

UC Santa Cruz

UC Santa Cruz Electronic Theses and Dissertations

Title

Hybrid Methods for Optimization with High Performance and Robustness

Permalink

<https://escholarship.org/uc/item/767788hs>

Author

Hustig-Schultz, Dawn

Publication Date

2022

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
SANTA CRUZ

**HYBRID METHODS FOR OPTIMIZATION WITH HIGH
PERFORMANCE AND ROBUSTNESS**

A dissertation submitted in partial satisfaction
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

by

Dawn M. Hustig-Schultz

September 2022

The Dissertation of Dawn M. Hustig-
Schultz
is approved:

Professor Ricardo G. Sanfelice, Chair

Professor Matthew Hale

Professor Dejan Milutinovic

Peter Biehl
Vice Provost and Dean of Graduate Studies

Copyright © by
Dawn M. Hustig-Schultz
2022

Table of Contents

List of Figures	vii
List of Tables	xv
List of Symbols	xvii
Abstract	xix
Dedication	xxii
Acknowledgments	xxiii
1 Introduction and Motivation	1
1.1 Overview of the Work	1
1.2 Uniting Heavy Ball Algorithms for Performance Improvement	3
1.2.1 Related Work	4
1.2.2 Motivation	5
1.2.3 Contributions	5
1.3 Uniting Nesterov’s Method and the Heavy Ball Method for Performance Improvement	8
1.3.1 Related Work	8
1.3.2 Motivation	11
1.3.3 Contributions For Strongly Convex L	14
1.3.4 Contributions For Nonstrongly Convex L	15
1.4 A Uniting Framework for Performance Improvement	19
1.4.1 Related Work	19
1.4.2 Motivation	20
1.4.3 Contributions	20
1.5 Hybrid Optimization for Nonconvex Problems	21
1.5.1 Related Work	22
1.5.2 Motivation	24
1.5.3 Contributions	27

1.6	A Totally Asynchronous, Block-Based Heavy Ball Algorithm for Convex Optimization	29
1.6.1	Related Work	30
1.6.2	Motivation	32
1.6.3	Contributions	33
1.7	Organization	34
2	Preliminaries	37
2.1	Hybrid Systems	37
2.2	Optimization	39
2.3	Morse Theory	41
2.4	Nonsmooth Lyapunov Functions	42
2.5	Mean Value Theorem	44
2.6	Properties of Sets	44
2.7	Difference Inclusions	45
3	Accelerated Gradient Algorithms Modeled as Dynamical Systems	47
3.1	Nesterov's Accelerated Gradient Descent Modeled as a Dynamical System	47
3.1.1	Strongly Convex L	48
3.1.2	Nonstrongly Convex L	58
3.1.3	Extensions of the Results for Nonstrongly Convex L	72
3.2	The Heavy Ball Method Modeled as a Dynamical System	73
3.2.1	Strongly Convex L	73
3.2.2	Nonstrongly Convex L	75
3.2.3	Extensions of the Results for Nonstrongly Convex L	84
3.2.4	Nonconvex L	85
4	Uniting Heavy Ball Algorithms	92
4.1	Problem Statement	92
4.2	Modeling	93
4.3	Uniting Heavy Ball Methods Using Measurements of L and ∇L	96
4.3.1	Design of the Sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$	97
4.3.2	Design of the Parameter λ_q	98
4.3.3	Well-posedness of the Hybrid Closed-Loop System \mathcal{H}	99
4.3.4	Existence of solutions to \mathcal{H}	100
4.3.5	Main Result	102
4.3.6	Numerical Example	107
4.4	Uniting Heavy Ball Methods Using Measurements of ∇L	109
4.4.1	Design of \mathcal{U}_0	111
4.4.2	Design of $\mathcal{T}_{1,0}$	113
4.4.3	Design of $\mathcal{T}_{0,1}$	114

4.4.4	Well-posedness of the Hybrid Closed-Loop System \mathcal{H} . . .	116
4.4.5	Existence of solutions to \mathcal{H}	117
4.4.6	Main Result	119
4.4.7	Numerical Examples	122
4.5	Extensions	129
5	Uniting Nesterov’s Method and the Heavy Ball Method	131
5.1	Strongly Convex L	131
5.1.1	Problem Statement	132
5.1.2	Modeling	132
5.1.3	Design of \mathcal{U}_0	135
5.1.4	Design of $\mathcal{T}_{1,0}$	136
5.1.5	Design of $\mathcal{T}_{0,1}$	138
5.1.6	Design of the Parameter λ	139
5.1.7	Well-posedness of the Hybrid Closed-Loop System \mathcal{H} . . .	139
5.1.8	Existence of Solutions to \mathcal{H}	140
5.1.9	Main Result	142
5.1.10	Numerical Examples	146
5.2	Nonstrongly Convex L	153
5.2.1	Problem Statement	154
5.2.2	Modeling	155
5.2.3	Design of the Set \mathcal{U}_0	159
5.2.4	Design of the Set $\mathcal{T}_{1,0}$	160
5.2.5	Design of the Set $\mathcal{T}_{0,1}$	161
5.2.6	Well-posedness of the Hybrid Closed-Loop System \mathcal{H} . . .	162
5.2.7	Existence of Solutions to \mathcal{H}	163
5.2.8	Main Result	165
5.2.9	Numerical Examples	170
5.2.10	Extensions	181
6	Uniting Framework for Accelerated Optimization	183
6.1	Problem Statement	184
6.2	Hybrid Uniting Framework for Accelerated Gradient Methods . .	185
6.2.1	Modeling	185
6.2.2	Design	188
6.2.3	Basic Properties of \mathcal{H}	190
6.3	Examples for Applying the framework	199
6.3.1	Uniting Heavy Ball Algorithms	199
6.3.2	Uniting Nesterov’s Method and the Heavy Ball Method for Strongly Convex L	202
6.3.3	Uniting Nesterov’s Method and the Heavy Ball Method for Nonstrongly Convex L	205

6.4	Uniting Other Gradient Algorithms	207
7	Hybrid Accelerated Optimization for Nonconvexity	210
7.1	Problem Statement	210
7.2	Design	211
7.3	Hybrid System Model of the Proposed Algorithm	213
7.4	Main Result	215
7.5	Numerical Example	226
8	Accelerated Multiagent Optimizaton	232
8.1	Problem Statement	232
8.2	Synchronous Heavy Ball	233
8.2.1	Modeling	233
8.2.2	Results for Algorithm 6	238
8.3	Synchronous, Double-Update Heavy Ball	248
8.3.1	Modeling	249
8.3.2	Results for Algorithm 7	251
8.4	Asynchronous, Double-Update Heavy Ball	257
8.4.1	Modeling	257
8.4.2	Forward Invariance of $(X \times X)^N$ for Algorithm 8	261
8.4.3	Convergence rate of Algorithm 8	263
8.5	Numerical Example	268
9	Conclusion	272
9.1	Summary	272
9.2	Future Directions	274
A	General Results on Hybrid Systems	277
A.1	Existence of Solutions, Stability, and Invariance	277
B	General Results on Totally Asynchronous Multiagent Algorithms	283
B.1	Totally Asynchronous Convergence	283
C	General Results on Optimality and Projection	288
C.1	Optimality Conditions and Projection Theorem	288
D	Code for Numerical Examples	290
	Bibliography	292

List of Figures

- 1.1 Comparison of the performance of the heavy ball method, with large and small values of λ , with the proposed logic-based algorithm for $L(\xi_1) = \frac{1}{4}\xi^2$. Top: when λ is large, heavy ball converges very slowly. Middle: when λ is small, heavy ball converges quickly, but with wild oscillations. Bottom: our proposed logic-based strategy yields fast convergence, with no oscillations. 6
- 1.2 Comparison of the performance of the heavy ball method, with large λ , Nesterov's accelerated gradient descent, and the proposed logic-based algorithm, for strongly convex L . The objective function is $L(\xi) = \xi^2$. Top left: the heavy ball algorithm, with large λ , converges very slowly. Top inset: zoomed out view of heavy ball. Middle left: Nesterov's accelerated gradient descent converges quickly, but with oscillations. Bottom left: our proposed logic-based algorithm yields fast convergence, with no oscillations. Right: comparison of the value of $L(\xi) - L^*$ (in log scale) versus time for each algorithm. Different tunings of the logic-based algorithm's parameters leads to modifications of the solution's profile. 12

- 1.3 Comparison of the performance of the heavy ball method, with large λ , Nesterov’s accelerated gradient descent, and the proposed logic-based algorithm, for nonstrongly convex L . The objective function is $L(\xi) = \xi^2$. Top left: the heavy ball algorithm, with large λ , converges very slowly. Top inset: zoomed out view of heavy ball. Middle left: Nesterov’s accelerated gradient descent converges quickly, but with oscillations. Bottom left: our proposed logic-based algorithm yields fast convergence, with no oscillations. Right: comparison of the value of $L(\xi) - L^*$ (in log scale) versus time for each algorithm. Different tunings of the logic-based algorithm’s parameters leads to modifications of the solution’s profile. 12
- 1.4 A comparison of the evolution of L over time for Nesterov’s method in (1.5), heavy ball, HAND-1, and our proposed uniting algorithm, for a function $L(\xi) := \xi^2$, with a single minimizer at $\xi^* = 0$. Nesterov’s method, shown in purple, settles to within 1% of ξ^* in about 8.8 seconds. The heavy ball algorithm, shown in green, settles to within 1% of ξ^* in about 138.1 seconds. HAND-1, shown in orange, settles to within 1% of ξ^* in about 14.3 seconds. The hybrid closed-loop system \mathcal{H} , shown in blue, settles to within 1% of z_1^* in about 2.4 seconds. As opposed to Figure 1.3, which uses $\zeta = 2$ for \mathcal{H}_1 , this example uses $\zeta = 1$, which results in slower convergence of solutions to \mathcal{H} and \mathcal{H}_1 than in Figure 1.3. 18

1.5	<p>Comparing performance of the proposed hybrid algorithm to other optimization methods, with small noise in measurements of the gradient, when the system starts near a local maximum, at $\xi_0 \approx 15$. For classic gradient descent (top left), a gradient-based optimization algorithm with discontinuous map f (top right), and simulated annealing, via Langevin diffusion (bottom left) the state ξ get pushed to the local maximum at $\xi = 15$, and stays there. All trajectories in the bottom left plot have the noise signal $\vartheta := \left(-\frac{\tau(\log(\tau))^2}{B_{SA}}\right) (\nabla L(\xi) + \Omega \text{sign}(\nabla L(\xi))(10^{-12}))$, where $\tau > 0$ is time, $B_{SA} > 0$ is large, and Ω is a normally distributed random number. The trajectory with red dotted line converging to the minimizer has an added constant of $B_\vartheta = 5 \times 10^{-13}$, such that $\vartheta + B_\vartheta$, while the other three trajectories represented by dashed lines have added constants B_ϑ equal to 3×10^{-13}, 10^{-13}, and 10^{-14}, respectively. The last trajectory, represented by the solid blue line, has no constant added to the noise signal ϑ. The hybrid algorithm (bottom right), with noise of the form $\left(-\frac{\tau(\log(\tau))^2}{B_{SA}}\right) (\nabla L(\xi) + \Omega \text{sign}(\nabla L(\xi))(10^{-12}))$ added to the gradient of L, where Ω is a normally distributed random number, is still able to escape the local maximum at $\xi = 15$ and converge to a local minimum at $\xi = 10$.</p>	25
1.6	<p>A comparison of the performance of our proposed double heavy ball algorithm with the asynchronous primal-dual algorithm for constrained gradient descent, with the dual variables fixed at zero. Convergence is twice as fast for double heavy ball (6 iterations) as it is for the gradient descent-based algorithm (12 iterations). . . .</p>	32
4.1	<p>Feedback diagram of the hybrid closed-loop system \mathcal{H}, in (4.3), uniting global and local optimization algorithms.</p>	94

- 4.2 A comparison of the evolution of L over time for \mathcal{H}_0 , \mathcal{H}_1 , and \mathcal{H} , for a function $L = \frac{1}{4}z_1^\top P z_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer at $z_1^* = (0, 0, \dots, 0)$. The heavy ball algorithm \mathcal{H}_1 uses $\lambda_1 = \frac{1}{2}$ (shown in purple) and settles to within 1% of z_1^* in about 16.3 seconds. The heavy ball algorithm \mathcal{H}_0 uses $\lambda_0 = 10.5$ (shown in green) and settles to within 1% of z_1^* in about 96.4 seconds. The hybrid closed-loop system \mathcal{H} (shown in blue) settles to within 1% of z_1^* in about 2.9 seconds. 107
- 4.3 An illustration of hysteresis in the design of the sets \mathcal{U}_0 , \mathcal{T}_0 , and $\mathcal{T}_{0,1}$ on \mathbb{R}^{2n} , via the constants $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$, $d_{1,0} \in (0, d_0)$, and $c_0 > 0$. Left: due to the design of \mathcal{U}_0 in (4.20), every $z \in \mathcal{U}_0$ belongs to the c_0 -sublevel set of the Lyapunov function V_0 , where V_0 is defined via (4.2). Hence, the same value of $c_0 > 0$ is also used to define $\mathcal{T}_{0,1}$ as the closed complement of a sublevel set of V_0 with level equal to c_0 . Right: the constants $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ and $d_{1,0} \in (0, d_0)$, defined via (4.22), are chosen such that the set $\mathcal{T}_{1,0}$ in (4.25) is contained in the interior of \mathcal{U}_0 115
- 4.4 Top: The evolution over time of z_1 , for the nominal hybrid closed-loop algorithm \mathcal{H} in (4.3), with C and D defined in (4.14), for a function $L(z_1) := \frac{1}{2}z_1^2$ with a single minimizer at $z_1^* = 0$. The time at which the solution settles to within 1% of z_1^* is marked with a dot and labeled in seconds. The jump is labeled with an asterisk. Bottom: Simulation results for perturbed solutions using zero mean Gaussian noise, with each simulation using a different value of the standard deviation σ . Results listed are for a large value of $t + j$. 123

