Temporal Variabilities Limit Convergence Rates in Gradient-Based Online Optimization

Bryan Van Scoy Gianluca Bianchin

***Abstract*— This paper investigates fundamental performance limits for gradient-based algorithms applied to time-varying optimization. Leveraging internal model principle and root locus techniques, we show that temporal variabilities impose intrinsic limits on the achievable rate of convergence. For a problem with condition ratio κ and temporal variability described by a model of degree n, we show that the worst-case convergence rate of any minimal-order gradient-based algorithm (having access to a model of temporal variation) is $\rho_{\text{TV}} = \left(\frac{\kappa-1}{\kappa+1}\right)^{1/n}$. This bound reveals a fundamental tradeoff between problem conditioning, temporal complexity, and rate of convergence. We further show how to construct algorithms that attain the bound for low-degree models of temporal variation.**

I. Introduction

Time-varying optimization problems provide a natural framework to describe decision-making tasks in which objectives and/or constraints evolve dynamically over time. Such problems arise in diverse engineering domains, including online learning and streaming data in machine learning [1], [2], adaptive filtering in signal processing [3], and trajectory planning or model predictive control in robotics [4].

Historically, the study of time-varying optimization has relied on continuity arguments with respect to static formulations: when the temporal variations of the problem are sufficiently slow, algorithms developed for static problems produce nearly optimal solutions in the time-varying setting [5]. This reasoning suggests that algorithms for time-varying problems may exhibit the same fundamental performance characteristics as those for time-invariant problems. For example, it is well established that the best achievable rate of gradient descent on time-invariant problems is $\rho_{\text{TI}} = \frac{\kappa-1}{\kappa+1}$, where κ denotes the condition ratio of the problem [6]. Accordingly, one may be tempted to conclude that algorithms for time-varying problems can attain the same rate, provided that the temporal variability is 'slow enough.' In contrast, we prove a fundamental relationship between the convergence rate ρ, the condition ratio κ, and the number n of modes that characterize the temporal variability of the problem. Since there is no direct relation between n and 'how fast' the problem varies, but rather with the *complexity* of its temporal structure, our results reveal fundamental differences between static optimization problems and their dynamic counterparts.

Contributions. Our main contributions are as follows:

1) We provide a fundamental bound on the worst-case convergence rate of minimal-degree controllers for unconstrained quadratic time-varying optimization, which is
$$\rho_{\text{TV}} := \left(\frac{\kappa-1}{\kappa+1}\right)^{1/n}, \tag{1}$$
where κ is the condition ratio of the objective, and n is the number of modes in the model of the time variation. Note that this limitation also applies to the broader class of smooth strongly-convex objectives.
2) We use root locus techniques to design explicit controllers for particular models of the time variation.

Related works. The literature on time-varying optimization methods can be broadly divided into two classes. The first class comprises approaches that ignore or do not exploit any model of the temporal variability, and instead solve a sequence of static problems [7]–[9]. These methods only react after changes are observed, and therefore incur a certain regret and achieve at best convergence to a neighborhood of the optimizer [5]. The second class, instead, leverages a model of the temporal evolution of the problem to track the optimal trajectory exactly. Indeed, such a model is necessary for exact tracking [10], [11]. A prominent example is the prediction-correction framework [12], [13], where each step combines a prediction of the optimizer's evolution with a correction based on the current problem. Extensions of these ideas have recently been studied under contraction analysis [4], sampling-based estimation of variability [14], and constrained formulations [15]; see also the survey [16].

There has been recent interest in using control tools to design optimization algorithms. Fundamental results have been derived in both discrete [10] and continuous [11] time. For quadratic problems, root locus and the internal model principle can be used to analyze and design optimization methods [17]. Recently, [18] investigated constrained settings and stochastic problems. Particularly relevant are [19], [20], which considered quadratic objectives with polynomial temporal variabilities. In [20], the authors use Nevanlinna–Pick interpolation to establish a fundamental lower bound of $\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{1/n}$ (cf. (1)). These results, however, are restricted to polynomial temporal variabilities and focus on accelerated methods, which may fail to achieve global convergence on more general objectives (beyond quadratics) [6], and are also susceptible to noise amplification in gradient evaluations

This material is based upon work supported in part by the National Science Foundation under Award No. 2347121 and in part by the FRFS WEL-T Investigator Programme. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

B. Van Scoy is with the Dept. of Electrical and Computer Engineering, Miami University, OH 45056, USA. Email: bvanscoy@miamioh.edu

G. Bianchin is with the ICTEAM institute and the Department of Mathematical Engineering (INMA) at the University of Louvain, Belgium. Email: gianluca.bianchin@uclouvain.be

[21]. We therefore restrict our attention to non-accelerated (minimal-order) methods and general temporal variabilities.

