Fast Gradient-Based Methods with Exponential Rate: A Hybrid Control Framework

Arman Sharifi Kolarijani [1] **Peyman Mohajerin Esfahani** [1] **Tamás Keciczky** [1]

Abstract

Ordinary differential equations, and in general a dynamical system viewpoint, have seen a resurgence of interest in developing fast optimization methods, mainly thanks to the availability of well-established analysis tools. In this study, we pursue a similar objective and propose a class of hybrid control systems that adopts a 2nd-order differential equation as its continuous flow. A distinctive feature of the proposed differential equation in comparison with the existing literature is a state-dependent, time-invariant damping term that acts as a feedback control input. Given a user-defined scalar α, it is shown that the proposed control input steers the state trajectories to the global optimizer of a desired objective function with a guaranteed rate of convergence $\mathcal{O}(e^{-\alpha t})$. Our framework requires that the objective function satisfies the so called Polyak–Łojasiewicz inequality. Furthermore, a discretization method is introduced such that the resulting discrete dynamical system possesses an exponential rate of convergence.

1. Introduction

The low computational and memory complexities of gradient-based optimization algorithms have made them an attractive alternative in many applications such as support vector machines (Allen-Zhu, 2016), signal and image processing (Becker et al., 2011), and networked-constrained optimization (Ghadimi et al., 2013) among others. Hence, extensive efforts have been made recently in order to bring more insight into these algorithms' properties.

One research direction that has been recently revitalized is the application of ordinary differential equations (ODEs) to the analysis and design of optimization algorithms. Consider an iterative algorithm that can be viewed as a discrete dynamical system, with the scalar s as its step size. As s decreases, one can observe that the iterative algorithm in fact recovers a differential equation, e.g., in the case of gradient descent method applied to an unconstrained optimization problem $\min_{X\in\mathbb{R}^n} \ f(X)$, one can inspect that

$$X^{k+1} = X^k - s\nabla f(X^k) \ \rightsquigarrow \ \dot{X}(t) = -\nabla f\big(X(t)\big)$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is a smooth function, X is the decision variable, $k\in\mathbb{Z}_{\geq 0}$ is the iteration index, and $t\in\mathbb{R}_{\geq 0}$ is the time. The main motivation behind this line of research has to do with well-established analysis tools in dynamical systems described by differential equations.

The slow rate of convergence of the gradient descent algorithm ($\mathcal{O}(\frac{1}{t})$ in continuous and $\mathcal{O}(\frac{1}{k})$ in discrete time), limits its application in large-scale problems. In order to address this shortcoming, many researchers resort to the following class of 2nd-order ODEs, which is also the focus of this study:

$$\ddot{X}(t) + \gamma(t)\dot{X}(t) + \nabla f\big(X(t)\big) = 0. \tag{1}$$

Increasing the order of the system dynamics interestingly helps improve the convergence rate of the corresponding algorithms to $\mathcal{O}(\frac{1}{k^2})$ in the discrete-time domain or to $\mathcal{O}(\frac{1}{t^2})$ in the continuous-time domain. Such methods are called *momentum*, *accelerated*, or *fast* gradient-based iterative algorithms in the literature. The time-dependent function $\gamma:\mathbb{R}_{\geq 0}\to\mathbb{R}_{>0}$ is a *damping* or a *viscosity* term, which has been also referred to as the *asymptotically vanishing viscosity* since $\lim_{t\to\infty}\ \gamma(t) = 0$ (Cabot, 2004).

Chronological developments of fast algorithms: It is believed that the application of (1) to speed-up optimization algorithms is originated from (Polyak, 1964) in which Polyak was inspired by a physical point of view (i.e., a heavy-ball moving in a potential field). Later on, Nesterov introduced his celebrated accelerated gradient method in (Nesterov, 1983) using the notion of "estimate sequences" and guaranteeing convergence rate of $\mathcal{O}(\frac{1}{k^2})$. Despite several extensions of Nesterov's method (Nesterov, 2004; 2005; 2013), the approach has not yet been fully understood. In this regard, many have tried to study the intrinsic properties of

[1]Delft Center for Systems and Control, Delft University of Technology, The Netherlands. Correspondence to: Arman Sharifi Kolarijani <a.sharifikolarijani@tudelft.nl>.

Proceedings of the 35^{th} International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018.

Nestrov's method such as (Drusvyatskiy et al., 2016; Bubeck et al., 2015; Drori & Teboulle, 2014; Lessard et al., 2016). Recently, the authors in (Su et al., 2014) and in details (Su et al., 2016) surprisingly discovered that Nesterov's method recovers (1) in its continuous limit, with the time-varying damping term $\gamma(t) = \frac{3}{t}$.

A dynamical systems perspective: Based on the observation suggested by (Su et al., 2014), several novel fast algorithms have been developed. Inspired by the mirror descent approach (Nemirovskii et al., 1983), the ODE (1) has been extended to non-Euclidean settings and to higher order methods using the Bregman Lagrangian in (Wibisono et al., 2016). Followed by (Wibisono et al., 2016), a "rate-matching" Lyapunov function is proposed in (Wilson et al., 2016) with its monotonicity property established for both continuous and discrete dynamics. Recently, the authors in (Lessard et al., 2016) make use of an interesting semidefinite programming framework developed by (Drori & Teboulle, 2014) and use tools from robust control theory to analyze the convergence rate of optimization algorithms. More specifically, the authors exploit the concept of integral quadratic constraints (IQCs) (Megretski & Rantzer, 1997) to design iterative algorithms under the strong convexity assumption. Later, the authors in (Fazlyab et al., 2017) extend the results of IQC-based approaches to quasiconvex functions. (Hu & Lessard, 2017) uses dissipativity theory (Willems, 1972) along with the IQC-based analysis to construct Lyapunov functions enabling rate analyses.

