Uniting Nesterov's Accelerated Gradient Descent and the Heavy Ball Method for Strongly Convex Functions with Exponential Convergence Rate

Dawn M. Hustig-Schultz and Ricardo G. Sanfelice

***Abstract*— We propose a hybrid control algorithm that guarantees fast convergence and uniform global asymptotic stability of the set of minimizers of a convex objective function. The algorithm, developed using hybrid system tools, employs a uniting control strategy, in which Nesterov's accelerated gradient descent is used "globally" and the heavy ball method is used "locally", relative to the set of minimizers. The proposed hybrid control strategy switches between these accelerated methods to ensure convergence to the set of minimizers without oscillations, with a (hybrid) convergence rate that preserves the convergence rates of the individual optimization algorithms. We analyze key properties of the resulting closed-loop system including existence of solutions, uniform global asymptotic stability, and convergence rate. Additionally, attractivity properties of Nesterov's algorithm are analyzed. Numerical results validate the findings.**

I. Introduction

We propose an algorithm that solves optimization problems of the form $\min_{\xi\in\mathbb{R}^n} L(\xi)$ with accelerated gradient methods. *Nesterov's accelerated gradient descent* is an accelerated method that guarantees convergence to the set of minimizers of a convex function L [1]. Nesterov's algorithm achieves a faster convergence rate than classical gradient descent by adding a velocity term to the gradient. More recently, there has been growing interest in analyzing Nesterov's algorithm from a dynamical systems perspective [2] [3] [4]. Due to its implications on robustness, we are particularly interested in achieving global asymptotic stability of the set of minimizers of L, for Nesterov's algorithm, which the literature shows is a hard problem to solve [5] [6].

One characterization of the dynamical system for Nesterov's algorithm, proposed in [4], is

$$\ddot{\xi} + 2d\dot{\xi} + \frac{1}{M\zeta^2}\nabla L(\xi + \beta\dot{\xi}) = 0, \tag{1}$$

where the quantities d and β take different forms depending on the convexity properties of L. The constant $M > 0$ is the Lipschitz constant of the gradient of L, and the constant $\zeta > 0$ rescales time in solutions to (1). As in [4], in this paper, we consider the case where $\zeta = 1$, for simplicity of analysis. The dynamical system in (1) models a mass-spring-damper with a curvature-dependent damping term. The authors in [4] characterize the convergence rate for (1) to be exponential when L is strongly convex, and show a convergence rate of $\frac{1}{(t+2)^2}$ when L is nonstrongly convex (for $t \geq 1$). The work in [4] assumes that the set of interest is the origin, at which L is zero. The stability properties of these algorithms are not studied in [4].

Another commonly used accelerated gradient method is the *heavy ball method* [7], with dynamical system

$$\ddot{\xi} + \lambda\dot{\xi} + \gamma\nabla L(\xi) = 0 \tag{2}$$

where λ and γ are positive tunable parameters, which represent friction and gravity, respectively; see [8] [9]. While algorithms based on acceleration converge quickly, such methods can suffer from oscillations. Heavy ball, for instance, converges very slowly when λ is large and very quickly, but with oscillations, when λ is small [8]. The top plot in Figure 1 shows the former behavior. Nesterov's algorithm converges quickly but also suffers from oscillations [2], as shown in the middle plot in Figure 1. This behavior motivates the logic-based algorithm proposed in this paper to exploit the main features of each accelerated gradient method. Our proposed logic-based algorithm, shown in the bottom of Figure 1 shows the improvement obtained by using Nesterov's algorithm "globally," namely, far away from the minimizer of L, and heavy ball "locally,"namely, nearby the minimizer of L.

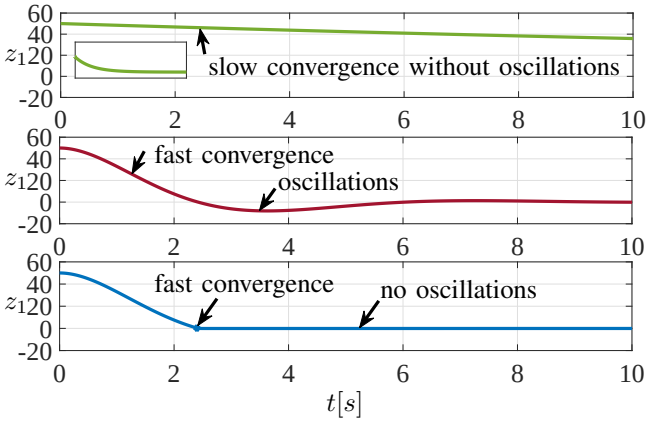

Fig. 1. Comparison of the performance of the heavy ball method, with large value of λ, Nesterov's accelerated gradient descent, and the proposed logic-based algorithm. The objective function is $L(z_1) = z_1^2$. Top: the heavy ball algorithm, with large λ, converges very slowly. Top inset: zoomed out view of heavy ball. Middle: Nesterov's accelerated gradient descent converges quickly, but with oscillations. Bottom: our proposed logic-based algorithm yields fast convergence, with no oscillations.

The main contributions of this paper are as follows.

