

Direct Synthesis of Iterative Algorithms With Bounds on Achievable Worst-Case Convergence Rate

Laurent Lessard¹

Peter Seiler²

Abstract

Iterative first-order methods such as gradient descent and its variants are widely used for solving optimization and machine learning problems. There has been recent interest in analytic or numerically efficient methods for computing worst-case performance bounds for such algorithms, for example over the class of strongly convex loss functions. A popular approach is to assume the algorithm has a fixed size (fixed dimension, or memory) and that its structure is parameterized by one or two hyperparameters, for example a learning rate and a momentum parameter. Then, a Lyapunov function is sought to certify robust stability and subsequent optimization can be performed to find optimal hyperparameter tunings. In the present work, we instead fix the constraints that characterize the loss function and apply techniques from robust control synthesis to directly search over algorithms. This approach yields stronger results than those previously available, since the bounds produced hold over algorithms with an arbitrary, but finite, amount of memory rather than just holding for algorithms with a prescribed structure.

1 Introduction

First-order methods are a widely used class of iterative algorithms for solving optimization problems of the form $\min_{x \in \mathbb{R}^n} f(x)$, where f is continuously differentiable and we only have access to first-order (gradient) measurements of f . An example of a first-order method is the *heavy ball* method [14], which uses iterations of the form

$$x_{k+1} = x_k - \alpha \nabla f(x_k) + \beta(x_k - x_{k-1}). \quad (1)$$

Here, α and β are parameters to be tuned. When $\beta = 0$, we recover the well-known *gradient descent* algorithm. Algorithms such as (1) are called *accelerated* because when they are appropriately tuned, they can achieve faster convergence than by using gradient descent alone.

To quantify convergence, we typically make assumptions about f , and then bound the worst-case convergence rate of the algorithm over all functions that satisfy

the assumptions. For example, suppose f is *strongly convex* and has *Lipschitz-continuous gradients*, which means there exists $0 < m \leq L$ such that [11, Thm. 2.1.12]

$$\begin{bmatrix} \nabla f(x) - \nabla f(y) \\ x - y \end{bmatrix}^\top \begin{bmatrix} -2mL & L + m \\ L + m & -2 \end{bmatrix} \begin{bmatrix} \nabla f(x) - \nabla f(y) \\ x - y \end{bmatrix} \geq 0$$

for all $x, y \in \mathbb{R}^n$. (2)

If $x^* := \arg \min_{x \in \mathbb{R}^n} f(x)$, then gradient descent with $0 \leq \alpha \leq \frac{2}{L+m}$ satisfies [11, §2.1.5]

$$\|x_{k+1} - x^*\| \leq \sqrt{1 - \frac{2\alpha m L}{L+m}} \|x_k - x^*\|. \quad (3)$$

In other words, the sequence of iterates $\{x_0, x_1, \dots\}$ converges to the global optimizer x^* at a geometric rate. The fastest rate bound is $\frac{L-m}{L+m}$, achieved when $\alpha = \frac{2}{L+m}$.

In an attempt to generalize these types of results and better understand the behavior of iterative algorithms and how they should be tuned, recent works have focused on a dynamical systems interpretation of optimization algorithms. In other words, the iterate sequence $\{x_k\}$ corresponds to the state, and the index k serves the role of *time*. Viewed in this light, convergence of an algorithm is equivalent to the stability of the associated equilibrium point of the dynamical system.

A recently proposed approach uses robust control to analyze worst-case convergence rates [4, 7, 8, 10]. Here, the algorithm itself plays the role of the controller, and the gradient mapping ∇f is the uncertain plant. This viewpoint has also been extended to analyze operator-splitting methods [12] and distributed optimization algorithms [16].

Main contributions. In the aforementioned robust control approach, the results obtained are general in the sense that they hold across all choices of parameters as in (3), but they are fundamentally constrained by the structure of the algorithm; the particular parametric form of the iterates such as (1).

In the present work, we apply tools from robust control synthesis to derive performance bounds that are *independent* of parametric forms such as (1) and hold over broad classes of algorithms. These results adapt the approach in [6] for the purpose of synthesizing algorithms that optimize the convergence rate.

Specifically, we consider the class of functions with *sector-bounded gradients* (a weaker version of (2)). We

¹L. Lessard is with the Department of Electrical and Computer Engineering and the Wisconsin Institute for Discovery, University of Wisconsin–Madison, USA. laurent.lessard@wisc.edu

²P. Seiler is with the Department of Electrical Engineering and Computer Science, University of Michigan, USA. pseiler@umich.edu

show that, for this function class, gradient descent with stepsize $\frac{2}{L+m}$ achieves the fastest possible worst-case convergence rate not only among all tunings of gradient descent, but among any algorithm representable as a linear time-invariant (LTI) system with finite state. This includes algorithms of the form (1) but where x_{k+1} depends linearly on $\{x_k, x_{k-1}, \dots, x_{k-\ell}\}$ for some fixed ℓ .

Our analysis also extends to the case where ∇f is characterized by a given and fixed integral quadratic constraint (IQC). In this case, we prove that when using the so-called *off-by-one IQC* introduced in [10], which is satisfied by the class of functions (2), the optimal rate over any LTI algorithm is achieved by a particular accelerated method known as the *triple momentum method* [17].

The remainder of the paper is organized as follows. In Section 2, we introduce the machinery from robust control synthesis that will be used throughout the rest of the paper. In Sections 4 and 5, we specialize the analysis to the sector-bounded and IQC cases, deriving the optimal LTI algorithms in both instances. We conclude in Section 7 with some closing remarks.

