Survey on Synthesis of Accelerated Gradient-Based Optimization Algorithms

Govind M. Chari

***Abstract*— Recently, optimization researchers started viewing optimization algorithms as dynamical systems. This view allowed them to apply techniques from system and control theory to analyze and even synthesize accelerated optimization algorithms. This article presents a survey of accelerated first-order algorithms for the minimization of smooth, strongly convex functions, an overview of recent developments in the synthesis of these algorithms, a review of a recent paper in this area, and some ideas for future research directions. All code used to generate figures is available here: https://github.com/govindchari/synthesis.**

***Index Terms*— Convex Optimization, Convex Synthesis, Linear Matrix Inequalities, Lyapunov Analysis**

I. Introduction

Numerical optimization is an area of applied mathematics which is concerned with developing algorithms for minimizing some function which may be subject to constraints. This field is widely applicable from developing trajectory optimization algorithms to land rockets to computing optimal allocations of assets in a financial portfolio to balance expected return and risk [1] [2].

One popular class of optimization algorithms are first-order or gradient-based algorithms. These algorithms aim to minimize the function using only gradient, or first-order information. Some examples of first-order algorithms are gradient descent or projected gradient descent. These are in contrast to second-order algorithms which use the function's Hessian, or second-order information. Some examples of second-order algorithms are Newton's method and interior point methods.

The advantage of first order methods is that they require less information about the function, and require less computational effort. Second-order algorithms requires Hessian information which is $\mathcal{O}(n^2)$ storage where n is the problem size, and naively requires $\mathcal{O}(n^3)$ floating-point operations per iteration due to the inversion of the Hessian matrix. On the other hand, first-order algorithms requires storing only gradient information which is $\mathcal{O}(n)$ storage, and requires $\mathcal{O}(n)$ floating-point operations per iteration. Of course, first-order algorithms take more iterations to converge since each iteration uses less information about the function, but a general rule of thumb is that first-order algorithms scale better than second-order algorithms in terms of storage and run-time as problem size increases.

Govind M. Chari is with the Aeronautics & Astronautics Department, University of Washington, Seattle, WA 98105 USA (e-mail: gchari@uw.edu).

Typical proofs for convergence and rates of convergence for gradient-based algorithms require applying numerous smoothness and strong convexity inequalities in a case-by-case basis for each algorithm, however recent work such as [3] treat optimization algorithms as dynamical systems and use ideas from robust control to prove convergence rates for first-order algorithms in a unified framework. These ideas were also used in [3] to synthesize accelerated gradient-based algorithms that are more robust to noisy gradients. Accelerated methods are variants of gradient-based methods which, when properly tuned, achieve faster convergence rates than gradient descent.

Paper [4], which is the focus of this survey, extends the idea of using Lyapunov analysis for algorithm synthesis to derive algorithms for optimization and saddle-point problems. This allows the synthesis of algorithms for minimizing strongly convex functions subject to linear equality constraints while also providing a certificate of convergence rate.

II. Background Information and Terminology

In this section we will review some basic concepts needed to understand the synthesis of gradient-based algorithms. In this section, we will consider unconstrained problems of the following form

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \tag{1}$$

where f is convex and continuously differentiable.

We will denote the optimizer of the function as x^* and the optimal objective value as f^*

A. Smoothness and Strong Convexity

Two important properties of convex functions that are are used to prove convergence of algorithms are strong convexity and smoothness.

A differentiable function $f(x)$ is said to be λ-smooth if the following holds

$$\|\nabla f(x) - \nabla f(y)\| \leq \lambda\|x - y\| \tag{2}$$

This condition is simply saying that the gradient of the function is λ-Lipschitz. Gradient based algorithms require functions to be λ-smooth to show global convergence for fixed step-sizes. A larger λ means the function's gradient changes quickly so the algorithm should take smaller steps so it does not overshoot the minimum.

If the function is also twice differentiable, (2) can be restated as follows

$$\nabla^2 f(x) \preccurlyeq \lambda I \tag{3}$$

where I is the identity matrix and $\preccurlyeq$ indicates the Loewner ordering. This condition is saying that the largest eigenvalue of the Hessian of the objective function is upper bounded by some constant λ.

From the definition of smoothness, we can also show the following

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{\lambda}{2}\|y - x\|_2^2 \tag{4}$$

This statement says that smooth functions can be upper bounded globally by some quadratic

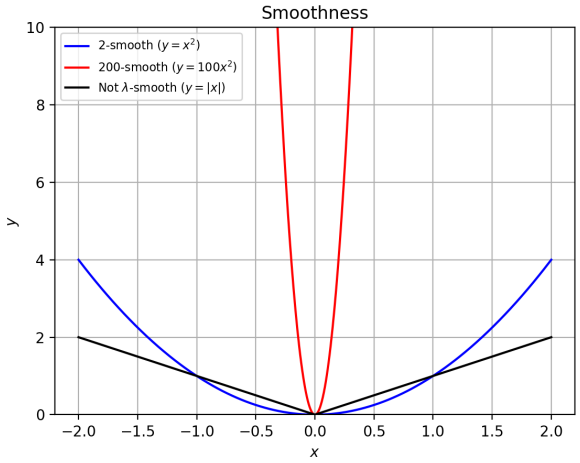

Fig. 1. Functions with different smoothness constants in one dimension

Convexity and λ-smoothness are the only requirements needed to show global convergence of fixed-stepsize gradient based algorithms, however we can show faster convergence rates if the function is also μ-strongly convex.

