A Robust Accelerated Optimization Algorithm for Strongly Convex Functions

Saman Cyrus[1,2] Bin Hu[2] Bryan Van Scoy[2] Laurent Lessard[1,2]

Abstract

This work proposes an accelerated first-order algorithm we call the Robust Momentum Method for optimizing smooth strongly convex functions. The algorithm has a single scalar parameter that can be tuned to trade off robustness to gradient noise versus worst-case convergence rate. At one extreme, the algorithm is faster than Nesterov's Fast Gradient Method by a constant factor but more fragile to noise. At the other extreme, the algorithm reduces to the Gradient Method and is very robust to noise. The algorithm design technique is inspired by methods from classical control theory and the resulting algorithm has a simple analytical form. Algorithm performance is verified on a series of numerical simulations in both noise-free and relative gradient noise cases.

1 Introduction

Consider the unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is L-smooth and m-strongly convex. The strong convexity of f guarantees that there exists a unique minimizer $x_\star$ satisfying $\nabla f(x_\star) = 0$. First-order methods are widely used for solving (1) when the Hessian is prohibitively expensive to compute, e.g., when the problem dimension is large. A simple first-order algorithm for solving (1) is the Gradient Method (GM),

$$x_{k+1} = x_k - \alpha\nabla f(x_k), \qquad x_0 \in \mathbb{R}^n.$$

For smooth and strongly convex f, the GM with a well-chosen stepsize converges linearly to the optimizer [1]. That is, for some $c \geq 0$ and $\rho \in [0, 1)$, we have

$$\|x_k - x_*\| \leq c\,\rho^k \quad \text{for all } k \geq 0.$$

For example, the standard choice $\alpha = 1/L$ leads to a linear rate $\rho = 1 - \frac{m}{L}$, while the choice $\alpha = \frac{2}{L+m}$ results in the improved linear rate $\rho = \frac{L-m}{L+m}$.

The issue with the Gradient Method, however, is that the convergence rate is slow, especially for ill-conditioned problems where the ratio $\frac{L}{m}$ is large. A common method of accelerating convergence is to use *momentum.* A well-established momentum algorithm for smooth and strongly convex f is Nesterov's Fast Gradient Method[1], (FGM) [2] described by the iteration

$$\begin{aligned} x_{k+1} &= y_k - \alpha\nabla f(y_k), \qquad x_0, x_{-1} \in \mathbb{R}^n \\ y_k &= x_k + \beta(x_k - x_{k-1}). \end{aligned}$$

The FGM tuned with $\alpha = \frac{1}{L}$ and $\beta = \frac{\sqrt{L}-\sqrt{m}}{\sqrt{L}+\sqrt{m}}$ converges with rate $\rho^2 < 1 - \sqrt{m/L}$, which is faster than the GM rate[2]. The rate can be improved to $\rho = 1 - \sqrt{m/L}$ using an accelerated algorithm called the Triple Momentum Method [4]. This is the fastest known worst-case convergence rate for this class of problems.

Robustness issues arise naturally in many optimization problems. For example, achieving the above rates associated with each first-order method requires knowledge of L and m, which may not be accurately accessible in practice. In addition, the gradient evaluation can be inexact for certain applications [5–7]. These issues motivate the need for accelerated first-order methods that are robust to underlying design assumptions.

As observed in [3, §5.2], optimization algorithm design involves a tradeoff between performance and robustness. For example, consider stepsize tuning for the GM. Using $\alpha = \frac{2}{L+m}$ optimizes the convergence rate, but makes the algorithm fragile to gradient noise. The more conservative choice $\alpha = \frac{1}{L}$ results in slower convergence, but more robustness to noise. This is consistent with the intuition that a smaller stepsize can improve the algorithm's robustness at the price of degrading its performance. For momentum methods, exploiting the tradeoff between performance and robustness is less straightforward, since one has to tune multiple algorithm parameters in a coupled manner to achieve acceleration. This tradeoff is exploited in [8] for first-order methods applied to smooth convex problems. In this work, we design a first-order method that exploits the tradeoff between robustness and performance for smooth strongly convex problems.

Notation. The set of functions that are m-strongly convex and L-smooth is denoted $\mathcal{F}(m, L)$. In particu-

[1] S. Cyrus and L. Lessard are with the Department of Electrical and Computer Engineering, University of Wisconsin–Madison.

[2] All authors are with the Wisconsin Institute for Discovery at the University of Wisconsin–Madison. Email addresses: {cyrus2,bhu38,vanscoy,laurent.lessard}@wisc.edu

[3] This material is based upon work supported by the National Science Foundation under Grants No. 1656951 and 1750162.

[1] Also called Neterov's accelerated gradient method.