Restarting schemes: A characteristic feature of fast methods is the non-monotonicity in the suboptimality measure $f - f^*$, where f^* refers to the optimal value of function f. The reason behind such an undesirable behavior can be intuitively explained in two ways: (i) a momentum based argument indicating as the algorithm evolves, the algorithm's momentum gradually increases to a level that it causes an oscillatory behavior (O'Donoghue & Candès, 2015); (ii) an acceleration-based argument indicating that the asymptotically vanishing damping term becomes so small that the algorithm's behavior drifts from an over-damped regime into an under-damped regime with an oscillatory behavior (Su et al., 2016). To prevent such an undesirable behavior in fast methods, an optimal fixed restart interval is determined in terms of the so-called condition number of function f such that the momentum term is restarted to a certain value, see e.g., (Nesterov, 2004; Nemirovski, 2005; Gu et al., 2013; Lan & Monteiro, 2013; Nesterov, 2013). It is worth mentioning that (O'Donoghue & Candès, 2015) proposes two heuristic adaptive restart schemes. It is numerically observed that such restart rules practically improve the convergence behavior of a fast algorithm.

Regularity for exponential convergence: Generally speaking, exponential convergence rate and the corresponding regularity requirements of the function f are two crucial metrics in fast methods. In what follows, we discuss about these metrics for three popular fast methods in the literature. When the objective functions are strongly convex with a constant σ_f and their gradient is Lipschitz with a constant L_f, (Su et al., 2016) proposes the "speed restarting" scheme

$$\sup\Big\{t > 0 : \ \forall \tau \in (0,t), \frac{d\|\dot{X}(\tau)\|^2}{d\tau} > 0\Big\},$$

to achieve the convergence rate of:

$$f\big(X(t)\big) - f^* \leq d_1 e^{-d_2 t}\|X(0) - X^*\|^2.$$

The positive scalars d_1 and d_2 depend on the constants σ_f and L_f. Assuming the convexity of the function f with a certain choice of parameters in their "ideal scaling" condition, (Wibisono et al., 2016) guarantees the convergence rate of $\mathcal{O}(e^{-ct})$ for some positive scalar c. However, in this general case, their approach requires to compute a matrix inversion in the Euler-Lagrange equation in the form of:

$$\ddot{X}(t) + c\dot{X}(t) \\ + c^2 e^{ct}\Big(\nabla^2 h\big(X(t) + \frac{1}{c}\dot{X}(t)\big)\Big)^{-1}\nabla f\big(X(t)\big) = 0,$$

where the function h is a distance generating function. Under uniform convexity assumption with a constant ν_f, it is further shown that

$$f\big(X(t)\big) - f^* \leq \Big(f\big(X(0)\big) - f^*\Big)e^{-\nu_f \frac{1}{p-1}t},$$

where $p-1$ is the order of smoothness of f. The authors in (Wilson et al., 2016) introduce the Lyapunov function

$$\mathcal{E}(t) = e^{\beta(t)}\left(f\big(X(t)\big) - f^* + \frac{\sigma_f}{2}\|X^* - Z(t)\|^2\right),$$

to guarantee the rate of convergence

$$\mathcal{E}(t) \leq \mathcal{E}(0)e^{-\int \dot{\beta}(s)ds},$$

where $Z(t) = X(t) + \frac{1}{\dot{\beta}(t)}\dot{X}$, and $\beta(t)$ is a user-defined function.

Our contribution: state-dependent damping coefficient It is evident that the damping term $\gamma(t)$ is unaware of how the dynamics (1) evolves. As a result, this term is also unaware of the non-monotonicity of the objective function along the trajectories of the dynamics (1). As such, several fast algorithms adopt restarting schemes to improve the theoretical and/or practical convergence rate. To some extent, the reason behind considering a time-dependent term γ may be due to the fact that the discretization process of the continuous-time dynamics (1) becomes less cumbersome. The above discussion strongly suggests that the term γ may

be treated as a feedback (or a control input) and thus allowing tools from control theory to synthesize γ, possibly based on the performance criterion a designer seeks for.

In this article we adopt this mindset and consider the controlled dynamics

$$\ddot{X}(t) + u\big(X(t), \dot{X}(t)\big)\, \dot{X}(t) + \nabla f(X(t)) = 0,$$

where the feedback control input $u\big(X(t), \dot{X}(t)\big)$ replaces the time-dependent damping coefficient $\gamma(t)$ in (1). Given a positive scalar α, we seek to achieve an exponential rate of convergence $\mathcal{O}(e^{-\alpha t})$ for an unconstrained, smooth optimization problem in the suboptimality measure $f\big(X(t)\big) - f^*$. Inspired by restarting techniques, in our proposed framework we extend the class of dynamics to hybrid control systems (see Definition 2.1 for further details) in which the above 2nd-order differential equation represents its *continuous flow*. To achieve the convergence rate of $\mathcal{O}(e^{-\alpha t})$, we propose the state-dependent feedback law

$$u_\alpha\big(X(t), \dot{X}(t)\big) := \alpha + \frac{\|\nabla f(X(t))\|^2 - \langle \nabla^2 f\big(X(t)\big)\dot{X}(t), \dot{X}(t)\rangle}{\langle \nabla f\big(X(t)\big), -\dot{X}(t)\rangle}.$$

We next suggest an admissible control input range $[u_{\min}, u_{\max}]$ that determines the *flow set* of the hybrid system. Given the model parameters α, $u_{\min}$, and $u_{\max}$, the *jump map* of the hybrid control system is defined through the mapping $\big(X^\top, -\beta\nabla^\top f(X)\big)^\top$ ensuring that the jump map's range is a subset of the flow set . Notice that the velocity restart scheme becomes $\dot{X} = -\beta\nabla f(X)$. We now summarize the contributions of our proposed approach in the context of continuous fast methods:

- We introduce a system-theoretic framework to design the damping term γ as a parametric state-dependent feedback control, as opposed to the customary choice of being time-dependent, whose parameter ensures the desired convergence rate (Theorem 3.1);
- Our framework requires that the objective function f satisfies the Polyak–Łojasiewicz (PL) inequality (Assumption (A2)). The PL inequality is in fact a weaker regularity assumption compared to the ones mentioned in the literature (e.g., strong convexity);
- We further provide a discretization method, as well as a discretization step size, leading to a discrete-time dynamical system (i.e., an optimization algorithm) that enjoys an exponential rate of convergence (Theorem 3.7).