A function $f(x)$ is μ-strongly convex if the following function is convex

$$g(x) = f(x) - \frac{\mu}{2}\|x\|_2^2 \tag{5}$$

This condition is saying that a strongly convex function must be curving upwards in all directions

Alternatively we can say that a μ-strongly convex function satisfies the following

$$f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|_2^2 \tag{6}$$

If $\mu = 0$ we can recover the definition of convexity which states that all convex functions are lower-bounded by their tangent line. When we have a strongly convex function, we can say that the function is globally lower-bounded by some quadratic function.

We will denote the class of λ-smooth and μ-strongly convex functions as $\mathcal{S}_{\mu,\lambda}$.

If the function is also twice differentiable, (5) can be restated as follows

$$\nabla^2 f(x) \succcurlyeq \mu I \tag{7}$$

This condition is saying that the smallest eigenvalue of the Hessian of the objective function is lower bounded by some constant μ.

Putting together (4) and (6) we can write the following inequality for $\mathcal{S}_{\mu,\lambda}$ functions which shows that we can globally upper and lower bound this class of functions with quadratics.

$$\frac{\mu}{2}\|y-x\|_2^2 \leq f(y)-f(x)-\nabla f(x)^\top (y-x) \leq \frac{\lambda}{2}\|y-x\|_2^2 \tag{8}$$

For twice differentiable $\mathcal{S}_{\mu,\lambda}$ functions we can bound the Hessian as follows

$$\lambda I \preccurlyeq \nabla^2 f(x) \preccurlyeq \mu I \tag{9}$$

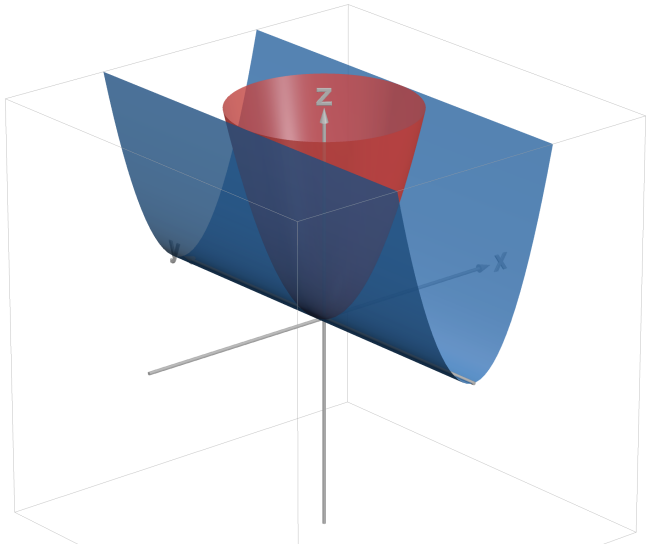

Fig. 2. Red denotes the graph of a strongly convex function, and blue denotes the graph of a non-strongly convex function

For $\mathcal{S}_{\mu,\lambda}$ functions we can define the condition number of the function as follows

$$\kappa = \frac{\lambda}{\mu} \tag{10}$$

Functions with larger condition number, also called ill-conditioned functions, have more elliptic level curves and are more difficult to minimize using gradient based methods.

B. Lyapunov Analysis

A classical result in nonlinear system theory is using Lyapunov functions to show global exponential stability for nonlinear dynamical systems.

Consider the following discrete-time nonlinear system:

$$x_{k+1} = g(x_k) \tag{11}$$

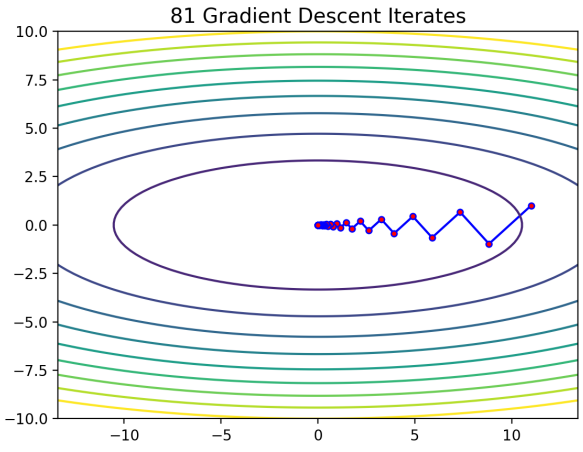

Fig. 3. Gradient Descent on ill-conditioned function

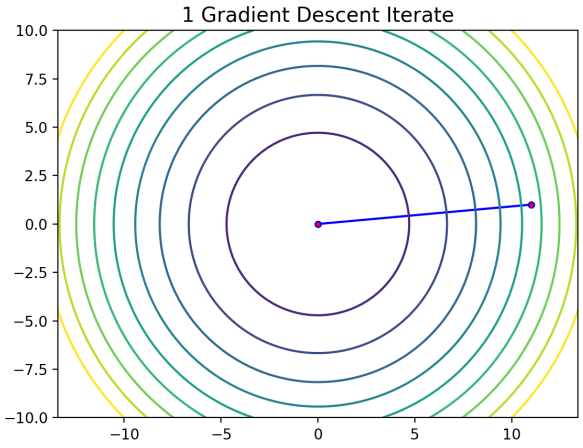

Fig. 4. Gradient Descent on well-conditioned function

where $x^* \in \mathbb{R}^n$ is a stationary point such that $x^* = g(x^*)$. If we can construct a Lyapunov function $V(x)$ such that