[2] A numerical study in [3] revealed that the standard rate bound for FGM derived in [2] is conservative. Nevertheless, the bound has a simple algebraic form and is asymptotically tight.

lar, $f \in \mathcal{F}(m, L)$ if for all $x, y \in \mathbb{R}^n$,

$$m\|x-y\|^2 \le (\nabla f(x) - \nabla f(y))^\mathsf{T} (x-y) \le L\|x-y\|^2.$$

The condition ratio is defined as $\kappa := L/m$.

2 Main result

2.1 Robust Momentum Method

Our proposed algorithm is parameterized by a scalar ρ that represents the worst-case convergence rate of the algorithm in the noise-free case. Specifically, the iteration is governed by the following recursion with arbitrary initialization $x_0, x_{-1} \in \mathbb{R}^n$

$$\begin{aligned} x_{k+1} &= x_k + \beta(x_k - x_{k-1}) - \alpha \nabla f(y_k), && \text{(2a)} \\ y_k &= x_k + \gamma(x_k - x_{k-1}). && \text{(2b)} \end{aligned}$$

where α, β, and γ depend directly on the parameter ρ as

$$\begin{aligned} \alpha &= \frac{\kappa(1-\rho)^2(1+\rho)}{L}, \qquad \beta = \frac{\kappa\rho^3}{\kappa - 1}, \\ \gamma &= \frac{\rho^3}{(\kappa-1)(1-\rho)^2(1+\rho)}. \end{aligned} \tag{3}$$

We now state the key convergence property of the Robust Momentum Method in the noise-free case.

Theorem 1. *Suppose $f \in \mathcal{F}(m, L)$ with $0 < m \le L$ and let $x_\star$ be the unique minimizer of f. Given the parameter $\rho \in [1 - 1/\sqrt{\kappa},\, 1 - 1/\kappa]$, the Robust Momentum Method* (2) *with parameter tuning* (3) *satisfies the bound*

$$\|x_k - x_\star\| \le c\,\rho^k \qquad \text{for } k \ge 1 \tag{4}$$

where $c > 0$ is a constant that does not depend on k.

The proof of Theorem 1 is provided in Section 2.2. Theorem 1 states that ρ directly controls the worst-case convergence rate of the Robust Momentum Method. We will see in Section 3 that although increasing ρ makes the algorithm slower, it also makes it more robust to gradient noise. In particular,

- The minimum value is $\rho = 1 - 1/\sqrt{\kappa}$. This is the fastest achievable convergence rate and also leads to the most fragile algorithm. This choice recovers the Triple Momentum Method [4].

- The maximum value is $\rho = 1 - 1/\kappa$. This is the slowest achievable convergence rate and also leads to the most robust algorithm. This choice recovers the Gradient Method with stepsize $\alpha = 1/L$.

To see why this last case reduces to the Gradient Method, substitute $\rho = 1 - 1/\kappa$ into (2) and (3). Then, (2a) reduces to $y_{k+1} = y_k - \frac{1}{L}\nabla f(y_k)$.

2.2 Convergence rate proof

In this section, we derive a proof for Theorem 1. The approach that follows is similar to the one used in [3], with one important difference. In addition to proving a rate bound as in [3], we also derive a Lyapunov function that yields intuition for the algorithm's behavior and robustness properties.

Proposition 2 (Co-coercivity)**.** *Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable. Further suppose f is L-smooth. Then for all $x, y \in \mathbb{R}^n$,*

$$f(y) \ge f(x) + \nabla f(x)^\mathsf{T}(y-x) + \frac{1}{2L}\|\nabla f(y) - \nabla f(x)\|^2.$$

The following lemma proves a key property of strongly convex functions. Parts of this result appear in [3] and we repeat them here for completeness.

Lemma 3. *Suppose $f \in \mathcal{F}(m, L)$. Let $x_\star$ be the unique minimizer of f (i.e., $\nabla f(x_\star) = 0$). Define the function $g(x) := f(x) - f(x_\star) - \frac{m}{2}\|x - x_\star\|^2$. Given any sequence of points $\{y_k\} \subseteq \mathbb{R}^n$,*

1. *If we define $q_k := (L-m)g(y_k) - \frac{1}{2}\|\nabla g(y_k)\|^2$, then*

$$q_k \ge 0 \quad \text{for all } k.$$

2. *If we define $u_k := \nabla f(y_k)$ and $\tilde{y}_k := y_k - x_\star$, then*

$$(u_k - m\tilde{y}_k)^\mathsf{T}(L\tilde{y}_k - u_k) \ge q_k \quad \text{for all } k.$$