D. M. Hustig-Schultz and R. G. Sanfelice are with the Department of Electrical and Computer Engineering, University of California, 1156 High Street, Santa Cruz, CA 95064, USA. dhustigs@ucsc.edu, ricardo@ucsc.edu. Research partially supported by NSF Grants no. ECS-1710621, CNS-1544396, and CNS-2039054, by AFOSR Grants no. FA9550-19-1-0053, FA9550-19-1-0169, and FA9550-20-1-0238, and by CITRIS and the Banatao Institute at the University of California.

Building from our previous work in [10], we propose a uniting algorithm, designed using hybrid system tools, that uses Nesterov's algorithm globally and the heavy ball method with large λ locally to guarantee fast convergence with uniform global asymptotic stability of the set of minimizers of L. The algorithm proposed in this paper not only renders the set of minimizers globally asymptotically stable, but also has a (hybrid) convergence rate that preserves the rates of the individual optimization algorithms for all (hybrid) time. Specifically, we show that our algorithm attains a (hybrid) exponential convergence rate when L is strongly convex. The proposed algorithm uses a switching strategy that measures the gradient of L, which is typically done via the method of finite differences, using measurements of L. Our algorithm, however, does not require measurements of the Hessian of L. In the process, we extend properties and convergence results for Nesterov's algorithm in [4]. In particular, we prove existence of solutions for (1) and global asymptotic stability of the set of minimizers for cost functions with a minimum value that is not necessarily zero.

The rest of the paper is organized as follows. Section II contains a brief explanation of notation, the hybrid systems framework, and definitions of strongly convex functions. Section III presents the motivation and problem statement. Section IV presents some of the nominal properties of Nesterov's algorithm and the heavy ball method. Section V introduces the hybrid algorithm, uniting Nesterov's algorithm globally and the heavy ball algorithm locally, and its nominal properties. Due to space constraints, detailed proofs of results will be published elsewhere.

II. Preliminaries

A. Notation

We denote the real, positive real, and natural numbers $\mathbb{R}$, $\mathbb{R}_{>0}$, and $\mathbb{N}$, respectively. The set C^n represents the family of n-th continuously differentiable functions. For vectors $v \in \mathbb{R}^n$ and $w \in \mathbb{R}^n$, $|v| = \sqrt{v^\top v}$ denotes the Euclidean vector norm of v, and $\langle v, w\rangle = v^\top w$ the inner product of v and w. The closure of a set S is denoted $\overline{S}$. The distance of a point $x \in \mathbb{R}^n$ to a set $S \in \mathbb{R}^n$ is defined by $|x|_S = \inf_{y\in S} |y - x|$. Given a set-valued mapping $M : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$, the domain of M is the set $\mathrm{dom}\ M = \{x \in \mathbb{R}^m \ : \ M(x) \neq \emptyset\ \}$.

B. Preliminaries on Hybrid Systems

In this paper, a hybrid system $\mathcal{H}$ has data (C, F, D, G) and is defined as [11, Definition 2.2]

$$\mathcal{H} = \begin{cases} \dot{x} \ \in F(x) & x \in C \\ x^+ \in G(x) & x \in D \end{cases} \tag{3}$$

where $x \in \mathbb{R}^n$ is the system state, $F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is the flow map, $C \subset \mathbb{R}^n$ is the flow set, $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is the jump map, and $D \subset \mathbb{R}^n$ is the jump set. The notation $\rightrightarrows$ indicates that F and G are set-valued maps. A solution ϕ is parameterized by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, where t is the amount of time that has passed and j is the number of jumps that have occurred. The domain of ϕ, namely, $\mathrm{dom}\phi \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a hybrid time domain, which is a set such that for each $(T, J) \in \mathrm{dom}\phi$, $\mathrm{dom}\phi \cap ([0, T] \times \{0, 1, \ldots, J\}) = \cup_{j=0}^{J}([t_j, t_{j+1}], j)$ for a finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \leq \ldots \leq t_{J+1}$. A hybrid arc ϕ is a function on a hybrid time domain that, for each $j \in \mathbb{N}$, $t \mapsto \phi(t, j)$ is absolutely continuous on the interval $I^j := \{t : (t, j) \in \mathrm{dom}\,\phi\}$. A solution ϕ to $\mathcal{H}$ is called maximal if it cannot be extended further. The set $\mathcal{S}_\mathcal{H}$ contains all maximal solutions to $\mathcal{H}$. A solution is called complete if its domain is unbounded. In the upcoming results, we will assume that our proposed hybrid closed-loop algorithm meets the hybrid basic conditions, as defined in [11, Assumption 6.5].

C. Preliminaries on Convex and Strongly Convex Functions

The algorithm proposed in this paper allows the cost function L to be strongly convex, as defined in [12].

Definition 2.1: (Strongly convex functions) A $\mathcal{C}^2$ function $L : \mathbb{R}^n \to \mathbb{R}$ is strongly convex if the following hold: there exists $\mu > 0$, such that for all $u_1, z_1 \in \mathbb{R}^n$,

(SC1) $\nabla^2 L(z_1) \geq \mu I$;

(SC2) $L(u_1) \geq L(z_1) + \langle \nabla L(z_1), u_1 - z_1\rangle + \frac{\mu}{2}|u_1 - z_1|^2$.

Additionally, some of the results in this paper employ the properties of convexity, quadratic growth, and the Polyak-Lojasiewicz condition, which are weaker conditions than strong convexity [13], [14], [15].