$$\alpha\|x - x^*\|_2^2 \leq V(x) \leq \beta\|x - x^*\|_2^2 \quad \forall x \in \mathbb{R}^n \tag{12a}$$

$$V(x_{k+1}) - \rho^2 V(x_k) \leq 0 \quad \forall x \in \mathbb{R}^n \tag{12b}$$

for some $\alpha > 0$, $\beta > 0$, and $\rho \in [0, 1]$, then the fixed point x^* of (11) is globally exponentially stable [5]. Mathematically, this can be expressed as

$$\|x^* - x_k\| \leq \sqrt{\frac{\beta}{\alpha}}\rho^k \|x^* - x_0\| \quad \forall x_0 \in \mathbb{R}^n \tag{13}$$

We can think of $V(x)$ as some function that quantifies the energy in the system for some state x. If this energy decreases as the system progresses forward, then we would expect the state to converge to some fixed point. This is the intuitive idea of what (12) and (13) are expressing.

This theory is used in [3], [4], and [6] to analyze convergence rates of first-order algorithms. Using Lyapunov theory to analyze convergence of optimization algorithms requires viewing the optimization algorithms as dynamical systems of the form (11), then applying ideas from system and control theory.

C. Convergence Rates

When showing convergence results for optimization algorithm we want to derive an upper bound on the algorithm's convergence to either x^* or f^*. Here we will focus on bounding convergence to x^*, since this is the bound we directly get by applying Lyapunov analysis as in (13).

Gradient based algorithms have linear convergence for functions that are in $\mathcal{S}_{\mu,\lambda}$.

Linear convergence take the following form:

$$\|x^* - x_k\| \leq c\rho^k \tag{14}$$

where $c > 0$ is some constant and $\rho \in [0, 1]$ is the linear convergence rate to the minimizer. This class of convergence rates is called linear since the distance to optimal is a line when plotted on a log plot. For the same constant c if ρ is smaller, then convergence is quicker.

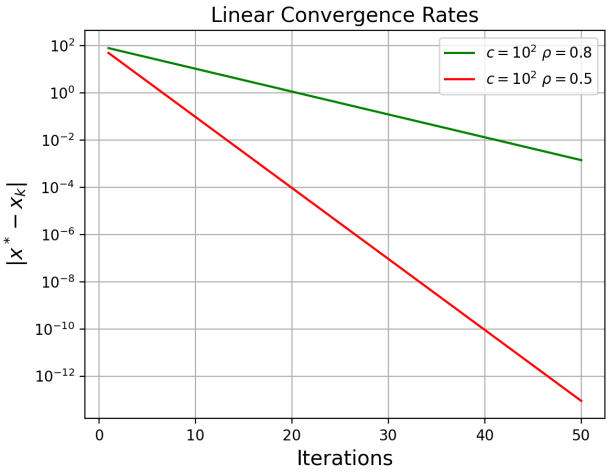

Fig. 5. Illustration of different classes of convergence rates

It is important to keep in mind that just because one algorithm has a better linear convergence rate (smaller ρ) than another does not mean it will be faster in practice for two reasons. Firstly, the constant c in (14) plays a large role in the proven convergence rate. Figure 6 demonstrates this well. Secondly, these proven rates are upper bounds on convergence rates, so an algorithm could perform much better in practice than the proven rates.

III. Accelerated First Order Methods

In this section we will only consider functions in $\mathcal{S}_{\mu,\lambda}$.

The first optimization algorithm proposed was gradient descent by Cauchy in 1846 [7]. However, for ill conditioned functions , this method can be painfully slow as the iterates will "zigzag" downhill as in 3.

The gradient descent update rule is given below

$$x_{k+1} = x_k - \eta\nabla f(x_k) \tag{15}$$

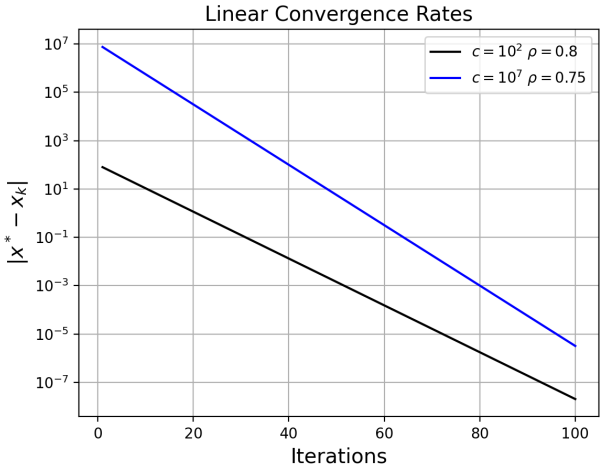

Fig. 6. Linear convergence with different constants

where $\eta = \frac{2}{\mu+\lambda}$ is the step-size.