3. *Using the same definitions as above, the following inequality holds for any $0 \le \rho \le 1$,*

$$\begin{aligned} &(u_k - m\tilde{y}_k)^\mathsf{T}\big(L(\tilde{y}_k - \rho^2\tilde{y}_{k-1}) \\ &\qquad - (u_k - \rho^2 u_{k-1})\big) \ge q_k - \rho^2 q_{k-1} \quad \text{for all } k. \end{aligned}$$

Proof. By the definition of strong convexity, g is convex and $(L-m)$-smooth. Moreover, $g(y) \ge g(x_\star) = 0$ for all $y \in \mathbb{R}^n$. Item 1 follows from applying Proposition 2 with $(f, x, y) \mapsto (g, x_\star, y_k)$. For Item 2, note that $u_k = \nabla f(y_k) = \nabla g(y_k) + m\tilde{y}_k$. We have

$$\begin{aligned} (u_k - m\tilde{y}_k)^\mathsf{T}(L\tilde{y}_k - u_k) &= \nabla g(y_k)^\mathsf{T}\big((L-m)\tilde{y}_k - \nabla g(y_k)\big) \\ &\ge (L-m)g(y_k) - \tfrac{1}{2}\|\nabla g(y_k)\|^2 \\ &= q_k \end{aligned}$$

where the inequality follows from applying Proposition 2 with $(f, x, y) \mapsto (g, y_k, x_\star)$. To prove Item 3, begin with the case $\rho = 1$. Using a similar argument to the one used to prove Item 2,

$$\begin{aligned} &(u_k - m\tilde{y}_k)^\mathsf{T}\big(L(\tilde{y}_k - \tilde{y}_{k-1}) - (u_k - u_{k-1})\big) \\ &= \nabla g(y_k)^\mathsf{T}\big((L-m)(\tilde{y}_k - \tilde{y}_{k-1}) - (\nabla g(y_k) - \nabla g(y_{k-1}))\big) \\ &\ge q_k - q_{k-1} \end{aligned}$$

where the inequality follows from applying Proposition 2 with $(f, x, y) \mapsto (g, y_k, y_{k-1})$. By combining the two previous results, we have

$$\begin{aligned}
&(u_k - m\tilde{y}_k)^\mathsf{T}\big(L(\tilde{y}_k - \rho^2\tilde{y}_{k-1}) - (u_k - \rho^2 u_{k-1})\big)\\
&\quad= (1-\rho^2)(u_k - m\tilde{y}_k)^\mathsf{T}\big(L\tilde{y}_k - u_k\big)\\
&\qquad\qquad + \rho^2(u_k - m\tilde{y}_k)^\mathsf{T}\big(L(\tilde{y}_k - \tilde{y}_{k-1}) - (u_k - u_{k-1})\big)\\
&\quad\geq (1-\rho^2)q_k + \rho^2(q_k - q_{k-1})\\
&\quad= q_k - \rho^2 q_{k-1}
\end{aligned}$$

and this completes the proof of Item 3. ■

Our next lemma provides a key algebraic property of the Robust Momentum Method (2). This result makes no assumptions about f.

Lemma 4. *Suppose $\{u_k, x_k, y_k\}$ is any sequence of vectors satisfying the constraints*

$$\begin{bmatrix} x_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} 1+\beta & -\beta & -\alpha \\ 1+\gamma & -\gamma & 0 \end{bmatrix} \begin{bmatrix} x_k \\ x_{k-1} \\ u_k \end{bmatrix} \quad \textit{for } k \geq 0 \quad (5)$$

where (α, β, γ) are given by (3), and thus depend on the parameters $0 < m \leq L$, $\kappa := L/m$, and $\rho \in (0,1)$. Define $z_k := (1-\rho^2)^{-1}\big(x_k - \rho^2 x_{k-1}\big)$ for $k \geq 0$. Then the following algebraic identity holds for $k \geq 1$,

$$\begin{aligned}
&(u_k - my_k)^\mathsf{T}\big(L(y_k - \rho^2 y_{k-1}) - (u_k - \rho^2 u_{k-1})\big)\\
&\qquad + \lambda\left(\|z_{k+1}\|^2 - \rho^2\|z_k\|^2\right) + \nu\|u_k - my_k\|^2 = 0 \quad (6)
\end{aligned}$$

where the constants λ and ν are defined as

$$\lambda := \frac{m^2\left(\kappa - \kappa\rho^2 - 1\right)}{2\rho(1-\rho)} \quad \textit{and} \quad (7)$$

$$\nu := \frac{(1+\rho)\left(1 - \kappa + 2\kappa\rho - \kappa\rho^2\right)}{2\rho}. \quad (8)$$

Proof. The algebraic identity may be verified by direct substitution of (3), (5), (7), and (8) into (6). Specifically, the constraints (5) allow us to express z_{k+1}, z_k, y_k, y_{k-1}, u_k, and u_{k-1} as linear functions of x_k, x_{k-1}, x_{k-2}, and u_k. Upon doing so, the resulting expression becomes identically zero. To express u_{k-1} as required, rearrange the first equation of (5) to obtain the expression $u_{k-1} = \alpha^{-1}((1+\beta)x_{k-1} - \beta x_{k-2} - x_k)$. ■

The algebraic identity (6) has three main terms. We will see how each serves a role in explaining the convergence and robustness properties of our algorithm. We are now ready to prove Theorem 1.