- 4.5 Simulation results for hybrid closed-loop algorithm \mathcal{H} in (4.3), with C and D defined in (4.14), for a function $L(z_1) := \frac{1}{2}z_1^2$ with a single minimizer at $z_1^* = 0$, with zero-mean Gaussian noise added to measurements of the gradient. Each subplot is labeled with the standard deviation used. Left subplots: the value of z_1 over time for each perturbed solution, with the jump in each solution labeled by an asterisk. Right subplots: the corresponding value of L over time for each perturbed solution. 125
- 4.6 A comparison of the evolution of L over time for \mathcal{H}_0 , \mathcal{H}_1 , and \mathcal{H} in (4.3), with C and D defined in (4.14), for a function $L = \frac{1}{4}z_1^\top P z_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer at $z_1^* = (0, 0, \dots, 0)$. The heavy ball algorithm \mathcal{H}_1 uses $\lambda_1 = \frac{1}{2}$ (shown in purple) and settles to within 1% of z_1^* in about 16.3 seconds. The heavy ball algorithm \mathcal{H}_0 uses $\lambda_0 = 10.5$ (shown in green) and settles to within 1% of z_1^* in about 96.4 seconds. The hybrid closed-loop system \mathcal{H} (shown in blue) settles to within 1% of z_1^* in about 2.9 seconds. 127
- 5.1 Top: The evolution over time of z_1 , for the nominal hybrid closed-loop algorithm \mathcal{H} in Sections 5.1.2-5.1.5, for a function $L(z_1) := z_1^2$ with a single minimizer at $z_1^* = 0$. The time at which the solution settles to within 1% of z_1^* is marked with a dot and labeled in seconds. The jump is labeled with an asterisk. Bottom: Simulation results for perturbed solutions using zero mean Gaussian noise, with each simulation using a different value of the standard deviation σ . Results listed are for a large value of $t + j$ 147

5.2	Simulation results for hybrid closed-loop algorithm \mathcal{H} in Sections 5.1.2-5.1.5, for a function $L(z_1) := z_1^2$ with a single minimizer at $z_1^* = 0$, with zero-mean Gaussian noise added to measurements of the gradient. Each subplot is labeled with the standard deviation used. Left subplots: the value of z_1 over time for each perturbed solution, with the jump in each solution labeled by an asterisk. Right subplots: the corresponding value of L over time for each perturbed solution.	149
5.3	A comparison of the evolution of L over time for \mathcal{H}_0 , \mathcal{H}_1 (both in (5.4)), HAND-2, and \mathcal{H} in (5.3) – with \mathcal{U} in (4.20), $\mathcal{T}_{1,0}$ in (5.12), and $\mathcal{T}_{0,1}$ in (5.15) – for a function $L(z_1) := z_1^2$, with a single minimizer at $z_1^* = 0$. Nesterov’s method, shown in purple, settles to within 1% of z_1^* in about 8.8 seconds. The heavy ball algorithm, shown in green, settles to within 1% of z_1^* in about 138.1 seconds. HAND-2, shown in orange, settles to within 1% of z_1^* in about 2.1 seconds. The hybrid closed-loop system \mathcal{H} , shown in blue, settles to within 1% of z_1^* in about 2.4 seconds.	152
5.4	Feedback diagram of the hybrid closed-loop system \mathcal{H} (on the right), in (5.22), uniting global and local optimization algorithms. An example optimization problem to solve is shown on the left and, for this example optimization problem, measurements of the gradient are used for the input of κ_q	158
5.5	Top: The evolution over time of z_1 , for the nominal hybrid closed-loop algorithm \mathcal{H} , for a function $L(z_1) := z_1^2$ with a single minimizer at $z_1^* = 0$. The time at which the solution settles to within 1% of z_1^* is marked with a dot and labeled in seconds. The jump is labeled with an asterisk. Bottom: Simulation results for perturbed solutions using zero mean Gaussian noise, with each simulation using a different value of the standard deviation σ . Results listed are for a large value of $t + j$	172

5.6	Simulation results for hybrid closed-loop algorithm \mathcal{H} , for a function $L(z_1) := z_1^2$ with a single minimizer at $z_1^* = 0$, with zero-mean Gaussian noise added to measurements of the gradient. Each subplot is labeled with the standard deviation used. Left subplots: the value of z_1 over time for each perturbed solution, with the jump in each solution labeled by an asterisk. Right subplots: the corresponding value of L over time for each perturbed solution.	173
5.7	The evolution of L over time, from different initial conditions, for \mathcal{H} (left) and HAND-1 (right). All solutions are for the objective function $L(z_1) := z_1^2$, and the parameters used for HAND-1 and \mathcal{H} are listed in Table 5.3, with different values of c_0 and $c_{1,0}$ for each solution of \mathcal{H} , leading to different values of d_0 calculated via (5.5) and $d_{1,0}$ calculated via (5.26), and different values of r and δ_{med} for each solution of HAND-1, leading to different values of T_{med} and T_{max}	178
7.1	The evolution of z_1 over time for the hybrid system \mathcal{H} , for the objective function $L(z_1) = \frac{z_1^2(z_1-10)^2(z_1-20)^2(z_1-30)^2}{10,000}$, with $\mathcal{A}_{1_{\text{min}}} = \{0, 10, 20, 30\}$, $\mathcal{A}_{1_{\text{max}}} = \{5(3 - \sqrt{5}), 15, 5(3 + \sqrt{5})\}$, and $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.06$, $\rho_1 = 0.05$, $\rho_2 = 0.06$, $\varsigma = 1$, $\lambda = 145$, and $\gamma = \frac{3}{4}$. This plot shows different solutions, starting from different initial conditions. Solutions start at local maxima at $z_1(0, 0) = 15$, $z_1(0, 0) = 5(3 - \sqrt{5})$, and $z_1(0, 0) = 5(3 + \sqrt{5})$, as well as at the points $z_1(0, 0) = 6$, $z_1(0, 0) = 24.5$, $z_1(0, 0) = -1$, and $z_1(0, 0) = 31$, which are neither maxima nor minima. All solutions start with $z_2(0, 0) = 0$ and $q(0, 0) = 0$. Times when each solution converges to within 0.01 of $\mathcal{A}_{1_{\text{min}}}$ are marked with black dots and labeled in seconds. Jumps are labeled with asterisks.	230

8.1 Comparing the effect of different computation rates on solutions, for the objective function in (8.81) with the constraint set $X_i = [1, 10]$. Top: A comparison of the evolution over time of $|z_1 - x^*|$, with $(D_{\text{async}}, G_{\text{async}})$ on the left and the asynchronous primal-dual algorithm in (8.80) on the right. Bottom: A comparison of the evolution over time of $|z_1(k) - z_1(k - 1)|$, with $(D_{\text{async}}, G_{\text{async}})$ on the left and the asynchronous primal-dual algorithm in (8.80) on the right. 270

List of Tables

4.1	Average times for which \mathcal{H} , \mathcal{H}_0 , and \mathcal{H}_1 settle to within 1% of z_1^* , and the average percent improvement of \mathcal{H} over each algorithm. Percent improvement is calculated via (4.13). The objective function used for this table is $L = \frac{1}{4}z_1^\top Pz_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer at $z_1^* = (0, 0, \dots, 0)$	108
4.2	Average times for which \mathcal{H} in (4.3), with C and D defined in (4.14), \mathcal{H}_0 , and \mathcal{H}_1 settle to within 1% of z_1^* , and the average percent improvement of \mathcal{H} over each algorithm. Percent improvement is calculated via (4.13). The objective function used for this table is $L = \frac{1}{4}z_1^\top Pz_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer at $z_1^* = (0, 0, \dots, 0)$	128
5.1	Average times for which \mathcal{H} , \mathcal{H}_0 , and \mathcal{H}_1 settle to within 1% of z_1^* , and the average percent improvement of \mathcal{H} over each algorithm. Percent improvement is calculated via (4.13). The objective function used for this table is $L(z_1) := z_1^2$	153
5.2	Average times for which \mathcal{H} , \mathcal{H}_0 , \mathcal{H}_1 , and HAND-1 settle to within 1% of z_1^* , and the average percent improvement of \mathcal{H} over each algorithm. Percent improvement is calculated via (5.39). The objective function used for this table is $L(z_1) := z_1^2$	177

5.3	Times for which \mathcal{H} and HAND-1 settle to within 1% of z_1^* , and percent improvement of \mathcal{H} over HAND-1, for solutions from different initial conditions, shown in Figure 5.7. The objective function used for this table is $L(z_1) := z_1^2$	178
5.4	Times for which \mathcal{H} , \mathcal{H}_0 , \mathcal{H}_1 , and HAND-1 settle to within 1% of z_1^* , and percent improvement of \mathcal{H} over each algorithm, as shown in Figure 1.4. Percent improvement is calculated via (5.39). The objective function used for this table is $L(z_1) := z_1^2$	180
7.1	Times in which different solutions converge to within 0.01 of \mathcal{A}_1	229

List of Symbols

\mathbb{R}	The set of all real numbers
$\mathbb{R}_{>0}$	The set of all positive real numbers
\mathbb{N}	The set of all positive integers including zero, i.e., $\{0, 1, 2, \dots\}$
\mathbb{B}	The closed unit ball, of appropriate dimension, in the Euclidean norm
C^n	The set representing the family of n -th continuously differentiable functions
(v, w)	Given vectors $v \in \mathbb{R}^m$ and $w \in \mathbb{R}^m$, $[v^\top, w^\top]^\top$ is equivalent to (v, w)
$\langle v, w \rangle$	The inner product of vectors $v \in \mathbb{R}^m$ and $w \in \mathbb{R}^m$, namely, $u^\top v$
$ v $	The Euclidean vector norm $ v = \ v\ _2 = \sqrt{v^\top v}$
$ v _\infty$	the max vector norm $ v _\infty = \max_i v_i $, which is the maximum of the absolute value of its components
$\Pi_X[v]$	The orthogonal projection, with respect to the Euclidean norm, of a vector v onto the convex set X , namely, $\Pi_X[v] = \arg \min_{w \in X} w - v $
$\Pi(S)$	The projection of S onto \mathbb{R}^m , i.e., $\Pi(S) := \{x \in \mathbb{R}^m : \exists y \text{ such that } (x, y) \in S\}$
\bar{S}	The closure of a set S
$ x _S$	The distance of a point $x \in \mathbb{R}^m$ to a set $S \in \mathbb{R}^m$, i.e., $ x _S = \inf_{y \in S} y - x $
x_\circ	The initial condition of a state x
$\text{dom } M$	The domain of $M : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$, i.e., $\text{dom } M = \{x \in \mathbb{R}^m : M(x) \neq \emptyset\}$

- $\text{rge } M$ The range of $M : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$, i.e.,
 $\text{rge } M = \{y \in \mathbb{R}^n : \exists x \in \mathbb{R}^m \text{ such that } y \in M(x)\}$
- \mathcal{K} A function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a class- \mathcal{K} function, also written $\alpha \in \mathcal{K}$, if α is zero at zero, continuous, and strictly increasing
- \mathcal{K}_∞ A function $\alpha : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a class- \mathcal{K}_∞ function, also written $\alpha \in \mathcal{K}_\infty$, if α is zero at zero, continuous, strictly increasing, and unbounded
- \mathcal{KL} A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a class- \mathcal{KL} function, also written $\beta \in \mathcal{KL}$, if it is nondecreasing in its first argument, nonincreasing in its second argument, $\lim_{r \rightarrow 0^+} \beta(r, s) = 0$ for each $s \in \mathbb{R}_{\geq 0}$, and $\lim_{s \rightarrow \infty} \beta(r, s) = 0$ for each $r \in \mathbb{R}_{\geq 0}$.

Abstract

Hybrid Methods for Optimization with High Performance and Robustness

by

Dawn M. Hustig-Schultz

Optimization has valuable applications in many areas of technology, including smart grids, transportation systems, multiagent systems, wireless sensor and communications networks, and deep learning. This dissertation focuses on developing hybrid algorithms for accelerated gradient descent, both for convex and nonconvex objective functions, with fast convergence, stability, and robustness.

The first two algorithms, developed using hybrid system tools, feature a uniting control strategy, in which two standard heavy ball algorithms, one used globally and another used locally, with properly designed gravity and friction parameters, are employed. The proposed hybrid control strategy switches the parameters to converge quickly to the minimizer of the nonstrongly convex objective function without oscillations. A hybrid control algorithm implementing a switching strategy that measures the objective function and its gradient, and another algorithm that only measures its gradient, are designed. Key properties of the resulting closed-loop systems, including existence of solutions, asymptotic stability, and convergence rate, are analyzed.

The second two algorithms – one for strongly convex objective functions and the other for nonstrongly convex objective functions – also employ 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 minimizer. 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.

Based on the uniting algorithms above, a uniting framework is designed, which allows the use of any accelerated gradient method for the global and local algorithms. Sufficient conditions are determined, which lead to general results on well-posedness, existence of solutions, and uniform global asymptotic stability for the hybrid closed-loop framework.

Then, we propose a hybrid algorithm for optimization, to ensure convergence to a local minimizer of a nonconvex Morse objective function L with a single, scalar argument. Developed using hybrid system tools, and based on the heavy ball method, the algorithm features switching strategies to detect whether the state is near a critical point and enable escape from local maximizer, using measurements of the gradient of L . Key properties of the resulting closed-loop system, including existence of solutions and practical global attractivity, are revealed.

In addition, we propose a totally asynchronous multiagent algorithm, based on the heavy ball method, that guarantees fast convergence to the minimizer of a \mathcal{C}^2 , convex objective function. The algorithm is parallelized in the sense that the decision variable is partitioned into blocks, each of which is updated only by a single agent. We consider two types of asynchrony: in agents' computations and in communication between agents. We show that, for certain parameter values, the heavy ball algorithm monotonically converges to a minimizer, even under asynchrony. Key properties of the algorithm, including existence of solutions, convergence rate, and asymptotic stability, are analyzed. Numerical results validate the findings.

Dedicated to my husband, Kevin L. Schultz.

Acknowledgments

I am deeply appreciative of my advisor, mentor, and friend, Dr. Ricardo Sanfelice, for his guidance, kindness, and patience in helping me to grow into an independent and creative thinker, and in helping me strive to finish my PhD. Although I embarked upon this journey with much apprehension and self-doubt, you believed in me and have helped me to grow in my belief of myself.