The convergence rate of gradient descent can be written as follows:

$$\|x^* - x_k\| \leq c\left(1 - \frac{1}{\kappa}\right)^{0.5k} \tag{16}$$

We can see that as the condition number κ of the function increases, the linear convergence rate approaches unity which indicates that very little progress is made during each iteration.

It is possible to come up with accelerated methods which achieve faster convergence rates than gradient descent. We will explore some accelerated methods in this section.

A. Polyak's Heavy-Ball Method

In 1964 Boris Polyak invented the heavy ball method which was an attempt to develop a gradient based algorithm that was faster than gradient descent and didn't suffer from the same "zigzag" problem as gradient descent [8]. This method makes use of momentum. With this addition of momentum it is difficult for iterates to sharply change directions which damps the oscillations in figure 3. Additionally it allows for longer steps to be taken in regions of low curvature.

The heavy-ball algorithm is as follows

$$x_{k+1} = (1+\beta)x_k - \beta x_{k-1} - \alpha\nabla f(x_k) \tag{17}$$

for some choice of α and $\beta \in [0, 1]$. It is possible to choose these parameters to achieve a local convergence rate of

$$\|x^* - x_k\| \leq c\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{0.5k} \tag{18}$$

We can see that as the condition number κ of the function increases, the linear convergence rate approaches unity much slower than the linear convergence rate for gradient descent. Thus, heavy-ball will converge much quicker than gradient descent locally, however it is not globally convergent for $\mathcal{S}_{\mu,\lambda}$ functions [3].

B. Nesterov's Accelerated Gradient Method

In 1983, Yuri Nesterov developed an accelerated gradient based algorithm that uses momentum similar to Polyak's Heavy-Ball method but is implemented slightly differently [9]. The update rule for this algorithm is below

$$x_{k+1} = y_k - \alpha\nabla f(y_k) \tag{19a}$$
$$y_{k+1} = (1+\beta)x_k - \beta x_{k-1} \tag{19b}$$

where $\alpha = 1/\lambda$ and $\beta = (\sqrt{\kappa}-1)/(\sqrt{\kappa}+1)$. This algorithm is globally convergent for $\mathcal{S}_{\mu,\lambda}$ functions. The convergence rate of Nesterov's accelerated gradient (NAG) method can be written as follows:

$$\|x^* - x_k\| \leq c\left(1 - \frac{1}{\sqrt{\kappa}}\right)^{0.5k} \tag{20}$$

This algorithm is a much better choice than gradient descent for functions in $\mathcal{S}_{\mu,\lambda}$.

C. Triple Momentum Method

In 2018, Bryan Van Scoy, Randy A. Freeman, and Kevin M. Lynch developed the triple momentum (TM) method for minimizing functions in $\mathcal{S}_{\mu,\lambda}$ [10]. This algorithm uses three momentum terms and is as follows:

$$\xi_{k+1} = (1+\beta)\xi_k - \beta\xi_{k-1} - \alpha\nabla f(y_k) \tag{21a}$$
$$y_k = (1+\gamma)\xi_k - \gamma\xi_{k-1} \tag{21b}$$
$$x_k = (1+\delta)\xi_k - \delta\xi_{k-1} \tag{21c}$$

If we define $\rho = 1 - 1/\sqrt{\kappa}$, the triple momentum parameters are

$$(\alpha, \beta, \gamma, \delta) = \left(\frac{1+\rho}{\lambda}, \frac{\rho^2}{2-\rho}, \frac{\rho^2}{(1+\rho)(2-\rho)}, \frac{\rho^2}{1-\rho^2}\right) \tag{22}$$

This algorithm, is globally convergent for $\mathcal{S}_{\mu,\lambda}$ functions, with rate given by

$$\|x^* - x_k\| \leq c\left(1 - \frac{1}{\sqrt{\kappa}}\right)^{k} \tag{23}$$

This algorithm is the fastest known globally convergent gradient-based algorithm for strongly convex, smooth functions. Notice that the linear convergence rate is double that of NAG.

D. Theoretical Lower Bound

In Nesterov's book, he also proves a lower bound on the convergence rate for any first-order method [11]. This lower bound states that any first-order algorithm must satisfy

$$\|x^* - x_k\| \geq c\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^{k} \tag{24}$$

This bound shows that it is impossible for any first-order algorithm to have a linear convergence rate ρ such that

$$\rho \leq \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right) \tag{25}$$

This theoretical lower bound is useful to keep in mind when comparing various first-order optimization algorithms.