The remainder of this paper is organized as follows. In Section 2, the mathematical notions are represented. The main results of the paper are introduced in Section 3. Section 4 contains the proofs of the main results. In Section 5, a numerical example is given.

Notations: The sets $\mathbb{R}^n$ and $\mathbb{R}^{m\times n}$ denote the n-dimensional Euclidean space and the space of $m \times n$ dimensional matrices with real entries, respectively. For a matrix $M \in \mathbb{R}^{m\times n}$, $M^\top$ is the transpose of M, $M \succ 0$ ($\prec 0$) refers to M is positive (negative) definite, $M \succeq 0$ ($\preceq 0$) refers to M is positive (negative) semi-definite, and $\lambda_{\max}(M)$ denotes the maximum eigenvalue of M. The $n \times n$ identity matrix is denoted by I_n. For a vector $v \in \mathbb{R}^n$ and $i \in \{1, \cdots, n\}$, v_i represents the i-th entry of v and $\|v\| := \sqrt{\Sigma_{i=1}^n v_i^2}$ is the Euclidean 2-norm of v. For two vectors $x, y \in \mathbb{R}^n$, $\langle x, y\rangle := x^\top y$ denotes the Euclidean inner product. For a matrix M, $\|M\| := \sqrt{\lambda_{\max}(A^\top A)}$ is the induced 2-norm. Given the set $S \subseteq \mathbb{R}^n$, ∂S and $\text{int}(S)$ represent the boundary and the interior of S, respectively.

2. Preliminaries

In this section, we recall the notion of hybrid control systems and then, formally present the problem statement. The following representation of a hybrid control system is adapted from (Goebel et al., 2012) that is sufficient in the context of this paper.

Definition 2.1 (Hybrid control system)**.** *A time-invariant hybrid control system $\mathcal{H}$ comprises a controlled ODE and a jump (or a reset) rule introduced as:*

$$\left\{ \begin{array}{lcll} \dot{x} & = & F\big(x, u(x)\big), & x \in \mathcal{C} \\ x^+ & = & G(x), & \textit{otherwise}, \end{array} \right. \qquad (\mathcal{H})$$

where x^+ is the state of the hybrid system after a jump, the function $u : \mathbb{R}^n \to \mathbb{R}^m$ denotes a feedback signal, the function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is the flow map, the set $\mathcal{C} \subseteq \mathbb{R}^n \times \mathbb{R}^m$ is the flow set, and the function $G : \partial C \to \text{int}(\mathcal{C})$ represents the jump map.

Throughout this study we assume the requirements under which the hybrid control system ($\mathcal{H}$) admits a well-defined solution, see Chapters 2 and 6 of (Goebel et al., 2012) for further details in this regard.

Consider the following class of unconstrained optimization problems:

$$f^* := \min_{X\in\mathbb{R}^n} f(X), \qquad (2)$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is an objective function. We proceed with the main problem in this article:

Problem 2.2. *Consider the unconstrained optimization problem (2) where the objective function f is twice differentiable. Given a positive scalar α, design a fast gradient-based method in the form of a hybrid control system ($\mathcal{H}$) with the α-exponential convergence rate, i.e. for any initial*

condition $X(0)$ and any $t \geq 0$ we have

$$f\big(X(t)\big) - f^* \leq e^{-\alpha t}\Big(f\big(X(0)\big) - f^*\Big),$$

where $\{X(t)\}_{t\geq 0}$ denotes the solution trajectory of the system ($\mathcal{H}$).

Assumption 2.3 (Regularity assumptions)**.** *We stipulate that the objective function $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable and fulfills the following*

- *The Hessian of function f, denoted by $\nabla^2 f(x)$, is uniformly bounded, i.e.,*

$$-\ell_f I_n \preceq \nabla^2 f(x) \preceq L_f I_n, \tag{A1}$$

  where ℓ_f and L_f are non-negative constants.

- *The function f satisfies the Polyak-Łojasiewicz inequality with a positive constant μ_f, i.e., for every x in $\mathbb{R}^n$ the following inequality holds:*

$$\frac{1}{2}\|\nabla f(x)\|^2 \geq \mu_f\big(f(x) - f^*\big), \tag{A2}$$

  where f^* is the minimum value of f on $\mathbb{R}^n$.

Remark 2.4 (Lipschitz gradient)**.** *Since the function f is twice differentiable, Assumption (A1) implies that the function f has also Lipschitz gradient with a positive constant L_f, i.e., for every x, y in $\mathbb{R}^n$ we have*

$$\|\nabla f(x) - \nabla f(y)\| \leq L_f\|x - y\|. \tag{3}$$

In what follows, we state interesting facts regarding the set of functions that satisfy (A2).

Remark 2.5 (PL functions and invexity)**.** *The PL inequality in general does not imply the convexity of a function but rather the invexity of it. The notion of invexity was first introduced by (Hanson, 1981). The PL inequality (A2) implies that the suboptimality measure $f - f^*$ grows at most as a quadratic function of ∇f.*

Remark 2.6 (Non-uniqueness of stationary points)**.** *While the PL inequality does not require the uniqueness of the stationary points of a function (i.e., $\{x : \nabla f(x) = 0\}$), it ensures that all stationary points of the function f are global minimizers (Craven & Glover, 1985).*

We close our preliminary section with a couple of popular examples borrowed from (Karimi et al., 2016).