Notation: We denote by $\mathbb{N}$ and $\mathbb{R}$ the sets of natural and real numbers, respectively; by $\mathbb{R}[z]$ the space of real-coefficient polynomials in z; and by $\mathbb{R}^d[z]$ the space of d-dimensional vector polynomials in z with real coefficients.

II. Problem Setup

We consider time-varying unconstrained optimization problems, consisting of minimizing the quadratic objective:

$$f_k(x) = \tfrac{1}{2} x^\mathsf{T} A x + b_k^\mathsf{T} x, \tag{2}$$

with time indexed by $k \in \mathbb{N}$. Here, $x \in \mathbb{R}^d$ is the decision variable, $A \in \mathbb{R}^{d\times d}$ is a symmetric time-invariant matrix, and $b = \{b_k\}_{k\in\mathbb{N}}$ with $b_k \in \mathbb{R}^d$ is a time-varying parameter.

Remark 1 (Quadratic objective functions): We focus on the class of quadratic objectives as they provide a structured framework to derive bounds on the worst-case convergence rate. Since quadratics are a special case of smooth strongly-convex objectives, and our goal is to establish lower bounds on the convergence rate, the forthcoming estimates also serve as fundamental limitations for this broader function class. This is in line with the time-invariant case, where [22, Sec. 2.1.4] shows that the optimal convergence rate achievable by any iterative algorithm on problems with smooth strongly-convex loss is attained for quadratics. □

We make the following assumptions throughout.

Assumption 1 (Eigenvalues of A): The matrix A has eigenvalues in the closed interval $[\mu, L]$ with $0 < \mu < L$. Moreover, the parameters μ and L are known. □

By Assumption 1, the cost (2) is strongly convex with parameter μ, and the gradient is Lipschitz smooth with parameter L. Considering problems of this class is a standard assumption in optimization [9], which has been widely used in related works [23]–[25]. In what follows, we let $\kappa := L/\mu$ denote the *condition ratio* of the objective in (2).

We make the following assumption on $B(z)$, the $\mathcal{Z}$-transform of the time-varying sequence $b = \{b_k\}_{k\in\mathbb{N}}$.

Assumption 2 (Model of time variation): $B(z)$ is a rational function of z with all poles in $\mathbb{D}_{=1} := \{z \in \mathbb{C} : |z| = 1\}$. □

Assumption 2 specifies the class of temporal variations of the parameter b under consideration. Intuitively, signals whose $\mathcal{Z}$-transform is rational in z are signals generated by finite-order linear recurrences (i.e., by LTI systems with a finite number of states) with poles that lie on the unit circle. This class includes a wide variety of signals including causal periodic sequences (sinusoids, square waves, etc.) and polynomial sequences (constant, ramp, parabolic, etc.). Similar assumptions have been employed in related works [17], [20].

By Assumption 2, $B(z)$ admits the representation

$$B(z) = \frac{B_N(z)}{m(z)}, \tag{3}$$

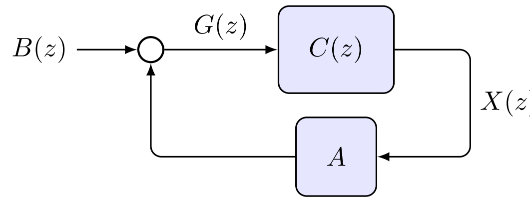

Fig. 1. Structure of gradient-based optimization algorithms as a block diagram in the frequency domain. The algorithm is characterized by a linear filter $C(s)$ applied to gradient evaluations of the cost function $(AX(z) + B(z))$; see (5) for the derivation.

where $B_N(z) \in \mathbb{R}^d[z]$ and $m(z) \in \mathbb{R}[z]$. Without loss of generality, we use the notation

$$m(z) = z^n + \sum_{i=0}^{n-1} m_i z^i, \qquad m_i \in \mathbb{R},$$

where n is the number of poles[1] of $B(z)$.

In line with the literature [22], we focus on gradient-type algorithms for minimizing (2); that is, algorithms that have access to oracle evaluations of the gradient of (2):

$$(k, x) \;\mapsto\; \nabla f_k(x) = Ax + b_k. \tag{4}$$

We restrict our attention to optimization algorithms whose iterates are obtained by processing (4) through Linear Time-Invariant (LTI) filters. Precisely, let $x_k \in \mathbb{R}^d$ denote the estimate for the minimizer of (2) generated by the algorithm[2] at time $k \in \mathbb{N}$, and by $X(z)$ its $\mathcal{Z}$-transform. Then, we consider algorithms that generate x_k as:

$$X(z) = C(z)G(z) = C(z)(AX(z) + B(z)), \tag{5}$$

where $C(z) \in \mathbb{R}^{d\times d}[z]$ is the transfer function of the filter, and $G(z) = AX(z)+B(z)$ is the $\mathcal{Z}$-transform of the gradient $\nabla f_k(x_k)$. A block diagram of the algorithm's structure is illustrated in Fig. 1. We make the following assumption on the filter throughout, which specifies that $C(z)$ is a linear filter, with the additional requirement that it be strictly proper to guarantee real-time implementability of the algorithm.