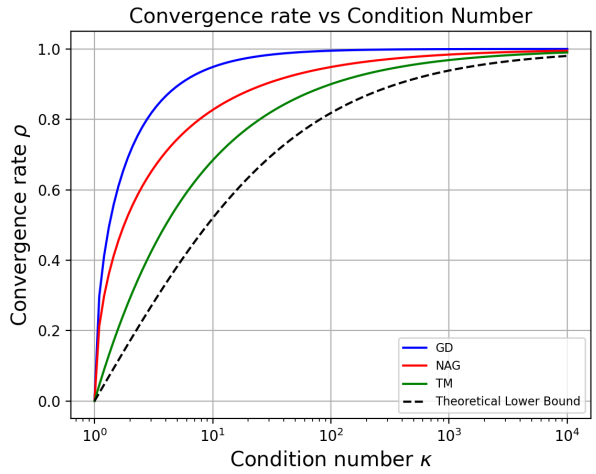

Fig. 7. Convergence Rates for different algorithms

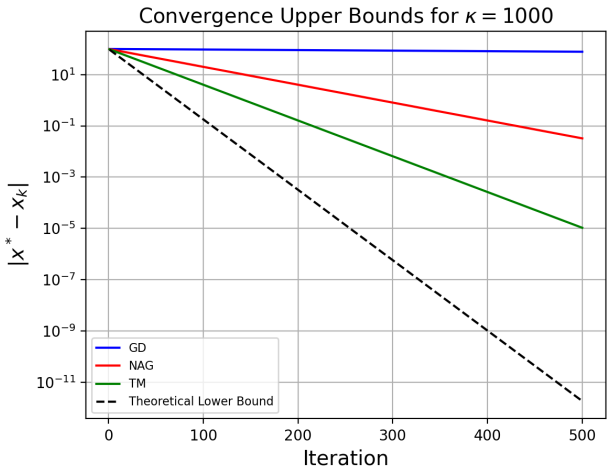

Fig. 8. Convergence Upper Bound for different algorithms with same κ

From figure 8 it appears that gradient descent is not making any progress, which shows us how critical it is to have accelerated algorithms such as Nesterov's Accelerated Gradient and Triple Momentum.

IV. Previous Work

The field of applying system and control theory to analyze and synthesize accelerated optimization algorithms is extremely new. The first paper to do this was by Laurent Lessard et al. in 2014 [3]. This paper introduces the idea of using robust control to analyze and synthesize simple gradient-based algorithms for smooth, strongly convex functions. In 2015, Nishihara et al. used these robust control ideas to provide a general convergence proof of ADMM with few

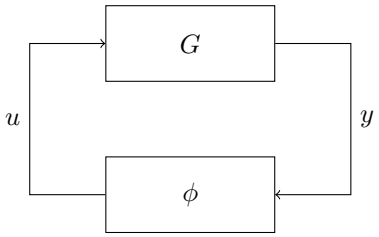

Fig. 9. Block diagram for algorithm analysis

assumptions on specific algorithm parameters [12]. In 2017, Cyrus et al. derived an accelerated first-order algorithm, they call the Robust Momentum Method, which has a single scalar parameter to trade off robustness to gradient noise and convergence rate. This algorithm was designed using ideas from control theory [13]. In 2017, Van Scoy et al. use the same IQC framework presented in [3] to derive and analyze the triple momentum algorithm. In 2018, Fazlyab et al. extended the work of using IQCs to analyze optimization algorithms for *non-strongly convex* functions and were able to certify sub-linear convergence of algorithms for this function class [14].

Since [3] was the paper that sparked this field, we will do a thorough review of its contributions in this section. The authors of [3] view optimization algorithms for $\mathcal{S}_{\mu,\lambda} : \mathbb{R}^d \to \mathbb{R}$ as a control system with a nonlinear block, which is the gradient of the function. If the function is a quadratic, then the gradient is linear in the decision variable, but for an arbitrary function in $\mathcal{S}_{\mu,\lambda}$, the gradient is nonlinear in the decision variable, but is a sector bounded non-linearity since we know the function is λ-smooth and μ-strongly convex.

We can think of the algorithm (G in Figure 9), as a controller for a nonlinear plant with bounded uncertainties (ϕ in Figure 9).

We can write this mathematically as follows:

$$\xi_{k+1} = A\xi_k + Bu_k \tag{26a}$$
$$y_k = C\xi_k + Du_k \tag{26b}$$
$$u_k = \nabla f(y_k) \tag{26c}$$

where the first two equations are G, and the last equation represents ϕ.

If this closed-loop system is stable, this means that the optimization algorithm converges to the optimal solution which is a fixed point of the dynamical system. This problem of assessing the stability of a forward linear path and a nonlinear feedback path is called the Lur'e problem [15], and was heavily studied in the mid to late 1900s.

All the algorithms we explored in Section III can be written in the form of (26) for different matrices A, B, C, and D. For example, gradient descent can be written as

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{c|c} I_d & -\eta I_d \\ \hline I_d & 0_d \end{array}\right] \tag{27}$$

We can write Polyak's heavy ball method as follows:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{ccc|c} (1+\beta)I_d & -\beta I_d & -\alpha I_d \\ I_d & 0_d & 0_d \\ \hline I_d & 0_d & 0_d \end{array}\right] \tag{28}$$

We can write Nesterov's accelerated gradient method as follows:

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{ccc|c} (1+\beta)I_d & -\beta I_d & -\alpha I_d \\ I_d & 0_d & 0_d \\ \hline (1+\beta)I_d & -\beta I_d & 0_d \end{array}\right] \tag{29}$$

The authors then apply techniques from robust control to analyze the stability of this closed-loop system. They first replace the nonlinear function ϕ with a quadratic constraint on the signals y and u, which is an integral quadratic constraint (IQC) [16]. The constraint on y and u comes from information about the gradient such as strong convexity and smoothness. Any property that can be concluded from this constrained system without ϕ will also hold for the original system.