Definition 2.2: (Convex functions) A $\mathcal{C}^1$ function $L : \mathbb{R}^n \to \mathbb{R}$ is (nonstrongly) convex if $L(u_1) \geq L(z_1) + \langle \nabla L(z_1), u_1 - z_1\rangle$ for all $u_1, z_1 \in \mathbb{R}^n$.

Definition 2.3: (Quadratic growth) A function $L : \mathbb{R}^n \to \mathbb{R}$ has quadratic growth away from its minimizer $\mathcal{A}_1 = \{z_1^*\}$ if there exists $\alpha > 0$ such that $L(z_1) - L^* \geq \alpha\,|z_1|^2_{\mathcal{A}_1}$ for all $z_1 \in \mathbb{R}^n$, where $L^* := L(z_1^*)$.

III. Motivation and Problem Statement

As illustrated in Figure 1, the performance of Nesterov's accelerated gradient descent commonly suffers from oscillations near the minimizer. This is also the case for the heavy ball method when $\lambda > 0$ is small. However, when λ is large, the heavy ball method converges slowly, albeit without oscillations. In Section I we discussed how Nesterov's algorithm guarantees an exponential convergence rate for strongly convex L. We desire to attain such a rate, while avoiding oscillations via the heavy ball algorithm with large λ. We state the problem to solve as follows:

Problem $(\star)$**:** Given a scalar, real-valued, continuously differentiable, strongly convex objective function L, design an optimization algorithm that preserves the convergence rate of Nesterov's accelerated gradient descent, without oscillations and with uniformity with respect to the compact sets of initial conditions, without knowing the function L or the location of its minimizer, and with robustness. □

We propose a logic-based algorithm that unites Nesterov's algorithm, used globally to converge uniformly, with the heavy ball method, with large λ, used locally to avoid oscillations. For such an algorithm, we want to preserve

the convergence rates of the individual heavy ball and Nesterov algorithms. One difficulty in designing such a uniting algorithm is that the objective function L and the set of minimizers are unknown, so the algorithm must be able to detect when to switch, and do so in a way that avoids chattering.

IV. Stability and Convergence Properties of Heavy Ball and Nesterov's Algorithms

In this section, we present some useful properties of Nesterov's algorithm in (1) and the heavy ball algorithm in (2). For the analysis to follow, we impose the following Assumption on L.

Assumption 4.1: The function L is $\mathcal{C}^2$ and strongly convex.

A. Results for Nesterov's Accelerated Gradient Descent

Every maximal solution to (1) is complete and bounded, when L satisfies Assumption 4.1, as shown in the following lemma.

Lemma 4.2: (Existence of solutions to (1)*) Let L satisfy Assumption 4.1. Then, every maximal solution to* (1) *is bounded, complete, and unique.*

The following theorem shows that the ODE in (1), for the strongly convex case, has the set $\mathcal{A}$, defined as

$$\mathcal{A} := \left\{ z \in \mathbb{R}^{2n} \ : \ \nabla L(z_1) = z_2 = 0 \ \right\} = \mathcal{A}_1 \times \{0\}, \quad (4)$$

uniformly globally asymptotically stable. To establish it, we use the invariance principle in [16, Corollary 4.2].

Theorem 4.3: (Global asymptotic stability of $\mathcal{A}$ for (1)*) Let L satisfy Assumption 4.1. Then, the set $\mathcal{A}$, defined via* (4)*, with d and β defined via* (9)*, is globally asymptotically stable for* (1)*.*

In Theorem 4.3, we show global asymptotic stability of the set $\mathcal{A}$ for (1), which was not proved in [4].

B. Results for the Heavy Ball Algorithm

When Assumption 4.1 is satisfied, then every maximal solution to (2) is complete and bounded, as stated in the following lemma.

Lemma 4.4: (Existence of solutions to (2)*) Let L satisfy Assumption 4.1. Then, every maximal solution to* (2) *is bounded, complete, and unique.*

The following result establishes that the closed-loop system resulting from (2) has a set $\mathcal{A}$, defined via 4, globally asymptotically stable. To prove it, we use the invariance principle in [16, Corollary 4.2].

Theorem 4.5: (Global asymptotic stability of $\mathcal{A}$ for (2)*.) Let L satisfy Assumption 4.1. Additionally, let $\lambda > 0$ and $\gamma > 0$. Then, the set $\mathcal{A}$ defined in 4, is globally asymptotically stable for* (2)*.*

V. Uniting Optimization Algorithm

In this section, we present a uniting optimization algorithm for $\mathcal{C}^2$ strongly convex objective functions, namely functions for which Assumption 4.1 holds. The algorithm exploits measurements of ∇L, which in practice are typically approximated using measurements of L.

A. Modeling

We interpret the ODEs in (1) and (2) as control systems consisting of a plant and a control algorithm [10] [17]. Defining z_1 as ξ and z_2 as $\dot{\xi}$, the plant for both ODEs is given by the double integrator

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ u \end{bmatrix} =: F_P(z,u) \quad (z,u) \in \mathbb{R}^{2n} \times \mathbb{R}^n \quad (5)$$

with outputs given by functions of the state, as defined below. The control algorithm leading to (2) is

$$u = \kappa_0(h_0(z)) = -\lambda z_2 - \gamma \nabla L(z_1) \quad (6)$$

and the control algorithm leading to (1) is

$$u = \kappa_1(h_1(z)) = -2dz_2 - \frac{1}{M}\nabla L(z_1 + \beta z_2) \quad (7)$$

where $M > 0$ is the Lipschitz constant for ∇L and

$$h_0(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1) \end{bmatrix}, \ h_1(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1 + \beta z_2) \end{bmatrix}. \quad (8)$$

where h_0 corresponds to the output for heavy ball and h_1 corresponds to the output for Nesterov's algorithm. The parameters $\lambda > 0$ and $\gamma > 0$ should be designed to achieve convergence without oscillations nearby the minimizer.

