E$ of the constraint matrix. Since the convergence rate of the recursion (12) is determined by $\max(\kappa_f, \overline{\kappa}_E)$, we can choose ℓ in Lemma 3 proportional to $O(\sqrt{\overline{\kappa}_E/\kappa_f})$. This choice effectively reduces the influence of $\overline{\kappa}_E$ to the level of κ_f, in sharp contrast to existing methods such as Scaman et al. (2017); Salim et al. (2022a), which require at least $O(\sqrt{\overline{\kappa}_E})$ Chebyshev iterations to eliminate dependence on $\overline{\kappa}_E$ regardless of whether $\kappa_f \le \overline{\kappa}_E$. The proof of Theorem 4 is given in Appendix C.

Theorem 4 *Let the cost function f in problem (1) be m-strongly convex with an L-Lipschitz continuous gradient, and let $W = E^\top E$ be a matrix whose singular values lie in $[\underline{\sigma}, \overline{\sigma}] \cup \{0\}$. For the step size $\alpha_1 \le 2/(m+L)$, the convergence rate of Algorithm 2 is upper bounded by $\rho_c = 1 - \alpha_1 m$ if the following conditions are satisfied*

$$\ell \;\ge\; \frac{\sqrt{\overline{\kappa}_E} + 1}{\sqrt{(2\alpha_1 m)^{-1}} - 1}, \quad \alpha_2 \;=\; \frac{1 + \zeta^{2\ell}}{(1 + \zeta^\ell)^2}, \quad \textit{where} \quad \zeta \;:=\; \frac{\sqrt{\overline{\kappa}_E} - 1}{\sqrt{\overline{\kappa}_E} + 1}, \quad \overline{\kappa}_E \;:=\; \frac{\overline{\sigma}}{\underline{\sigma}}.$$

The upper bound on the convergence rate is minimized by setting $\alpha_1 = 2/(m + L)$, which yields $\rho = (\kappa_f - 1)/(\kappa_f + 1)$ and $\ell \ge 2(\sqrt{\overline{\kappa}_E} + 1)/(\sqrt{\kappa_f} - 2)$.

This completes the characterization of the convergence properties of Algorithm 2.

3. Computational experiments

Inspired by Salim et al. (2022a), we conduct a compressed-sensing type computational experiment[1]. with the cost function $f(x) = \sum_{i=1}^n \bar{f}(x_i)$, and

$$\bar{f}(x_i) \;=\; (x_i^2 \;+\; 1/(L \;-\; m)^2)^{1/2} \;+\; \tfrac{m}{2}\, x_i^2$$

which can be interpreted as an m-strongly convex and L-smooth approximation of the ℓ_1-norm. For the constraint $Ex = q$, we randomly sample a sparse vector $\overline{x}$ with 50 nonzero entries (each equal

1. Source code to reproduce experiments is available at `https://github.com/ibrahimkurban/I-GM.git`

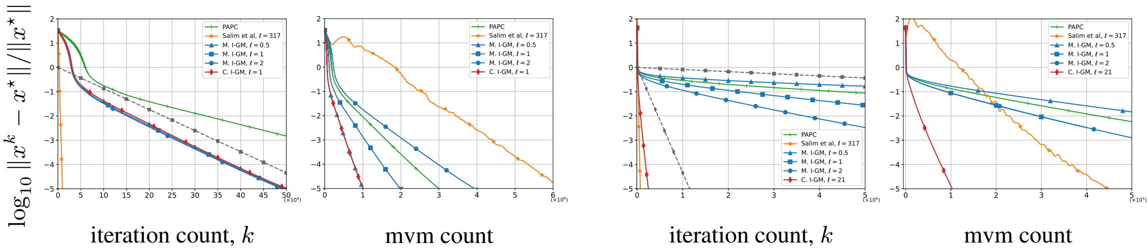

Figure 2: Error plots versus the number of iterations and matrix–vector multiplications (mvm) for: (left) $(\kappa_f, \kappa_E) = (10^5, 10^5)$; (right) $(\kappa_f, \kappa_E) = (10^3, 10^5)$. The gray dashed line with square markers indicates the theoretical convergence rate ρ_m of the Monomial I-GM for $\ell = 1$, while the line with diamond markers shows the theoretical rate ρ_c of the Chebyshev I-GM. The two rates coincide in the left plot.

to 1) and generate the matrix $E \in \mathbb{R}^{d\times n}$ with entries drawn from the standard normal distribution. After scaling the singular values of E to lie within the interval $[\sqrt{\underline{\sigma}}, \sqrt{\overline{\sigma}}] \cup \{0\}$, we set $q = E\overline{x}$.