Using these IQCs, the authors then derive a Linear Matrix Inequality (LMI) which, if feasible for a given ρ, certifies linear convergence of the algorithm under consideration with rate ρ as shown in (30). An LMI is a generalized inequality constraint for matrices with respect to the positive semi-definite cone [17]. It is important to note that the size of the LMI is independent of the size of the optimization problem. This is because A, B, and C in (27), (28), and (29) have repeated diagonal blocks and so do the IQCs. Thus, the LMI decouples and becomes dimension-independent.

$$\|\xi_k - \xi^*\| \leq \sqrt{\text{cond}(P)}\rho^k \|\xi_0 - \xi^*\| \tag{30}$$

where $P \succ 0$ is a matrix that shows up in the LMI, whose condition number, $\text{cond}(P)$, is within a constant factor of the condition number of the function being optimized.

The authors then go on to consider the case of analyzing the stability of accelerated algorithms with noisy gradients which requires a slightly different IQC.

Finally with all of this tooling in hand, the authors attack the problem of synthesizing an optimization algorithm. They first assume a template for their algorithm which is given by (31). Then for a fixed condition number κ and gradient noise strength, they generate a grid of tuples $(\alpha, \beta_1, \beta_2)$ and for each tuple they solve a sequence of feasibility problems with their LMI to find the $(\alpha, \beta_1, \beta_2)$ combination that results in the smallest feasible ρ.

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{ccc|c} 1+\beta_1 & -\beta_1 & -\alpha \\ 1 & 0 & 0 \\ \hline 1+\beta_1 & -\beta_2 & 0 \end{array}\right] \tag{31}$$

Since they assume a very specific form for their algorithm, they will not be able to discover new algorithms, just the right parameters $(\alpha, \beta_1, \beta_2)$ to maximize the convergence rate of algorithms in their proposed form.

One major result from the IQC framework is the triple momentum method by Van Scoy et al. [10]. In this paper, the authors mention that IQCs were used to motivate the design of the triple momentum algorithm. In the appendix of their paper, they show how they can use IQCs to analyze this algorithm. As as previously mentioned, this algorithm is the fastest known globally convergent gradient-based algorithm for smooth, strongly convex functions.

V. Paper Review

This section will review paper [4]: *Synthesis of accelerated gradient algorithms for optimization and saddle point problems using Lyapunov functions and LMIs*, discuss its contribution to the field of accelerated gradient methods, show some numerical results of the proposed algorithm, and provide a critique of the paper.

A. Summary

This paper presents a procedure to synthesize accelerated algorithms to minimize smooth, strongly convex functions and solve saddle problems. The class of objectives functions under consideration is defined by a generalized sector condition. By also considering saddle point problems, the synthesized algorithm can be used to perform equality constrained minimization if the objective function is the Lagrangian of the equality constrained problem.

The generalized sector condition that defines the class of objective functions $f : \mathbb{R}^d \mapsto \mathbb{R}$ can be written as

$$\frac{1}{2}\|y-x\|_M^2 \leq f(y) - f(x) - \nabla f(x)^\top (y-x) \leq \frac{1}{2}\|y-x\|_L^2 \tag{32}$$

where $M \in \mathbb{S}^d$, $L \in \mathbb{S}^d$ are symmetric matrices. $\|\cdot\|_M$ is the norm with respect to M. If f is twice differentiable we can write the sector condition as follows

$$M \preccurlyeq \nabla^2 f(x) \preccurlyeq L \tag{33}$$

We will denote this class of functions as $\mathcal{S}_{M,L}$. Notice that if $M = \mu I$ and $L = \lambda I$, this sector condition defines $\mathcal{S}_{\mu,\lambda}$ functions.

The paper aims to design gradient-based algorithms for $\mathcal{S}_{M,L}$ of the following form.

$$z_{k+1} = Az_k + B\nabla f(Cz_k) \tag{34}$$

where $z_k \in \mathbb{R}^n$, $f : \mathbb{R}^d \to \mathbb{R}$, and the matrices $A \in \mathbb{R}^{d\times d}$, $B \in \mathbb{R}^{n\times d}$, $C \in \mathbb{R}^{d\times n}$ are algorithm parameters to be designed. We can see that this is a condensed form of (26). Notice that for acceleration, we require that $d > n$, since our state z must include the momentum term. For NAG we can see that $z = (x, y)$ where x and y are defined in (19).

The synthesis procedure involves constructing a Lyapunov function and defining a Linear Matrix Inequality (LMI) based on (12) and solving a sequence of feasibility Semidefinite Programs (SDPs) to find the smallest possible ρ such that

$$\|z_k - z^*\| \leq c\rho^k \tag{35}$$

where $x^* = Cz^*$ is the minimizer of f or the saddle point of f.

B. Contribution

The analysis and synthesis of gradient-based algorithms for $\mathcal{S}_{\mu,\lambda}$ had already been introduced prior to the publication of [4] in papers such as [3] and [10].