As in [4], we have set $\zeta = 1$ in (1) for simplicity of analysis. For strongly convex L, the constants d and β are typically chosen as follows:

$$d := \frac{1}{(\sqrt{\kappa}+1)}, \quad \beta := \frac{(\sqrt{\kappa}-1)}{\sqrt{\kappa}+1} \quad (9)$$

where $\kappa := \frac{M}{\mu} \geq 1$ is the condition number associated with L; see [12] [18]. The constants defined as in (9) satisfy $2d + \beta = 1$.

The proposed logic-based algorithm "unites" the two individual controllers, or, equivalently, optimization algorithms defined by κ_0 and κ_1. The algorithm defined by κ_1 plays the role of the global algorithm in uniting control (see, e.g., [17]), while the algorithm defined by κ_0 plays the role of the local algorithm. A logic variable $q \in Q := \{0, 1\}$ indicates which algorithm is currently used.

The design of the logic and parameters of the individual algorithms is done using Lyapunov functions. The Lyapunov function used for heavy ball is

$$V_0(z) := \gamma \left(L(z_1) - L^* \right) + \frac{1}{2} \left| z_2 \right|^2 \quad (10)$$

defined for each $z \in \mathbb{R}^{2n}$. The Lyapunov function used for the Nesterov algorithm is

$$V_1(z) := \frac{1}{2} \left| a(z_1 - z_1^*) + z_2 \right|^2 + \frac{1}{M} \left(L(z_1) - L^* \right) \quad (11)$$

defined for each $z \in \mathbb{R}^{2n}$, where $a > 0$ is properly chosen.

To encapsulate the plant and static state-feedback laws, we define a hybrid closed-loop system $\mathcal{H}$ with state $x := (z, q) \in \mathbb{R}^{2n} \times Q$ and data (C, F, D, G) as follows:

$$F(x) := \begin{bmatrix} \begin{bmatrix} z_2 \\ \kappa_q(h_q(z)) \end{bmatrix} \\ 0 \end{bmatrix} \qquad \forall x \in C := C_0 \cup C_1 \tag{12a}$$

$$G(x) := \begin{bmatrix} z \\ 1 - q \end{bmatrix} \qquad \forall x \in D := D_0 \cup D_1 \tag{12b}$$

where C_0, C_1, D_0, and D_1 are to be defined. We denote, for each $q \in Q := \{0, 1\}$, the closed-loop systems resulting from the individual optimization algorithms as $\mathcal{H}_q$. Namely, the closed-loop resulting from using the global algorithm (Nesterov's algorithm, κ_1) is denoted as $\mathcal{H}_1$, and the closed-loop resulting from using the local algorithm (heavy ball, κ_0) is denoted as $\mathcal{H}_0$.

The switching rules in this section rely on quadratic growth of L, in Definition 2.3, which is a weaker property than strong convexity, even for $\mathcal{C}^1$ functions [13]. Exploiting Definition 2.3, the following lemma from [10] relates the size of the gradient at a point to the distance from the point to the minimizer z_1^*.

Lemma 5.1: (ε-suboptimality) Let L satisfy Assumption 4.1, and let $\alpha > 0$ come from Definition 2.3. For some $\varepsilon > 0$, if $z_1 \in \mathbb{R}^n$ is such that $|\nabla L(z_1)| \leq \varepsilon\alpha$, then $|z_1|_{\mathcal{A}_1} \leq \varepsilon$.

Proof. Since L satisfies Assumption 4.1, this implies that L also satisfies the properties in Definition 2.2 and Definition 2.3. Then, combining the properties in these definitions for $u_1 = z_1^*$ yields

$$\begin{aligned} \alpha \left|z_1\right|_{\mathcal{A}_1}^2 \leq \left|L(z_1) - L^*\right| &\leq \left|\nabla L(z_1)\right| \left|z_1\right|_{\mathcal{A}_1} \\ \implies \left|z_1\right|_{\mathcal{A}_1} &\leq \frac{1}{\alpha} \left|\nabla L(z_1)\right|. \end{aligned} \tag{13a}$$

This is true since $L(z_1) \geq L^*$. From (13a), we can deduce that $|\nabla L(z_1)| \leq \varepsilon\alpha$ implies $|z_1|_{\mathcal{A}_1} \leq \frac{1}{\alpha}(\varepsilon\alpha) = \varepsilon$. □

The ε-suboptimality condition from Lemma 5.1 is typically used as a stopping condition for optimization [12], as it indicates that the argument of L is close enough to the set of minimizers. We will exploit Lemma 5.1 to determine when the state component z_1 of the hybrid closed-loop system $\mathcal{H}_1$ is close enough to the global minimizer to switch to the local optimization algorithm, κ_0, in this way activating $\mathcal{H}_0$.