2 Preliminaries

Notation. We use $\succ$ and $\succeq$ to denote matrix inequality in the definite and semi-definite sense, respectively. $X^\top$ denotes the matrix transpose. We use the short-hand notation: $\text{sym } X := X + X^\top$ with the convention that products are expanded as $\text{sym } XY = XY + Y^\top X^\top$.

Linear matrix inequalities. Most of the algebraic manipulations in the sequel involve linear matrix inequalities (LMI), which have now become a widespread tool in optimization and control. We refer the reader to [2] and references therein. We will make use of three fundamental results involving LMIs. The first is a basic property of the Schur complement, which can convexify certain semidefinite programs by converting them into an LMI.

Proposition 1. *Let A, B, C be matrices of compatible dimensions such that we can define the block matrix: $X := \begin{pmatrix} A & B \\ B^\top & C \end{pmatrix}$ and suppose $X = X^\top$ and $C \succ 0$. Then, $X \succeq 0$ if and only if $A - BC^{-1}B^\top \succeq 0$. The result also holds when $\succeq$ is replaced by $\succ$.*

The second fundamental result is called the *matrix elimination lemma*, by Gahinet and Apkarian [6]. This result applies to matrix inequalities with affine dependence on a matrix of decision variables and is useful for eliminating controller parameters to obtain results independent of controller state dimension.

Lemma 2 ([6]). *Given a symmetric matrix $\Psi \in \mathbb{R}^{n \times n}$ and two matrices P, Q of column dimension n , consider the problem of finding some matrix Θ of compatible dimensions such that*

$$\Psi + P^\top \Theta^\top Q + Q^\top \Theta P \prec 0. \quad (4)$$

Denote by W_P, W_Q any matrices whose columns form bases for the null spaces of P and Q respectively. Then there exists Θ satisfying (4) if and only if

$$W_P^\top \Psi W_P \prec 0 \quad \text{and} \quad W_Q^\top \Psi W_Q \prec 0.$$

The third fundamental result we will use is called the *matrix completion lemma*, by Packard et al. [13, Lem. 7.5]. This result describes how a pair of matrices can be augmented to satisfy an inverse relationship.

Lemma 3 ([13]). *Suppose $X, Y \in \mathbb{R}^{n \times n}$, with $X = X^\top \succ 0$ and $Y = Y^\top \succ 0$. Let m be a positive integer. Then there exist matrices $X_2 \in \mathbb{R}^{n \times m}$, $X_3 \in \mathbb{R}^{m \times m}$ such that $X_3 = X_3^\top$, and*

$$\begin{bmatrix} X & X_2 \\ X_2^\top & X_3 \end{bmatrix} \succ 0 \quad \text{and} \quad \begin{bmatrix} X & X_2 \\ X_2^\top & X_3 \end{bmatrix}^{-1} = \begin{bmatrix} Y & ? \\ ? & ? \end{bmatrix}$$

if and only if $X - Y^{-1} \succeq 0$, and $\text{rank}(X - Y^{-1}) \leq m$.

Remark 4. *Applying Proposition 1, the condition in Lemma 3 can be expressed as an LMI in X and Y :*

$$X - Y^{-1} \succeq 0 \quad \text{if and only if} \quad \begin{bmatrix} X & I \\ I & Y \end{bmatrix} \succeq 0.$$

Also note that the condition $\text{rank}(X - Y^{-1}) \leq m$ is trivially satisfied for any $m \geq n$.

Exponential stability with IQCs. Finally, we will make use of an LMI for bounding the worst-case exponential convergence rate of a linear dynamical system subject to integral quadratic constraints (IQCs). Similar LMIs have appeared in [1, 9, 15].

Lemma 5. *Consider the discrete-time LTI system G , which maps $u \mapsto y$ and obeys the dynamics:*

$$\xi_{k+1} = A_G \xi_k + B_G u_k \quad (5a)$$

$$y_k = C_G \xi_k + D_G u_k \quad (5b)$$

Fix $\rho \in (0, 1]$, and consider the auxiliary system Ψ_ρ , which maps $(y, u) \mapsto z$ and obeys the dynamics:

$$\zeta_{k+1} = A_\Psi \zeta_k + B_{\Psi 1} y_k + B_{\Psi 2} u_k \quad (6a)$$

$$z_k = C_\Psi \zeta_k + D_{\Psi 1} y_k + D_{\Psi 2} u_k \quad (6b)$$

Suppose (y, u) satisfies the ρ -Integral Quadratic Constraint (ρ -IQC) defined by (Ψ_ρ, M) . That is,

$$\sum_{k=0}^T \rho^{-2k} z_k^\top M z_k \geq 0 \quad \text{for } T = 0, 1, \dots \quad (7)$$

Define state-space matrices for the map $u \mapsto z$:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] := \left[\begin{array}{cc|c} A_G & 0 & B_G \\ B_{\Psi 1} C_G & A_\Psi & B_{\Psi 2} + B_{\Psi 1} D_G \\ \hline D_{\Psi 1} C_G & C_\Psi & D_{\Psi 2} + D_{\Psi 1} D_G \end{array} \right].$$

If there exists some $P \succ 0$ such that

$$\begin{aligned} & [A \quad B]^\top P [A \quad B] - \rho^2 [I \quad 0]^\top P [I \quad 0] \\ & + [C \quad D]^\top M [C \quad D] \preceq 0, \end{aligned} \quad (8)$$

then the state of G converges to zero at an exponential rate of ρ . Specifically, $\lim_{k \rightarrow \infty} \rho^{-k} \xi_k = 0$ for all initial states (ξ_0, ζ_0) .

3 Algorithms as control systems

Following the line of research developed in [7, 10], we formulate the problem of iterative algorithm analysis as a control problem. Specifically, we will consider algorithms set up as in Figure 1.