I would like to thank my other committee members, Dr. Dejan Milutinovic and Dr. Matthew Hale. I appreciate your wisdom, the input you have offered on my dissertation at various times, and for your challenging questions and encouraging words.

I am thankful for my family and friends who helped support me during my graduate studies. I am thankful for my dad, Charles Hustig, for instilling a love and sense of wonder for the sciences in me when I was still very young. I owe a huge debt of gratitude to my husband and the love of my life, Kevin Schultz. Thank you for loving me and always believing in my potential, and for reminding me to have compassion for and to take care of myself during difficult times. I would also like to thank my cats Yuki and Yoshi for offering additional emotional support during my studies. Thank you to my friend Dr. Mynga Futrell, for encouraging me to persevere. Samira Zare, thank you for all the friendship, conversations, soccer, and movie watching.

I would like to thank my collaborators, Katherine Hendrickson and Paul Wintz, for all your help, insight, support, and kindness. May our collaborations continue to be fruitful and rewarding.

Thank you to my friends in the Hybrid Systems Laboratory the University of California, Santa Cruz: Jun Chai, Sean Phillips, Berk Altin, Hyejin Han, Mal-ladi BharaniPrabha, Nathalie Risso, Yegeta Zeleke, Marcello Guarro, Mohamed

Maghenem, Nan Wang, Haoyue Gao, Roger Berman, David Kooi, Parmita Ojaghi, Santiago Jiminez Leudo, Ryan Johnson, Masoumeh Ghanbarpour, Adam Ames, and Harsh Bhakta. You not only encouraged and supported me intellectually, but helped to make the lab feel like a second home. I hope that our friendships will continue into the future.

Chapter 1

Introduction and Motivation

1.1 Overview of the Work

Accelerated gradient methods have many applications in automation, such as controlling a single autonomous agent, multiagent systems [1], [2], wireless sensor or communication networks [2], [3], smart grids [4], [5], [6] transportation systems [4], [7], [8] and machine learning [2], [9] [10], to name a few examples. In the context of such applications, there has been growing interest in analyzing accelerated gradient methods from a dynamical systems perspective [11], [12]. A dynamical systems perspective permits the use of well established analysis tools, such as Lyapunov theory, to study convergence and stability properties of accelerated algorithms [13], [14], [15], [16], [17], [18], [19].

In this dissertation, we develop algorithms for global optimization with fast convergence, reduced oscillations, and robustness, using hybrid system tools, as described in [20], [21], [22]. In particular, as in [20], [21], [22], a *hybrid dynamical system* is a dynamical system which exhibits behavior characteristic of both continuous-time and discrete-time systems. It consists of a flow map, which is a set-valued map that governs the continuous change of state variables, a flow set,

which is a subset of the state space on which solutions can evolve continuously, a jump map, which is another set-valued map which governs the discrete change of state variables, and a jump set, which is a subset of the state space where jumps can occur.

Our approach to optimization incorporates both hybridity for performance and hybridity for nonconvexity. A class of optimization algorithms, referred to as *accelerated gradient methods*, involve a “velocity” term in addition to a gradient term, to speed up optimization. For certain parameter values of the coefficient of the gradient term, however, these types of algorithms can elicit solutions with oscillatory behavior as the system gets closer to a minimum of an objective function, L . Hybridity for performance for such algorithms can ensure fast convergence to the global minimum, without oscillations, by enabling the system to switch between two algorithms with different coefficient values: one which is more effective when the system is still far from the minimum, and one which is more effective when the system is close to the minimum. We extend hybridity for performance to a general framework for hybrid optimization, which can switch between two of any kind of gradient method.

One potential problem which can arise in optimization, when the objective function L has multiple isolated local minima and maxima (for example, L is a *Morse* function), is the possibility that the system starts at a local maximum ξ^* , when the velocity term equals zero. When this occurs, the system cannot converge to a minimum, since $\nabla L(\xi^*) = 0$. Hybridity for nonconvexity, in this case, can enable the system to detect that it is at a maximum, using hysteresis, and then push the system state term away from this maximum while still avoiding oscillations at the local minimum.

In the last part of this dissertation, we pivot from the hybrid system approach

to develop a totally asynchronous, block-based multiagent algorithm, based on the discrete-time version of the heavy ball method. We show that, for certain parameter values, the heavy ball algorithm monotonically converges to a minimizer, even under asynchrony. It is established that such an algorithm has an exponential convergence rate.

1.2 Uniting Heavy Ball Algorithms for Performance Improvement

The *heavy ball method* is an accelerated gradient method that guarantees convergence to the minimizer of a nonstrongly convex function L [23], and that achieves a faster convergence rate than classical gradient descent by adding a “velocity” term to the gradient. The dynamical system characterization for this method is

$$\ddot{\xi} + \lambda \dot{\xi} + \gamma \nabla L(\xi) = 0 \tag{1.1}$$

where λ and γ are positive tunable parameters that represent friction and gravity, respectively; see [14], [13]. This system resembles the dynamics of a particle sliding on a profile defined by L , with friction. In such a setting, the parameter λ represents the ratio between the viscous friction coefficient and the mass of the particle, and γ represents the gravity constant. The performance of the heavy ball method is highly dependent on λ and γ . Specifically, when λ is large, heavy ball converges very slowly, and when λ is small, heavy ball converges quickly, but with oscillations near the minimum [14].

1.2.1 Related Work

In [14], convergence is established for the continuous-time heavy ball method, both when L is convex and when L is a Morse function, but global asymptotic stability is not established. For the case when L is strongly convex, and inspired by the heavy ball algorithm, two algorithms with a resettable velocity term are proposed in [24] and shown to guarantee exponential convergence. In [25], however, it was demonstrated that the heavy ball algorithm converges exponentially for nonstrongly convex L when such an objective function also has the property of quadratic growth away from its minimizer.

Global asymptotic stability of the minimizer, which is the property that all solutions that start close to the minimizer stay close, and solutions from all initial conditions converge to the minimizer, is demonstrated in [1], [26], when L is nonstrongly convex and smooth. The work in [13] provides several Lyapunov functions to establish global asymptotic stability of the minimizer and convergence rates for the heavy ball method, both when L is strongly convex and when L is nonstrongly convex.

Contrary to classical gradient descent, accelerated gradient methods suffer from error accumulation. In [27], [28], and [29], it is suggested that the heavy ball method is sensitive to perturbations, due to its acceleration component. In [29], the effect of white noise on the discrete-time heavy-ball method is analyzed, and robustness to such noise is attained through the use of varying step-sizes. In [1], a perturbed continuous-time heavy ball system is analyzed and shown to be robust, but at the expense of the system measuring the Hessian of L . In [26], a continuous-time heavy ball with perturbations is also formulated and analyzed, where the system employs an observer to measure these perturbations.

1.2.2 Motivation

The performance of the heavy ball method, defined by the dynamical system in (1.1), depends highly on the choice of λ and γ . In particular, for a fixed value of γ , the choice of the “friction parameter” λ significantly affects the asymptotic behavior of the solutions to (1.1). For rather simple choices of the function L , the literature on this method indicates that large values of λ are seen to give rise to slowly converging solutions resembling solutions yielded by steepest descent while smaller values give rise to fast solutions with oscillations getting wilder as λ decreases [14]. The top and middle plots of Figure 1.1 demonstrates the behaviors of solutions with large values of λ and small values of λ , respectively¹. Such a compromise between damping the oscillations and converging fast motivates the logic-based algorithms proposed in this paper. Both algorithms select heavy ball with small λ to converge quickly to nearby the minimizer and, once solutions reach a neighborhood of the minimizer, switch to the heavy ball method with large λ to avoid oscillations. The first such algorithm uses measurements of L and ∇L and requires knowledge of L^* , while the second algorithm uses measurements of ∇L and does not require any knowledge of the minimizer. An example solution to our proposed logic-based algorithm, shown in the third plot from the top in Fig. 1.1, demonstrates the improvement obtained by using small λ globally and large λ locally, under relatively mild assumptions on the objective function L . The proposed algorithm guarantees UGAS and a (hybrid) convergence rate that holds for all hybrid time.

1.2.3 Contributions

The main contributions of the forthcoming Chapter 4 are as follows.

¹Code at github.com/HybridSystemsLab/UnitingMotivationHBF.

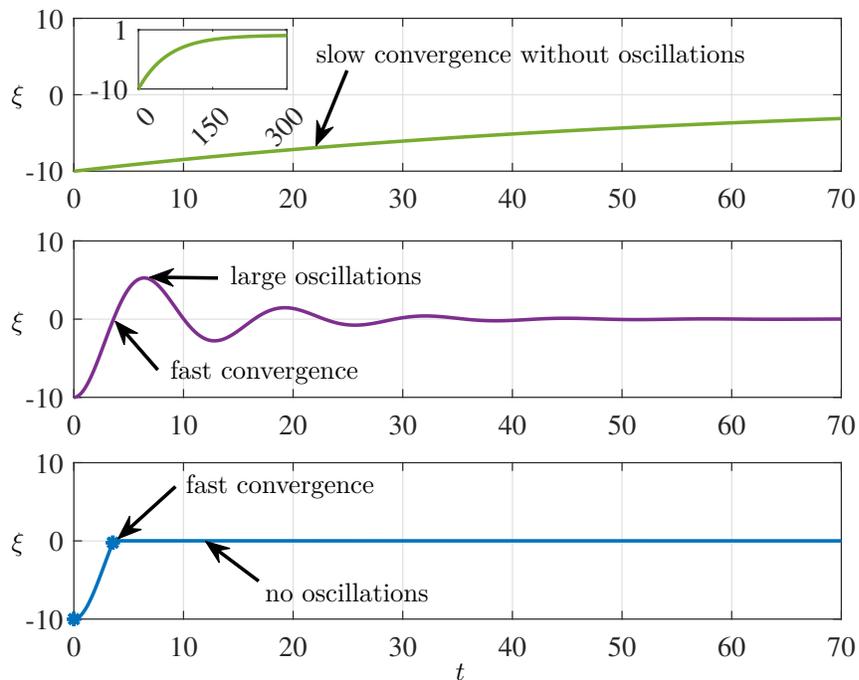


Figure 1.1: Comparison of the performance of the heavy ball method, with large and small values of λ , with the proposed logic-based algorithm for $L(\xi_1) = \frac{1}{4}\xi^2$. Top: when λ is large, heavy ball converges very slowly. Middle: when λ is small, heavy ball converges quickly, but with wild oscillations. Bottom: our proposed logic-based strategy yields fast convergence, with no oscillations.

- 1) *A uniting algorithm for fast convergence and UGAS of the minimizer:* In Section 4.2 we propose heavy ball control algorithms for optimization of a convex objective function L , with fast convergence and reduced oscillations. The algorithms utilize a uniting control strategy, developed using hybrid system tools (see Section 2.1), which switches between two standard heavy ball algorithms with different gravity and friction parameters. We design a hybrid control algorithm implementing a switching strategy that measures both L and its gradient (see Section 4.3), and then extend it to the case where it measures only the gradient of L (see Section 4.4). The algorithm in Section 4.3 requires no knowledge of ξ^* , but requires knowledge of $L^* := L(\xi^*)$. The algorithm in Section 4.4 requires no knowledge of L^* or ξ^* . Both algorithms require no measurements of the Hessian of L . UGAS of the minimizer ξ^* is guaranteed for both algorithms; see Sections 4.3.5 and 4.4.5.

- 2) *Well-posedness and existence of solutions:* In Sections 4.3.3 and 4.4.4 we prove well-posedness and in Sections 4.3.4 and 4.4.5 we prove existence of solutions for the proposed hybrid closed-loop algorithm. Hybrid systems that are *well-posed* are defined to be those hybrid systems, vaguely speaking, for which graphical limits of graphically convergent sequences of solutions, with no perturbations and with vanishing perturbations, respectively, are still solutions [21, Chapter 6]. It is important for our algorithm to be well-posed as we want to ensure robustness to small noise in measurements of the gradient of L .

- 3) *Robustness to small perturbations:* Due to the well-posedness of the proposed hybrid uniting algorithms, we show that the established UGAS property is robust to small perturbations in measurements of the gradient of L [21, Theorem 7.21]. We illustrate this robustness for the second algorithm in Section 4.4.7 via numerical simulations that include small noise in measurements of

the gradient.

- 4) *A (hybrid) convergence rate preserving the rate of the heavy ball method:* The algorithms proposed in Sections 4.3 and 4.4 have a (hybrid) convergence rate that preserves the rates of the individual optimization algorithms for all (hybrid) time. Specifically, we show that our algorithms attain a (hybrid) exponential convergence rate, both globally and locally.

1.3 Uniting Nesterov’s Method and the Heavy Ball Method for Performance Improvement

Another powerful accelerated gradient method is *Nesterov’s accelerated gradient descent*. Nesterov’s method is an accelerated method that guarantees convergence to the set of minimizers of a convex function L [30]. Nesterov’s algorithm achieves a faster convergence rate than classical gradient descent by adding a velocity term to the gradient.

1.3.1 Related Work

One characterization of the dynamical system for Nesterov’s algorithm for strongly convex L , proposed in [12], is

$$\ddot{\xi} + 2d\dot{\xi} + \frac{1}{M\zeta^2}\nabla L(\xi + \beta\xi) = 0, \tag{1.2}$$

where $M > 0$ is the Lipschitz constant for ∇L and where the constant $\zeta > 0$ rescales time in solutions to (1.2). The dynamical system in (1.3) resembles the model of a mass-spring-damper, with a curvature-dependent damping term where the total damping is a linear combination of d and β , which are time-invariant and

defined in a later section. In [12], the convergence rate for (1.2) is characterized as exponential when $\zeta = 1$ and when L is strongly convex.