We set the dimensions (N, d, n) to $(10^3, 250, 10^3)$, the rank of E to $r = 200$, and run the algorithms on two problem instances with different condition numbers; see Figure 2. In the first instance, we set $(L, m, \overline{\sigma}, \underline{\sigma}) = (10^4, 10^{-1}, 10^2, 10^{-3})$, yielding $\kappa_f = \kappa_E = 10^5$. In the second instance, we use $(10^1, 10^{-2}, 10^2, 10^{-3})$, yielding $\kappa_f = 10^3$ and $\kappa_E = 10^5$.

Consistent with our analysis, increasing ℓ reduces the number of Monomial I-GM iterations required to reach a given level of accuracy. For instance, increasing ℓ from 1 to 2 approximately halves the iteration count, in close agreement with the worst-case convergence rate ρ_m predicted by (13). Notably, the theoretical bound ρ_m matches the empirical convergence rate with remarkable accuracy, suggesting that the bound is tight.

Moreover, while both the algorithm proposed in Salim et al. (2022a) and the Chebyshev I-GM achieve the optimal lower bound on communication complexity, our Chebyshev I-GM requires fewer matrix–vector multiplications (which correspond to broadcast and synchronization operations in distributed settings) to attain the same level of accuracy. This advantage becomes even more pronounced when $\kappa_f \geq \kappa_E$. In this regime, the convergence rate of Chebyshev I-GM no longer benefits from additional matrix–vector multiplications, and only one is performed per iteration. In contrast, the method of Salim et al. (2022a) requires $O(\sqrt{\kappa_E})$ multiplications per iteration regardless of the relation between the condition numbers and empirically diverges if the number of iterations is reduced. On the other hand, the iteration complexity of the I-GM family is at best $O(\kappa_f \log \frac{1}{\epsilon})$, requiring additional gradient evaluations relative to the method of Salim et al. (2022a). Which metric dominates ultimately depends on the application: in some settings, communication is the primary bottleneck, whereas in others, the cost of gradient evaluations is the main concern.

4. Discussion and conclusion

Despite achieving optimal communication complexity, the iteration complexity of the I-GM family remains suboptimal. This limitation arises from the conservatism of the circle criterion. A natural remedy is to incorporate Zames–Falb multipliers into the analysis (Carrasco et al., 2016). While they integrate seamlessly into our framework, these multipliers restrict the class of admissible nonlinearities. Removing these restrictions while preserving the benefits of Zames–Falb analysis is part of our ongoing work. We are also extending the framework to handle nonsmooth problems through proximal operators which integrate naturally into the proposed synthesis procedure.

References

S. A. Alghunaim and A. H. Sayed. Linear convergence of primal-dual gradient methods and their performance in distributed optimization. *Automatica*, 117:109003, 2020.

R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, 1994.

S. Boyd, T. Parshakova, E. Ryu, and J. J. Suh. Optimization algorithm design via electric circuits. *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 37:68013–68081, 2024.

J. Carrasco, M. C. Turner, and W. P. Heath. Zames–Falb multipliers for absolute stability: From O'Shea's contribution to convex searches. *Eur. J. Control*, 28:1–19, 2016.

A. Chambolle and T. Pock. An introduction to continuous optimization for imaging. *Acta Numer.*, 25:161–319, 2016.

L. Condat. A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *J. Optim. Theory Appl.*, 158(2):460–479, 2013.

D. Ding and M. R. Jovanović. Global exponential stability of primal-dual gradient flow dynamics based on the proximal augmented Lagrangian: A Lyapunov-based approach. In *Proc. IEEE Conf. Decis. Control*, pages 4836–4841, 2020.