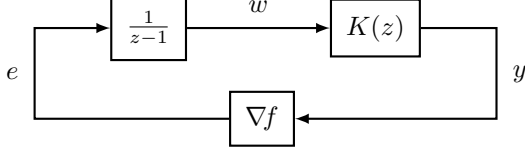


Figure 1: Block diagram representing an iterative optimization algorithm. The integrator is present so that the algorithm can converge with no steady-state error for any initial condition of K .

The iterative algorithm must contain a pure integrator, i.e. it must take the form $K(z)\frac{1}{z-1}$ where $K(z)$ is an LTI system that represents the algorithm. Assume $K(z)$ has a state-space realization (A_K, B_K, C_K, D_K) . Let $w \in \mathbb{R}$ and $q \in \mathbb{R}^{n_K}$ be the state of integrator and $K(z)$, respectively. The order of $K(z)$, denoted n_K , is unspecified at this point, i.e. the algorithm may have a finite but arbitrary amount of memory. A realization for $K(z)\frac{1}{z-1}$ from input e_k to output y_k is given by:

$$w_{k+1} = w_k + e_k \quad (9a)$$

$$q_{k+1} = A_K q_k + B_K w_k \quad (9b)$$

$$y_k = C_K q_k + D_K w_k \quad (9c)$$

This representation is very general. For example, the heavy-ball method is represented by $K(z) = \frac{-\alpha z}{(z-\beta)}$, which corresponds to $A_K = C_K = \beta$ and $B_K = D_K = -\alpha$. Meanwhile, the gradient descent algorithm corresponds to the static gain $K(z) = -\alpha$.

Dimensionality reduction. Throughout this paper, we assume for convenience that $K(z)$ is a single-input, single-output (SISO) system. For algorithms such as heavy-ball (1) where α and β are scalars, this assumption is not restrictive. Indeed, the transfer function for the heavy-ball algorithm takes the form $I_d \otimes K(z)\frac{1}{z-1}$, where $x_k \in \mathbb{R}^d$ in (1). The IQCs we will use to describe ∇f and hence the LMIs (8) will also have a block-Kronecker form. These facts allow us to restrict our attention to SISO $K(z)$ without any loss of generality. For further discussion on this type of dimensionality reduction, see [10, §4.2].

Loop transformation. The case of interest in this paper is where the function f is strongly convex and has Lipschitz gradients (2). This implies, in the language of robust control, that ∇f is sector-bounded and slope-restricted in the interval (m, L) . In order to streamline the rest of the paper, we will perform a loop transformation to shift this interval to $(-1, 1)$. The transformation is shown in Figure 2.

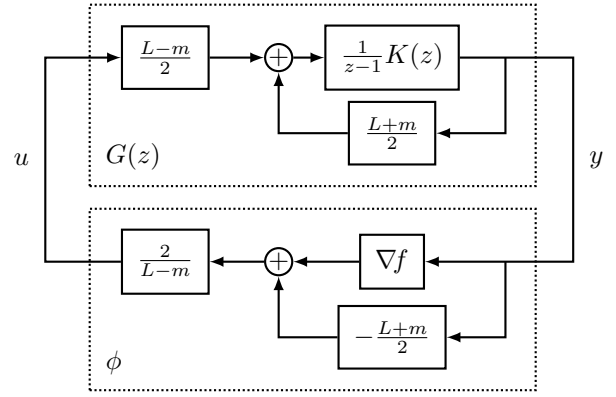


Figure 2: Loop Transformation. If ∇f is in the sector (m, L) , then ϕ is in the sector $(-1, 1)$.

The loop-shifting transformation normalizes the nonlinearity ϕ to lie in the sector $(-1, 1)$. As a result, $u_k = \phi(y_k)$ satisfies the pointwise-in-time constraint:

$$\begin{bmatrix} y_k \\ u_k \end{bmatrix}^\top \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} y_k \\ u_k \end{bmatrix} \geq 0. \quad (10)$$

The input to $K(z)\frac{1}{z-1}$ is transformed by the following algebraic equation: $e_k = \frac{L-m}{2}u_k + \frac{L+m}{2}y_k$. A realization for the transformed system $G(z)$ is obtained by combining this relation with (9). Eliminating e_k yields the following state-space realization with state (q_k, w_k) :

$$G(z) := \left[\begin{array}{c|c} A_G & B_G \\ \hline C_G & D_G \end{array} \right] = \left[\begin{array}{c|c} A_K & B_K \\ \hline \frac{L+m}{2}C_K & 1 + \frac{L+m}{2}D_K \end{array} \middle| \begin{array}{c} 0 \\ \frac{L-m}{2} \end{array} \right] \quad (11)$$

4 Sector-bounded nonlinearities

In this section we treat the analysis problem for the special case where ∇f is *sector-bounded*. This assumption corresponds to relaxing (2) to hold only between an arbitrary point and the optimizer of f . In other words, (2) reduces to:

$$\begin{bmatrix} \nabla f(x) \\ x - x^* \end{bmatrix}^\top \begin{bmatrix} -2mL & L+m \\ L+m & -2 \end{bmatrix} \begin{bmatrix} \nabla f(x) \\ x - x^* \end{bmatrix} \geq 0 \quad \text{for all } x \in \mathbb{R}^n.$$

From the IQC standpoint, this means that we assume the transformed nonlinearity ϕ only satisfies the pointwise-in-time constraint (10).