The switch between κ_0 and κ_1 is governed by a *supervisory* algorithm implementing switching logic. The supervisor selects between these two optimization algorithms, based on the plant's output and the optimization algorithm currently applied.

To that end, let $\varepsilon_0 > 0$, $\alpha > 0$, $c_0 > 0$, and $\gamma > 0$ from κ_0 be such that

$$\tilde{c}_0 := \varepsilon_0\alpha > 0 \tag{14a}$$

$$d_0 := c_0 - \gamma\left(\frac{\tilde{c}_0^2}{\alpha}\right) > 0. \tag{14b}$$

Then, V_0 in (10) can be upper bounded, using Definition 2.2 with $u_1 = z_1^*$, as follows: for each $z \in \mathbb{R}^{2n}$,

$$\begin{aligned} V_0(z) &= \gamma\left(L(z_1) - L^*\right) + \frac{1}{2}\left|z_2\right|^2 \\ &\leq \gamma\left|\nabla L(z_1)\right| \left|z_1\right|_{\mathcal{A}_1} + \frac{1}{2}\left|z_2\right|^2 \end{aligned} \tag{15}$$

Then, when $|\nabla L(z_1)| \leq \tilde{c}_0$, the ε-suboptimality condition in Lemma 5.1 implies $|z_1|_{\mathcal{A}_1} \leq \frac{\tilde{c}_0}{\alpha}$, from where we get

$$V_0(z) \leq \gamma\left(\frac{\tilde{c}_0^2}{\alpha}\right) + \frac{1}{2}\left|z_2\right|^2 \tag{16}$$

Then, by defining the set $\widetilde{\mathcal{U}}_0$ as

$$\widetilde{\mathcal{U}}_0 := \left\{ z \in \mathbb{R}^{2n} \ : \ \left|\nabla L(z_1)\right| \leq \tilde{c}_0, \frac{1}{2}\left|z_2\right|^2 \leq d_0 \right\}, \tag{17}$$

every $z \in \widetilde{\mathcal{U}}_0$ belongs to the c_0-sublevel set of V_0. In fact, using the conditions in (14) and (16), we have that for each $z \in \widetilde{\mathcal{U}}_0$,

$$V_0(z) \leq \gamma\left(\frac{\tilde{c}_0^2}{\alpha}\right) + \frac{1}{2}\left|z_2\right|^2 \leq c_0. \tag{18}$$

The parameters $\tilde{c}_0$, d_0, λ, and γ are designed so that $\widetilde{\mathcal{U}}_0$ is in the region where κ_0 is to be used. In this design, λ is large to avoid oscillations when converging to the minimum.

The set $\widetilde{\mathcal{U}}_0$ is contained in the basin of attraction induced by κ_0, due to the global attractivity property guaranteed by Theorem 4.5.

Let $\varepsilon_{1,0} \in (0, \varepsilon_0)$, $\alpha > 0$, and $c_{1,0} \in (0, c_0)$ such that

$$\tilde{c}_{1,0} := \varepsilon_{1,0}\alpha \in (0, \tilde{c}_0) \tag{19a}$$

$$d_{1,0} := c_{1,0} - a^2\left(\frac{\tilde{c}_{1,0}}{\alpha}\right)^2 - \frac{1}{M}\left(\frac{\tilde{c}_{1,0}^2}{\alpha}\right) \in (0, d_0) \tag{19b}$$

Then, with V_1, namely, with V_1 given in (11) and using Definition 2.2 with $u_1 = z_1^*$,

$$V_1(z) \leq a^2\left|z_1\right|_{\mathcal{A}_1}^2 + \left|z_2\right|^2 + \frac{1}{M}\left|\nabla L(z_1)\right| \left|z_1\right|_{\mathcal{A}_1} \tag{20}$$

Then, when $|\nabla L(z_1)| \leq \tilde{c}_{1,0}$, the ε-suboptimality condition in Lemma 5.1 implies $|z_1|_{\mathcal{A}_1} \leq \frac{\tilde{c}_{1,0}}{\alpha}$, from where we get

$$V_1(z) \leq a^2\left(\frac{\tilde{c}_{1,0}}{\alpha}\right)^2 + \left|z_2\right|^2 + \frac{1}{M}\left(\frac{\tilde{c}_{1,0}^2}{\alpha}\right). \tag{21}$$

Then, by defining

$$\widetilde{\mathcal{T}}_{1,0} := \left\{ z \in \mathbb{R}^{2n} \ : \ \left|\nabla L(z_1)\right| \leq \tilde{c}_{1,0}, \left|z_2\right|^2 \leq d_{1,0} \right\} \tag{22}$$

which, by construction, is contained in the interior of $\widetilde{\mathcal{U}}_0$, every $z \in \widetilde{\mathcal{T}}_{1,0}$ belongs to the $c_{1,0}$-sublevel set of V_1. In fact, using the conditions in (19) and (21), we have for each $z \in \widetilde{\mathcal{T}}_{1,0}$,

$$V_1(z) \leq a^2\left(\frac{\tilde{c}_{1,0}}{\alpha}\right)^2 + \left|z_2\right|^2 + \frac{1}{M}\left(\frac{\tilde{c}_{1,0}^2}{\alpha}\right) \leq c_{1,0}. \tag{23}$$

When $q = 1$, $|\nabla L(z_1)| \leq \tilde{c}_{1,0}$, and $|z_2|^2 \leq d_{1,0}$, the supervisor will switch from the global algorithm κ_1 to the

local algorithm κ_0. The constants c_0 and $c_{1,0}$, $\tilde{c}_0$, $\tilde{c}_{1,0}$, d_0, and $d_{1,0}$ comprise the hysteresis necessary to avoid chattering at the switching boundary.