J. C. Doyle, B. A. Francis, and A. R. Tannenbaum. *Feedback Control Theory*. Macmillan, New York, NY, 1992.

B. A. Francis and W. M. Wonham. The internal model principle of control theory. *Automatica*, 12 (5):457–465, 1976.

H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, NJ, 2002.

D. Kovalev, A. Salim, and P. Richtarik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 33:18342–18352, 2020.

Guanghui Lan, Soomin Lee, and Yi Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *Math. Program.*, 180(1):237–284, 2020.

L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. Control Optim.*, 26(1):57–95, 2016.

H. Li and Z. Lin. Revisiting extra for smooth distributed optimization. *SIAM J. Optim.*, 30(3): 1795–1821, 2020.

T. Li, A. K. Sahu, A. Talwalkar, and V. Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.*, 37(3):50–60, 2020.

S. Michalowsky, C. Scherer, and C. Ebenbauer. Robust and structure exploiting optimisation algorithms: An integral quadratic constraint approach. *Internat. J. Control*, 94(11):2956–2979, 2021.

M. Muehlebach and M. I. Jordan. A dynamical systems perspective on Nesterov acceleration. In *Proc. Int. Conf. Mach. Learn.*, pages 4656–4662, 2019.

A. Nedic, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM J. Optim.*, 27(4):2597–2633, 2017.

Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Springer, New York, NY, 2003.

I. K. Ozaslan and M. R. Jovanović. Accelerated forward–backward and Douglas–Rachford splitting dynamics. *Automatica*, 175:112210, 2025.

I. K. Ozaslan, M. Pilanci, and O. Arikan. M-IHS: An accelerated randomized preconditioning method avoiding costly matrix decompositions. *Linear Algebra Appl.*, 678:57–91, 2023.

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.

I. K. Ozaslan, T. T. Georgiou, and M. R. Jovanović. Automated algorithm design via Nevanlinna–Pick Interpolation. *arXiv:2509.21416*, 2025a. (to appear in NeurIPS DynaFront Workshop).

I. K. Ozaslan, W. Wu, J. Chen, T. T. Georgiou, and M. R. Jovanović. Automated algorithm design for convex optimization problems with linear equality constraints. *arXiv:2509.20746*, 2025b. (to appear in IEEE Conf. Decis. Control).

G. Peyre and M. Cuturi. Computational optimal transport: With applications to data science. *Found. Trends Mach. Learn.*, 11(5-6):355–607, 2019.

G. Qu and N. Li. On the exponential stability of primal-dual gradient dynamics. *IEEE Control Syst. Lett.*, 3(1):43–48, 2019a.

G. Qu and N. Li. Accelerated distributed Nesterov gradient descent. *IEEE Trans. Automat. Control*, 65(6):2566–2581, 2019b.

Walter Rudin. *Real and Complex Analysis*. McGraw-Hill, New York, 3rd edition, 1987.

Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

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.*, pages 4482–4498, 2022a.

A. Salim, L. Condat, K. Mishchenko, and P. Richtarik. Dualize, split, randomize: Toward fast nonsmooth optimization algorithms. *J. Optim. Theory Appl.*, 195(1):102–130, 2022b.

K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulie. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proc. Int. Conf. Mach. Learn.*, pages 3027–3036. PMLR, 2017.

C. Scherer and C. Ebenbauer. Convex synthesis of accelerated gradient algorithms. *SIAM J. Control Optim.*, 59(6):4615–4645, 2021.

C. Scherer, C. Ebenbauer, and T. Holicki. Optimization algorithm synthesis based on integral quadratic constraints: A tutorial. In *Proc. IEEE Conf. Decis. Control*, pages 2995–3002, 2023.

W. Shi, Q. Ling, G. Wu, and W. Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM J. Optim.*, 25(2):944–966, 2015.

H. Taha, C. Gonzalez, and M. Shorbagy. A minimization principle for incompressible fluid mechanics. *Phys. Fluids*, 35(12), 2023.

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*, 155:111129, 2023.

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.*, 2(1):49–54, 2017.