Assumption 3 (Structure of the optimization filter): $C(z)$ is a rational, strictly proper function of z. □

Example 1 (Gradient descent): It is immediate to verify (by applying the $\mathcal{Z}$-transform to both sides of the equation) that the gradient-descent algorithm:

$$x_{k+1} = x_k - \alpha \nabla f_k(x_k),$$

is a particular instance of (5) with $C(z) = -\frac{\alpha}{z-1} I_d$. □

In what follows, we focus on designing optimization filters with optimal rate of convergence and that are of minimal order; we make these two notions formal next.

[1]Note that (3) does not restrict the poles of $B(z)$ to be identical for each component, since $m(z)$ can be chosen as the polynomial whose root set is the union of the root sets of the individual components of $B(z)$.

[2]Since our focus is on characterizing the asymptotic convergence rate of the method, we henceforth assume that the internal state of the optimization filter is initialized to zero, noting that the framework can be extended to nonzero initial conditions by accounting for the free response of $C(z)$.

Definition 1 (Asymptotic tracking): We say that (5) *asymptotically tracks the minimizer of* (2) if $\{x_k\}$ satisfies

$$\lim_{k\to\infty} \|x_k - x_k^\star\| = 0,$$

where $x_k^\star := -A^{-1}b_k$ is the minimizer of (2). Moreover, the *root-convergence factor* (or, simply, *convergence rate*) is

$$\rho := \limsup_{k\to\infty} \|x_k - x_k^\star\|^{1/k}. \qquad \square$$

Note that Definition 1 formalizes a notion of *exact* tracking, whereby x_k reaches $x_k^\star$ with *zero error*, asymptotically.

Remark 2: The minimizer $x_k^\star$ of the quadratic objective (2) has $\mathcal{Z}$-transform $X^\star(z) = -A^{-1}B(z) = -\frac{1}{m(z)}A^{-1}B_N(z)$. Consequently, $B(z)$ and $X^\star(z)$ share the same poles. It follows that Assumption 2 can equivalently be formulated in terms of a model for $x_k^\star$ rather than b_k, aligning our formulation with other models in the literature (e.g., [20]). $\square$

Definition 2 (Optimization filters of minimal order): Let $C(z)$ be an optimization filter that asymptotically tracks the minimizer of (2). We say that $C(z)$ is of *minimal order* if the degree of its denominator polynomial is minimal among all filters that asymptotically track the minimizer. $\square$

Although higher-order filters could be employed, yielding accelerated algorithms [26], our focus here is on the class of *non-accelerated* gradient methods [22]. Motivations for studying non-accelerated methods include that accelerated methods are known to possibly fail to achieve global convergence on more general objectives (beyond quadratics) [6] and also amplify noise in gradient evaluations [21].

We are now ready to formalize the objective of this work.

Problem 1: Determine the optimal worst-case convergence rate achievable by any minimal-order optimization algorithm of the form (5), where *optimal* is with respect to all optimization filters of minimal order satisfying Assumption 3, and *worst-case* is with respect to all objectives of the form (2) satisfying Assumptions 1 and 2. In addition, construct an optimization filter that attains this rate. $\square$

III. Preliminaries

In this section, we provide an instrumental characterization of the convergence rate that will enable us to address Problem 1. We begin with the following result.

Lemma 1 (Structure of tracking filters): Let Assumptions 1 to 3 hold, and consider optimization filters of the form

$$C(z) = \frac{C_N(z)}{m(z)}, \qquad C_N(z) \in \mathbb{R}^{d\times d}[z]. \qquad (6)$$

(L1) Suppose $C(z)$ is an optimization filter that achieves exact asymptotic tracking. Then, $C(z)$ is of minimal order only if it has the form (6).

(L2) Suppose $C(z)$ has the form (6) and the roots of $\det(m(z)I - AC_N(z))$ are in $\mathbb{D}_{<1} := \{z : |z| < 1\}$. Then, $C(z)$ is an optimization filter of minimal order that achieves exact asymptotic tracking.