To make the switch back to κ_1, choose $\hat{c}_0 > c_0$ and define the set

$$\left\{ z \in \mathbb{R}^{2n} \ : \ \gamma \left(L(z_1) - L^* \right) + \frac{1}{2} \left| z_2 \right|^2 \geq \hat{c}_0 \right\}. \tag{24}$$

This set defines the (closed) complement of a sublevel set with level larger than c_0. It is used for the design of the set D_0, which triggers the jumps from using κ_0 to use κ_1, so that when, in particular, the state z_1 is far from the set $\mathcal{A}_1$, then κ_1 is used to steer it back to nearby it.

Employing $\widetilde{\mathcal{U}}_0$, $\widetilde{\mathcal{T}}_{1,0}$, and the set in (24), the flow and jump sets C and D are defined as follows:

$$C_0 := \widetilde{\mathcal{U}}_0 \times \{0\}, \ C_1 := \overline{\mathbb{R}^{2n} \setminus \widetilde{\mathcal{T}}_{1,0}} \times \{1\} \tag{25a}$$

$$D_0 := \widetilde{\mathcal{T}}_{0,1} \times \{0\}, \ D_1 := \widetilde{\mathcal{T}}_{1,0} \times \{1\} \tag{25b}$$

where

$$\widetilde{\mathcal{T}}_{0,1} \subset \left\{ z \in \mathbb{R}^{2n} \ : \ \gamma \left(L(z_1) - L^* \right) + \frac{1}{2} \left| z_2 \right|^2 \geq \hat{c}_0 \right\}. \tag{26}$$

Remark 5.2: The algorithm has no knowledge of the particular objective function L; however, however, it uses knowledge of $L(z_1) - L^*$ to trigger jumps from $q = 0$ to $q = 1$, which would only occur due to wrong initializations of the logic variable or due to large measurement noise. Though somewhat restrictive, having knowledge of L^* is justified since a wide range of optimization problems require steering the value of the objective function to zero. Additionally, L^* – and, for that matter, objective functions in general – can be learned online [19].

B. Main Results

Under Assumption 4.1, the hybrid closed-loop system $\mathcal{H}$ in (12), with C and D defined via (25), is well-posed as it satisfies the hybrid basic conditions.

When, Assumption 4.1 holds, every maximal solution to $\mathcal{H}$ is complete and bounded, as stated in the following lemma.

Lemma 5.3: (Existence of solutions to $\mathcal{H}$) Let L satisfy Assumptions 4.1. Then, every maximal solution to the hybrid closed-loop system $\mathcal{H}$ in (12)*, with C and D defined via* (25)*, is bounded and complete.*

The following result establishes that the hybrid closed-loop system $\mathcal{H}$ with data as in (12) has the set

$$\mathcal{A} := \left\{ z \in \mathbb{R}^{2n} : \nabla L(z_1) = z_2 = 0 \right\} \times \{0\} = \mathcal{A}_1 \times \{0\} \times \{0\} \tag{27}$$

globally asymptotically stable, with convergence that is exponential. The last $\{0\}$ component in $\mathcal{A}$ is due to the logic state ending with value $q = 0$, namely using κ_0 as the state z reaches the set of minimizers of L. To prove the following result, we use the invariance principle in [16, Corollary 4.2].

Theorem 5.4: (Global asymptotic stability of $\mathcal{A}$ and convergence rate for $\mathcal{H}$) Let L satisfy Assumption 4.1. Additionally, let $\lambda > 0$, $\gamma > 0$, $\varepsilon_{1,0} \in (0, \varepsilon_0)$, $c_{1,0} \in (0, c_0)$, $c_0 > 0$, $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ from (14a) *and* (19a)*, $d_{1,0} \in (0, d_0)$ from* (14b) *and* (19b)*, and $\hat{c} > c_0$. Then, the set $\mathcal{A}$, defined in* (27)*, is globally asymptotically stable for $\mathcal{H}$. Furthermore, each maximal solution $(t, j) \mapsto x(t, j) = (z_1(t, j), z_2(t, j), q(t, j))$ of the hybrid closed-loop algorithm $\mathcal{H}$ starting from C_1 satisfies,*

$$L(z_1(t, 0)) - L^* \leq \left(L(z_1(0, 0)) - L^* \right) \exp(-at) \tag{28}$$

for each $t \in I^0$ where q is equal to 1, and

$$L(z_1(t, 1)) - L^* \leq \left(L(z_1(t_1, 1)) - L^* \right) \exp(-2\mu t) \tag{29}$$

for each $t \in I^1$ where q is equal to 0, where t_1 is the time at which the first jump occurs, where $\mu > 0$, and where $a > 0$ is defined, for $\kappa := \frac{M}{\mu} \geq 1$, as $a := d + \frac{\beta}{2\kappa} = \frac{1}{\sqrt{\kappa}} - \frac{1}{2\kappa}$.