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

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.

R. Xin, D. Jakovetic, and U. A. Khan. Distributed Nesterov gradient methods over arbitrary graphs. *IEEE Signal Process. Lett.*, 26(8):1247–1251, 2019.

J. Xu, Y. Tian, Y. Sun, and G. Scutari. Distributed algorithms for composite optimization: Unified framework and convergence analysis. *IEEE Trans. Signal Process.*, 69:3555–3570, 2021.

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*, 69(12):8539–8554, 2024.

T. Zheng, L. Zhu, A. M. So, J. Blanchet, and J. Li. Universal gradient descent ascent method for nonconvex-nonconcave minimax optimization. *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 36:54075–54110, 2023.

Appendix A. Algorithm representation

In this section, we demonstrate how formulation (2) encompasses a wide range of algorithms including those with primal-dual updates. As a concrete instance, consider Proximal Alternating Predictor-Corrector (PAPC) algorithm (Salim et al., 2022a):

$$x^{k+1/2} = x^k - \alpha_1 \nabla f(x^k) - \alpha_1 E^\top y^k \tag{16a}$$
$$y^{k+1} = y^k + \alpha_2 (E x^{k+1/2} - q) \tag{16b}$$
$$x^{k+1} = x^k - \alpha_1 \nabla f(x^k) - \alpha_1 E^\top y^{k+1}. \tag{16c}$$

Replacing the equality constraint with $E^\top E x = E^\top q$, i.e., multiplying the dual update (16b) by $E^\top$, we introduce a new "dual" variable $v^k = E^\top y^k$, which reduces the PAPC algorithm to

$$x^{k+1/2} = x^k - \alpha_1 \nabla f(x^k) - \alpha_1 v^k \tag{17a}$$
$$v^{k+1} = v^k + \alpha_2 E^\top (E x^{k+1/2} - q) \tag{17b}$$
$$x^{k+1} = x^k - \alpha_1 \nabla f(x^k) - \alpha_1 v^{k+1}. \tag{17c}$$

Notably, all variables in (17) have the same dimension as primal variable x. While matching dimensions is not required for our representation (2), it substantially simplifies the transfer function structure by avoiding block matrix forms. For clarity of exposition, we also assume $q = 0$ (this assumption can be avoided by defining an affine operator $W(x) = E^\top(Ex - q)$ as discussed in Section (2.7)). Substituting the intermediate variable $x^{k+1/2}$ into (17b), we obtain the following primal-dual updates:

$$v^{k+1} = (I - \alpha_1 \alpha_2 E^\top E) v^k + \alpha_2 E^\top E (x^k - \alpha_1 \nabla f(x^k)) \tag{18a}$$
$$x^{k+1} = x^k - \alpha_1 \nabla f(x^k) - \alpha_1 v^{k+1}. \tag{18b}$$

In the most general form, any primal-dual update (for solving problem (1)) that is linear both in optimization variable and gradients, can be represented as

$$z \widehat{x} = \mathcal{K}_1(z,E) \widehat{x} + \mathcal{K}_2(z,E) \widehat{\nabla f} + \mathcal{K}_0(z,E) \widehat{v} \tag{19a}$$
$$z \widehat{v} = \mathcal{L}_0(z,E) \widehat{v} + \mathcal{L}_1(z,E) \widehat{x} + \mathcal{L}_2(z,E) \widehat{\nabla f}. \tag{19b}$$

For instance, the primal version of PAPC algorithm (18), a.k.a., PriLiCo (Salim et al., 2022b), corresponds to the following transfer function definitions,

$$\begin{aligned} \mathcal{K}_0(z,E) &= -\alpha_1 z I, & \mathcal{L}_0(z,E) &= I - \alpha_1 \alpha_2 E^\top E \\ \mathcal{K}_1(z,E) &= I, & \mathcal{L}_1(z,E) &= \alpha_2 E^\top E \\ \mathcal{K}_2(z,E) &= \alpha_1 I, & \mathcal{L}_2(z,E) &= -\alpha_1 \alpha_2 E^\top E. \end{aligned}$$