Proof of Theorem 1. Choose x_0 and x_{-1} arbitrarily and consider the sequence $\{u_k, x_k, y_k, z_k\}$ defined by setting $u_k := \nabla f(y_k)$ and propagating for all $k \geq 0$ using (5). This sequence is precisely a trajectory of our algorithm. Let $x_\star$ be the unique minimizer of f. Define the shifted sequences $\tilde{x}_k := x_k - x_\star$, $\tilde{y}_k := y_k - x_\star$, and $\tilde{z}_k := z_k - x_\star$ where z_k is defined in Lemma 4. Note that the constraints (5) still hold when we use the shifted sequence $\{u_k, \tilde{x}_k, \tilde{y}_k, \tilde{z}_k\}$. Applying Lemma 4 with Item 3 of Lemma 3, we conclude that for $k \geq 1$,

$$\begin{aligned}
\lambda(\|\tilde{z}_{k+1}\|^2 - \rho^2\|\tilde{z}_k\|^2) + (q_k - \rho^2 q_{k-1})&\\
+ \nu\,\|u_k - m\tilde{y}_k\|^2 \leq 0, &\quad (9)
\end{aligned}$$

where λ and ν are defined in (7)–(8). When $1 - 1/\sqrt{\kappa} \leq \rho \leq 1 - 1/\kappa$, we have $mL \geq \lambda \geq \frac{1}{2}mL$ and $0 \leq \nu \leq 1 - \frac{1}{2\kappa}$. As we increase ρ, the parameter λ decreases monotonically while ν increases monotonically. Define the sequence $\{V_k\}$ by $V_k := \lambda\|\tilde{z}_k\|^2 + q_{k-1}$. If we choose ρ in the interval specified above, then $\nu \geq 0$ and $\lambda > 0$. Since $q_k \geq 0$, V_k can serve as a Lyapunov function. In particular, it follows from (9) that

$$V_{k+1} \leq \rho^2\, V_k \qquad \text{for } k \geq 1. \quad (10)$$

Iterating this relationship, we find that $V_{k+1} \leq \rho^{2k}\, V_1$. The reason we do not iterate down to zero is because V_k is not defined at $k = 0$. Substituting the definitions and simplifying, we obtain the bound

$$\|\tilde{z}_{k+1}\| \leq \rho^k\sqrt{\|\tilde{z}_1\|^2 + \lambda^{-1}q_0} \qquad \text{for } k \geq 1. \quad (11)$$

The bound (11) therefore captures two effects. As we increase ρ, the linear rate ρ^k becomes slower and the constant factor in the rate bound also grows.

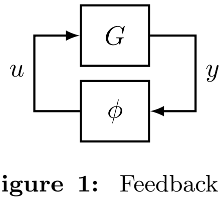
Next, we show that $\{\tilde{x}_k\}$ goes to zero at the same rate ρ^k, but with different constant factors. Note that because $\tilde{z}_k = (1-\rho^2)^{-1}\big(\tilde{x}_k - \rho^2\tilde{x}_{k-1}\big)$, we can form the telescoping sum

$$\tilde{x}_k = \rho^{2(k-1)}\tilde{x}_{-1} + (1-\rho^2)\sum_{t=0}^{k-1}\rho^{2(k-t)}\tilde{z}_t \quad \text{for } k \geq 0. \quad (12)$$

Taking the norm of both sides of (12), applying the triangle inequality, and substituting (11), we obtain a geometric series. Upon simplification, we find that $\|\tilde{x}_k\|$ is bounded above by a constant times ρ^k, as required.

3 Control design interpretations

In this section, we cast the problem of algorithm analysis as a robust control problem. Specifically, we can view the problem of algorithm analysis as being equivalent to solving a Lur'e problem [9]. The Lur'e setup is illustrated in Figure 1, where a linear dynamical system G (13) is in feedback with a static nonlinearity ϕ.



$$\begin{aligned}
\xi_{k+1} &= A\xi_k + Bu_k, &(13a)\\
y_k &= C\xi_k, &(13b)\\
u_k &= \phi(y_k). &(13c)
\end{aligned}$$

Figure 1: Feedback interconnection of a linear system G with a troublesome (nonlinear or uncertain) component ϕ. We use the positive feedback convention in this block diagram.