Our goal is to find an algorithm (A_K, B_K, C_K, D_K) such that the convergence rate ρ is minimized. The sector IQC (10) has no dynamics, so let $z = (\frac{y}{u})$. In this case, $\Psi = I_2$ and the state matrices for the map $u \mapsto z$ in Lemma 5 simplify to:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] := \left[\begin{array}{c|c} A_G & B_G \\ \hline [C_G] & [D_G] \end{array} \right].$$

By Lemma 5, iterates converge with rate $\rho \in (0, 1]$ if there exists $P \succ 0$ such that:

$$\begin{aligned} & \begin{bmatrix} A_G & B_G \end{bmatrix}^\top P \begin{bmatrix} A_G & B_G \end{bmatrix} - \rho^2 \begin{bmatrix} I & 0 \end{bmatrix}^\top P \begin{bmatrix} I & 0 \end{bmatrix} \\ & + \begin{bmatrix} C_G & D_G \end{bmatrix}^\top \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} C_G & D_G \end{bmatrix} \preceq 0. \end{aligned} \quad (12)$$

We can rearrange the LMI above to the form:

$$\begin{bmatrix} A_G & B_G \\ C_G & D_G \end{bmatrix}^\top \begin{bmatrix} P & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} A_G & B_G \\ C_G & D_G \end{bmatrix} - \begin{bmatrix} \rho^2 P & 0 \\ 0 & 1 \end{bmatrix} \preceq 0. \quad (13)$$

The design corresponds to synthesizing the algorithm (A_K, B_K, C_K, D_K) , Lyapunov matrix $P \succ 0$, and convergence rate bound $\rho \in (0, 1]$. The matrix inequality (13) is nonlinear due to products of these variables. Applying Proposition 1, we can write the equivalent condition:

$$\begin{bmatrix} \rho^2 P & 0 & A_G^\top & C_G^\top \\ 0 & 1 & B_G^\top & D_G^\top \\ A_G & B_G & P^{-1} & 0 \\ C_G & D_G & 0 & 1 \end{bmatrix} \succeq 0, \quad \text{and} \quad P \succ 0. \quad (14)$$

From (11), we can write (A_G, B_G, C_G, D_G) as an affine function of the controller parameters. In the equation below, dashed lines indicate block structure (not state-space notation).

$$\begin{aligned} \begin{bmatrix} A_G & B_G \\ C_G & D_G \end{bmatrix} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & \frac{L-m}{2} \\ 0 & 0 & 0 \end{bmatrix} \\ &+ \begin{bmatrix} I & 0 \\ 0 & \frac{L+m}{2} \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}}_K \begin{bmatrix} I & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \end{aligned} \quad (15)$$

Substituting (15) into (14), we obtain:

$$\begin{aligned} & \underbrace{\begin{bmatrix} \rho^2 P & \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 1 & \begin{bmatrix} 0 & \frac{L-m}{2} \end{bmatrix} & 0 \\ \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 \\ \frac{L-m}{2} \end{bmatrix} & P^{-1} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 & \begin{bmatrix} 0 & 0 \end{bmatrix} & 1 \end{bmatrix}}_\Psi \\ & + \text{sym} \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \begin{bmatrix} I \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ \frac{L+m}{2} \end{bmatrix} \\ 0 & 1 \end{bmatrix} K \begin{bmatrix} \begin{bmatrix} I & 0 \end{bmatrix} & 0 & 0 & 0 \\ \begin{bmatrix} 0 & 1 \end{bmatrix} & 0 & 0 & 0 \end{bmatrix} \succeq 0, \end{aligned} \quad (16)$$

where sym is the shorthand notation we defined in Section 2 and K is defined in (15). By Lemma 2, the LMI (16) is feasible if and only if a pair of conditions hold. In this case, the conditions are:

$$\begin{aligned} & \begin{bmatrix} I & 0 & \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \\ 0 & 1 & \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \\ 0 & 0 & \begin{bmatrix} 0 & 1 \end{bmatrix} & -\frac{L+m}{2} \end{bmatrix} \Psi \begin{bmatrix} I & 0 & \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \\ 0 & 1 & \begin{bmatrix} 0 & 0 \end{bmatrix} & 0 \\ 0 & 0 & \begin{bmatrix} 0 & 1 \end{bmatrix} & -\frac{L+m}{2} \end{bmatrix}^\top \succeq 0 \\ & \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix}^\top \Psi \begin{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ I \\ 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} \succeq 0 \end{aligned}$$

where Ψ is defined in (16). Carrying out the matrix multiplications and simplifying yields:

$$\begin{bmatrix} \rho^2 P & \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ \begin{bmatrix} 0 & 0 \end{bmatrix} & 1 & \frac{L-m}{2} \\ \begin{bmatrix} 0 & 1 \end{bmatrix} & \frac{L-m}{2} & \begin{bmatrix} 0 \\ 1 \end{bmatrix}^\top P^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \left(\frac{L+m}{2}\right)^2 \end{bmatrix} \succeq 0, \quad (17a)$$

$$\begin{bmatrix} 1 & \begin{bmatrix} 0 & \frac{L-m}{2} \end{bmatrix} & 0 \\ \begin{bmatrix} 0 \\ \frac{L-m}{2} \end{bmatrix} & P^{-1} & \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ 0 & \begin{bmatrix} 0 & 0 \end{bmatrix} & 1 \end{bmatrix} \succeq 0 \quad (17b)$$

Applying Proposition 1 to the first 2×2 block of (17a) we obtain the following equivalent condition:

$$r + \left(\frac{L+m}{2}\right)^2 - (\rho^{-2}r + \left(\frac{L-m}{2}\right)^2) \geq 0$$

where $r := \begin{bmatrix} 0 \\ 1 \end{bmatrix}^\top P^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} > 0$. This simplifies to:

$$mL \geq (\rho^{-2} - 1)r \quad (18)$$

Next, Equation (17b) is feasible if and only if the first 2×2 block is feasible. Applying Proposition 1 to the upper left corner of this 2×2 block yields: $r \geq \left(\frac{L-m}{2}\right)^2$. Combining with (18) yields $4mL \geq (\rho^{-2} - 1)(L - m)^2$. Solving for ρ yields: $\rho \geq \frac{L-m}{L+m}$, which is precisely the optimal gradient rate bound described in Section 1. Indeed, if we also set $P^{-1} = \begin{bmatrix} \varepsilon & 0 \\ 0 & \left(\frac{L-m}{2}\right)^2 \end{bmatrix}$ for any $\varepsilon > 0$, Equation (17) is satisfied. Setting $D_K = -\frac{2}{m+L}$ (and A_K empty), which is the gradient method with optimal stepsize, renders (16) feasible too.