Since the set $\mathcal{A}$ is compact and $\mathcal{H}$ satisfies the hybrid basic conditions, the global asymptotic stability in Theorem 5.4 is both uniform and robust [11].

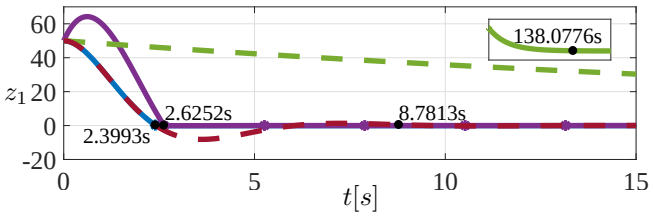

Fig. 2. A comparison of the evolution of z_1 and z_2 over time for $\mathcal{H}_0$, $\mathcal{H}_1$, HAND-2, and $\mathcal{H}$, defined via (12) with C and D defined in (25), for a function $L(z_1) := z_1^2$ with a single minimizer at $\mathcal{A}_1 = \{0\}$. Nesterov's accelerated gradient descent ($\mathcal{H}_1$), shown in red, settles to within 1% of $\mathcal{A}_1$ in about 8.8 seconds. The heavy ball algorithm $\mathcal{H}_0$, shown in green, settles to within 1% of $\mathcal{A}_1$ in about 138.1 seconds. HAND-2, shown in purple, settles to within 1% of $\mathcal{A}_1$ in about 2.6 seconds. The hybrid closed-loop system $\mathcal{H}$, shown in blue, settles to within 1% of $\mathcal{A}_1$ in about 2.4 seconds.

Example 5.5: To show the effectiveness of the uniting algorithm, we compare it in simulation to the optimization algorithms $\mathcal{H}_0$, $\mathcal{H}_1$, and the HAND-2 algorithm from [5] for strongly convex functions L satisfying Assumption 4.1. Using an alternate state space representation, namely, $z_1 := \xi$ and $z_2 := \xi + \frac{\tau}{2}\dot{\xi}$, the HAND-2 algorithm has state $(z, \tau) \in \mathbb{R}^{2n+1}$ and data (C, F, D, G)

$$F(z, \tau) := \begin{bmatrix} \frac{2}{\tau}(z_2 - z_1) \\ -2c\tau \nabla L(z_1) \\ 1 \end{bmatrix} \qquad \forall (z, \tau) \in C \tag{30a}$$

$$G(z, \tau) := \begin{bmatrix} G_z(z, \tau) \\ T_{\min} \end{bmatrix} \qquad \forall (z, \tau) \in D \tag{30b}$$

where $c > 0$, $G_z(z, \tau) := [z_1^\top z_1^\top]^\top$, $C := \left\{ (z, \tau) \in \mathbb{R}^{2n+1} \ : \ \tau \in [T_{\min}, T_{\max}] \right\}$, and $D := \left\{ (z, \tau) \in \mathbb{R}^{2n+1} \ : \ \tau \geq T_{\max} \right\}$, where $0 < T_{\min} < T_{\max} < \infty$. It is shown therein that each maximal solution $(t, j) \mapsto (z_1(t, j), z_2(t, j), \tau(t, j))$ of HAND-2 satisfies

$$L(z_1(t, j)) - L^* \leq k_a \left| \tilde{z}_1(0, 0) \right|^2 \exp\left(-\tilde{k}_b \tilde{\alpha} \left(t + j \right) \right) \tag{31}$$

for all $(t, j) \in \operatorname{dom}(z, \tau)$, where $k_a := 0.5 k_1 M$, $M > 0$, $k_1 := \frac{(c\mu)^{-1} + T_{\min}^2}{\Delta T^2}$, $\Delta T := T_{\max} - T_{\min}$, $0 < T_{\min} < T_{\max}$, $c > 0$, $\frac{1}{c\mu} < T_{\max}^2 - T_{\min}^2$, $\tilde{k}_b := 1 - k_0$,

$k_0 := \frac{(c\mu)^{-1}+T_{\min}^2}{T_{\max}^2}$, $j \geq \tilde{\alpha}(t+j) := \frac{\max\{t+j-\Delta T,0\}}{\Delta T+1}$, and $|\tilde{z}_1(0,0)| := |z_1(0,0) - z_1^*|$. This bound holds when $z_1(0,0) = z_2(0,0)$ and $\tau(0,0) = T_{\min}$.

To compare these algorithms, we use the objective function $L(z_1) = z_1^2$, with a single minimizer at $\mathcal{A}_1 = \{0\}$. This objective function is strongly convex with $\mu = 2$ and its gradient is Lipschitz continuous with $M = 2$, which results in $\kappa = \frac{M}{\mu} = 1$. For simulation, we used the heavy ball parameter values $\gamma = \frac{2}{3}$ and $\lambda = 40$. For HAND-2, we used the parameter values $T_{\max} = 5.63$, $T_{\min} = 3$, and $c = 0.25$. The parameter values for the uniting algorithm are $c_0 = 1000$ and $c_{1,0} = 400$, $\varepsilon_0 = 20$, $\varepsilon_{1,0} = 15$, and $\alpha_0 = \alpha_{1,0} = 1$, which yield the values $\tilde{c}_0 = 20$, $\tilde{c}_{1,0} = 15$, $d_0 = 733.3$, and $d_{1,0} = 231.25$, which are calculated via (14) and (19). We also pick $\hat{c} = 21$. Initial conditions for $\mathcal{H}_0$, $\mathcal{H}_1$, and $\mathcal{H}$ are $z_1(0,0) = 50$, $z_2(0,0) = 0$, and $q(0,0) = 1$, and for HAND-2 are $z_1(0,0) = 50$, $z_2(0,0) = 50$, and $\tau(0,0) = T_{\min}$.