The first contribution of this paper is introducing an analysis and synthesis framework for the more general class of functions, $\mathcal{S}_{M,L}$. This class of functions includes saddle-point problems which allows for the synthesis of equality constrained optimization problems. Secondly, the synthesis of the algorithm is not more conservative than its analysis framework. In some other works, some assumptions such as fixed IQC multipliers or quadratic Lyapunov functions are necessary to go from analysis to synthesis. In [4], the analysis to synthesis step is lossless meaning the analysis LMI is feasible if and only if the synthesis LMI is feasible. In the framework of this paper a very general algorithm template is presented as shown in (34). In papers such as [3], a very particular block diagonal structure is assumed such as in (31). This more general template allows the recovery of triple momentum which is not possible in the synthesis framework presented in [3].

A final smaller contribution of this paper is the ability to analyze and synthesize optimization algorithms with just Lyapunov theory. Papers such as [3] and [18] use Integral Quadratic Constraints which is more technical than just Lyapunov theory.

C. Proof Sketch

The formal problem statement of the paper is to find A, B, C, and the smallest possible ρ such that the algorithm given by (34) is globally convergent to the unique stationary point z^* of $f \in \mathcal{S}_{L,M}$ with linear convergence rate ρ.

The authors first reformulate the problem to an equivalent problem with $z^* = 0$. This is likely done to ease notational burden for the following proofs. Next, the authors propose a non-quadratic Lyapunov function, which contains information about the sector bounded gradients, and write a Linear Matrix Inequality (LMI) which is sufficient for the condition in (12) to hold. This is a key difference from earlier works such as [3] where a quadratic Lyapunov function is chosen but the LMI it generates is infeasible and an IQC is needed to capture the sector bounds on the gradient of the function.

This LMI is linear in the Hessian of the Lyapunov function P for fixed ρ, but becomes nonlinear if A, B, and C are optimization variables. This LMI is called the analysis LMI and for a given algorithm parameterized by A, B, C and a rate ρ, feasibility of the analysis LMI certifies that the algorithm converges with rate ρ.

The next step is to derive an LMI that is linear in A, B, and C such that we can pick a rate ρ and if the LMI is feasible with that ρ, we are given the algorithm parameters A, B, and C that achieves convergence with rate ρ. The authors then derive this LMI using Schur complements and other tools. This synthesis LMI is the key result of the paper.

D. Numerical Results

Here we will show some numerical results of the synthesized algorithm from [4]. Firstly, we synthesize algorithms for a range of condition numbers in order to compare the synthesized convergence rate to the fastest known rate (Triple Momentum). The results are in figure 10. We are able to replicate the results in the paper which showed that the synthesized algorithms for minimizing strongly convex functions achieve the same convergence rate as triple momentum.

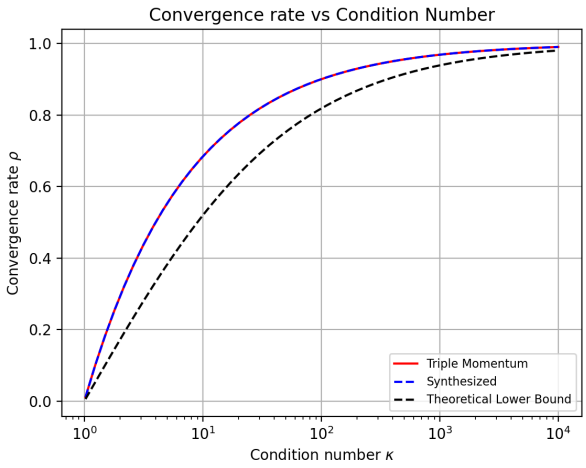

Fig. 10. Convergence Rate of Triple Momentum and Synthesized algorithm are identical

We then test the practical performance of the synthesized algorithm on a bivariate quadratic function with a condition number $\kappa = 1000$ and a random initial guess. We run gradient descent, Nesterov's accelerated gradient, triple momentum, and the synthesized algorithm on this function, and the distance to optimal along with the proven convergence bounds.

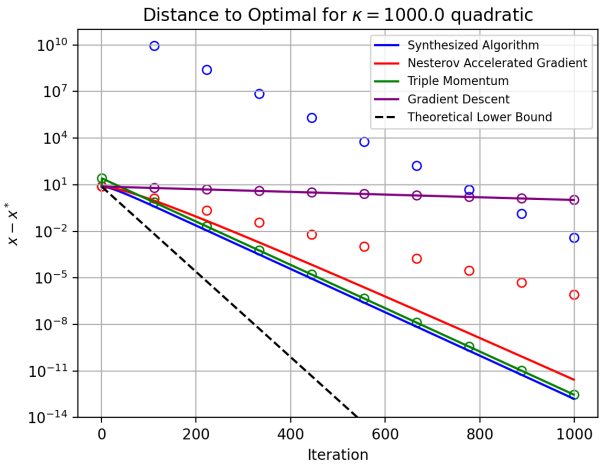

Fig. 11. Convergence of various algorithms; circles denote theoretical upper bounds for each algorithm

There are a number of conclusions to draw from this graph. Firstly, the proven upper bound for the synthesized algorithm is extremely conservative, since the constant in the upper

bound is proportional to the condition number of the derived Lyapunov function which is near singular. However, we see that the algorithm drastically outperforms its upper bound in this example. This is an unsubstantiated claim made in [4], but here is some evidence to back their claim.

We know that the upper bounds of triple momentum and the synthesized algorithm have the same convergence rate from figure 10, but we notice that at least for this example, the two algorithms have almost the same practical performance.