Discussion. It is known that for gradient descent, choosing $\alpha = \frac{2}{m+L}$ optimizes the rate bound (3) for strongly convex functions with Lipschitz continuous gradients. Similar results have shown that the same upper bound (and optimal stepsize) holds if we constrain f to be quadratic or if we relax f to merely have sector-bounded gradients [10]. It is also straightforward to show that the rate bound of $\frac{L-m}{L+m}$ is tight, because it is attained when using this algorithm to optimize the scalar quadratic function $f(x) := \frac{m}{2}(x - x^*)^2$.

This bound assumes the algorithm in question is gradient descent. In this section, we showed that for the class of functions with sector-bounded gradients, gradient descent with $\alpha = \frac{2}{L+m}$ is in fact optimal over all algorithms that can be expressed as LTI systems (of any fixed order) in feedback with ∇f , as in Figure 2. Put another way, when using sector-bounded gradients, we cannot improve our upper bound by using more complex algorithms.

5 Sector-bounded and slope restricted

We now extend our approach from Section 4 to the case where ∇f is both sector-bounded and *slope-restricted*. Specifically, we assume that f is strongly convex with Lipschitz-continuous gradients so that ∇f satisfies the

constraints in (2). The transformed nonlinearity ϕ thus satisfies the normalized constraint for all $x, y \in \mathbb{R}^n$:

$$\begin{bmatrix} x - y \\ \phi(x) - \phi(y) \end{bmatrix}^\top \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x - y \\ \phi(x) - \phi(y) \end{bmatrix} \geq 0.$$

Such functions satisfy a general class of ρ -IQCs defined by Zames–Falb multipliers [1, 3, 5]. For now, we will assume a general IQC of the form (6), and we further break apart z into its two components, thereby resulting in joint plant-IQC dynamics:

$$\begin{aligned} \begin{bmatrix} \xi_{k+1} \\ \zeta_{k+1} \end{bmatrix} &= \begin{bmatrix} A_G & 0 \\ B_\Psi^1 C_G & A_\Psi \end{bmatrix} \begin{bmatrix} \xi_k \\ \zeta_k \end{bmatrix} + \begin{bmatrix} B_G \\ B_\Psi^1 D_G + B_\Psi^2 \end{bmatrix} u_k \\ \begin{bmatrix} z_k^1 \\ z_k^2 \end{bmatrix} &= \begin{bmatrix} D_\Psi^{11} C_G & C_\Psi^1 \\ D_\Psi^{21} C_G & C_\Psi^2 \end{bmatrix} \begin{bmatrix} \xi_k \\ \zeta_k \end{bmatrix} + \begin{bmatrix} D_\Psi^{11} D_G + D_\Psi^{12} \\ D_\Psi^{21} D_G + D_\Psi^{22} \end{bmatrix} u_k \end{aligned}$$

Substituting (A_G, B_G, C_G, D_G) from (11), obtain:

$$\begin{bmatrix} q_{k+1} \\ w_{k+1} \\ \zeta_{k+1} \end{bmatrix} = \begin{bmatrix} A_K & B_K & 0 \\ \frac{L+m}{2} C_K & 1 + \frac{L+m}{2} D_K & 0 \\ B_\Psi^1 C_K & B_\Psi^1 D_K & A_\Psi \end{bmatrix} \begin{bmatrix} q_k \\ w_k \\ \zeta_k \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{L-m}{2} \\ B_\Psi^2 \end{bmatrix} u_k \quad (19a)$$

$$\begin{bmatrix} z_k^1 \\ z_k^2 \end{bmatrix} = \begin{bmatrix} D_\Psi^{11} C_K & D_\Psi^{11} D_K & C_\Psi^1 \\ D_\Psi^{21} C_K & D_\Psi^{21} D_K & C_\Psi^2 \end{bmatrix} \begin{bmatrix} q_k \\ w_k \\ \zeta_k \end{bmatrix} + \begin{bmatrix} D_\Psi^{12} \\ D_\Psi^{22} \end{bmatrix} u_k \quad (19b)$$

This closed loop includes the algorithm (integrator and controller) and the IQC states. The closed-loop state matrices are denoted $\left(A, B, \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}, \begin{bmatrix} D_1 \\ D_2 \end{bmatrix} \right)$. In this case the LMI in Lemma 5 looks different because the IQC term is quadratic in (z^1, z^2) instead of (y, u) . We obtain:

$$\begin{bmatrix} A & B \\ C_1 & D_1 \end{bmatrix}^\top \begin{bmatrix} P & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} A & B \\ C_1 & D_1 \end{bmatrix} - \begin{bmatrix} I & 0 \\ C_2 & D_2 \end{bmatrix}^\top \begin{bmatrix} \rho^2 P & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ C_2 & D_2 \end{bmatrix} \preceq 0$$

As before, we can use Proposition 1 to convert this to a single LMI:

$$\begin{bmatrix} \rho^2 P + C_2^\top C_2 & C_2^\top D_2 & A^\top & C_1^\top \\ D_2^\top C_2 & D_2^\top D_2 & B^\top & D_1^\top \\ A & B & P^{-1} & 0 \\ C_1 & D_1 & 0 & 1 \end{bmatrix} \succeq 0, \quad P \succ 0 \quad (20)$$

Equation (20) will be affine in the algorithm parameters (A_K, B_K, C_K, D_K) and everything will carry through as before if C_2 and D_2 do not depend on the algorithm parameters. If we examine the formulas of (19b), this amounts to requiring for example that $D_\Psi^{21} = 0$ and $D_\Psi^{22} = 1$. So the question becomes: is it possible to find a factorization Ψ of the IQC with this property?