Proof: Solving (5), the $\mathcal{Z}$-transform of the gradient is $G(z) = (I - AC(z))^{-1}B(z)$. By Assumption 3, we can write $C(z) = \frac{C_N(z)}{c_D(z)}$ with $C_N(z) \in \mathbb{R}^{d\times d}[z]$ and $c_D(z) \in \mathbb{R}[z]$. Substituting this form and (3) into the expression for the gradient yields

$$G(z) = \frac{c_D(z)}{m(z)}\big(c_D(z)I - AC_N(z)\big)^{-1} B_N(z). \qquad (7)$$

The filter achieves exact asymptotic tracking iff all poles of $G(z)$ are strictly inside the unit circle. Each entry of $(c_D(z)I - AC_N(z))^{-1}$ is a rational function of z, and the denominator polynomial is $\det(c_D(z)I - AC_N(z))$. Since all roots of $m(z)$ are marginally stable by Assumption 2, the poles of $G(z)$ would lie on the unit circle unless the poles introduced by $m(z)$ were canceled by either $c_D(z)$, $B_N(z)$, or the adjugate of $c_D(z)I - AC_N(z)$. Since the roots of $m(z)$ are poles of $B(z)$, they are not canceled by each component of $B_N(z)$. Moreover, they cannot be canceled by the adjugate for all A satisfying Assumption 1. Thus, for $C(z)$ to achieve exact asymptotic tracking with minimal order, it's denominator must be $c_D(z) = m(z)$. Finally, the choice $C(z)$ in (6) includes this necessary factor and no additional pole factors, and hence is of minimal order. ■

Lemma 1 provides necessary and sufficient conditions for asymptotic tracking. The statement (L1) provides a necessary condition for an optimization filter to be of minimal order; the property that $C(z)$ is required to incorporate precisely the same poles as $B(z)$ can be interpreted as an instance of the *internal model principle of time-varying optimization* [10], [11], as it captures the requirement that the optimization filter must embed an internal model of the temporal variability of the problem (encoded by $m(z)$). Conversely, the statement (L2) provides a sufficient condition for an optimization filter to be of minimal order and achieve exact tracking, requiring that all roots of $\det(m(z)I - AC_N(z))$ to be in the open unit disk. This condition will be used later in this work to construct algorithms that address Problem 1.

Because we search within the class of optimization filters of minimal order, driven by the conclusion of Lemma 1, we assume the optimization filter has the form (6). Moreover, we also assume that the same filter is applied to each component of the objective function.

Assumption 4 (Minimal order optimization filter): The optimization filter has the form $C(z) = c(z)I_d$, where $c(z) = \frac{d(z)}{m(z)}$ for some $d(z) \in \mathbb{R}[z]$ of degree at most $n-1$. $\square$

We now provide a formal restatement for Problem 1.

Lemma 2 (Characterization of the convergence rate): Suppose Assumptions 1 to 4 hold. The optimal worst-case convergence rate achievable by any minimal-order optimization algorithm of the form (5) is given by the optimal value of the following min-max problem:

$$\rho \;=\; \min_{d(z)\in\mathbb{R}[z]} \; \max_{\lambda\in[\mu,L]} \; |z| \quad \text{subject to} \quad m(z) - \lambda d(z) = 0. \qquad (8)$$

Proof: Let $A = V\Lambda V^\top$ be an eigendecomposition of A, with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ and V an orthonormal matrix of (real) eigenvectors. Let $\tilde{x}_k := V^\top x_k$ and $\tilde{X}(z)$ denote the corresponding $\mathcal{Z}$-transform. By projecting (5) onto the range space of V, we obtain the update:

$$\tilde{X}(z) = \frac{d(z)}{m(z)}(\Lambda \tilde{X}(z) + V^\top B(z)).$$

With this decomposition, the iterates of (5) separate into d decoupled equations index by $i = 1, \ldots, n$, given by

$$\tilde{X}_i(z) = \frac{d(z)}{m(z)}(\lambda_i \tilde{X}_i(z) + v_i^\top B(z)).$$

Rewriting this equation in input-output form gives

$$\tilde{X}_i(z) = \frac{d(z)}{m(z) - \lambda_i d(z)} v_i^\top B(z). \tag{9}$$

Therefore, the closed-loop poles of (5) are the roots of $m(z) - \lambda_i d(z)$. Since the convergence rate of an LTI system is the maximum modulus of the poles of its transfer function, this gives the formulation (8). ∎

Lemma 2 provides a mathematical reformulation of the optimal worst-case convergence rate achievable by any minimal-order optimization algorithm. The outer minimization over the polynomial $d(z)$ captures a search over all optimization filters of minimal order that satisfy Assumption 3, while the inner maximization over $\lambda \in [\mu, L]$ accounts for the worst-case scenario over the entire class of objective functions consistent with Assumption 1.

IV. Bound on the worst-case convergence rate

In this section, we provide a bound on the worst-case convergence rate achievable by any minimal-order algorithm that asymptotically tracks the minimizer of the quadratic objective (2). The following is our main result.