Example 1 (PL functions). The composition of a strongly convex function and an exponential function satisfies the PL inequality. This class includes a number of important problems such as least squares, i.e., $f(x) = \|Ax - b\|$ (obviously, strongly convex functions also satisfy the PL inequality). Any strictly convex function over a compact set satisfies the PL inequality. As such, the log-loss objective function in logistic regression, i.e., $f(x) = \Sigma_{i=1}^n \log\big(1 + \exp(b_i a_i^\top x)\big)$, locally satisfies the PL inequality.

3. Main Results

The main results of this paper are presented in this section along with several remarks highlighting their implications. The underlying idea and the corresponding technical proofs are provided in Section 4. In what follows we introduce the notation $x := (x_1, x_2)$ such that the variables x_1 and x_2 represent the system trajectories X and $\dot{X}$, respectively.

In the first step we provide a type of parameterization for the hybrid system ($\mathcal{H}$). Given a positive scalar α, the proposed parameterization denoted by $u_\alpha(x)$ enables achieving the rate of convergence $\mathcal{O}(e^{-\alpha t})$ in the suboptimality measure $f\big(X(t)\big) - f^*$. Motivated by the dynamics of fast gradient methods (Su et al., 2016), we start with a 2nd-order ODE as the continuous evolution (or the flow map) $F : \mathbb{R}^{2n} \times \mathbb{R} \to \mathbb{R}^{2n}$ defined as

$$F\big(x, u_\alpha(x)\big) = \begin{pmatrix} x_2 \\ -\nabla f(x_1) \end{pmatrix} + \begin{pmatrix} 0 \\ -x_2 \end{pmatrix} u_\alpha(x). \tag{4a}$$

The feedback law $u_\alpha : \mathbb{R}^{2n} \to \mathbb{R}$ is given by

$$u_\alpha(x) = \alpha + \frac{\|\nabla f(x_1)\|^2 - \langle \nabla^2 f(x_1)x_2, x_2\rangle}{\langle \nabla f(x_1), -x_2\rangle}. \tag{4b}$$

The important feature of the proposed control structure is to ensure achieving an α-exponential convergence rate, see Subsection 4.1 for more details. In the next step, we consider an admissible interval $[u_{\min}\ u_{\max}]$ to characterize a candidate flow set $\mathcal{C} \subset \mathbb{R}^{2n}$, i.e.,

$$\mathcal{C} = \big\{x \in \mathbb{R}^{2n} :\ u_\alpha(x) \in [u_{\min}, u_{\max}]\big\}, \tag{4c}$$

where $u_{\min}, u_{\max}$ represent the range of acceptable control values. Notice that the flow set $\mathcal{C}$ is the domain in which the hybrid system ($\mathcal{H}$) can evolve continuously. Finally, we introduce the jump map $G : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ parameterized by a constant β

$$G(x) = \begin{pmatrix} x_1 \\ -\beta\nabla f(x_1) \end{pmatrix}. \tag{4d}$$

The parameter β ensures that the range space of the jump map G is a strict subset of $\text{int}(\mathcal{C})$. By construction, one can inspect that any neighborhood of the the optimizer x_1^* has a non-empty intersection with the flow set $\mathcal{C}$. That is, there always exist paths in the set $\mathcal{C}$ that allow the continuous evolution of the Hybrid system to approach arbitrarily close to the optimizer.

The first result of this section introduces a mechanism to compute the hybrid system's parameters $u_{\min}$, $u_{\max}$, and β in (4c) and (4d) to achieve the desired exponential convergence rate $\mathcal{O}(e^{-\alpha t})$.

Theorem 3.1 (Continuous-time hybrid dynamics)**.** *Consider a positive scalar α and a smooth function $f : \mathbb{R}^n \to \mathbb{R}$*

satisfying Assumption 2.3. Then, the solution trajectory of the continuous-time hybrid control system ($\mathcal{H}$) with the respective parameters (4) and starting from any initial condition $x_1(0)$ satisfies

$$f\big(x_1(t)\big) - f^* \leq e^{-\alpha t}\Big(f\big(x_1(0)\big) - f^*\Big), \quad \forall t \geq 0, \quad (5)$$

if the scalars $u_{\min}$, $u_{\max}$, and β are chosen such that

$$u_{\min} < \alpha + \beta^{-1} - L_f\beta, \quad (6a)$$
$$u_{\max} > \alpha + \beta^{-1} + \ell_f\beta, \quad (6b)$$
$$\alpha \leq 2\mu_f\beta. \quad (6c)$$

Remark 3.2 (Weaker regularity than strong convexity)**.** *The PL inequality is a weaker requirement than the strong convexity, which is often assumed in similar contexts (Su et al., 2016; Wibisono et al., 2016; Wilson et al., 2016). It is worth noting that such a condition has also been used in the context of 1st-order algorithms (Karimi et al., 2016).*

Remark 3.3 (Hybrid embedding of restarting)**.** *The hybrid framework intrinsically captures a restarting scheme through the jump map. The scheme is a weighted gradient where the weight factor β is essentially characterized by the given data α, μ_f, ℓ_f, and L_f. One may inspect that the constant β can be in fact introduced as a state-dependent weight factor to potentially improve the performance. Nonetheless, for the sake of simplicity of exposition, we do not pursue this level of generality in this paper.*