The Robust Momentum Method (as well as the Fast Gradient Method and ordinary Gradient Method) can be written in this way by setting $\phi = \nabla f$ and choosing A, B, and C appropriately. For example, the Robust Momentum Method (2) is given by

$$A = \begin{bmatrix} 1+\beta & -\beta \\ 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -\alpha \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 1+\gamma & -\gamma \end{bmatrix}.$$

Here, we shifted all signals so they are measured relative to the steady-state value $x_\star$ and therefore assumed that $\nabla f(0) = 0$. We also assumed without loss of generality that u_k and y_k are scalars. This interpretation was used in [3, 10, 11] to provide a unified analysis framework.

Traditionally, Lur'e systems were analyzed in the frequency domain rather than the time domain. For the case of the Robust Momentum Method, the (discrete-time) transfer function of the linear block is given by

$$G(z) = -\alpha \frac{(1+\gamma)z - \gamma}{(z-1)(z-\beta)}. \tag{14}$$

It was observed in Section 2.1 that the Robust Momentum Method becomes the Gradient Method if $\rho = 1 - 1/\kappa$. This fact can be directly verified using the transfer function. Substituting this ρ and the parameter values (3) into (14), there is a pole-zero cancellation and we obtain $G(z) = \frac{-1}{L(z-1)}$, which is the transfer function for the Gradient Method with stepsize $\alpha = \frac{1}{L}$.

Frequency-domain condition. Continuing with the frequency-domain interpretation, Lur'e systems can be analyzed using the formalism of Integral Quadratic Constraints (IQCs) [12]. To this end, the nonlinearity is characterized by a quadratic inequality that holds between its input and output

$$\int_{|z|=1} \begin{bmatrix} \hat{y}(z) \\ \hat{u}(z) \end{bmatrix}^* \Pi(z) \begin{bmatrix} \hat{y}(z) \\ \hat{u}(z) \end{bmatrix} \mathrm{d}z \geq 0$$

where $\hat{y}$ and $\hat{u}$ are the z-transforms of $\{y_k\}$ and $\{u_k\}$, respectively, and $\Pi(z)$ is a para-Hermitian matrix. For convenience, we use a loop-shifting transformation to move the nonlinearity $\phi = \nabla f$ from the sector (m, L) to the sector $(0, \kappa - 1)$. We also scale the frequency variable z by a factor of ρ so that we can reduce the problem of certifying exponential stability (finding a linear rate) to that of certifying BIBO stability. This procedure is described in [13].

The nonlinearity of interest is sector-bounded and slope-restricted because it is the gradient of a function $g \in \mathcal{F}(0, \kappa - 1)$. We may therefore represent the nonlinearity with a Zames–Falb IQC as in [13], leading to

$$\Pi(z) := \begin{bmatrix} 0 & (\kappa-1)(1-\rho^2 \bar{z}^{-1}) \\ (\kappa-1)(1-\rho^2 z^{-1}) & -2 + \rho^2(z^{-1} + \bar{z}^{-1}) \end{bmatrix}.$$

The transformed transfer function is

$$\tilde{G}(z) = \frac{-\alpha m(1+\gamma)z + \alpha m \gamma}{z^2 - (1 + \beta - \alpha m(1+\gamma))z + \beta - \alpha m \gamma}. \tag{15}$$

To certify stability of the feedback interconnection, we must have $\tilde{G}(\rho z)$ stable and for all $|z| = 1$,

$$\operatorname{Re}\Big((1 - \rho z^{-1})\big((\kappa - 1)\tilde{G}(\rho z) - 1\big)\Big) < 0. \tag{16}$$

Equation (16) has a graphical interpretation; that the Nyquist plot of $F(z) := (1 - \rho z^{-1})\big((\kappa - 1)\tilde{G}(\rho z) - 1\big)$ should lie entirely in the left half-plane.

Graphical design for robustness. The frequency-domain condition (16) can provide useful intuition for the design of robust accelerated optimization methods. We can visualize different algorithms by choosing the parameters α, β, γ appropriately in (15).

In Figure 2 (left panel), we show the Nyquist plot for the Gradient Method using the sector IQC [3, 13]. To this effect, we set $\beta = \gamma = 0$ and use either $\alpha = \frac{2}{L+m}$ or $\alpha = \frac{1}{L}$. As we increase ρ, the Nyquist plots become ellipses in the left half-plane. At the fastest certifiable rate (smallest ρ), the plots become vertical lines. When $\alpha = \frac{2}{L+m}$, the vertical line coincides with the imaginary axis, whereas when $\alpha = \frac{1}{L}$, the vertical line is shifted left. This result confirms our intuition that since the imaginary axis is the stability boundary, *robust* stability is achieved as the Nyquist contour moves further left, away from the boundary.