Theorem 1 (Bound on worst-case convergence rate): Let Assumptions 1 to 4 hold. The worst-case convergence rate achievable by any minimal-order optimization algorithm is bounded below by ρ_{TV} defined in (1), where $\kappa := L/\mu$ is the condition ratio and n is the degree of the model $m(z)$.

Proof: Suppose Assumptions 1 to 4 hold, and suppose the controller $c(z)$ is such that the algorithm in Fig. 1 asymptotically tracks a critical trajectory with rate $\rho \in (0, 1)$ for all quadratic objectives (2) satisfying these assumptions. Since the controller is minimal degree (by Assumption 4), it must have the form $c(z) = d(z)/m(z)$, where $d(z)$ and $m(z)$ are polynomials, the degree of the numerator $d(z)$ is strictly less than that of the denominator $m(z)$, and the model $m(z)$ is monic of degree n with all roots on the unit circle. Define the characteristic polynomial of the subsystem in (9) for a general eigenvalue $\lambda \in [\mu, L]$ as $p_\lambda(z) := m(z) - \lambda\, d(z)$. Since the degree of $d(z)$ is strictly less than that of $m(z)$ and the model is monic, $p_\lambda(z)$ is also monic of degree n. Our goal is then to construct a lower bound on the convergence rate ρ such that all roots of the polynomial $p_\lambda(z)$ are in the disk $\mathbb{D}_{\le\rho} := \{z \in \mathbb{C} : |z| \le \rho\}$ for all $\lambda \in [\mu, L]$.

Write the numerator of the optimization filter in terms of its coefficients as $d(z) = d_{n-1}z^{n-1} + \ldots + d_0$, where each of the d_i coefficients may be zero. Also, denote the roots of the characteristic polynomial as $\zeta_1, \ldots, \zeta_n \in \mathbb{C}$, which all have modulus at most ρ by assumption. In terms of its roots and the coefficients of the controller polynomials, the characteristic polynomial is then

$$p_\lambda(z) \quad = \quad z^n + \sum_{i=0}^{n-1}(m_i - \lambda\, d_i) z^i \quad = \quad \prod_{i=1}^{n}(z - \zeta_i).$$

Define the corresponding elementary symmetric polynomials

$$e_k(p_\lambda) \quad = \quad \sum_{1 \le i_1 < i_2 < \cdots < i_k \le n} \zeta_{i_1}\zeta_{i_2}\cdots\zeta_{i_k}$$

for $k = 1, \ldots, n$, which is the sum of all distinct products of k roots. Vieta's formulas relate the characteristic polynomial coefficients to the elementary symmetric polynomials by

$$e_k(p_\lambda) \quad = \quad (-1)^k (m_{n-k} - \lambda d_{n-k}).$$

Since $e_k(p_\lambda)$ is a sum of products of k roots, each root has modulus at most ρ, and there are $c_k := \binom{n}{k}$ terms in the summation, the elementary symmetric polynomials satisfy the bound

$$|e_k(p_\lambda)| \le c_k \rho^k, \qquad \forall \lambda \in [\mu, L] \text{ and } k = 1, \ldots, n. \tag{10}$$

We can then bound the size of the numerator polynomial coefficients by

$$\begin{aligned}(L - \mu)|d_{n-k}| &= |(m_{n-k} - \mu\, d_{n-k}) - (m_{n-k} - L\, d_{n-k})|, \\ &\le |m_{n-k} - \mu\, d_{n-k}| + |m_{n-k} - L\, d_{n-k}|, \\ &\le 2c_k \rho^k,\end{aligned}$$

where the first inequality follows from the triangle inequality and the second inequality follows from the bound (10) applied to both endpoints $\lambda = \mu$ and $\lambda = L$. Again using the bound (10) at the endpoint $\lambda = \mu$ along with the (reverse) triangle inequality, we bound the size of the coefficients by

$$|m_{n-k}| - \mu\, |d_{n-k}| \le |m_{n-k} - \mu\, d_{n-k}| \le c_k \rho^k.$$

Combining this with the bound on $|d_{n-k}|$ above yields

$$\begin{aligned}|m_{n-k}| &\le c_k \rho^k + \mu\, |d_{n-k}|, \\ &\le c_k \rho^k + \mu \frac{2c_k}{L - \mu} \rho^k, \\ &= c_k \rho^k \frac{\kappa + 1}{\kappa - 1},\end{aligned}$$

where $\kappa := L/\mu$ is the condition ratio. Isolating the convergence rate and using that this holds for all $k = 1, \ldots, n$ yields the lower bound

$$\rho \quad \ge \quad \max_{k=1,\ldots,n} \left(\frac{|m_{n-k}|}{c_k} \cdot \frac{\kappa - 1}{\kappa + 1} \right)^{1/k}. \tag{11}$$