Another characterization of the dynamical system for Nesterov's method, for nonstrongly convex L , proposed in [12], is

$$\ddot{\xi} + 2\bar{d}(t)\dot{\xi} + \frac{1}{M\zeta^2}\nabla L(\xi + \bar{\beta}(t)\dot{\xi}) = 0, \quad (1.3)$$

where $M > 0$ is the Lipschitz constant of the gradient of L and where the constant $\zeta > 0$ rescales time in solutions to (1.3). The dynamical system in (1.3) also resembles the model of a mass-spring-damper, with a curvature-dependent damping term where the total damping is a linear combination of $\bar{d}(t)$ and $\bar{\beta}(t)$, which are defined in a later section. In [12], the convergence rate of Nesterov's method is characterized as $\frac{1}{(t+2)^2}$ for (1.3) (for $t \geq 1$), when $\zeta = 1$, and when the minimizer is the origin, at which L is zero. In this dissertation, for both the strongly convex ODE in (1.2) and the nonstrongly convex ODE in (1.3), we relax such a constant to $\zeta > 0$. The work in [12] assumes that the set of interest for (1.2) and (1.3) is the origin, at which L is zero. The stability properties of (1.2) and (1.3) are not studied in [12].

Whereas [12] started with the ODEs in (1.2) and (1.3), and subsequently showed that the discrete-time analog of Nesterov's method arises from discretizing (1.3) with a semi-implicit Euler integration scheme, one of the earliest analyses of a dynamical system characterization for Nesterov's method, in [15], started with the discrete-time analog of Nesterov's method and showed that for a vanishing step size the trajectories of such an accelerated gradient scheme approach the solutions of the ODE

$$\ddot{\xi} + \frac{3}{t}\dot{\xi} + \nabla L(\xi) = 0 \quad (1.4)$$

for all $t > 0$. Such an ODE does not have a local curvature dependent damping term, as (1.2) and (1.3) do, and which [12] argues is instrumental to the intuition behind the acceleration phenomenon. The development in [15] includes the analysis of a variation of the dynamical system in (1.4) for higher friction, and show that their dynamical system characterizations have a convergence rate of $\frac{1}{t^2}$. In [16], the analysis of the dynamical system in [15] is extended to include optimization of objective functions L with non-Euclidean geometries, using a Bregman divergence to characterize the distance of the state ξ from the minimizer. The work in [16] combines this dynamical system with mirror descent to design an accelerated mirror descent ODE, with a convergence rate of $\frac{1}{t^2}$. In [17], a dynamical system, consisting of an Euler-Lagrange equation, is derived for Nesterov’s algorithm via a Bregman Lagrangian. In [17] an exponential rate of convergence for such a system under ideal scaling is provided, and, for a polynomial class of dynamical systems, a convergence rate of $\frac{1}{t^p}$ with $p \geq 2$ is shown.

In [31] and [19], two hybrid algorithms based on the ODE in (1.4) are presented: one with a state-dependent, time-invariant damping input and another with an input that controls the magnitude of the gradient term. The algorithms require the objective function to satisfy the Polyak-Łojasiewicz inequality, which includes a subclass of nonconvex functions in which all stationary points are global minimizers. The authors in [32] propose two hybrid reset algorithms based on the ODE in (1.4), HAND-1 and HAND-2, which yield an exponential convergence rate for strongly convex L and a rate of $\frac{1}{t^2}$ for nonstrongly convex L , with the latter rate only assured until the first reset.

While the results in [12], [15], [16], [17], [31], and [19] characterize the convergence properties of Nesterov’s method (or a variation of), the stability properties of the method are not revealed. While stability properties for such methods were

studied in [13], a particularly useful property for optimization algorithms, called *uniform global asymptotic stability* (UGAS), requires that solutions reach a neighborhood of the minimizer in time that is uniform on the set of initial conditions. After finite time, the error of such solutions becomes smaller than a given threshold [22]. Due to such a guarantee for solutions, UGAS is typically useful for certifying robustness to small perturbations in time-varying dynamical and hybrid systems [21], [22]. Remarkably, the algorithms with resets in the velocity term proposed in [24] and [33] can be shown to induce UGAS of the minimizer (with zero velocity term) and reduced oscillations, for the particular case when L is strongly convex. Unfortunately, as shown in [32], via a counterexample, Nesterov-like algorithms do not necessarily assure UGAS of the minimizer when L is nonstrongly convex. In response to this, [32] proposes the HAND-1 and HAND-2 reset algorithms, and prove UGAS of the minimizer for both algorithms. The exponential convergence rate of HAND-2, however, only applies to strongly convex L , and the convergence rate of $\frac{1}{t^2}$ for HAND-1, for nonstrongly convex L , only holds up until the first reset.

1.3.2 Motivation

The work in this dissertation is motivated by the lack of an accelerated gradient algorithm assuring UGAS, with a convergence rate that holds for all time and that resembles that of Nesterov’s method (at least far from the minimizer), when the objective function is nonstrongly convex. However, attaining such a rate is expected to lead to oscillations, which are typically seen in accelerated gradient methods. As described in Section 1.2.2, the performance of the heavy ball method, for instance, depends highly on the choice of λ when γ is fixed, with large values of λ resulting in slowly converging solutions resembling solutions yielded by steepest

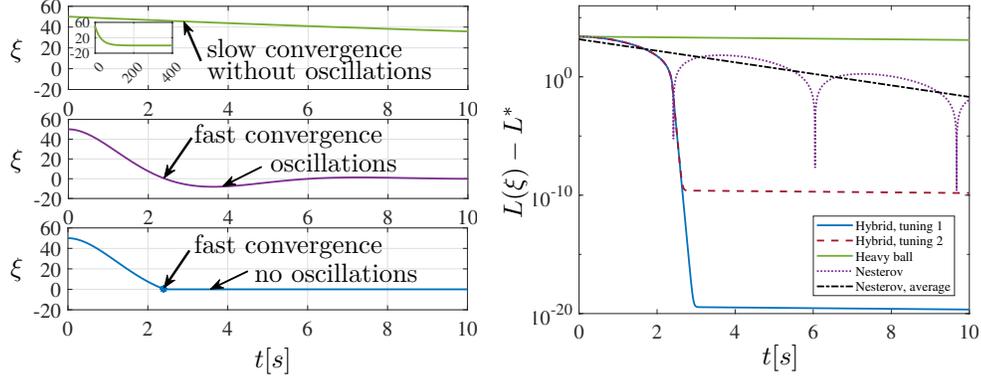


Figure 1.2: Comparison of the performance of the heavy ball method, with large λ , Nesterov’s accelerated gradient descent, and the proposed logic-based algorithm, for strongly convex L . The objective function is $L(\xi) = \xi^2$. Top left: the heavy ball algorithm, with large λ , converges very slowly. Top inset: zoomed out view of heavy ball. Middle left: Nesterov’s accelerated gradient descent converges quickly, but with oscillations. Bottom left: our proposed logic-based algorithm yields fast convergence, with no oscillations. Right: comparison of the value of $L(\xi) - L^*$ (in log scale) versus time for each algorithm. Different tunings of the logic-based algorithm’s parameters leads to modifications of the solution’s profile.

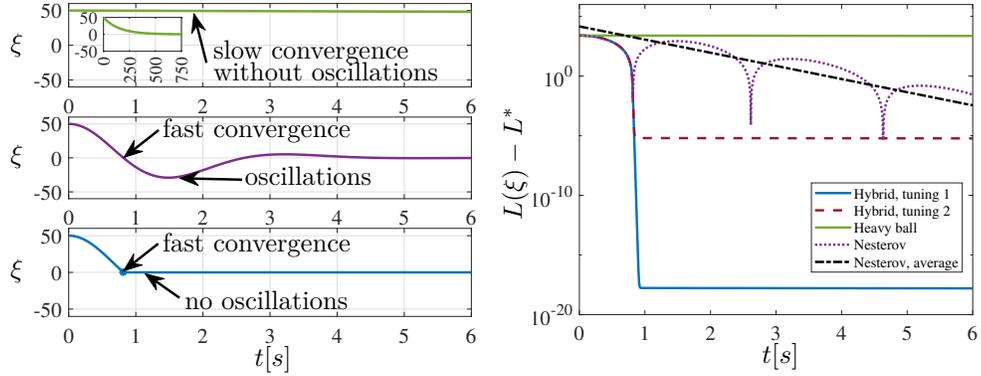


Figure 1.3: Comparison of the performance of the heavy ball method, with large λ , Nesterov’s accelerated gradient descent, and the proposed logic-based algorithm, for nonstrongly convex L . The objective function is $L(\xi) = \xi^2$. Top left: the heavy ball algorithm, with large λ , converges very slowly. Top inset: zoomed out view of heavy ball. Middle left: Nesterov’s accelerated gradient descent converges quickly, but with oscillations. Bottom left: our proposed logic-based algorithm yields fast convergence, with no oscillations. Right: comparison of the value of $L(\xi) - L^*$ (in log scale) versus time for each algorithm. Different tunings of the logic-based algorithm’s parameters leads to modifications of the solution’s profile.

descent [14], as in the top plot² on the left in Figure 1.2 and the top plot³ on the left in Figure 1.3, and with smaller values of λ resulting in fast solutions with oscillations [14], as shown in the middle plot on the left in Figures 1.2 and 1.3. Nesterov’s method converges quickly but also suffers from oscillations [15]. The oscillatory behavior of Nesterov’s method for strongly convex L in (1.2), with $\zeta = 1$, is shown in the middle plot on the left in Figure 1.2. The oscillatory behavior of Nesterov’s method for nonstrongly convex L , with $\zeta = 2$, is shown in the middle plot on the left in Figure 1.3.

Due to its implications on robustness, we are particularly interested in an algorithm that assures uniform global asymptotic stability of the minimizer of L with a rate of convergence that holds for all time, and without the undesired oscillations. As pointed out in Section 1.3.1, these properties are not guaranteed by Nesterov’s method. The behavior shown in the top and middle plots in Figure 1.3 motivates the logic-based algorithm proposed in this dissertation. The proposed algorithm exploits the main features of heavy ball and Nesterov’s method to achieve fast convergence and UGAS of the minimizer. More precisely, without knowledge of the location of the minimizer, it selects Nesterov’s method to converge quickly to nearby the minimizer and, once solutions reach a neighborhood of the minimizer, switches to the heavy ball method with large λ to avoid oscillations. An example solution to our proposed logic-based algorithm for strongly convex L , shown in the bottom plot on the left of Figure 1.2, demonstrates the improvement obtained by using Nesterov’s method globally and the heavy ball method locally, under relatively mild assumptions on the strongly convex objective function L . The proposed algorithm guarantees UGAS and a (hybrid) convergence rate that holds for all hybrid time. An example solution to our proposed logic-based algorithm for

²Code at github.com/HybridSystemsLab/UnitingMotivationSC.

³Code at [gitHub.com/HybridSystemsLab/UnitingMotivation](https://github.com/HybridSystemsLab/UnitingMotivation).

nonstrongly convex L , shown in the bottom plot on the left of Figure 1.3, demonstrates the improvement obtained by using Nesterov’s method globally and the heavy ball method locally, under relatively mild assumptions on the nonstrongly convex objective function L . The proposed algorithm guarantees UGAS and a (hybrid) convergence rate that holds for all $t \geq 0$.

1.3.3 Contributions For Strongly Convex L

The main contributions of Section 5.1 are as follows.

- 1) *A uniting algorithm for fast convergence and UGAS of the minimizer:* In Section 5.1.2, we propose a uniting algorithm, designed using hybrid system tools (see Section 2.1), 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 minimizer of L , without knowledge of $L^* := L(\xi^*)$ or ξ^* ; see Section 5.1.9. 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 .
- 2) *Well-posedness and existence of solutions:* In Section 5.1.7 we prove well-posedness and in Section 5.1.8 we prove existence of solutions for the proposed hybrid closed-loop algorithm. Hybrid systems that are *well-posed* are defined to be those hybrid systems, vaguely speaking, for which graphical limits of graphically convergent sequences of solutions, with no perturbations and with vanishing perturbations, respectively, are still solutions [21, Chapter 6]. It is important for our algorithm to be well-posed as we want to ensure robustness to small noise in measurements of the gradient of L .

- 3) *Robustness to small perturbations:* Due to the well-posedness of the proposed hybrid uniting algorithm, we show that the established UGAS property is robust to small perturbations in measurements of the gradient of L [21, Theorem 7.21]. We illustrate this robustness in Section 5.1.10 via numerical simulations that include small noise in measurements of the gradient.
- 4) *A (hybrid) convergence rate preserving the rates of Nesterov’s method and heavy ball:* The algorithm proposed in Section 5.1 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, both globally and locally, when L is strongly convex.
- 5) *Extension of the results on Nesterov’s method in [12]:* In the process, in Section 3.1.1, we extend the properties and convergence results for Nesterov’s method in [12]. In particular, while the convergence rate results in [12] assume that $L(\xi_1^*) = 0$ at $\xi^* = 0$ for (1.2), here we prove uniform global asymptotic stability (UGAS) of the minimizer, with an exponential convergence rate, for cost functions with a minimum value that is not necessarily zero. As in [12], however, we set $\zeta = 1$ in (1.2) for simplicity of analysis.

1.3.4 Contributions For Nonstrongly Convex L

The main contributions for Section 5.2 are as follows.

- 1) *A uniting algorithm for fast convergence and UGAS of the minimizer:* In Section 5.2 we propose a uniting algorithm that solves optimization problems of the form $\min_{\xi \in \mathbb{R}^n} L(\xi)$ with accelerated gradient methods. Designed using hybrid system tools (see Section 2.1), the algorithm unites Nesterov’s method in (1.5) globally and the heavy ball method in (1.1) with large λ locally to guar-

antee fast convergence with UGAS of the minimizer ξ^* of a nonstrongly convex objective function L ; see Sections 5.2.2 and 5.2.8. The establishment of UGAS solves the difficult problem of achieving such a property for Nesterov-like algorithms [32], [35]. The algorithm we propose exploits measurements of ∇L and requires no knowledge of $L^* := L(\xi^*)$ or ξ^* . In practice, such measurements of ∇L are typically approximated from measurements of L . The algorithm, however, does not require measurements of the Hessian of L .