Remark 3.4 (Fundamental limits on control input)**.** *In order to guarantee the rate of convergence of $\mathcal{O}\big(e^{-\alpha t}\big)$, Theorem 3.1 asserts the following theoretical limits on $u_{\min}$ and $u_{\max}$: (i) The upper-bound on the admissible input interval $u_{\max}$ is required to be larger than α, and (ii) the lower-bound on the admissible input interval $u_{\min}$ has to be* negative *if the geometrical property $\alpha > \big(\frac{2\mu_f}{\sqrt{\max\{L_f - 2\mu_f, 0\}}}\big)$ holds based on the given α. As a result, it is required to* inject energy *to the dynamical system through* negative *damping in order to achieve an exponential rate of convergence.*

Remark 3.5 (Connection to time dilation)**.** *The authors in (Wibisono et al., 2016) show that in the continuous-time domain an arbitrary rate of convergence can be achieved through a change of variable on the time variable, to which they refer as "*time dilation*". Notice that such a technique may yields a time-varying dynamical system. Theorem 3.1 indeed addresses the exact same objective in a more explicit fashion through the parameter α, representing the desired convergence rate, in the control law of the damping term as defined in* (4b).

Remark 3.6 (2nd-order information)**.** *Although our proposed framework requires 2nd-order information, i.e., the Hessian $\nabla^2 f$, this requirement only appears in a mild form as an evaluation in the same spirit as the modified Newton step proposed in (Nesterov & Polyak, 2006). Furthermore, we emphasize that our results still hold true if one replaces $\nabla^2 f(x_1)$ with its upper-bound $L_f I_n$ following essentially the same analysis. For further details we refer the reader to the proof of Theorem 3.1.*

In the following, we use the *forward-Euler* method to discretize the continuous-time hybrid control system ($\mathcal{H}$). This technique leads to an iterative optimization algorithm that enjoys an exponential rate of convergence in $f(x_1^k) - f^*$ where k is the iteration index. Define the parameter s as the step size of the discretization. Consider

$$\mathcal{H}_d := \begin{cases} x^{k+1} = F_d\big(x^k, u_{\alpha,d}(x^k)\big), & x^k \in \mathcal{C}_d \\ x^{k+1} = G_d(x^k), & \text{otherwise}, \end{cases} \quad (7)$$

where the discrete flow map $F_d : \mathbb{R}^{2n} \times \mathbb{R} \to \mathbb{R}^{2n}$ is given by

$$F_d\big(x^k, u_{\alpha,d}(x^k)\big) = \begin{pmatrix} x_1^k + s\, x_2^k \\ \big(1 - s\, u(x^k)\big)\, x_2^k - s\nabla f(x_1^k) \end{pmatrix}, \quad (8a)$$

the discrete state-dependent feedback $u_{\alpha,d} : \mathbb{R}^{2n} \to \mathbb{R}$ is given by

$$u_{\alpha,d}(x^k) = \alpha + \frac{\|\nabla f(x_1^k)\|^2 - \langle \nabla^2 f(x_1^k) x_2^k, x_2^k\rangle}{\langle \nabla f(x_1^k), -x_2^k\rangle}, \quad (8b)$$

the discrete flow set $\mathcal{C}_d \subset \mathbb{R}^{2n}$ is

$$\mathcal{C}_d := \big\{(x_1^k, x_2^k) \in \mathbb{R}^{2n} : c_1\|x_2^k\|^2 \leq \|\nabla f(x_1^k)\|^2 \leq c_2\langle \nabla f(x_1^k), -x_2^k\rangle\big\}, \quad (8c)$$

and the discrete jump map $G_d : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ is

$$G_d(x^{k+1}) = \begin{pmatrix} x_1^k \\ -\beta\nabla f(x_1^k) \end{pmatrix}. \quad (8d)$$

Due to technical difficulties mainly caused by the discretization of the control input $u_\alpha(x)$, we need to appropriately modify the definition of the discrete-time flow set $\mathcal{C}_d$ in comparison with the continuous-time flow set $\mathcal{C}$ so that the stability of the process can be ensured. Based on the discrete dynamics (7) with the parameterization (8), the upper-bound on the step size s is determined such that an exponential rate of convergence is guaranteed in Theorem 3.7.

Theorem 3.7 (Stable discretization)**.** *Consider a smooth function $f : \mathbb{R}^n \to \mathbb{R}$ satisfying Assumption 2.3. The solution trajectory of the discrete-time hybrid control system (7) with the respective parameters (8) and starting from any initial condition x_1^0 satisfies*

$$f(x_1^{k+1}) - f^* \leq \lambda(s, c_1, c_2, \beta)\big(f(x_1^k) - f^*\big), \quad (9)$$

Algorithm 1 Sate Dependent Scheme

Input: data x_1^0, ℓ_f, L_f, μ_f, $\alpha \in \mathbb{R}^+$, $k_{\max} \in \mathbb{N}^+$
Set: $\sqrt{c_1} = c_2 = \beta^{-1} = L_f s$, $x_2^0 = -\beta \nabla f(x_1^0)$
$x^0 = (x_1^0, x_2^0)$
for $k = 1$ **to** $k_{\max}$ **do**
  if $c_1\|x_2^k\|^2 \leq \|\nabla f(x_1^k)\|^2 \leq c_2\langle \nabla f(x_1^k), -x_2^k\rangle$ **then**
    $x^{k+1} \leftarrow F_d(x^k)$
  else
    $x^{k+1} \leftarrow G_d(x^k)$
  end if
end for

with $\lambda(s, c_1, c_2, \beta) \in (0,1)$ given by

$$\lambda(s, c_1, c_2, \beta) := 1 + 2\mu_f\big(-\frac{s}{c_2} + \frac{L_f}{2c_1}s^2\big) \tag{10}$$

if the set of parameters s, c_1 ,c_2, and β satisfies the following:

$$\sqrt{c_1} \leq c_2, \tag{11a}$$
$$\beta^2 c_1 \leq 1 \leq \beta c_2, \tag{11b}$$
$$c_2 L_f s < 2c_1. \tag{11c}$$

Remark 3.8 (Naive discretization)**.** *We stress that our proposed discretization effectively exploits only the dynamics of x_1. Namely, the dynamics of x_2 as well as the control law u_α play no active role in our proposed method, see Subsection 4.2 for more details. Thus, a more in-depth analysis is due in this regard.*

Corollary 3.9 (Optimal guaranteed rate)**.** *The optimal convergence rate guaranteed by Theorem 3.7 for the discrete-time dynamics is $\lambda^* := \big(1 - \frac{\mu_f}{L_f}\big)$ and*

$$\sqrt{c_1^*} = c_2^* = \frac{1}{\beta^*} = L_f s^*.$$

In Algorithm 1, we provide the pseudocode to implement Corollary 3.9 using the discrete-time dynamics (7) with the respective parameters (8).