The general Zames–Falb ρ -IQC for a nonlinearity that is sector-bounded and slope-restricted on $(-1, 1)$ is:

$$\Pi(z) := \frac{1}{2} \begin{bmatrix} 2 - \hat{h} - \hat{h}^* & \hat{h} - \hat{h}^* \\ \hat{h}^* - \hat{h} & -2 + \hat{h} + \hat{h}^* \end{bmatrix}$$

Where $\hat{h}(z)$ is a discrete-time transfer function with impulse response h_k satisfying:

$$\sum_{k=-\infty}^{\infty} \max(1, \rho^{-2k}) |h_k| \leq 1$$

If we assume for now that the Zames–Falb ρ -IQC is causal, then a possible factorization is:

$$\Pi(z) = \Psi(z)^* \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \Psi(z), \quad \text{with } \Psi(z) = \frac{1}{2} \begin{bmatrix} 2 - \hat{h} & \hat{h} \\ \hat{h} & 2 - \hat{h} \end{bmatrix}.$$

Assume $\hat{h}$ is a finite impulse response filter, i.e. $\hat{h}(z) = h_1 z^{-1} + h_2 z^{-2} + \dots + h_k z^{-k}$. Consider the realization:

$$\Psi(z) = \left[\begin{array}{cccc|cc} 0 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 & 0 \\ 0 & 0 & \cdots & 0 & \frac{1}{2} & -\frac{1}{2} \\ \hline -h_k & -h_{k-1} & \cdots & -h_1 & 1 & 0 \\ h_k & h_{k-1} & \cdots & h_1 & 0 & 1 \end{array} \right]$$

We see that $D_\Psi^{21} = 0$ and $D_\Psi^{22} = 1$, as desired. Therefore, Equation (20) holds, but the $\hat{h}$ coefficients only appear in C_1 and C_2 , and the algorithm parameters only appear in an affine fashion and in the terms A, B, C_1, D_1 . The lower-left corner of (20) can be written in terms of the algorithm parameters by substituting (19). The result is (again, dashed lines indicate block structure)

$$\begin{aligned} \left[\begin{array}{c|c} A & B \\ \hline C_1 & D_1 \end{array} \right] &= \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & \frac{L-m}{2} \\ 0 & 0 & A_\Psi & B_\Psi^2 \\ 0 & 0 & C_\Psi^1 & D_\Psi^{12} \end{bmatrix}}_{\Phi} \\ &+ \underbrace{\begin{bmatrix} I & 0 \\ 0 & \frac{L+m}{2} \\ 0 & B_\Psi^1 \\ 0 & D_\Psi^{11} \end{bmatrix}}_S \underbrace{\begin{bmatrix} A_K & B_K \\ C_K & D_K \end{bmatrix}}_K \underbrace{\begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_T \end{aligned} \quad (21)$$

Let columns of $S^\perp$ and $T^\perp$ be bases for the left and right nullspaces of S and T , respectively. These are given by:

$$\begin{aligned} S^\perp &= \begin{bmatrix} I & 0 \\ 0 & \frac{L+m}{2} \\ 0 & B_\Psi^1 \\ 0 & 1 \end{bmatrix}^\perp = \begin{bmatrix} 0 & 1 & 0 & -\frac{L+m}{2} \\ 0 & 0 & B_1^\perp & B_2^\perp \end{bmatrix}, \\ T^\perp &= \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^\perp = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ I & 0 \\ 0 & 1 \end{bmatrix} \end{aligned}$$

where $[B_1^\perp \ B_2^\perp] \begin{bmatrix} B_\Psi^1 \\ 1 \end{bmatrix} = 0$. Substituting (21) into (20) and applying Lemma 2, the LMI condition (20) is feasible

if and only if:

$$\begin{aligned} \begin{bmatrix} I & 0 \\ 0 & S^\perp \end{bmatrix} \begin{bmatrix} \begin{bmatrix} \rho^2 P + C_2^\top C_2 & C_2^\top \\ C_2 & 1 \end{bmatrix} & \Phi^\top \\ & \Phi \end{bmatrix} \begin{bmatrix} P^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & S^\perp \end{bmatrix}^\top \succeq 0 \\ \begin{bmatrix} T^\perp & 0 \\ 0 & I \end{bmatrix}^\top \begin{bmatrix} \begin{bmatrix} \rho^2 P + C_2^\top C_2 & C_2^\top \\ C_2 & 1 \end{bmatrix} & \Phi^\top \\ & \Phi \end{bmatrix} \begin{bmatrix} P^{-1} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} T^\perp & 0 \\ 0 & I \end{bmatrix} \succeq 0 \end{aligned}$$

Expanding and factoring the (1,1) block to expose P :

$$\begin{aligned} \begin{bmatrix} \begin{bmatrix} I & 0 \\ C_2 & 1 \end{bmatrix}^\top \begin{bmatrix} \rho^2 P & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ C_2 & 1 \end{bmatrix} & \Phi^\top S^{\perp\top} \\ S^\perp \Phi & S^\perp \begin{bmatrix} P^{-1} & 0 \\ 0 & 1 \end{bmatrix} S^{\perp\top} \end{bmatrix} \succeq 0 \\ \begin{bmatrix} T^{\perp\top} \begin{bmatrix} I & 0 \\ C_2 & 1 \end{bmatrix}^\top \begin{bmatrix} \rho^2 P & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ C_2 & 1 \end{bmatrix} T^\perp & T^{\perp\top} \Phi^\top \\ \Phi T^\perp & \begin{bmatrix} P^{-1} & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix} \succeq 0 \end{aligned}$$