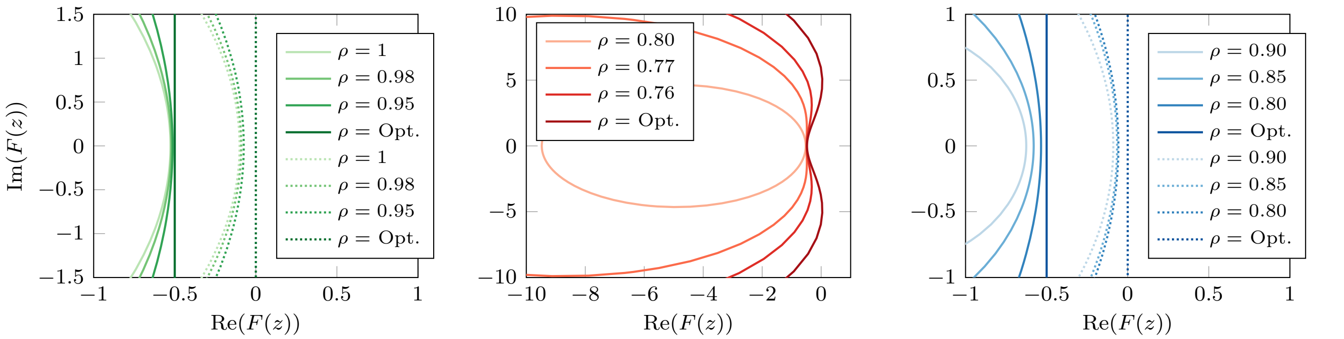
The Robust Momentum Method (2) was designed such that the Nyquist diagram forms a vertical line passing through the point $(-\nu, 0)$. In other words, we solved for (α, β, γ) such that (16) holds with the right-hand side replaced by $-\nu$. Constraining the Nyquist plot as such directly leads to the choice (3) with ν related to ρ via (8). In Figure 2 (right panel), we show the Nyquist plot for the Robust Momentum Method using the Zames–Falb IQC (for $\nu = 0$ and $\nu = \frac{1}{2}$). We also show Nyquist plots that certify a convergence rate of ρ that is larger than the corresponding algorithm parameter. This leads to ellipses as with the Gradient Method. Note that although the RMM and GM plots look similar, the RMM ρ-values are generally smaller due to acceleration. In contrast, the FGM (center panel) does not produce a vertical line in the Nyquist plot but still touches the stability boundary at the optimal ρ.

Further robustness interpretations. The parameter ν can be interpreted as the *input feed-forward passivity index* (IFP) [14], which is a measure of the shortage or excess of passivity of the system $F(z)$ defined above. In the frequency domain, the discrete-time definition of the IFP index is given by[3]

$$\nu(F(z)) := -\tfrac{1}{2} \max_{|z|=1} \lambda_{\max}\big(F(z) + F(z)^*\big), \tag{17}$$

where $\lambda_{\max}(\cdot)$ denotes the largest eigenvalue and F^* is the conjugate transpose of F. For the SISO case, (17)

[3] Most sources use a negative feedback convention. The definition we give in (17) uses the positive feedback convention.



(a) Gradient Method

(b) Fast Gradient Method

(c) Robust Momentum Method

Figure 2: Frequency-domain plots of various algorithms for $\kappa = 10$ and different values of the convergence rate ρ. The system is stable if the entire curve lies in the left half-plane. **(a)** Gradient Method for $\alpha = 1/L$ (solid) and $\alpha = 2/(L+m)$ (dashed). The latter is right on the stability boundary while the former is shifted left (more robust). **(b)** Fast Gradient Method. **(c)** Robust Momentum Method for $\nu = 1/2$ (solid) and $\nu = 0$ (dashed). Again, the latter is right on the stability boundary while the former is shifted left (more robust).

reduces to $\nu = -\max_{|z|=1} \operatorname{Re}(F(z))$, which is the shortest distance between each curve and the imaginary axis in Figure 2.

We can also interpret ν as a robustness margin in the time domain using the Lyapunov function defined in (8). In the proof of Theorem 1, when we substitute the definition for V_k into (9), we obtain

$$V_{k+1} \leq \rho^2\, V_k - \nu\, \|\nabla g(y_k)\|^2.$$

Proving the desired rate bound only requires (10) to hold, so the term $\nu\,\|\nabla g(y_k)\|^2$ can be interpreted as an additional margin that ensures the inequality $V_{k+1} \leq \rho^2 V_k$ will hold even if underlying assumptions such as exactness in gradient evaluations or accurate knowledge of L and m are violated. As we increase ρ, the linear rate becomes slower, but ν also increases via (8), which serves to increase the robustness margin in the inequality (10).