4. Underlying idea and technical proofs

4.1. Proof of Theorem 3.1

We start with explanation on why the chosen structure for $u_\alpha(x)$ guarantees the desired convergence rate α. Let us define the set $\mathcal{E}_\alpha := \Big\{x \in \mathbb{R}^{2n} : \alpha\big(f(x_1) - f^*\big) < \langle \nabla f(x_1), -x_2\rangle\Big\}$. In the first step, we argue that the objective function f decreases at the rate α (i.e., (5)) along any solution trajectory of the dynamical system (4a) that is contained in the set $\mathcal{E}_\alpha$. To see this, observe that if $\big(x_1(t), x_2(t)\big) \in \mathcal{E}_\alpha$, we then have

$$\begin{aligned}\frac{d}{dt}\Big(f\big(x_1(t)\big) - f^*\Big) &= \big\langle \nabla f\big(x_1(t)\big), x_2(t)\big\rangle \\ &\leq -\alpha\big(f(x_1) - f^*\big).\end{aligned}$$

The direct application of Gronwall's inequality, see Lemma A.1 in (Khalil, 2002), to the above inequality yields the desired convergence claim (5). In the light of the above observation, it suffices to ensure that the solution trajectory does not leave the set $\mathcal{E}_\alpha$. Let us define the quantity

$$\sigma(t) := \big\langle \nabla f\big(x_1(t)\big), x_2(t)\big\rangle + \alpha\Big(f\big(x_1(t)\big) - f^*\Big).$$

By definition, if $\sigma(t) < 0$, it is then readily guaranteed that $\big(x_1(t), x_2(t)\big) \in \mathcal{E}_\alpha$. By virtue of this implication, if $\dot\sigma(t) \leq 0$ along the solution trajectory of (4a), we ensure that the value of $\sigma(t)$ does not increase, and as such

$$\big(x_1(t), x_2(t)\big) \in \mathcal{E}_\alpha,\ \forall t \geq 0 \iff \big(x_1(0), x_2(0)\big) \in \mathcal{E}_\alpha.$$

To ensure non-positivity property of $\dot\sigma(t)$, note that we have

$$\begin{aligned}\dot\sigma(t) &= \big\langle \nabla^2 f\big(x_1(t)\big)x_2, x_2(t)\big\rangle + \big\langle \nabla f\big(x_1(t)\big), \dot x_2(t)\big\rangle \\ &\quad + \alpha\big\langle \nabla f\big(x_1(t)\big), x_2(t)\big\rangle \\ &= \big\langle \nabla^2 f\big(x_1(t)\big)x_2(t), x_2(t)\big\rangle - \|\nabla f\big(x_1(t)\big)\|^2 \\ &\quad + \Big(\alpha - u_\alpha\big(x(t)\big)\Big)\big\langle \nabla f\big(x_1(t)\big), x_2(t)\big\rangle = 0,\end{aligned}$$

where the last equality follows from the definition of the proposed control law (4b). It is worth noting that one can simply replace the information of the Hessian $\nabla^2 f\big(x_1(t)\big)$ with the upper bound L_f and still arrives at the desired inequality, see also Remark 3.6 in regard to the 1st-order information oracle. Thus far, we have showed how the designed feedback control preserves the α-rate of convergence along the continuous flow of the hybrid system. Consider the initial state $x_2(0) = -\beta\nabla f\big(x_1(0)\big)$. To ensure $x(0) \in \mathcal{E}_\alpha$, notice that

$$\begin{aligned}\alpha\Big(f\big(x_1(0)\big) - f^*\Big) &\leq \frac{\alpha}{2\mu_f}\big\|\nabla f\big(x_1(0)\big)\big\|^2 \\ &= \frac{\alpha}{2\mu_f\beta}\big\langle -x_2(0), \nabla f\big(x_1(0)\big)\big\rangle \\ &\leq \big\langle \nabla f\big(x_1(0)\big), -x_2(0)\big\rangle,\end{aligned}$$

where in the first line we use (A2), and in the last line the condition (6c). Introducing the proposed $x_2(0)$ as the jump x^+ one can see that the range space of the jump map (4d) is indeed contained in the set $\mathcal{E}_\alpha$. Finally, we need to ensure that such a jump policy is well-defined, that is the trajectory lands in the interior of the flow set $\mathcal{C}$ defined as in (4c), i.e., the control values also belong to the admissible set $[u_{\min}, u_{\max}]$. In this view, we only need to take the initial control value into consideration, as the switching

law is continuous in the states and serves the purpose by design. Suppose that $x \in \mathcal{C}$, we then have the sufficient requirements

$$u_{\min} < \alpha + \frac{\|\nabla f(x_1^+)\|^2 - L_f\beta^2\|\nabla f(x_1^+)\|^2}{\beta\|\nabla f(x_1^+)\|^2} \leq u_\alpha(x^+) \leq \alpha + \frac{\|\nabla f(x_1^+)\|^2 + \ell_f\beta^2\|\nabla f(x_1^+)\|^2}{\beta\|\nabla f(x_1^+)\|^2} < u_{\max},$$

where the relations (4b) and (A1) are considered. Canceling the term $\|\nabla f(x_1^+)\|^2$ concludes the sufficient requirements in (6a) and (6a).