Again using Vieta's formulas, each model coefficient m_{n-k} is (up to a sign) the sum of all distinct products of k roots, each of which has unit modulus by Assumption 2. Therefore, $|m_{n-k}| = e_k(m) \le c_k$ for all $k = 1, \ldots, n$. Moreover, this holds with equality when $k = n$, which produces the lower bound in (1). ■

Theorem 1 provides a fundamental lower bound on the rate of convergence attainable by any minimal-order optimization filter. It is worth noting that the bound is depends solely on the properties of the optimization (i.e., the condition ratio κ) and the degree n of the temporal variability model $m(z)$. The dependence on κ is classical and in line with time-invariant counterparts: as the problem becomes more ill-conditioned (i.e., κ grows), the lower bound approaches one, indicating arbitrarily slow convergence in the worst case. The presence of n outlines a new, fundamental bound intrinsic to time-varying problems: as the temporal variability to be tracked becomes more complex (i.e., as n increases), the algorithm's convergence rate necessarily degrades, showing that *high-order internal models inherently preclude fast convergence.* Thus, (1) characterizes an intrinsic tradeoff between problem conditioning and model complexity, highlighting that even in the best-case design, the convergence rate cannot be improved beyond this limit.

Remark 3 (Comparison with the literature): In the special case $m(z) = (z-1)^n$, the lower bound for *non-minimal* controllers was derived in [20] as: $\rho \geq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{1/n}$. Since $\frac{\kappa-1}{\kappa+1} \geq \frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ for $\kappa > 1$, restricting attention to minimal-order (non-accelerated) algorithms shows that these algorithms suffer from the same fundamental limitation as accelerated ones. Moreover, our result naturally generalizes the classical time-invariant case: when $n = 1$, it reduces exactly to $\rho_{\text{TI}} = \frac{\kappa-1}{\kappa+1}$, the optimal rate of non-accelerated methods on static problems [6]. □

V. Controller Design via Root Locus

We now shift to designing optimization algorithms that attain the fundamental bound (1). Our approach is based on root locus techniques, which enables us to construct closed-form solutions to the optimization (8). Due to the complexity of this task, we focus on three specific cases: $n = 1, 2, 3$.

A. Case: $n = 1$

When the model has only a single pole, it must be either ± 1 since the poles are on the unit circle and the model coefficients are real (so complex roots must appear in conjugate pairs). We now consider these two cases.

First, suppose the model is $m(z) = z - 1$. Then we can use the standard gradient descent controller

$$c(z) = \frac{-\alpha}{z-1} \qquad \text{and} \qquad \alpha = \frac{2}{L+\mu},$$

which achieves the optimal worst-case rate $\rho = \frac{\kappa-1}{\kappa+1}$. In this case, the root locus of $1 - \lambda\, c(z)$ starts at the open loop pole of $z = 1$ when $\lambda = 0$ and moves to the left on the real axis as λ increases, crossing $z = \rho$ when $\lambda = \mu$ and $z = -\rho$ when $\lambda = L$. If the model is $m(z) = z + 1$, then the controller $c(z) = \frac{\alpha}{z+1}$ with the same stepsize α achieves the same rate. The root locus for both cases is shown in Fig. 2 (top).

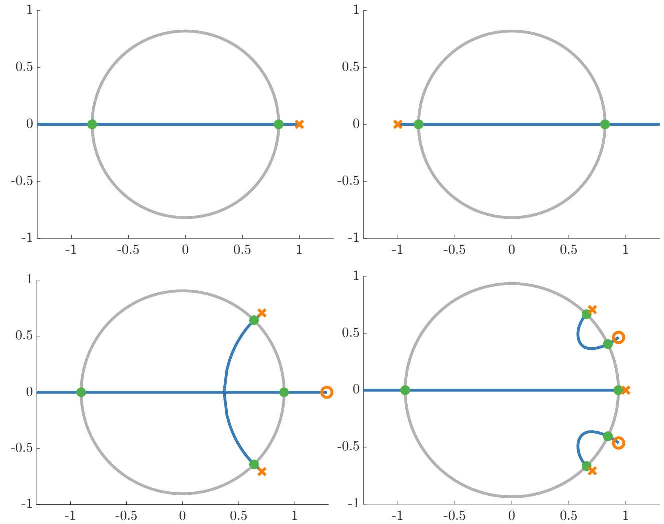

Fig. 2. Root locus of the controller for a model with a constant (top left), the single frequency $\theta = \pi$ (top right), the single frequency $\theta = \pi/4$ (bottom left), both the frequency $\theta = \pi/4$ and a constant (bottom right). Each locus (blue) starts at the open-loop poles (×) and ends at the open-loop zeros (○). The pole locations at gains $\lambda = \mu$ and $\lambda = L$ are shown (●). For all $\lambda \in [\mu, L]$, the root locus is entirely contained in the ρ circle (gray).