Figure 2 shows the z_1 and z_2 components over time for each of the algorithms[1]. Black dots with times labeled in seconds denote when each solution settles within 1% of $\mathcal{A}_1$. Algorithm $\mathcal{H}_1$, shown in red, reaches the set $\mathcal{A}_1$ quickly with a rise time of about 2.4 seconds. However, it overshoots to about -8.15, at a peak time of 3.6 seconds. Then it continues oscillating until it settles within 1% of $\mathcal{A}_1$ in about 8.8 seconds. Algorithm $\mathcal{H}_0$, shown in green, slowly settles within 1% of $\mathcal{A}_1$ in about 138.1 seconds, which is the same as its rise time. The HAND-2 algorithm settles to within 1% of $\mathcal{A}_1$ in about 2.6 seconds. The hybrid closed-loop system $\mathcal{H}$, shown in blue, settles within 1% of $\mathcal{A}_1$ in about 2.4 seconds, which is also the same as its rise time. This is a 8.6% improvement over HAND-2, a 72.7% improvement over $\mathcal{H}_1$ and a 98.3% improvement over $\mathcal{H}_0$.

The different choice of initial conditions between HAND-1 and $\mathcal{H}$ is essential to the improved performance of the hybrid closed-loop algorithm $\mathcal{H}$ over HAND-2. Namely, the bound in (31) is guaranteed only when $z_1(0,0) = z_2(0,0)$. This means that the initial velocity for HAND-2 will be nonzero, unless the state starts at the set of minimizers, which can lead to overshoot in the transients of solutions. In comparison, $z_2(0,0)$ can be set to zero for $\mathcal{H}$, which helps to avoid overshoot. Such overshoot in HAND-2 means this algorithm reaches the set of minimizers later than $\mathcal{H}$, as seen in Figure 2.

VI. Conclusion

We presented an algorithm, designed using hybrid system tools, that properly unites Nesterov's accelerated algorithm and the heavy ball algorithm to ensure fast convergence and uniform global asymptotic stability. Future work will extend our results characterizing convergence rate and UGAS to include a generic parameter $\zeta > 0$. Application of the algorithm to learning is also part of future work.

[1]Code at
gitHub.com/HybridSystemsLab/UnitingGradientsSC

References

[1] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate $\mathcal{O}\left(\frac{1}{k^2}\right)$," in *Dokl. Akad. Nauk SSSR*, vol. 269, 1983, pp. 543–547.

[2] W. Su, S. Boyd, and E. Candes, "A differential equation for modeling nesterov's accelerated gradient method: Theory and insights," in *Advances in Neural Information Processing Systems*, 2014, pp. 2510–2518.

[3] A. S. Kolarijani, P. M. Esfahani, and T. Keviczky, "Continuous-time accelerated methods via a hybrid control lens," *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp. 3425–3440, 2020.

[4] M. Muehlebach and M. I. Jordan, "A dynamical systems perspective on Nesterov acceleration," *arXiv preprint arXiv:1905.07436*, 2019.

[5] J. I. Poveda and N. Li, "Inducing uniform asymptotic stability in time-varying accelerated optimization dynamics via hybrid regularization," *arXiv preprint arXiv:1905.12110*, 2019.

[6] J. I. Poveda and A. R. Teel, "The heavy-ball ode with time-varying damping: Persistence of excitation and uniform asymptotic stability," in *2020 American Control Conference (ACC)*. IEEE, 2020, pp. 773–778.

[7] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[8] H. Attouch, X. Goudou, and P. Redont, "The heavy ball with friction method, I. the continuous dynamical system: global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system," *Communications in Contemporary Mathematics*, vol. 2, no. 01, pp. 1–34, 2000.

[9] B. Polyak and P. Shcherbakov, "Lyapunov functions: An optimization theory perspective," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 7456–7461, 2017.

[10] D. M. Hustig-Schultz and R. G. Sanfelice, "A robust hybrid heavy ball algorithm for optimization with high performance," in *2019 American Control Conference (ACC)*, 2019, pp. 151–156.

[11] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. New Jersey: Princeton University Press, 2012.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[13] D. Drusvyatskiy and A. S. Lewis, "Error bounds, quadratic growth, and linear convergence of proximal methods," *Mathematics of Operations Research*, 2018.

[14] A. S. Kolarijani, P. M. Esfahani, and T. Keviczky, "Continuous-time accelerated methods via a hybrid control lens," *IEEE Transactions on Automatic Control*, 2019.

[15] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 795–811.

[16] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, New Jersey: Prentice Hall, 2002.

[17] R. Sanfelice, *Hybrid Feedback Control*. New Jersey: Princeton University Press, 2021.

[18] Y. Nesterov, "Introductory lectures on convex optimization, vol. 87," 2004.

[19] M. Wimmer, F. Stulp, S. Pietzsch, and B. Radig, "Learning local objective functions for robust face model fitting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 8, pp. 1357–1370, 2008.