4.2. Proof of Theorem 3.7

Let us first introduce our proposed discretization method applied to the continuous-time hybrid system ($\mathcal{H}$) with the parameters (4). Applying the forward-Euler method, the velocity $\dot{x}_1$ is replaced with

$$\frac{x_1^{k+1} - x_1^k}{s} = x_2^k\,. \tag{12}$$

Similarly, the discretized version of the acceleration $\dot{x}_2$ gives rise to

$$\frac{x_2^{k+1} - x_2^k}{s} = -\nabla f(x_1^k) - u_{\alpha,d}(x^k)x_2^k,$$

where the discrete input $u_{\alpha,d}$ is given by (8b). Based on the above discussion, the corresponding discrete dynamics of ($\mathcal{H}$), (4) becomes (7), (8).

The definition of the flow set $\mathcal{C}_d$ (8c) implies

$$c_1\|x_2^k\|^2 \leq \|\nabla f(x_1^k)\|^2 \leq c_2\langle\nabla f(x_1^k), -x_2^k\rangle \leq c_2\|\nabla f(x_1^k)\| \cdot \|x_2^k\|,$$

where the extra inequality follows from the Cauchy-Schwarz inequality ($\forall\, a, b \in \mathbb{R}^n$, $\langle a, b\rangle \leq \|a\| \cdot \|b\|$). In order to guarantee that the flow set $\mathcal{C}_d$ is non-empty the relation (11a) should hold between the parameters c_1 and c_2 since $\sqrt{c_1} \leq \frac{\|\nabla f(x_1^k)\|}{\|x_2^k\|} \leq c_2$. Next, suppose that the parameters c_1, c_2, and β satisfy (11b). Multiplying (11b) by $\|\nabla f(x_1^k)\|$, one can observe that the range space of the jump map G_d (8d) is inside the flow set $\mathcal{C}_d$ (8c).

The discrete dynamics (7) is forced to evolve respecting the the flow set $\mathcal{C}_d$ defined in (8c). This observation yields

$$\begin{aligned} f(x_1^{k+1}) - f(x_1^k) &\leq \langle\nabla f(x_1^k), x_1^{k+1} - x_1^k\rangle + \frac{L_f}{2}\|x_1^{k+1} - x_1^k\|^2 \\ &\leq -s\langle\nabla f(x_1^k), -x_2^k\rangle + \frac{L_f s^2}{2}\|x_2^k\|^2 \\ &< -\frac{s}{c_2}\|\nabla f(x_1^k)\|^2 + \frac{L_f s^2}{2c_1}\|\nabla f(x_1^k)\|^2 \\ &= \big(-\frac{s}{c_2} + \frac{L_f}{2c_1}s^2\big)\|\nabla f(x_1^k)\|^2 \\ &\leq 2\mu_f\big(-\frac{s}{c_2} + \frac{L_f}{2c_1}s^2\big)\big(f(x_1^k) - f^*\big), \end{aligned}$$

where we made use of the relation (3), the definition (12), the relation (8c), and the assumption (A2), respectively. Then, considering the inequality implied by the first and last terms given above and adding $f(x_1^k) - f^*$ to both sides of the considered inequality, we arrive at

$$f(x_1^{k+1}) - f^* \leq \lambda(s, c_1, c_2, \beta)\,\big(f(x_1^k) - f^*\big)$$

where $\lambda(s, c_1, c_2, \beta)$ is given by (10). As a result, if the step size s is chosen such that $s < \frac{2c_1}{c_2 L_f}$ then $\lambda(s, c_1, c_2, \beta) \in (0, 1)$. Hence, the claim follows.

5. Numerical Example

In this section, a numerical example is provided to illustrate the results presented in preceding sections. We consider a quadratic objective function $f(x_1) = x_1^\top Q x_1$ where $x_1 \in \mathbb{R}^5$ with the matrix $Q = \text{diag}\{0.1, 0.2, \cdots, 0.5\}$. It is not difficult to verify that for quadratic objective functions we have $L_f = 2\lambda_{max}(Q) = 1$, $\mu_f = 2\lambda_{\min}(Q) = 0.2$, and due to the convexity we consider the lower bound $\ell_f = 0$. In what follows, we compare the performance of Algorithm 1 (denoted by **HD**) with that of Nesterov's accelerated method using the speed restarting scheme proposed in (Su et al., 2016) (denoted by **NSR**). We set $s = 1/L_f$ in Algorithm 1 and the rest of the parameters are computed according to Corollary 3.9.

The **NSR** algorithm requires a tuning parameter $k_{\min}$ that is the minimum number of iterations between two consecutive restart instants (i.e., no restarting is allowed unless the number of iterations after the last restarting instant is larger than or equal to $k_{\min}$). The motivation behind adjusting such a parameter is to avoid potentially frequent restarts in the discrete-time domain, which may have significant impact on the practical convergence rate (Su et al., 2016). The **NSR** scheme exhibits an improved rate of convergence although the rate analysis provided in (Su et al., 2016) holds true only for $k_{\min} = 1$. However, setting $k_{\min} > 1$ suffers from a shortcoming that **NSR** may lose the desirable monotonicity property. Figure 1 reports the performance of **NSR** for two