- 2) *Well-posedness and existence of solutions:* In Section 5.2.6 we prove well-posedness and in Section 5.2.7 we prove existence of solutions for the proposed hybrid closed-loop algorithm. Hybrid systems that are *well-posed* are defined to be those hybrid systems, vaguely speaking, for which graphical limits of graphically convergent sequences of solutions, with no perturbations and with vanishing perturbations, respectively, are still solutions [21, Chapter 6]. It is important for our algorithm to be well-posed as we want to ensure robustness to small noise in measurements of the gradient of L .
- 3) *Robustness to small perturbations:* Due to the well-posedness of the proposed hybrid uniting algorithm, we show that the established UGAS property is robust to small perturbations in measurements of the gradient of L [21, Theorem 7.21]. We illustrate this robustness in Section 5.2.9 via numerical simulations that include small noise in measurements of the gradient.
- 4) *A (hybrid) convergence rate preserving the rates of Nesterov's method and heavy ball:* In Section 5.2.8 we show that our uniting algorithm attains a rate of $\frac{1}{(t+2)^2}$ for the global algorithm and $\exp(-(1-m)\psi t)$, where $m \in (0, 1)$ and $\alpha > 0$ are such that $\psi := \frac{m\alpha\gamma}{\lambda} > 0$ and $\nu := \psi(\psi - \lambda) < 0$, for the local algorithm. The latter rate holds under the mild assumption on L of quadratic growth away

from the minimizer. As mentioned in Section 1.3.1, Nesterov-like algorithms do not necessarily assure UGAS of the minimizer. The HAND-1 algorithm for nonstrongly convex L , proposed in [32], provides UGAS via a hybrid restarting mechanism that yields a convergence rate $\frac{1}{t^2}$. However, this convergence rate holds only until the first reset. The algorithm we propose not only renders the minimizer UGAS, but also has a (hybrid) convergence rate that preserves the rates of the individual optimization algorithms for all (hybrid) time such that $t \geq 0$. Moreover, the global rate of our algorithm is commensurate with that of HAND-1. In Figure 1.4 and Section 5.2.9, our uniting algorithm is shown via numerical simulations⁴ to have improved performance over the HAND-1 algorithm in [32].

- 5) *Extension of the results on Nesterov’s method in [12]:* In the process, in Section 3.1.2, we extend the properties and convergence results for Nesterov’s method in [12]. In particular, while the convergence rate results in [12] assume that $L(\xi_1^*) = 0$ at $\xi^* = 0$, and $\zeta = 1$ for (1.3), here we prove uniform global asymptotic stability of the minimizer, with a convergence rate of $\frac{1}{(t+2)^2}$ for all $t \geq 0$, for cost functions with a minimum value that is not necessarily zero, which holds for a generic parameter $\zeta > 0$. We achieve the relaxation on ζ by moving it into the numerator of the coefficient of the gradient, effectively decoupling ζ and M . This leads to the ODE

$$\ddot{\xi} + 2\bar{d}(t)\dot{\xi} + \frac{\zeta^2}{M}\nabla L(\xi + \bar{\beta}(t)\dot{\xi}) = 0. \quad (1.5)$$

Such a modification leads to faster convergence as ζ increases, and slower convergence as $\zeta \rightarrow 0$.

⁴Code at github.com/HybridSystemsLab/UnitingTradeoff.

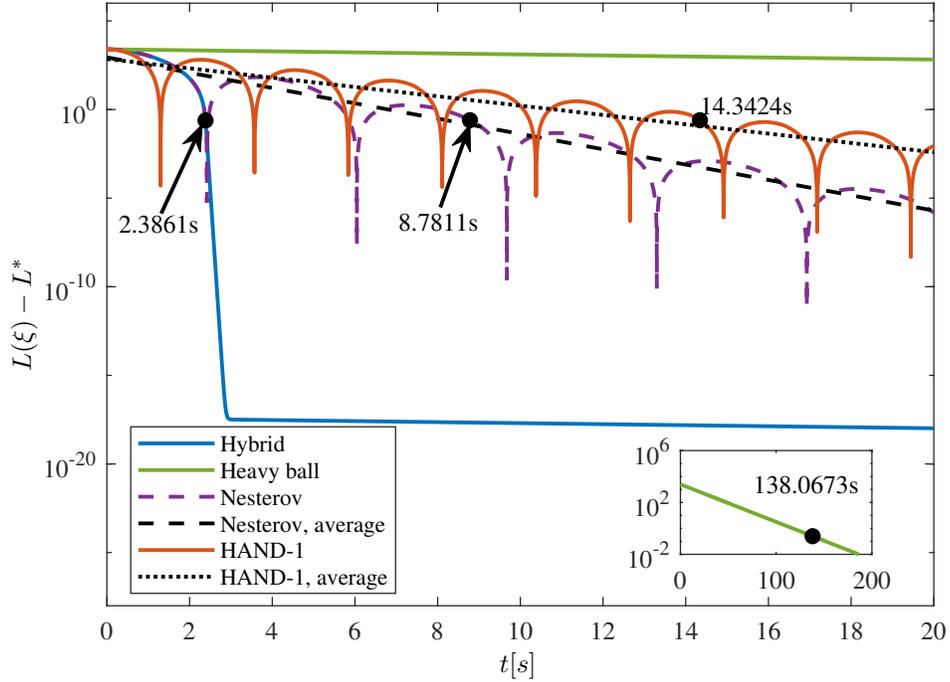


Figure 1.4: A comparison of the evolution of L over time for Nesterov’s method in (1.5), heavy ball, HAND-1, and our proposed uniting algorithm, for a function $L(\xi) := \xi^2$, with a single minimizer at $\xi^* = 0$. Nesterov’s method, shown in purple, settles to within 1% of ξ^* in about 8.8 seconds. The heavy ball algorithm, shown in green, settles to within 1% of ξ^* in about 138.1 seconds. HAND-1, shown in orange, settles to within 1% of ξ^* in about 14.3 seconds. The hybrid closed-loop system \mathcal{H} , shown in blue, settles to within 1% of z_1^* in about 2.4 seconds. As opposed to Figure 1.3, which uses $\zeta = 2$ for \mathcal{H}_1 , this example uses $\zeta = 1$, which results in slower convergence of solutions to \mathcal{H} and \mathcal{H}_1 than in Figure 1.3.

1.4 A Uniting Framework for Performance Improvement

We propose a general framework for algorithms that solve optimization problems of the form

$$\min_{\xi \in \mathbb{R}^n} L(\xi) \tag{1.6}$$

with gradient methods. The proposed framework, designed using hybrid system tools, utilizes a more general version of the uniting strategy discussed in Sections 1.2 and 1.3.

1.4.1 Related Work

Frameworks for the analysis and design of algorithms have been proposed in the past. In [19] the two hybrid algorithms based on the ODE in (1.4), discussed in Section 1.3.1, are presented as a framework, involving a hybrid system model with similar flow and jump maps for each algorithm. An exponential convergence rate is established for this framework. In [18], an analysis framework is proposed for a family of Euler-Lagrange ODEs for accelerated optimization, and exponential rates of convergence are established within this framework. In [36] a Hamiltonian-based framework is proposed, to generalize Nesterov’s accelerated gradient descent and Polyak’s heavy ball method to a broad class of momentum methods in the setting of (possibly) constrained minimization in Euclidean and non-Euclidean normed vector spaces. Convergence of the continuous-time dynamics is established, and the resulting discretized class of methods converges at a rate of $\frac{1}{k^2}$ for the Nesterov-like algorithms and at a rate of $\frac{1}{k}$ for the heavy ball-like algorithms.

While the results for the frameworks in [19], [18], [36] characterize the con-

vergence properties of accelerated gradient methods, the stability properties of such methods are not revealed. In [37] a framework allowing for four gradient-free accelerated optimization algorithms is proposed, for optimization problems of the following types: unconstrained nonstrongly convex, unconstrained strongly convex, strongly convex with linear equality constraints, and strongly convex with inequality constraints. The resulting nonstrongly convex algorithm has the set of interest semiglobally practically asymptotically stable, and the resulting strongly convex algorithms have the set of interest semiglobally practically exponentially stable.

1.4.2 Motivation

Sections 1.2.2 and 1.3.2 discussed the motivation for designing logic-based algorithms that assure UGAS of the minimizer of L with a hybrid rate of convergence that holds for all time, without the undesired oscillations. Figure 1.1 illustrates the benefit of a logic-based algorithm uniting two heavy ball algorithms with properly designed λ and γ . Figure 1.2 illustrates the benefit of a logic-based algorithm, for strongly convex L , uniting Nesterov’s algorithm globally and the heavy ball algorithm with large λ locally. Figure 1.3 illustrates the benefit of a logic-based algorithm, for nonstrongly convex L , uniting Nesterov’s algorithm globally and the heavy ball algorithm with large λ locally. Such a pattern suggests the potential to design a general framework for uniting local and global accelerated gradient methods.

1.4.3 Contributions

- 1) *A general uniting framework for gradient methods:* In Section 6.2.1, we propose a general framework, designed using hybrid system tools (see Section 2.1),

for uniting local and global optimization algorithms, which allows either the local or global algorithms to be any gradient method, including Nesterov’s algorithm, the heavy ball algorithm, classic gradient descent, or the triple momentum method [38] [39].

- 2) *Sufficient conditions for UGAS:* In Section 6.2.3, we determine sufficient conditions leading to general results on well-posedness, existence of solutions, and uniform global asymptotic stability for the hybrid closed-loop framework.
- 3) *Robustness to small perturbations:* Due to the well-posedness of the proposed hybrid uniting framework, we can show that the uniform global asymptotic stability that we guarantee is robust to small perturbations [21, Theorem 7.21]. Such a property is illustrated for special cases in Sections 4.4.7, 5.1.10, and 5.2.9.
- 4) *Examples of applying the framework:* In Section 6.3, we show special cases of uniting algorithms in Sections 1.2 and 1.3 satisfy the basic properties of the uniting framework.

1.5 Hybrid Optimization for Nonconvex Problems

In the forthcoming Chapter 7, we consider the problem of finding a local minimizer of a scalar, continuously differentiable objective function L with a single, scalar argument, which is not necessarily convex, and may have multiple local minimizers. In particular, we are interested in algorithms capable of solving optimization problems of the form

$$\min_{\xi \in \mathbb{R}} L(\xi), \tag{1.7}$$

with a guarantee of *global* attractivity of the set of minimizers. By *global*, we mean “from any initial condition (or guess).” This is different from the typical use of the term global in the optimization literature, which corresponds to the guarantee that an optimization algorithm converges to the global minimizer rather than to a local minimizer. In fact, the objective functions considered in this section may have multiple isolated critical points, which are known to impose challenges to optimization algorithms.

For the type of nonconvex optimization problem in which we are interested, and approaching the problem from a control theory viewpoint, it is infeasible to design an algorithm of the form

$$\dot{\xi} = f(\xi, \nabla L(\xi)), \tag{1.8}$$

that solves the problem with attractivity and robustness when small measurement noise exists in measurements of the gradient. This infeasibility suggests the need of an algorithm that is robust to measurement noise. Such an algorithm would detect when the state ξ is close to a local maximum, and then implement a strategy that moves the state away from that maximum. Instead of algorithms of the form $\dot{\xi} = f(\xi, \nabla L(\xi))$, we propose an algorithm conveniently modeled and designed using hybrid system tools, based on the heavy ball method in (1.1), for convergence to a local minimum of a nonconvex Morse objective function L .

1.5.1 Related Work

To the best of our knowledge, we propose the first algorithm based on the heavy ball method for which the set of minimizers of a nonconvex objective function L , with a single, scalar argument, is practically globally attractive, and for which we observe robustness to small noise in simulation. In contrast, the previous literature

establishes only the convergence rate for the heavy ball method. In particular, the heavy ball method was first analyzed in a nonconvex setting in [40]. In [14], the convergence bounds for the heavy ball method, when L is a Morse function, are derived. In [41], almost sure convergence on nonconvex objective functions is proved for a stochastic heavy ball algorithm, but the properties of stability and robustness to arbitrarily small noise were not addressed.

Many inertial forward-backward (FB) optimization methods incorporate the heavy ball method, and are commonly used to solve nonconvex optimization problems. Examples of FB algorithms, including [42], [43], and [44], have established convergence rates of these algorithms to a local minimum, but have not demonstrated whether these algorithms render the set of minimizers globally asymptotically stable, or whether these algorithms are robust to small noise in measurements of the gradient.

There has been a surge of interest in utilizing hybrid systems tools for gradient-based optimization. In [45], a hybrid gradient descent algorithm using an adjustable diffeomorphism is proposed, to ensure global asymptotic stability to the minimum of a compact manifold that is a circle. This result is then extended to manifolds with an equal number of maxima and minima, and then a model-free version of the algorithm is proposed. In [46], a class of hybrid stochastic gradient descent algorithms is proposed, to solve nonconvex optimization problems on smooth manifolds. Uniform global asymptotic stability in probability is established, and then such results are extended to a partially multiagent setting. In [31] and [19], two hybrid algorithms based on Nesterov's accelerated gradient descent are proposed: one with a state-dependent, time-invariant damping input and another with an input that controls the magnitude of the gradient term. The algorithms require the objective function to satisfy the Polyak-Łojasiewicz inequality,

which includes a subclass of nonconvex functions in which all stationary points are global minimizers. Although an exponential convergence rate is established in [19] for these two algorithms, the attractivity properties of such algorithms are not explored.

1.5.2 Motivation

As mentioned in Section 1.5, it is infeasible to design an algorithm of the form (1.8) that solves nonconvex optimization problems of the form (1.7) with attractivity and robustness. To illustrate this point, consider the function L given by $L(\xi) = \frac{\xi^2(\xi-10)^2(\xi-20)^2(\xi-30)^2}{10,000}$ for each $\xi \in \mathbb{R}$, for which each $\xi \in \{0, 10, 20, 30\}$ is a local minimizer and each $\xi \in \{5(3 - \sqrt{5}), 15, 5(3 + \sqrt{5})\}$ is a local maximizer. Classic gradient descent, which corresponds to $f(\xi, \nabla L(\xi)) = -\nabla L(\xi)$, does not render the set of minimizers of this function globally attractive, since when the state ξ starts at a local maximizer and the initial value of $\dot{\xi}$ is zero, we have that ∇L is zero and the algorithm remains stuck at such a local maximizer. Moreover, when the state ξ starts close to the local maximizer and there is small noise added to the measurements of the gradient, then the algorithm cannot always push ξ away from the maximizer, even when the noise signal is arbitrarily small. This can be seen in the top left plot of Figure 1.5, where arbitrarily small noise in the gradient keeps the state close to the local maximizer of L at $\xi = 15$.⁵

Algorithms of the form (1.8) with a static, discontinuous map f , for which the nominal system has the set of minimizers of L globally asymptotically stable, are not robust to arbitrarily small measurement noise. Such a system is not well-posed⁶, due to discontinuities in the map f , at local maximizers. In fact,

⁵Code at github.com/HybridSystemsLab/RobustnessHeavyBall.