Apply Proposition 1 to the (1,1) block of the first LMI above to obtain an equivalent LMI in $Q := P^{-1}$. Next apply Proposition 1 to the (2,2) block of the second LMI above to obtain an equivalent LMI in P . This yields the following two conditions:

$$\begin{aligned} S^\perp \begin{bmatrix} Q & 0 \\ 0 & 1 \end{bmatrix} S^{\perp\top} \\ - S^\perp \Phi \begin{bmatrix} I & 0 \\ -C_2 & 1 \end{bmatrix} \begin{bmatrix} \rho^{-2} Q & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ -C_2 & 1 \end{bmatrix}^\top \Phi^\top S^{\perp\top} \succeq 0 \\ T^{\perp\top} \begin{bmatrix} I & 0 \\ C_2 & 1 \end{bmatrix}^\top \begin{bmatrix} \rho^2 P & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ C_2 & 1 \end{bmatrix} T^\perp \\ - T^{\perp\top} \Phi^\top \begin{bmatrix} P & 0 \\ 0 & 1 \end{bmatrix} \Phi T^\perp \succeq 0 \end{aligned}$$

These LMIs are block 4×4 and the dimensions are $(n_K + 1 + n_\Psi + 1)$. To proceed further, we need to break P and Q into their 3×3 blocks. These blocks have dimensions $(n_K + 1 + n_\Psi)$. Substitute for $(S^\perp, T^\perp)$ and note that C_2 simplifies to: $C_2 = [D_\Psi^{21} C_K \quad D_\Psi^{21} D_K \quad C_\Psi^2] = [0 \quad 0 \quad C_\Psi^2]$. Upon further simplification, we find the first block-rows and block-columns of P and Q (which had dimension n_K cancel out. The resulting LMIs are:

$$\begin{aligned} \begin{bmatrix} 1 & 0 & -\frac{L+m}{2} \\ 0 & B_1^\perp & B_2^\perp \end{bmatrix} \begin{bmatrix} Q_{22} & Q_{23} & 0 \\ Q_{32} & Q_{33} & 0 \\ 0 & 0 & 1 \end{bmatrix} [\star]^\top \\ - \begin{bmatrix} 1 & 0 & -\frac{L+m}{2} \\ 0 & B_1^\perp & B_2^\perp \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{L-m}{2} \\ 0 & A_\Psi & B_\Psi^2 \\ 0 & C_\Psi^1 & D_\Psi^{12} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & I & 0 \\ 0 & -C_\Psi^2 & 1 \end{bmatrix} \\ \times \begin{bmatrix} \rho^{-2} Q_{22} & \rho^{-2} Q_{23} & 0 \\ \rho^{-2} Q_{32} & \rho^{-2} Q_{33} & 0 \\ 0 & 0 & 1 \end{bmatrix} [\star]^\top \succeq 0 \quad (22a) \end{aligned}$$

$$\begin{aligned} [\star]^\top \begin{bmatrix} \rho^2 P_{22} & \rho^2 P_{23} & 0 \\ \rho^2 P_{32} & \rho^2 P_{33} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & I & 0 \\ 0 & C_\Psi^2 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & 1 \end{bmatrix} \\ - [\star]^\top \begin{bmatrix} P_{22} & P_{23} & 0 \\ P_{32} & P_{33} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & \frac{L-m}{2} \\ 0 & A_\Psi & B_\Psi^2 \\ 0 & C_\Psi^1 & D_\Psi^{12} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & 1 \end{bmatrix} \succeq 0 \quad (22b) \end{aligned}$$

These conditions are two LMIs involving (sub-blocks) of P and Q but coupled by the non-convex constraint $PQ = I$. Apply Lemma 3 to replace the non-convex constraint by a rank constraint and the following LMI:

$$\begin{bmatrix} P_{22} & P_{23} & 1 & 0 \\ P_{32} & P_{33} & 0 & I \\ 1 & 0 & Q_{22} & Q_{23} \\ 0 & I & Q_{32} & Q_{33} \end{bmatrix} \succeq 0 \quad (22c)$$

The rank constraint is trivially satisfied if $n_K \geq 2$ and hence can be neglected. In this case (22) provides a set of three LMI conditions in P and Q . In particular, $n_K = 2$ corresponds to a third-order algorithm (two states in K plus the additional integrator).

Discussion. For each fixed value of ρ , if (22) is feasible for some P and Q , then it is possible to synthesize a controller (A_K, B_K, C_K, D_K) for which the rate of convergence ρ is certified over the class of sector-bounded and slope-restricted functions.

Unlike previous analyses where the algorithm is fixed and one seeks to certify a certain convergence rate by searching over IQCs of a given class [4, 8, 10, 12, 16], here we fix the particular Zames-Falb IQC and instead certify a rate of convergence over all possible algorithms. This synthesis is performed numerically in the next section for the class of off-by-one IQCs.