B. Case: $n = 2$

Now suppose the model has a single pair of complex conjugate poles on the unit circle with angle θ so that $n = 2$. We can then parameterize the controller with $c_1, c_2 \in \mathbb{R}$ as

$$c(z) = \frac{c_1 z - c_2}{z^2 - 2\cos(\theta) z + 1}.$$

The parameters that yield the optimal worst-case rate are

$$c_1 = \frac{-2\cos\theta}{L} \qquad \text{and} \qquad c_2 = \frac{-2}{L+\mu},$$

which are the unique solutions to the condition that the root locus pass through both $z = -\rho$ and $z = \rho$ when $\lambda = L$. The root locus of this controller is shown in Fig. 2 (bottom left).

The previous case assumes the roots are complex conjugates. If instead the model has real roots at $z = +1$ and $z = -1$ so that $m(z) = z^2 - 1$, the optimal worst-case rate is achieved by the parameters $c_1 = 0$ and $c_2 = \frac{2}{L+\mu}$ for which the root locus passes through both $z = -\rho$ and $z = \rho$ when $\lambda = \mu$.

C. Case: $n = 3$

Now suppose the model has both a pair of complex conjugate poles and a pole at one; that is,

$$m(z) = (z-1)(z^2 - 2\cos(\theta) z + 1). \tag{12}$$

The controller has the form $c(z) = (c_2 z^2 + c_1 z + c_0)/m(z)$, and the optimal worst-case rate is achieved by the parameters

$$c_0 = \frac{-1+\rho^3}{\mu},$$
$$c_1 = \frac{(-L+\mu+(L+\mu)\rho)(1+\rho^2)+2\rho(L+\mu-\rho(L-\mu))\cos(\theta)}{2\mu L\rho},$$
$$c_2 = -\frac{(-L+\mu+(L+\mu)\rho^2)(1+\rho)+2\rho(-L+\mu+(L+\mu)\rho)\cos(\theta)}{2\mu L\rho^2},$$

which yields the root locus in Fig. 2 (bottom right).

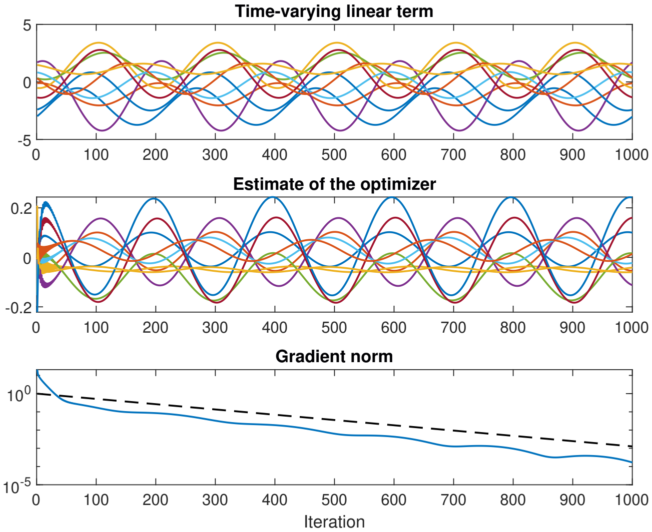

Fig. 3. Numerical simulation of the optimal worst-case controller from Section V-C to minimize the quadratic objective in (2). (Top) Trajectories of each component in the time-varying linear term b_k. (Middle) Trajectories of each component in the time-varying estimate x_k of the optimizer. (Bottom) Gradient norm $\|Ax_k + b_k\|$ and the bound on the worst-case rate ρ^k (dashed); see Section VI for details.

VI. Numerical Simulation

To illustrate our results, we simulate the trajectory of (5) using the optimization filter $C(z)$ designed in Section V to the quadratic objective in (2). We chose parameters $\mu = 1$, $L = 10$, and $d = 10$. The matrix A was randomly generated with eigenvalues uniformly distributed in $[\mu, L]$, and the time-varying linear term b_k was generated by an LTI system corresponding to the model $m(z)$ in (12) with $\theta = 0.01\pi$. The system trajectories are shown in Fig. 3. In our simulations, the controller proposed in [17] could not be constructed, as the associated LMIs were infeasible.

VII. Conclusions

We have established fundamental limits on the attainable convergence rates of gradient-based algorithms for time-varying quadratic optimization. By leveraging tools from control theory, in particular, the internal model principle and root locus techniques, we have shown that the optimal worst-case convergence rate necessarily degrades with the complexity of the temporal variability, quantified by the degree n of the underlying model. Overall, the results of this paper contribute to a deeper understanding of the intrinsic performance limits of gradient-based methods in time-varying optimization, showing that temporal variability fundamentally constrains the achievable rate of convergence.