⁶For a purely continuous-time algorithm, well-posed means that solutions depend “continuously” with respect to initial conditions.

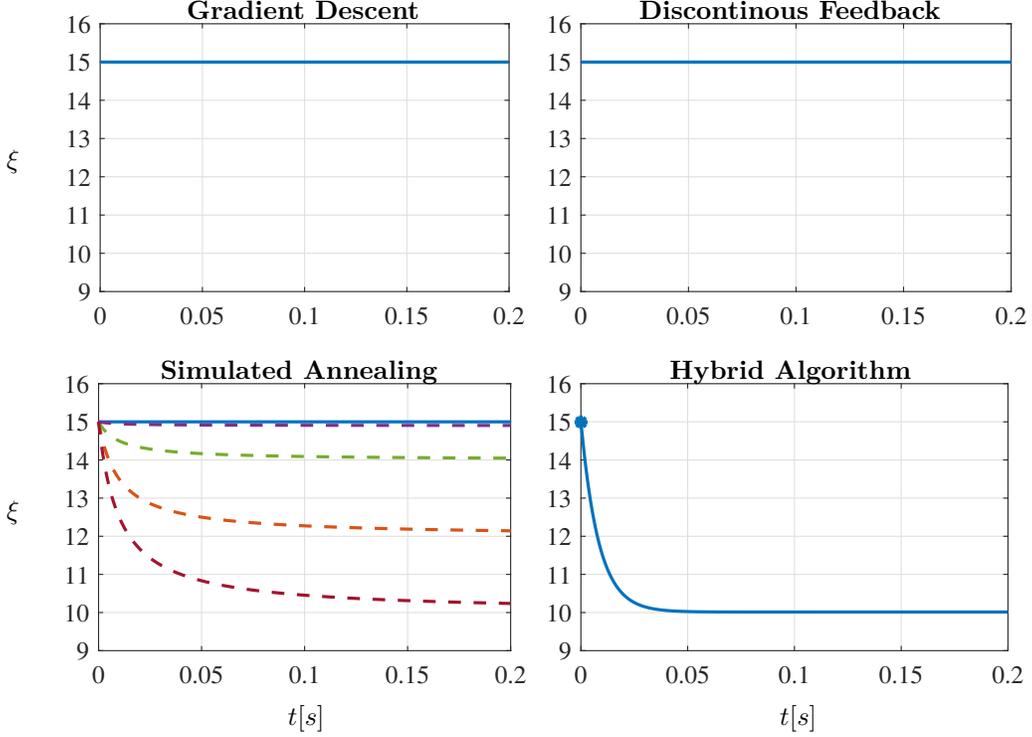


Figure 1.5: Comparing performance of the proposed hybrid algorithm to other optimization methods, with small noise in measurements of the gradient, when the system starts near a local maximum, at $\xi_0 \approx 15$. For classic gradient descent (top left), a gradient-based optimization algorithm with discontinuous map f (top right), and simulated annealing, via Langevin diffusion (bottom left) the state ξ get pushed to the local maximum at $\xi = 15$, and stays there. All trajectories in the bottom left plot have the noise signal $\vartheta := \left(-\frac{\tau(\log(\tau))^2}{B_{SA}}\right) (\nabla L(\xi) + \Omega \text{sign}(\nabla L(\xi))(10^{-12}))$, where $\tau > 0$ is time, $B_{SA} > 0$ is large, and Ω is a normally distributed random number. The trajectory with red dotted line converging to the minimizer has an added constant of $B_\vartheta = 5 \times 10^{-13}$, such that $\vartheta + B_\vartheta$, while the other three trajectories represented by dashed lines have added constants B_ϑ equal to 3×10^{-13} , 10^{-13} , and 10^{-14} , respectively. The last trajectory, represented by the solid blue line, has no constant added to the noise signal ϑ . The hybrid algorithm (bottom right), with noise of the form $\left(-\frac{\tau(\log(\tau))^2}{B_{SA}}\right) (\nabla L(\xi) + \Omega \text{sign}(\nabla L(\xi))(10^{-12}))$ added to the gradient of L , where Ω is a normally distributed random number, is still able to escape the local maximum at $\xi = 15$ and converge to a local minimum at $\xi = 10$.

when the state ξ starts close to one of the points of discontinuity, and when small noise is added to the measurements of the gradient, there will exist a solution that remains nearby such a point, even when the noise is arbitrarily small. The limit of such a solution as the noise goes to zero is a solution to the differential inclusion $\dot{\xi} \in F(\xi, \nabla L(\xi))$, where F is the *Krasovskii regularization* of $\xi \mapsto f(\xi, \nabla L(\xi))$. Such a solution, when the right-hand side is bounded, is also a Hermes solution [21, Theorem 4.3], and represents an equilibrium point of $\dot{\xi} \in F(\xi, \nabla L(\xi))$, from which the state ξ cannot converge to a local minimizer. Therefore, the Krasovskii regularization does not have the set of minimizers of L globally attractive. According to [21], the attractivity of the original system $\dot{\xi} = f(\xi, \nabla L(\xi))$ with f discontinuous is not robust. This behavior can be seen in the top right plot of Figure 1.5, where arbitrarily small noise induces an equilibrium point at the maximizer located at $\xi = 15$, at which $f(\xi, \nabla L(\xi))$ is discontinuous.

Simulated annealing [47], via Langevin diffusion, is a popular alternative used to find the global minimizer of a nonconvex function. Langevin diffusion, which corresponds to $\dot{\xi} = -\nabla L(\xi) + B(t)\vartheta(t)$ combines classic gradient descent with a noise signal ϑ , such as Brownian motion, for which the magnitude is controlled by the “temperature” function B . Although such a noise signal is used to help the state find the global minimum, it can also be detrimental to performance. It can be shown that when the state ξ starts close to a local maximizer the algorithm cannot always push ξ away from the maximizer, due to this noise signal, no matter how large the initial annealing temperature is. This is even the case when the noise is arbitrarily small. This behavior is shown by the solid line in the bottom left plot of Figure 1.5, where noise keeps the state ξ close to the local maximizer at $\xi \approx 15$. The dashed lines show the effect of adding a small constant to the noise, which causes the state ξ to drift away from the local maximum, and eventually

converge to a local minimum. Essentially, as the size of such a small constant decreases, the more likely simulated annealing is to be stuck at a local maximizer.

The issues depicted in the top left, top right, and bottom left of Figure 1.5 show that nonconvex optimization problems cannot be efficiently solved with existing line search algorithms or stochastic algorithms. On the contrary, Figure 1.5 demonstrates the need of an algorithm, modeled and designed using hybrid system tools, that in simulation demonstrates robustness to measurement noise. Its performance is shown in the bottom right of Figure 1.5, starting at $\xi \approx 15$ with zero velocity, and converging despite the presence of noise in measurements of the gradient, as is present for the other algorithms in Figure 1.5.

1.5.3 Contributions

The main contributions of the forthcoming Chapter 7 are as follows.

- 1) *A hybrid algorithm for nonconvex optimization:* In Section 7.2, we develop an optimization algorithm, based on the heavy ball method, for convergence to a local minimum of a nonconvex Morse objective function L with a single, scalar argument. We emphasize that our proposed algorithm is not designed to find all the local minimizers, but rather to converge to an element in the set of local minimizers. The algorithm employs a switching strategy, developed using hybrid system tools (see Section 2.1), to detect whether the state ξ is near a critical point and ensure escape from a local maximizer, depicted in Figure 1.5. Such a switching strategy employs measurements of the gradient of L – which in practice are typically approximated from measurements of L – and hysteresis to determine whether the state ξ needs to be pushed away from a nearby critical point, or whether the state ξ is far enough away from a critical point to resume use of the heavy ball method. The algorithm does

not need to distinguish between local maximizers and local minimizers, and therefore does not need information about the Hessian.

- 2) *Well-posedness and existence of solutions:* In Section 7.4 we prove well-posedness and existence of solutions for the proposed hybrid closed-loop algorithm. Hybrid systems that are *well-posed* are defined to be those hybrid systems, vaguely speaking, for which graphical limits of graphically convergent sequences of solutions, with no perturbations and with vanishing perturbations, respectively, are still solutions [21, Chapter 6]. It is important for our algorithm to be well-posed as we want to ensure robustness to small noise in measurements of the gradient of L .
- 3) *Practical global attractivity of the set of minimizers of L :* In Section 7.4, we establish practical global attractivity of the set of minimizers of L for the closed-loop system.
- 4) *Robustness to small perturbations:* Due to the well-posedness of the proposed hybrid uniting algorithm, we show that the established practical global attractivity property is robust to small perturbations in measurements of the gradient of L [21, Theorem 7.21]. We illustrate this robustness in Figure 1.5 via numerical simulations that include small noise in measurements of the gradient.

1.6 A Totally Asynchronous, Block-Based Heavy Ball Algorithm for Convex Optimization

We are interested in asynchronously solving problems of the form

$$\begin{aligned} & \text{minimize } f(\xi) \\ & \text{subject to } \xi \in X \end{aligned} \tag{1.9}$$

using the discrete-time version of the heavy ball method. Given $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $X \subset \mathbb{R}^n$, the constrained heavy ball algorithm is defined as follows:

$$\xi(k+1) = \Pi_X [\xi(k) - \gamma \nabla f(\xi(k)) + \lambda(\xi(k) - \xi(k-1))] \tag{1.10}$$

for each $k \in \{1, 2, 3, \dots\} \subset \mathbb{N}_{\geq 0}$, where $\xi(k) \in \mathbb{R}^n$ is the state of the algorithm at discrete time k , $\Pi_X[v] = \arg \min_{w \in X} |w - v|$ is the orthogonal projection of a vector v onto the convex set X with respect to the Euclidean norm, and $\lambda > 0$ and $\gamma > 0$ are tunable parameters representing friction and gravity, respectively; see [23]. We interpret (1.10) as a control system consisting of a plant and a control algorithm. Let $z_1 := \xi(k)$, $z_2 := \xi(k-1)$, and $z := (z_1, z_2)$. Then, the plant associated with (1.10) is given by

$$\begin{bmatrix} z_1^+ \\ z_2^+ \end{bmatrix} = \begin{bmatrix} u \\ z_1 \end{bmatrix} =: G_P(z, u) \quad (z, u) \in X \times X \times \mathbb{R}^n =: D_P \tag{1.11}$$

with output $y = z$. The control algorithm leading to (1.10) is given by

$$u = \kappa(z) := \Pi_X [z_1 - \gamma \nabla f(z_1) + \lambda(z_1 - z_2)]. \tag{1.12}$$

1.6.1 Related Work

In [9], it is shown that the unconstrained heavy ball method converges exponentially when f is strongly convex and converges with rate $\frac{1}{k}$ when f is nonstrongly convex. A convergence rate of $\frac{1}{(k+1)^2}$ for the constrained heavy ball method is shown in [48] for convex, Lipschitz continuous functions, with some additional assumptions on the parameters λ and γ . In [49], it is shown that the constrained heavy ball method converges exponentially for nonsmooth, strongly convex functions.

Since we are interested in asynchronously solving large-scale convex problems – such as problems found in machine learning or consisting of coordinating large numbers of autonomous agents – one challenge involves reducing potential disagreements between agents, resulting from generating and communicating different information at different times [50]. One approach to reducing disagreements, dating back several decades, consists of repeated averaging of the agents’ iterates [51]. Examples of multiagent algorithms with an averaging-based approach and utilizing accelerated gradient methods – such as the heavy ball method and Nesterov’s accelerated gradient descent (see [30], [52]), can be found in [53], [54], [55], [56], [57], [58]. A synchronous, unconstrained multiagent algorithm based on the heavy ball method is proposed in [53], for strongly convex f with a Lipschitz continuous gradient and an undirected, connected graph. A convergence rate of $k^{\frac{-2s}{s+1}}$ is established for the algorithm. In [54], Nesterov’s method and the heavy ball method are combined in a double accelerated, unconstrained, asynchronous algorithm for strongly convex f with a Lipschitz continuous gradient and a directed graph. Convergence for such an algorithm is exponential, under the condition that the largest step size and momentum parameters are positive and less than an explicitly stated upper bound. A constrained, synchronous, Nesterov-like sub-

gradient algorithm for convex f is proposed in [55], with a convergence rate of $\frac{1}{k}$, where k is the number of communication rounds, and there are multiple communication steps per iteration. In [56], a hybrid, unconstrained, asynchronous, algorithm motivated by Nesterov’s method and with a distributed reset mechanism is proposed for strongly convex f . The algorithm features a complete dual approach with Laplacian dependent restarting. Uniform global asymptotic stability of the minimizer is established, and the hybrid convergence rate is exponential when ∇f is Lipschitz continuous and $\frac{1}{t}$ otherwise.

The drawback to averaging-based methods, such as the the methods described above, is that such methods require bounded delays, due to the requirement of connectedness of the agents’ communication graph over intervals of specified length [59, Chapter 7]. In some applications, such delay bounds cannot be reliably checked. Moreover, averaging-based methods can be prohibitive in large-scale problems. Due to such drawbacks, in this paper we design a totally asynchronous, parallelized algorithm, based on the heavy ball method, for solving large constrained convex optimization problems. The term “totally asynchronous” was first used by Bertsekas [59] and includes both computations and communications being performed asynchronously and without need for a uniform upper bound on the length of communication delays. The term “parallelized” means that the decision vector is partitioned into blocks, each of which is updated only by a single agent. Block-based algorithms date back several decades [60], [51], and have been shown to tolerate arbitrarily long delays in some unconstrained problems [60], [61], [62]. For constrained problems, block-based methods have been utilized for primal-dual algorithms with centralized updates [63], [64], for synchronous primal updates [65], and for convex problems solved by totally asynchronous primal-dual algorithms based on classic gradient descent [50].