We also see that gradient descent is painfully slow as expected and its upper bound on convergence is tight for this example. A final interesting observation is that Nesterov's accelerated gradient drastically outperforms its upper bound for this example.

VI. Future Work

A. Critiques & Suggestions

The most major concern with the framework the paper introduces for synthesis is that the synthesis LMI scales with the dimension of the optimization problem. For large-scale applications such as machine learning, signal processing, or image processing we can have hundreds of thousands or even millions of optimization variables. Synthesizing algorithms for these large problems using the framework provided in the paper requires solving a series of SDPs of the same dimension which is intractable. This constraint limits the utility of the introduced framework to small problems. This is in contrast to the IQC framework introduced in [3] where the SDP to be solved is independent of the problem dimension since a more restrictive algorithm template is chosen.

A second concern is that the upper bound is proportional to the condition number of the Lyapunov function. Typically the constraint that the Lyapunov function must be positive definite is active and thus the condition number of the Lyapunov function can be arbitrarily large and result in bounds of form (14) with extremely large c.

Another concern is that numerical results of synthesized algorithms are not presented at all in this paper. There are results comparing the linear convergence rate ρ from the synthesized algorithm to other algorithms, but no actual test results are presented where the synthesized algorithm is used to minimize or find the saddle point of the function. This concern is compounded by the fact that the upper bound provided has a constant that is very large. Thus, the authors do not present evidence that the synthesized algorithms perform well in practice, contrary to their claim "The true transient behavior of designed optimization algorithms was usually much better than the bound".

A final criticism is that the authors have not shared their code to do algorithm synthesis, which makes it difficult for someone reading the paper to verify the claim that synthesized algorithms performs better than the provided bounds in practice.

B. Open Problems

Extending the synthesis of optimization algorithms for more general cases such as inequality constraints and non-smooth functions would be beneficial. One immediate challenge is that we cannot obtain linear convergence for these function and constraint classes, so we will not be able to satisfy the Lyapunov decrement condition. However, deriving a generic synthesis framework for more general function classes would be a major contribution.

This framework can also be used to synthesize preconditioners for optimization problems. A preconditioner is a change of primal and dual variables that result in faster convergence for a given algorithm. Currently, preconditioners are mostly designed via heuristics with little math to back up intuition, so preconditioner design using Lyapunov theory would be a great contribution to this field. There are some issues with directly applying this framework, namely, we would have to restrict ourselves to equality constrained optimization of strongly convex functions, but this is a promising direction for future research.

Similar to [3], this framework should also be extended to synthesize algorithms for saddle point problems in the presence of noisy gradients.

VII. Conclusion

Although the dynamical system perspective of optimization algorithms is fairly new, it has delivered some very promising results such as a more general analysis framework for optimization algorithms based on dissapativity theory and the derivation of the fastest globally convergent algorithm for the minimization of strongly convex functions [6] [10]. There remains a lot of interesting extensions of this framework to design new preconditioners or new algorithms.

References

[1] Lars Blackmore. Autonomous precision landing of space rockets. In *in Frontiers of Engineering: Reports on Leading-Edge Engineering from the 2016 Symposium*, volume 46, pages 15–20, 2016.

[2] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77, March 1952.

[3] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM J. Optim.*, 26(1):57–95, January 2016.

[4] Dennis Gramlich, Christian Ebenbauer, and Carsten W Scherer. Synthesis of accelerated gradient algorithms for optimization and saddle point problems using lyapunov functions and LMIs. *Syst. Control Lett.*, 165:105271, July 2022.

[5] Alexander Mikhailovich Lyapunov. The general problem of the stability of motion. 1994.

[6] Laurent Lessard. The analysis of optimization algorithms: A dissipativity approach. *IEEE Control Syst.*, 42(3):58–72, June 2022.

[7] Augustin-Louis Cauchy. *Méthode générale pour la résolution des systèmes d'équations simultanées*. 1847.

[8] Boris Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 1964.

[9] Y E Nesterov. A method of solving a convex programming problem with convergence rate. *Dokl. Akad. Nauk*, 1983.

[10] Bryan Van Scoy, Randy A Freeman, and Kevin M Lynch. The fastest known globally convergent first-order method for minimizing strongly convex functions. *IEEE Control Syst. Lett.*, 2(1):49–54, January 2018.

[11] Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition, 2014.

[12] Robert Nishihara, Laurent Lessard, Benjamin Recht, Andrew Packard, and Michael I Jordan. A general analysis of the convergence of ADMM. February 2015.

[13] Saman Cyrus, B Hu, Bryan Van Scoy, and Laurent Lessard. A robust accelerated optimization algorithm for strongly convex functions. 2017.

[14] Mahyar Fazlyab, Alejandro Ribeiro, Manfred Morari, and Victor M Preciado. Analysis of optimization algorithms via integral quadratic constraints: Nonstrongly convex problems. *SIAM J. Optim.*, 28(3):2654–2689, January 2018.

[15] A. I. Lur'e and V. N. Postnikov. On the theory of stability of control systems. *Applied Math Mechanics*, 1944.

[16] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Transactions on Automatic Control*, 42(6):819–830, 1997.

[17] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[18] Laurent Lessard and Peter Seiler. Direct synthesis of iterative algorithms with bounds on achievable worst-case convergence rate, 2020.