References

[1] Y. Chen and Y. Zhou, "Machine learning based decision making for time varying systems: Parameter estimation and performance optimization," *Knowledge-Based Systems*, vol. 190, p. 105479, 2020.

[2] A. Rakhlin and K. Sridharan, "Online learning with predictable sequences," in *Conference on Learning Theory*, 2013, pp. 993–1019.

[3] F. Y. Jakubiec and A. Ribeiro, "D-map: Distributed maximum a posteriori probability estimation of dynamic systems," *IEEE Trans Signal Processing*, vol. 61, no. 2, pp. 450–466, 2012.

[4] A. Davydov, V. Centorrino, A. Gokhale, G. Russo, and F. Bullo, "Time-varying convex optimization: A contraction and equilibrium tracking approach," *IEEE Trans Automatic Ctrl*, vol. 70, no. 11, pp. 7446–7460, 2025.

[5] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 2032–2048, 2020.

[6] L. Lessard, B. Recht, and A. Packard, "Analysis and design of optimization algorithms via integral quadratic constraints," *SIAM J Optim*, vol. 26, no. 1, pp. 57–95, 2016.

[7] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *International conference on machine learning*, 2003, pp. 928–936.

[8] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2, pp. 169–192, 2007.

[9] E. Hazan, "Introduction to online convex optimization," *Foundations and Trends in Optimization*, vol. 2, no. 3-4, pp. 157–325, 2016.

[10] G. Bianchin and B. Van Scoy, "The discrete-time internal model principle of time-varying optimization," in *Proc CDC*, Rio de Janeiro, Dec. 2025, pp. 3958–3956.

[11] ——, "The internal model principle of time-varying optimization," *IEEE Trans Automatic Ctrl*, 2026, (In Press).

[12] Y. Zhao and M. N. S. Swamy, "A novel technique for tracking time-varying minimum and its applications," in *IEEE Canadian Conference on Electrical and Computer Engineering*, vol. 2, 1998, pp. 910–913.

[13] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Prediction-correction interior-point method for time-varying convex optimization," *IEEE Trans Automatic Ctrl*, vol. 63, no. 7, pp. 1973–1986, 2017.

[14] M. Marchi, J. Bunton, J. P. Silvestre, and P. Tabuada, "A framework for time-varying optimization via derivative estimation," in *Proc ECC*, 2024, pp. 2730–2735.

[15] A. Simonetto and E. Dall'Anese, "Prediction-correction algorithms for time-varying constrained optimization," *IEEE Trans Signal Processing*, vol. 65, no. 20, pp. 5481–5494, 2017.

[16] A. Hauswirth, Z. He, S. Bolognani, G. Hug, and F. Dörfler, "Optimization algorithms as robust feedback controllers," *Annual Reviews in Control*, vol. 57, p. 100941, 2024.

[17] N. Bastianello, R. Carli, and S. Zampieri, "Internal model-based online optimization," *IEEE Trans Automatic Ctrl*, vol. 69, no. 1, pp. 689–696, 2024.

[18] U. Casti, N. Bastianello, R. Carli, and S. Zampieri, "A control theoretical approach to online constrained optimization," *Automatica*, vol. 176, p. 112107, 2025.

[19] A. X. Wu, I. R. Petersen, V. Ugrinovskii, and I. Shames, "An online optimization algorithm for tracking a linearly varying optimal point with zero steady-state error," in *Proc ACC*, 2025, pp. 930–934.

[20] A. X. Wu, I. R. Petersen, and I. Shames, "A fundamental convergence rate bound for gradient based online optimization algorithms with exact tracking," in *arXiv:2508.21335 [math.OC]*, 2025.

[21] H. Mohammadi, M. Razaviyayn, and M. R. Jovanović, "Tradeoffs between convergence rate and noise amplification for momentum-based accelerated optimization algorithms," *IEEE Trans Automatic Ctrl*, vol. 70, no. 2, pp. 889–904, 2025.

[22] Y. Nesterov, *Lectures on convex optimization*. Springer, 2018, vol. 137.

[23] G. Bianchin, J. Cortés, J. I. Poveda, and E. Dall'Anese, "Time-varying optimization of LTI systems via projected primal-dual gradient flows," *IEEE Trans Control Netw Syst*, vol. 9, no. 1, pp. 474–486, Mar. 2022.

[24] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Timescale separation in autonomous optimization," *IEEE Trans Automatic Ctrl*, vol. 66, no. 2, pp. 611–624, 2021.

[25] G. Carnevale, N. Mimmo, and G. Notarstefano, "Nonconvex distributed feedback optimization for aggregative cooperative robotics," *Automatica*, vol. 167, p. 111767, 2024.

[26] A. d'Aspremont, D. Scieur, and A. Taylor, "Acceleration methods," *Foundations and Trends in Optimization*, vol. 5, no. 1–2, pp. 1–245, 2021.