1.6.2 Motivation

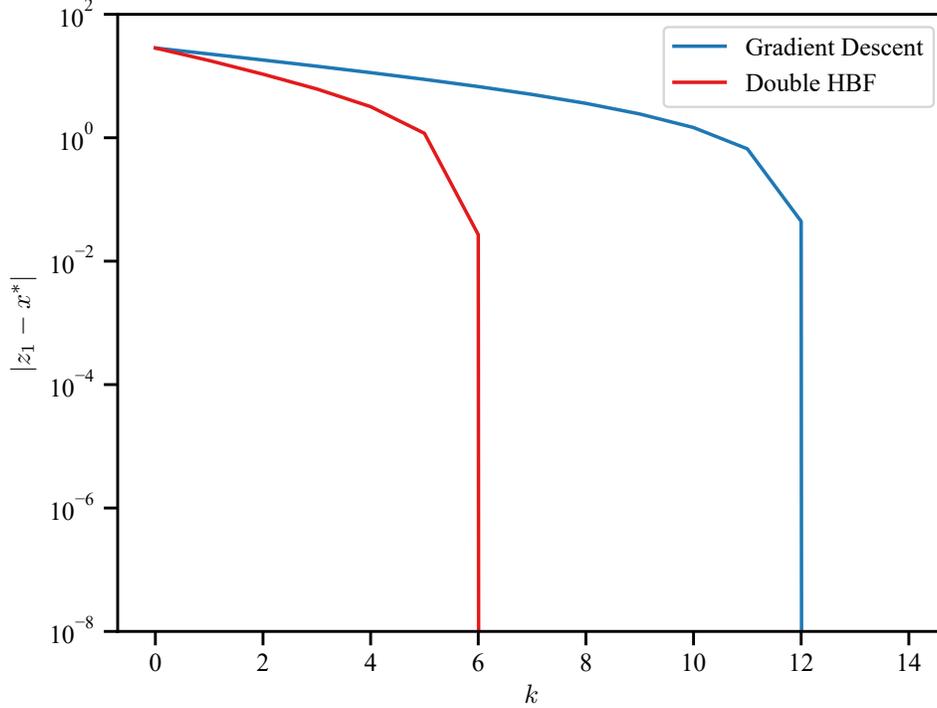


Figure 1.6: A comparison of the performance of our proposed double heavy ball algorithm with the asynchronous primal-dual algorithm for constrained gradient descent, with the dual variables fixed at zero. Convergence is twice as fast for double heavy ball (6 iterations) as it is for the gradient descent-based algorithm (12 iterations).

Our work is motivated by the lack of a totally asynchronous, block-based algorithm based on the heavy ball method. We wish to achieve fast performance with such an algorithm, without oscillations. An essential property that any update law must have, for totally asynchronous convergence, is to be contractive with respect to a block-maximum norm [59]. Since the heavy ball typically converges quickly but exhibits oscillations as $\lambda > 0$ gets smaller, as discussed in Sections 1.2.2 and 1.3.2, we design an update law consisting of computing (1.12) twice, which in the forthcoming Chapter 8 is contractive. Figure 1.6 compares the performance

of such an algorithm⁷, in comparison to totally asynchronous, block-based algorithm based on classic gradient descent, namely, the asynchronous primal-dual algorithm in [50, Theorem 2], with the dual variables fixed to zero. As can be seen in Figure 1.6, our proposed algorithm is twice as fast.

1.6.3 Contributions

The main contributions of the forthcoming Chapter 8 are as follows.

- 1) *A totally asynchronous heavy ball algorithm:* In Section 8.4.1, we propose a totally asynchronous, block-based optimization algorithm, utilizing two constrained heavy ball computations per agent update. The proposed algorithm guarantees fast convergence to the unique minimizer of f , without knowledge of $f^* := f(\xi^*)$ or ξ^* .
- 2) *Existence of solutions and forward invariance of the constraint set:* In Section 8.4.2, we prove existence of solutions and forward invariance of the constraint set. It is important for the constraint set to be forward invariant, as such a property ensures that convergence guarantees hold even when the initial conditions are outside of the constraint set.
- 3) *An exponential convergence rate:* In Section 8.4.3, we show that our algorithm has an exponential convergence rate under the assumption that f is \mathcal{C}^2 , convex, and the Hessian of f is diagonally dominant. Although such an exponential convergence rate is theoretically no better than the primal convergence rate, with a fixed dual variable, in [50, Theorem 2] for the asynchronous primal-dual algorithm⁸ in [50], we demonstrate in simulation that our algorithm is twice

⁷Code at github.com/HybridSystemsLab/MultiagentHBF.

⁸Such a comparison is fair, as the dual variables in [50] can be fixed to zero, leading to a primal-only convergence rate and algorithm.

as fast; see Section 8.5.

1.7 Organization

The contents of this dissertation are organized into the following chapters.

Chapter 2: Preliminaries In this chapter, we present the hybrid framework and its basic properties. Additionally, we include definitions of properties of objective functions for optimization. We also present Morse theory, as well as some of its basic properties. We also include preliminary information on nonsmooth Lyapunov functions and Clarke’s generalized derivative for hybrid systems, the Mean Value Theorem, and properties of sets. Finally, we present information on difference inclusions and their basic properties.

Chapter 3: Accelerated Gradient Algorithms Modeled as Dynamical Systems In this chapter, we interpret the ODEs in (1.1), (1.2), and (1.5) as control systems consisting of a plant and a control algorithm, and then we analyze key properties of the ODEs in (1.1), (1.2), and (1.5). In Section 3.1.1, we establish UGAS of the minimizer for (1.2) and an exponential convergence rate. In Section 3.1.2, we establish UGAS of the minimizer and a convergence rate of $\frac{1}{(t+2)^2}$ for (1.5). In Section 3.2.2, we establish UGAS of the minimizer for nonstrongly convex objective functions L for (1.1), and we establish exponential convergence rates for both strongly and nonstrongly convex L for (1.1) in Sections 3.2.1 and 3.2.2, respectively. In Section 3.2.4, we also establish almost global asymptotic stability of a local minimizer for nonconvex Morse functions L , for (1.1).

Chapter 4: Uniting Heavy Ball Algorithms In this chapter, we propose two logic-based algorithms uniting two heavy ball algorithms with properly designed parameters $\lambda > 0$ and $\gamma > 0$. The first such algorithm, in Section 4.3, utilizes measurements of L and ∇L , and the second algorithm, in Section 4.4,

uses measurements of ∇L . Key properties of both algorithms, including UGAS of the minimizer and an exponential (hybrid) convergence rate, are analyzed.

Chapter 5: Uniting Nesterov’s Method and the Heavy Ball Method

In this chapter, we propose two logic-based algorithms uniting Nesterov’s method globally and the heavy ball method locally, with large $\lambda > 0$. The first such algorithm, in Section 5.1, allows for L to be strongly convex and restricts $\zeta = 1$, and the second such algorithm, in Section 5.2, relaxes these conditions such that L is nonstrongly convex and $\zeta > 0$. Key properties – such as UGAS of the minimizer and a hybrid exponential convergence rate when L is strongly convex, and UGAS of the minimizer with a hybrid convergence rate consisting of $\frac{1}{(t+2)^2}$ globally and exponential locally when L is nonstrongly convex – are established.

Chapter 6: Uniting Framework for Accelerated Optimization

In this chapter, we propose a hybrid framework for uniting any type of accelerated gradient method. We determine sufficient conditions leading to general results on well-posedness, existence of solutions, and uniform global asymptotic stability for the hybrid closed-loop framework. We show that the algorithms in Chapters 4 and 5 hold for this framework, and discuss the potential for other gradient methods to be used within the proposed framework.

Chapter 7: Hybrid Accelerated Optimization for Nonconvexity

In this chapter, we propose a hybrid algorithm for nonconvex Morse objective functions L . The algorithm is based on the heavy ball algorithm in (1.1). Key properties, such as practical global attractivity of the set of minimizers, are analyzed.

Chapter 8: Accelerated Multiagent Optimization

In this chapter, Section 8.2 introduces a completely synchronous algorithm which employs one constrained heavy ball update per agent update, and presents the nominal properties of the algorithm. Section 8.3 introduces a completely synchronous algorithm which

employs two constrained heavy ball updates per agent update, and presents nominal properties of the algorithm. Section 8.4 introduces the asynchronous algorithm which employs two constrained heavy ball computations per agent update, and we establish the algorithm's nominal properties.

Chapter 9: Conclusion In this chapter, we summarize the results in this dissertation and discuss potential future directions.

Chapter 2

Preliminaries

2.1 Hybrid Systems

In this dissertation, we use the hybrid systems framework to design many of our proposed algorithms since such a framework allows for the combination of continuous-time behavior with discrete-time. A hybrid system \mathcal{H} has data (C, F, D, G) and is defined as [21, Definition 2.2]

$$\mathcal{H} = \begin{cases} \dot{x} \in F(x) & x \in C \\ x^+ \in G(x) & x \in D \end{cases} \quad (2.1)$$

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 x to \mathcal{H} 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 x , namely, $\text{dom}x \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$, is a hybrid time domain, which is a set such that for each $(T, J) \in \text{dom}x$, $\text{dom}x \cap ([0, T] \times \{0, 1, \dots, J\}) = \cup_{j=0}^J ([t_j, t_{j+1}], j)$ for a

finite sequence of times $0 = t_0 \leq t_1 \leq t_2 \leq \dots \leq t_{J+1}$. A hybrid arc x is a function on a hybrid time domain that, for each $j \in \mathbb{N}$, $t \mapsto x(t, j)$ is locally absolutely continuous on the interval $I^j := \{t : (t, j) \in \text{dom } x\}$. A solution x 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.

The following definitions, from [21] and [22], will be used in the analysis of the hybrid closed-loop system, obtained with the proposed hybrid control algorithms.

Definition 2.1.1 (Hybrid basic conditions). *A hybrid system \mathcal{H} is said to satisfy the hybrid basic conditions if its data (C, F, D, G) is such that*

(A1) *C and D are closed subsets of \mathbb{R}^n ;*

(A2) *$F : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is outer semicontinuous and locally bounded relative to C , $C \subset \text{dom } F$, and $F(x)$ is convex for every $x \in C$;*

(A3) *$G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is outer semicontinuous and locally bounded relative to D , and $D \subset \text{dom } G$.*

The notions of stability, uniform global stability, pre-attractivity, uniform global pre-attractivity, and uniform global pre-asymptotic stability (UGpAS) are listed in the following definition, from [22] and [21].

Definition 2.1.2 (Stability and attractivity notions). *Given a hybrid closed-loop system \mathcal{H} as in (2.1), a nonempty set $\mathcal{A} \subset \mathbb{R}^n$ is said to be*

- *Stable for \mathcal{H} if for each $\varepsilon > 0$ there exists $\delta > 0$ such that each solution x to \mathcal{H} with $|x(0, 0)|_{\mathcal{A}} \leq \delta$ satisfies $|x(t, j)|_{\mathcal{A}} \leq \varepsilon$ for all $(t, j) \in \text{dom } x$;*
- *Uniformly globally stable for \mathcal{H} if there exists a class- \mathcal{K}_{∞} function α such that any solution x to \mathcal{H} satisfies $|x(t, j)|_{\mathcal{A}} \leq \alpha(|x(0, 0)|_{\mathcal{A}})$ for all $(t, j) \in \text{dom } x$;*

- Pre-attractive for \mathcal{H} if there exists $\mu > 0$ such that every solution x to \mathcal{H} with $|x(0,0)|_{\mathcal{A}} \leq \mu$ is such that $(t,j) \mapsto |x(t,j)|_{\mathcal{A}}$ is bounded and if x is complete then $\lim_{(t,j) \in \text{dom } x, t+j \rightarrow \infty} |x(t,j)|_{\mathcal{A}} = 0$;
- Uniformly globally pre-attractive for \mathcal{H} if for each $\varepsilon > 0$ and $\delta > 0$ there exists $T > 0$ such that, for any solution x to \mathcal{H} with $|x(0,0)|_{\mathcal{A}} \leq \delta$, $(t,j) \in \text{dom } x$ and $t+j \leq T$ imply $|x(t,j)|_{\mathcal{A}} \leq \varepsilon$;
- Uniformly globally pre-asymptotically stable (UGpAS) for \mathcal{H} if it is both uniformly globally stable and uniformly globally pre-attractive.

In the notions involving convergence in Definition 2.1.2, when every maximal solution is complete, then the prefix “pre” is dropped to obtain attractivity, uniform global attractivity (UGA), and uniform global asymptotic stability (UGAS). The prefix “pre” is in the notions involving convergence in Definition 2.1.2 to allow for maximal solutions that are not complete. When every maximal solution is complete, such a property guarantees that nontrivial solutions exist from each initial point in $C \cup D$ to the hybrid system resulting from using our proposed uniting algorithms.

As was mentioned in Section 1.3.1, establishing UGAS for Nesterov’s algorithm is a difficult problem to solve, due to its time-varying nature, as some solutions converge in a non-uniform way. We show in Section 5.1.9 5.2.8 that our proposed uniting algorithm overcomes such a difficulty.

2.2 Optimization

Some of the algorithms proposed in this dissertation allow the cost function L to be strongly convex, as defined in [66].

Definition 2.2.1 (Strongly convex functions). *A \mathcal{C}^2 function $L : \mathbb{R}^n \rightarrow \mathbb{R}$ is strongly convex if the following hold: there exists $\mu > 0$, such that for all $u_1, w_1 \in \mathbb{R}^n$,*

$$(SC1) \quad \nabla^2 L(w_1) \geq \mu I;$$

$$(SC2) \quad L(u_1) \geq L(w_1) + \langle \nabla L(z_1), u_1 - w_1 \rangle + \frac{\mu}{2} |u_1 - w_1|^2.$$

Other algorithms proposed in this dissertation allow the cost function L to be convex (also referred to by some as “nonstrong convexity”), as defined in [66].

Definition 2.2.2 (Convex functions). *A \mathcal{C}^1 function $L : \mathbb{R}^n \rightarrow \mathbb{R}$ is (nonstrongly) convex if $L(u_1) \geq L(w_1) + \langle \nabla L(z_1), u_1 - w_1 \rangle$ for all $u_1, w_1 \in \mathbb{R}^n$.*

Additionally, some of the results in this dissertation employ the property of quadratic growth, which is a weaker condition than strong convexity [67], [19], [68], [69], [70].