4 Robustness to gradient noise

The Robust Momentum Method has a single parameter, which can be used to tune the performance. In this section, we provide both simulations and numerical rate analyses to verify the performance of the algorithm when the gradient is subject to relative deterministic noise [1]. Specifically, we will suppose that instead of measuring the gradient $\nabla f(y_k)$, we measure $u_k = \nabla f(y_k) + r_k$ where $r_k \in \mathbb{R}^n$ satisfies $\|r_k\| \leq \delta\,\|\nabla f(y_k)\|$. For a given fixed $\delta \geq 0$, we will bound the worst-case performance of the algorithm over all $f \in \mathcal{F}(m, L)$ and feasible $\{r_k\}$.

Numerical rate analysis. To find the worst-case performance, we adopt the methodology from [3, Eq. 5.1]. There, the authors formulate a linear matrix inequality parameterized by $\hat{\rho}$ and δ whose feasibility provides a sufficient condition for convergence with linear rate $\hat{\rho}$.

In Figure 3, we plot the computed convergence rate as a function of noise strength δ for the Gradient Method, Fast Gradient Method, and Robust Momentum Method. Note that the worst-case rate in closed form for the Gradient Method is given in [15, 16].

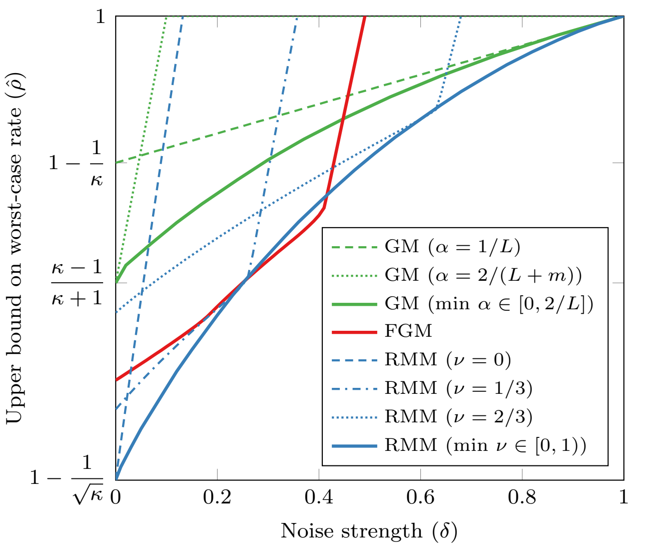
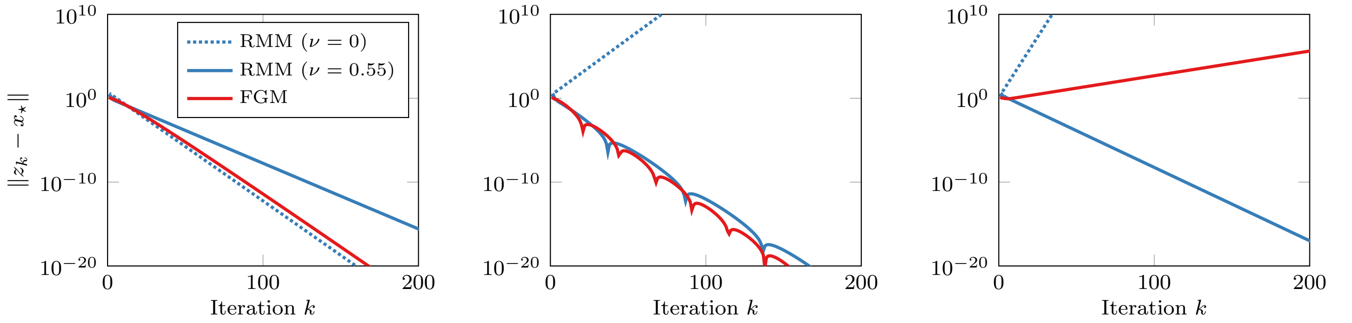


Figure 3: Upper bound on the worst-case linear convergence rate as a function of the noise level δ for $\kappa = 10$ (the figure looks similar for other choices of κ). We used a relative noise model, where the measured gradient u_k satisfies $\|u_k - \nabla f(y_k)\| \leq \delta\,\|\nabla f(y_k)\|$ for the Gradient Method (GM), Fast Gradient Method (FGM), and Robust Momentum Method (RMM). By tuning the parameter ν, the RMM trades off robustness to gradient noise with convergence rate.