To establish a connection with our representation (2), we express $\widehat{v}$ in terms of $\widehat{x}$ and $\widehat{\nabla f}$,

$$\widehat{v} = (zI - \mathcal{L}_0)^{-1} (\mathcal{L}_1 \widehat{x} + \mathcal{L}_2 \widehat{\nabla f}) \tag{20a}$$
$$= \widetilde{\mathcal{L}}_1 \widehat{x} + \widetilde{\mathcal{L}}_2 \widehat{\nabla f} \tag{20b}$$

where, omitting the arguments, $\widetilde{\mathcal{L}}_1 := (zI - \mathcal{L}_0)^{-1}\mathcal{L}_1$ and $\widetilde{\mathcal{L}}_2 := (zI - \mathcal{L}_0)^{-1}\mathcal{L}_2$. Substituting (20b) into (19a) yields

$$\begin{aligned} z\widehat{x} &= (\mathcal{K}_1 + \mathcal{K}_0\widetilde{\mathcal{L}}_1)\widehat{x} + (\mathcal{K}_2 + \mathcal{K}_0\widetilde{\mathcal{L}}_2)\widehat{\nabla f} && \text{(21a)} \\ &= \widetilde{\mathcal{K}}_1\widehat{x} + \widetilde{\mathcal{K}}_2\widehat{\nabla f} && \text{(21b)} \end{aligned}$$

where $\widetilde{\mathcal{K}}_1 := \mathcal{K}_1 + \mathcal{K}_0\widetilde{\mathcal{L}}_1$ and $\widetilde{\mathcal{K}}_2 := \mathcal{K}_2 + \mathcal{K}_0\widetilde{\mathcal{L}}_2$.

The reduction from (19) to (21) confirms that algorithms following the primal-dual structure (19) can be equivalently cast in the primal-only form (21), thereby establishing the generality of representation (2).

Appendix B. Necessity of the internal model principle

Proposition 5 *Let the feedback interconnection in Figure 1 be stable for any sector bounded nonlinearity Δ in $[m, L]$. The error asymptotically converges to zero, i.e., $\lim_{k\to\infty} e^k = 0$, if and only if the transformed scalar transfer function satisfies*

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

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

$$\widehat{e}(z) + \tfrac{z}{z-1}\, x^\star = \mathcal{H}(z, W)\big(\widehat{\Delta}(z) + \tfrac{z}{z-1}\,\nabla f(x^\star)\big)$$

where $\widehat{e}(z)$ and $\widehat{\Delta}(z)$ are the $\mathcal{Z}$-transforms of the sequences $\{e^k\}_{k=0}^\infty$ and $\{\Delta(e^k)\}_{k=0}^\infty$, respectively. Relation (5) yields

$$\begin{aligned} &(m\overline{\mathcal{H}}(z, W) - LI)\big((z-1)\widehat{e} + zx^\star\big) \\ &\qquad = (\overline{\mathcal{H}}(z, W) - I)\big((z-1)\widehat{\Delta}(z) + z\nabla f(x^\star)\big). \end{aligned}$$

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

$$(m\overline{\mathcal{H}}(1, W) - LI)x^\star = (\overline{\mathcal{H}}(1, W) - I)\nabla f(x^\star).$$

Coordinate transformation (4) in conjunction with the optimality conditions gives

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

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

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

■

Appendix C. Proof of Theorem 4

The convergence rate of recursion (12) for any polynomial $\rho_1(\cdot)$ satisfying conditions (10a) is, by construction, upper bounded by $\rho = \max(\bar{\rho}_1, 1 - \alpha_1 m)$. The polynomial (15) satisfies (10a)-(ii) since $\mathcal{C}_\ell(0) = 1$ for any $\ell > 0$. Moreover, constraint (10a)-(i) enforces