6 Off-by-one IQC certification

A special instance of the general class of Zames-Falb multipliers discussed in Section 5 is the weighted off-by-one ρ -IQC [10]. For a given $\rho \in (0, 1]$ and any $h_1 \in [0, \rho^2]$, the function ϕ satisfies the ρ -IQC defined by:

$$\begin{aligned} A_\Psi = 0, \quad B_\Psi = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \end{bmatrix}, \quad C_\Psi = \begin{bmatrix} -h_1 \\ h_1 \end{bmatrix}, \\ D_\Psi = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad M = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (23) \end{aligned}$$

The filter Ψ is parameterized by ρ and the free variable h_1 . It has one timestep of memory and the corresponding ρ -IQC captures the slope conditions that must hold between data $(y_k, u_k) = (y_k, \phi(y_k))$ at one timestep and the previous timestep (y_{k-1}, u_{k-1}) . If $h_1 = 0$ then the constraint reduces to the sector IQC. In other words, this form of the weighted off-by-1 subsumes the sector bound used in Section 4. The more general Zames-Falb multipliers uses a filter Ψ with additional memory to incorporate the slope constraints across multiple time steps.

Substituting the IQC (23) into the semidefinite program (22) and using $h_1 = \rho^2$ as in [10], we obtain a convex program in (P, Q) for each fixed value of ρ (and for fixed values of L and m). Performing a bisection search on ρ , we can obtain the minimum ρ such that the LMI is feasible. We plot this result in Figure 3, where we can see that the numerical results agree perfectly with $\rho = 1 - \sqrt{m/L}$, which is the fastest known convergence rate achievable for strongly convex functions with Lipschitz-continuous gradients, and it is achieved by the *triple momentum method* [17].

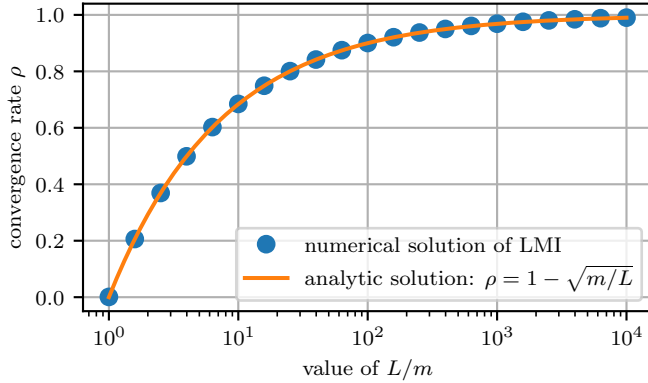


Figure 3: The circular dots are numerical solutions of the LMI (22) for different values of L/m , using the off-by-one IQC and performing a bisection search on ρ . The solid curve is a plot of the function $\rho = 1 - \sqrt{m/L}$, which is the fastest known achievable worst-case rate [17].

7 Concluding remarks

We demonstrated that tools from robust synthesis can be leveraged for algorithm design. Unlike prior works that fix the algorithm size (fixed memory) and seek a Lyapunov candidate that guarantees a certain convergence rate over a class of functions, we instead fix the IQC used to characterize the class of functions, and then search over all possible algorithms (with finite but arbitrary memory) for the fastest certifiable rate.

We show that if ∇f is sector-bounded, no algorithm of any size can outperform gradient descent. We then show that if ∇f satisfies the weighted off-by-one ρ -IQC [10], no algorithm of any (finite) memory can outperform the triple momentum method. In short, we show that eliminating the control variables from the associated LMIs can lead to novel algorithm performance bounds that are independent of algorithm dimension.

References

- [1] R. Boczar, L. Lessard, A. Packard, and B. Recht. Exponential stability analysis via integral quadratic constraints, 2017. arXiv:1706.01337 [cs.SY].
- [2] S. P. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994.
- [3] J. Carrasco, M. C. Turner, and W. P. Heath. Zames–Falb multipliers for absolute stability: From O’Shea’s contribution to convex searches. *European Journal of Control*, 28:1–19, 2016.
- [4] S. Cyrus, B. Hu, B. V. Scoy, and L. Lessard. A robust accelerated optimization algorithm for strongly convex functions. In *American Control Conference*, pages 1376–1381, 2018.
- [5] R. Freeman. Noncausal Zames–Falb multipliers for tighter estimates of exponential convergence rates. In *American Control Conference*, pages 2984–2989, 2018.
- [6] P. Gahinet and P. Apkarian. A linear matrix inequality approach to H_∞ control. *International Journal of Robust and Nonlinear Control*, 4(4):421–448, 1994.
- [7] B. Hu and L. Lessard. Control interpretations for first-order optimization methods. In *American Control Conference*, pages 3114–3119, 2017.
- [8] B. Hu and L. Lessard. Dissipativity theory for Nesterov’s accelerated method. In *International Conference on Machine Learning*, pages 1549–1557, 2017.
- [9] B. Hu and P. Seiler. Exponential decay rate conditions for uncertain linear systems using integral quadratic constraints. *IEEE Transactions on Automatic Control*, 61(11):3631–3637, 2016.
- [10] L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.
- [11] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004.
- [12] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan. A general analysis of the convergence of ADMM. In *International Conference on Machine Learning*, pages 343–352, 2015.
- [13] A. Packard, K. Zhou, P. Pandey, and G. Becker. A collection of robust control problems leading to LMIs. In *IEEE Conference on Decision and Control*, pages 1245–1250, 1991.
- [14] B. T. Polyak. Introduction to optimization. *Optimization Software, Inc.*, 1987.
- [15] P. Seiler. Stability analysis with dissipation inequalities and integral quadratic constraints. *IEEE Transactions on Automatic Control*, 60(6):1704–1709, 2015.
- [16] A. Sundararajan, B. Hu, and L. Lessard. Robust convergence analysis of distributed optimization algorithms. In *Allerton Conference on Communication, Control, and Computing*, pages 1206–1212, 2017.
- [17] B. Van Scoy, R. A. Freeman, and K. M. Lynch. The fastest known globally convergent first-order method for minimizing strongly convex functions. *IEEE Control Systems Letters*, 2(1):49–54, 2018.