First, consider the Robust Momentum Method. When $\nu = 0$ and there is no gradient noise ($\delta = 0$), the method achieves the fast convergence rate $1 - 1/\sqrt{\kappa}$. Increasing the noise level above $\delta > 0.13$, however, leads to a loss of convergence guarantee. As we increase ν, the convergence rate becomes slower but the method is capable of tolerating larger noise levels. In the limiting case as



(a) $\delta = 0$ **(b)** $\delta = 0.25$ **(c)** $\delta = 0.5$

Figure 4: Simulation of the Robust Momentum Method (RMM) and the Fast Gradient Method (FGM) with relative gradient noise of strength δ and condition ratio $\kappa = 10$. The objective function is the two-dimensional quadratic with gradient (18). The measured gradient at each iteration is $u_k = (1-\delta)\nabla f(y_k)$. **(a)** With no noise, all methods are stable and the RMM with $\nu = 0$ is the fastest. **(b)** With more noise, the RMM with $\nu = 0$, the most fragile possible tuning, is unstable. **(c)** With high noise, only the RMM with $\nu = 0.55$ remains stable. Even FGM is unstable with this much noise.

$\nu = 1 - \frac{1}{2\kappa}$ the Robust Momentum Method becomes the Gradient Method with $\alpha = \frac{1}{L}$ (dashed black line).

It is interesting to note that the Fast Gradient Method has a faster convergence bound than the Robust Momentum Method for noise levels $0.26 < \delta < 0.41$. However, the Fast Gradient Method is also unstable for $\delta > 0.5$ while the Robust Momentum Method can be tuned so that it converges with noise levels up to $\delta \to 1$.

Numerical simulations. To illustrate the noise robustness properties of different tunings of the Robust Momentum Method, we compared it to the Fast Gradient Method when applied to a simple two-dimensional quadratic function. We used the gradient

$$\nabla f(y_k) = \begin{bmatrix} m & 0 \\ 0 & L \end{bmatrix} (y_k - x_\star) \tag{18}$$

where the gradient noise is $r_k = -\delta\, \nabla f(y_k)$. See Figure 4. The RMM with $\nu = 0$ has the fastest convergence rate in the noiseless case ($\delta = 0$), but quickly diverges when noise is present. The FGM is more robust to noise, but also diverges when the noise magnitude δ is too large. The RMM with $\nu = 0.55$ remains stable for large amounts of noise, although in the absence of noise the convergence rate is slower than both other methods.

References

[1] B. T. Polyak, *Introduction to optimization.* New York: Optimization Software, Publications Division, 1987.

[2] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, ser. Applied Optimization. Boston, MA: Kluwer Academic Publishers, 2004, vol. 87.

[3] 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.

[4] 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*, vol. 2, no. 1, pp. 49–54, 2018.

[5] A. d'Aspremont, "Smooth optimization with approximate gradient," *SIAM J. Optim.*, vol. 19, no. 3, pp. 1171–1183, 2008.

[6] M. Schmidt, N. L. Roux, and F. R. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," in *Advances in neural information processing systems*, 2011, pp. 1458–1466.

[7] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Math. Program.*, vol. 146, no. 1-2, pp. 37–75, 2014.

[8] ——, "Intermediate gradient methods for smooth convex problems with inexact oracle," CORE Discussion Paper 2013/17, Tech. Rep., 2013.

[9] A. Lur'e and V. Postnikov, "On the theory of stability of control systems," *Applied mathematics and mechanics*, vol. 8, no. 3, pp. 246–248, 1944.

[10] B. Hu and L. Lessard, "Dissipativity theory for Nesterov's accelerated method," in *International Conference on Machine Learning*, vol. 70, 2017, pp. 1549–1557.

[11] B. Hu, P. Seiler, and A. Rantzer, "A unified analysis of stochastic optimization methods using jump system theory and quadratic constraints," in *Conference on Learning Theory*, vol. 65, 2017, pp. 1157–1189.

[12] A. Megretski and A. Rantzer, "System analysis via integral quadratic constraints," *IEEE Trans. Automat. Control*, vol. 42, no. 6, pp. 819–830, 1997.

[13] R. Boczar, L. Lessard, and B. Recht, "Exponential convergence bounds using integral quadratic constraints," in *IEEE Conf. Decision Control*, 2015, pp. 7516–7521.

[14] J. Bao and P. L. Lee, *Process control: The passive systems approach.* London: Springer-Verlag, 2007.

[15] E. de Klerk, F. Glineur, and A. B. Taylor, "On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions," *Optimization Letters*, vol. 11, no. 7, pp. 1185–1199, 2017.

[16] ——, "Worst-case convergence analysis of gradient and Newton methods through semidefinite programming performance estimation," *arXiv:1709.05191*, 2017.