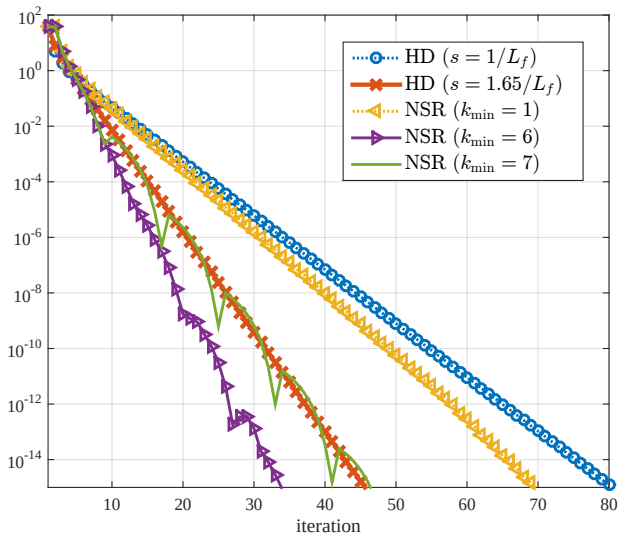

Figure 1. Comparison of suboptimality decay $f(x_1^k) - f^*$ between the discrete-time hybrid system (**HD**) employing Algorithm 1 and Nesterov' accelerated scheme with the speed restarting scheme (**NSR**).

values $k_{\min} \in \{1, 6, 7\}$. We note that when $k_{\min} = 6, 7$, **NSR** is no longer monotone, while it remains monotone for $k_{\min} \leq 5$. We remark that the best performance is achieved in case of $k_{\min} = 6$ as depicted in Figure 1. In regard with the proposed method, the monotonicity property is always preserved as long as the step size s respects the inequalities (11). We observe that among these admissible options, in this numerical case study, the best performance is achieved when $s = 1.65/L_f$. As illustrated in this numerical example, the step size proposed by Corollary 3.9 is practically outperformed by a bigger step size. This observation suggests that further analysis is required to prescribe a more intelligent step size that can carry useful dynamical features of the continuous-domain to the discrete-time counterpart.

References

Allen-Zhu, Z. Katyusha: The first direct acceleration of stochastic gradient methods. *arXiv preprint arXiv:1603.05953*, 2016.

Becker, S., Bobin, J., and Candès, E. J. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

Bubeck, S., Lee, Y. T., and Singh, M. A geometric alternative to nesterov's accelerated gradient descent. *arXiv preprint arXiv:1506.08187*, 2015.

Cabot, A. The steepest descent dynamical system with control. applications to constrained minimization. *ESAIM: Control, Optimisation and Calculus of Variations*, 10(2): 243–258, 2004.

Craven, B. D. and Glover, B. M. Invex functions and duality. *Journal of the Australian Mathematical Society*, 39(1): 1–20, 1985.

Drori, Y. and Teboulle, M. Performance of first-order methods for smooth convex minimization: a novel approach. *Mathematical Programming*, 145(1-2):451–482, 2014.

Drusvyatskiy, D., Fazel, M., and Roy, S. An optimal first order method based on optimal quadratic averaging. *arXiv preprint arXiv:1604.06543*, 2016.

Fazlyab, M., Ribeiro, A., Morari, M., and Preciado, V. M. Analysis of optimization algorithms via integral quadratic constraints: Non-strongly convex problems. *arXiv preprint arXiv:1705.03615*, 2017.

Ghadimi, E., Shames, I., and Johansson, M. Multi-step gradient methods for networked optimization. *IEEE Transactions on Signal Processing*, 61(21):5417–5429, 2013.

Goebel, R., Sanfelice, R. G., and Teel, A. R. *Hybrid dynamical systems: modeling, stability, and robustness*. Princeton University Press, 2012.

Gu, M., Lim, L.-H., and Wu, C. J. Parnes: a rapidly convergent algorithm for accurate recovery of sparse and approximately sparse signals. *Numerical Algorithms*, 64 (2):321–347, 2013.

Hanson, M. A. On sufficiency of the Kuhn-Tucker conditions. *Journal of Mathematical Analysis and Applications*, 80(2):545–550, 1981.

Hu, B. and Lessard, L. Dissipativity theory for Nesterov's accelerated method. *arXiv preprint arXiv:1706.04381*, 2017.

Karimi, H., Nutini, J., and Schmidt, M. *Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition*, pp. 795–811. Springer International Publishing, 2016.

Khalil, H. S. *Nonlinear systems*. Prentice Hall, 3rd edition, 2002.

Lan, G. and Monteiro, R. Iteration-complexity of first-order penalty methods for convex programming. *Mathematical Programming*, 138(1-2):115–139, 2013.

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

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

Nemirovski, A. Efficient methods in convex programming. 2005.

Nemirovskii, A., Yudin, D. B., and Dawson, E. R. Problem complexity and method efficiency in optimization. 1983.

Nesterov, Y. A method of solving a convex programming problem with convergence rate $\mathcal{O}(1/k^2)$. In *Soviet Mathematics Doklady*, volume 27, pp. 372–376, 1983.

Nesterov, Y. *Introductory lectures on convex optimization: a basic course*. Springer Science and Business Media, 2004.

Nesterov, Y. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

Nesterov, Y. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

Nesterov, Y. and Polyak, B. T. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

O'Donoghue, B. and Candès, E. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15(3):715–732, 2015.

Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Su, W., Boyd, S., and Candès, E. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems (NIPS 2014)*, pp. 2510–2518, 2014.

Su, W., Boyd, S., and Candès, E. A differential equation for modeling Nesterov's accelerated gradient method: theory and insights. *Journal of Machine Learning Research*, 17 (153):1–43, 2016.

Wibisono, A., Wilson, A. C., and Jordan, M. I. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47): E7351–E7358, 2016.

Willems, J. C. Dissipative dynamical systems part i: General theory. *Archive for Rational Mechanics and Analysis*, 45 (5):321–351, 1972.

Wilson, A. C., Recht, B., and Jordan, M. I. A Lyapunov analysis of momentum methods in optimization. *arXiv preprint arXiv:1611.02635*, 2016.