$$\begin{aligned}
\rho_1(\sigma) = (1 \,-\, \alpha_2(1 \,-\, \widehat{C}_\ell(\sigma)))^{1/2} \in (0,1) \quad \forall \sigma \in [\underline{\sigma}, \overline{\sigma}]& \\
\iff \max_{\sigma\in[\underline{\sigma},\overline{\sigma}]} \alpha_2(1 \,-\, \widehat{C}_\ell(\sigma)) \,\le\, 1& \\
\iff \alpha_2(1 \,+\, \max_{\sigma\in[\underline{\sigma},\overline{\sigma}]} \widehat{C}_\ell(\sigma)) \,\le\, 1& \\
\iff \alpha_2 \,\le\, \left(1 \,+\, \max_{\sigma\in[\underline{\sigma},\overline{\sigma}]} \widehat{C}_\ell(\sigma)\right)^{-1}& \\
\iff \alpha_2 \,\le\, (1+\zeta^{2\ell})/(1+\zeta^\ell)^2&
\end{aligned}$$

where the last equivalence follows from Lemma 3. To minimize the upper bound on the convergence rate, therefore, we set $\alpha_2 = (1+\zeta^{2\ell})/(1+\zeta^\ell)^2$. While we can achieve an arbitrarily small $\bar{\rho}_1$ by choosing sufficiently large ℓ, the best convergence rate achieved by (12) is restricted by $1 - \alpha_1 m$. Hence, in what follows, we find a lower bound on ℓ such that $\bar{\rho}_1 \le 1 - \alpha_1 m$, which suffices to guarantee the worst case convergence rate of $\rho_c = 1 - \alpha_1 m$:

$$\begin{aligned}
\bar{\rho}_1 \le 1 - \alpha_1 m \iff\ & \bar{\rho}_1^2 \le (1-\alpha_1 m)^2 \\
\iff\ & 1 \,-\, \alpha_2(1 \,-\, \widehat{C}_\ell(\sigma)) \,\le\, 1 \,-\, 2\alpha_1 m + \alpha_1^2 m^2 \\
\Longleftarrow\ & 1 \,-\, \alpha_2(1 \,-\, \widehat{C}_\ell(\sigma)) \,\le\, 1 \,-\, 2\alpha_1 m \\
\iff\ & \mathcal{X} \,:=\, 2\alpha_1 m \,\le\, \alpha_2(1 \,-\, \widehat{C}_\ell(\sigma)) \\
\Longleftarrow\ & \mathcal{X} \,\le\, \alpha_2(1 \,-\, \max_{\sigma\in[\underline{\sigma},\overline{\sigma}]} \widehat{C}_\ell(\sigma))) \\
\iff\ & \sqrt{\mathcal{X}} \,\le\, \frac{1-\zeta^\ell}{1+\zeta^\ell} \,:=\, \tanh(\theta) \quad \text{where} \quad \ell \ln \zeta \,=\, -2\theta \\
\iff\ & \operatorname{arctanh}(\sqrt{\mathcal{X}}) \,\le\, \theta \\
\iff\ & \frac{2\operatorname{arctanh}(\sqrt{\mathcal{X}})}{\ln(\zeta^{-1})} \,\le\, \ell \\
\iff\ & \frac{\ln\left(\dfrac{1+\sqrt{\mathcal{X}}}{1-\sqrt{\mathcal{X}}}\right)}{\ln(\zeta^{-1})} \,\le\, \ell \\
\Longleftarrow\ & \frac{\dfrac{2\sqrt{\mathcal{X}}}{1-\sqrt{\mathcal{X}}}}{\dfrac{2}{\sqrt{\kappa_E}+1}} \,=\, \frac{\sqrt{\kappa_E}+1}{\sqrt{(2\alpha_1 m)^{-1}} \,-\, 1} \le \ell
\end{aligned}$$

where the last sufficient condition is obtained by using the following two inequalities,

$$\ln\left(\frac{1+\sqrt{\mathcal{X}}}{1-\sqrt{\mathcal{X}}}\right) \,\le\, \frac{2\sqrt{\mathcal{X}}}{1-\sqrt{\mathcal{X}}} \quad \forall \mathcal{X} \in (0,1), \qquad \ln\left(\frac{\sqrt{\kappa_E}+1}{\sqrt{\kappa_E}-1}\right) \,\ge\, \frac{2}{\sqrt{\kappa_E}+1} \quad \forall \kappa_E \ge 1.$$