Definition 2.2.3 (Quadratic growth). *A function $L : \mathbb{R}^n \rightarrow \mathbb{R}$ has quadratic growth away from its minimizer z_1^* if there exists $\alpha > 0$ such that $L(z_1) - L^* \geq \alpha |z_1 - z_1^*|^2$ for all $z_1 \in \mathbb{R}^n$, where $L^* := L(z_1^*)$.*

The following condition, from [68], will be used in some of the results in this dissertation.

Definition 2.2.4. *A function $L : \mathbb{R}^n \rightarrow \mathbb{R}$ satisfies the Polyak-Łojasiewicz condition if there exists $\theta > 0$ such that, for all $z_1 \in \mathbb{R}^n$,*

$$|\nabla L(z_1)|^2 \geq 2\mu |L(z_1) - L^*|. \tag{2.2}$$

Definition 2.2.4 means that the gradient grows faster than a quadratic function as we move away from the minimizer of L . Note that this inequality implies

that every stationary point is a global minimum [68]. The Polyak-Łojasiewicz inequality is a weaker condition than strong convexity [68] [71].

2.3 Morse Theory

In Chapter 7, we will restrict the objective function L to the class of Morse functions [72].

Definition 2.3.1 (Morse function). *The function $L : \mathbb{R}^n \rightarrow \mathbb{R}$ is a Morse function if none of its critical points is degenerate.*

For functions $L : \mathbb{R}^n \rightarrow \mathbb{R}$, a critical point is degenerate if its Hessian is singular. The following lemma, from [72, Theorem 1.3.1], describes the behavior near a critical point of a Morse function.

Lemma 2.3.2. *(The Morse Lemma) Let the function $L : \mathbb{R}^n \rightarrow \mathbb{R}$ be defined on a compact manifold and let z_1^* be a nondegenerate critical point of L . There exists an open neighborhood U of z_1^* and a diffeomorphism $\varphi : (U, z_1^*) \rightarrow (\mathbb{R}^n, 0)$ such that*

$$L \circ \varphi^{-1}(x_1, \dots, x_n) = L(z_1^*) - \sum_{j=1}^i x_j^2 + \sum_{j=i+1}^n x_j^2. \quad (2.3)$$

The Morse Lemma shows how a real-valued function $L : \mathbb{R}^n \rightarrow \mathbb{R}$ behaves on a manifold near a nondegenerate critical point, facilitating classification of an area around that critical point according to the index of L . For instance, the indices of minima, saddle points, and maxima are 0, 1, and 2, respectively. An immediate corollary of the Morse Lemma [72, Corollary 1.3.2] is as follows.

Corollary 2.3.3. *The nondegenerate critical points of a Morse function are isolated.*

The critical points of a Morse function are isolated, which means that critical points are single points, i.e., a Morse function cannot have a continuum of critical points. Note that although Definition 2.3.1 and Lemma 2.3.2 refer more generally to manifolds, we will restrict our analysis to Morse functions on the one-dimensional manifold \mathbb{R} , namely, we consider Morse functions with a single, scalar argument. For \mathcal{C}^2 functions with a single argument in \mathbb{R} , a saddle point is a stationary point that is also an inflection point. For such inflection points, the determinant of the Hessian is always singular [73, Theorem 4.8], and therefore degenerate. Therefore, saddle points never occur in \mathcal{C}^2 for Morse functions on the one-dimensional manifold \mathbb{R} . See Section 9.2 for more details on possible extensions to higher dimensions, where saddle points can occur, i.e., for $L : \mathbb{R}^n \rightarrow \mathbb{R}$ where $n > 1$.

2.4 Nonsmooth Lyapunov Functions

For the analysis of the ODE in (1.1) for nonconvex objective functions L in Chapter 3, and for the analysis of the proposed hybrid algorithm for nonconvex morse functions in Chapter 7, we will use Clarke’s generalized directional derivative.

Given a hybrid system \mathcal{H} with data (C, F, D, G) , let $V : \text{dom } V \mapsto \mathbb{R}$ be piecewise continuous on $\text{dom } V$ and locally Lipschitz on an open neighborhood of $(C \cap \mathcal{U})$. Following [74], the generalized gradient in the sense of Clarke of V at a point $z \in (C \cap \mathcal{U})$, denoted by $\partial V(z)$, is a closed, convex, and nonempty set equal to the convex hull of all limits of the sequence $\nabla V(z_i)$, where z_i is any sequence converging to z that avoids any set with zero Lebesgue measure that contains points at which V is nondifferentiable – since V is locally Lipschitz, then ∇V exists almost everywhere. Then, Clarke’s generalized directional derivative

of V at a point z in the direction of χ is given by

$$V^\circ(z, \chi) = \max_{\zeta \in \partial V(z)} \langle \zeta, \chi \rangle. \quad (2.4)$$

Then, for any solution $t \mapsto z(t)$ to $\dot{z} \in F(z)$,

$$\frac{d}{dt}V(z(t)) \leq V^\circ(z(t), \dot{z}(t)) \quad (2.5)$$

for almost all t in the domain of the definition of the function z , where the derivative $\frac{d}{dt}V(z(t))$ is understood in the standard sense since V is locally Lipschitz. The reader is referred to [74] for more details on the generalized gradient and Clarke's generalized directional derivative.

Following [75], a bound on the increase of the function V along solutions to the hybrid system \mathcal{H} is obtained by defining the function $u_C : \text{dom } V \rightarrow [-\infty, \infty)$ as

$$u_C(z) := \begin{cases} \max_{\chi \in F(z)} \max_{\zeta \in \partial V(z)} \langle \zeta, \chi \rangle & z \in C \cap \mathcal{U} \\ -\infty & \text{otherwise} \end{cases} \quad (2.6)$$

Then, for each solution ϕ to \mathcal{H} and each t at which $\frac{d}{dt}V(\phi(t, j))$ exists, the following bound holds:

$$\frac{d}{dt}V(\phi(t, j)) \leq u_C(\phi(t, j)). \quad (2.7)$$

Similarly, to obtain a bound on the change in V at jumps, the following quantity is defined:

$$u_D(z) := \begin{cases} \max_{\chi \in G(z)} V(\chi) - V(z) & z \in D \cap \mathcal{U} \\ -\infty & \text{otherwise} \end{cases} \quad (2.8)$$

Then, for any solution ϕ to \mathcal{H} and for any $(t_{j+1}, j), (t_{j+1}, j+1) \in \text{dom } \phi$, it follows

that

$$V(\phi(t_{j+1}, j+1)) - V(\phi(t_{j+1}, j)) \leq u_D(\phi(t_{j+1}, j)). \quad (2.9)$$

Note that when F is a single-valued map, $u_C(z) = V^\circ(z, F(z))$ for each $z \in C \cap \mathcal{U}$.

When G is a single-valued map, $u_D(z) = V(G(z)) - V(z)$ for each $z \in D \cap \mathcal{U}$.

2.5 Mean Value Theorem

For the analysis of the proposed multiagent heavy ball algorithm in Chapter 8, we use the following version of the Mean Value Theorem (MVT), from [59, Proposition A.30].

Proposition 2.5.1. (*Mean value theorem*): *If $L : \mathbb{R}^m \rightarrow \mathbb{R}$ is continuously differentiable, then for every $x, y \in \mathbb{R}^m$, there exists some $c \in [0, 1]$ such that*

$$L(y) - L(x) = \nabla L(cx + (1-c)y)^\top (y - x). \quad (2.10)$$

2.6 Properties of Sets

In this section, we give some basic definitions and properties that we use to characterize sets, in the analysis of the proposed multiagent heavy ball algorithm in Chapter 8.

Definition 2.6.1 (Inner and outer limit). *For a sequence of sets $\{T_i\}_{i=0}^\infty$ in \mathbb{R}^n*

- *The inner limit of the sequence $\{T_i\}_{i=0}^\infty$, denoted $\liminf_{i \rightarrow \infty} T_i$, is the set of all $x \in \mathbb{R}^n$ for which there exist points $x_i \in T_i$, $i \in \mathbb{N}$, such that $\lim_{i \rightarrow \infty} x_i = x$.*

- The outer limit of the sequence $\{T_i\}_{i=0}^{\infty}$, denoted $\limsup_{i \rightarrow \infty} T_i$ is the set of all $x \in \mathbb{R}^n$ for which there exists a subsequence $\{T_{i_k}\}_{k=0}^{\infty}$ of $\{T_i\}_{i=0}^{\infty}$ and points $x_k \in T_{i_k}$, $k \in \mathbb{N}$, such that $\lim_{k \rightarrow \infty} x_k = x$.

The *limit* of the sequence exists if the outer and inner limit sets are equal, namely,

$$\lim_{i \rightarrow \infty} T_i = \liminf_{i \rightarrow \infty} T_i = \limsup_{i \rightarrow \infty} T_i.$$

The inner and outer limit of a sequence always exist and are closed, although the limit itself might not exist.

The following definition of convergence of a sequence of sets comes from [76].

Definition 2.6.2 (Convergence of a sequence of sets). *When the limit of the sequence $\{T_i\}_{i=0}^{\infty}$ in \mathbb{R}^m exists in the sense of Definition 2.6.1 and is equal to T , the sequence of sets is said to converge to the set T .*

2.7 Difference Inclusions

In Chapter 8, we consider discrete-time systems with data (D, G) and defined as

$$z^+ \in G(z) \quad z \in D \tag{2.11}$$

where $z \in \mathbb{R}^m$ is the system state, $G : \mathbb{R}^m \rightrightarrows \mathbb{R}^m$ is the right-hand side, and $D \subset \mathbb{R}^m$ is the constraint set. The notation \rightrightarrows indicates that G is a set-valued map.

The following definitions, from [22] [21], will be used in the analysis of the closed-loop system, obtained with the proposed control algorithm.

Definition 2.7.1 (Basic conditions). *System (2.11) is said to satisfy the basic conditions if its data (D, G) is such that*

(A1) D is a closed subset of \mathbb{R}^m ;

(A2) $G : \mathbb{R}^m \rightrightarrows \mathbb{R}^m$ is outer semicontinuous and locally bounded relative to D ,
and $D \subset \text{dom } G$.

The following definition of forward invariance, from [22, Definition 3.13], will be used in the forthcoming result on asymptotic stability in Chapter 8.

Definition 2.7.2 (Forward invariance). *Given a system with data (D, G) , defined in (2.11), a nonempty set $Z \subset \mathbb{R}^m$ is said to be forward invariant for (D, G) if each maximal solution z to (D, G) starting from $z_0 \in Z$ is complete and satisfies $z(k) \in Z$ for all $k \in \text{dom } z$.*

Chapter 3

Accelerated Gradient Algorithms Modeled as Dynamical Systems

3.1 Nesterov's Accelerated Gradient Descent Modeled as a Dynamical System

We interpret the ODEs in (1.1), (1.2), and (1.5) as control systems consisting of a plant and a control algorithm [34] [22]. Defining z_1 as ξ and z_2 as $\dot{\xi}$, the plant associated with such ODEs is given by the double integrator

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ u \end{bmatrix} =: F_P(z, u) \quad (z, u) \in \mathbb{R}^{2n} \times \mathbb{R}^n \quad (3.1)$$

with an output given by a function of the state, as defined below. With this model, the optimization algorithms that we consider assign u to a function of the state that involves the cost function, and such a function of the state may be time dependent. The control algorithm u , leading to each of the ODEs, will be defined in the subsequent sections.

3.1.1 Strongly Convex L

For the analysis in this section, we impose the following Assumption on L .

Assumption 3.1.1. *The function L is \mathcal{C}^2 and strongly convex.*

Remark 3.1.2. *Assumption 3.1.1, which is a common assumption used in the analysis of optimization algorithms [66] [52], ensures that the objective function is continuously differentiable, which is necessary for well-posedness of the proposed uniting algorithms. Additionally, the strongly convex property in Assumption 3.1.1 restricts the objective function to having a unique minimizer, and rules out the possibility of the objective function having a continuum of minimizers or multiple isolated minimizers.*

The control algorithm leading to (1.2) is

$$u = \kappa(h(z)) = -2dz_2 - \frac{1}{M}\nabla L(z_1 + \beta z_2) \quad (3.2)$$

where $M > 0$ is the Lipschitz constant for ∇L and d and β are defined as

$$d := \frac{1}{(\sqrt{\kappa_c} + 1)} > 0, \quad \beta := \frac{(\sqrt{\kappa_c} - 1)}{(\sqrt{\kappa_c} + 1)} \geq 0. \quad (3.3)$$

where

$$\kappa_c := \frac{M}{\mu} \geq 1 \quad (3.4)$$

is the condition number associated with L ; see [66] [52]. We define h as

$$h(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1 + \beta z_2) \end{bmatrix}. \quad (3.5)$$

Using the plant in (3.1), we denote the closed-loop system resulting from κ in (3.2) as

$$\dot{z} = \begin{bmatrix} z_2 \\ \kappa(h(z)) \end{bmatrix} \quad z \in \mathbb{R}^{2n}. \quad (3.6)$$

For some of the results to follow, we impose the following assumption on ∇L .

Assumption 3.1.3 (Lipschitz Continuity of ∇L). *The function ∇L is Lipschitz continuous with constant $M > 0$, namely, for all $w_1, u_1 \in \mathbb{R}^n$,*

$$|\nabla L(w_1) - \nabla L(u_1)| \leq M |w_1 - u_1|. \quad (3.7)$$

Remark 3.1.4. *Assumption 3.1.3 is commonly used in nonlinear analysis to ensure that the differential equations of the individual optimization algorithms, for example, the ODE in (1.2), does not have solutions that escape in finite time, which is used to guarantee existence and completeness of maximal solutions to (3.6) [77, Theorem 3.2].*

Each maximal solution to (3.6) is complete and bounded, when L satisfies Assumptions 3.1.1 and 3.1.3, as shown in the following lemma.

Proposition 3.1.5. *(Existence of solutions to (3.6)) Let L satisfy Assumptions 3.1.1 and 3.1.3. Let the functions d and β be defined as in (3.3). Let κ be defined via (3.2). Then, each maximal solution $t \mapsto z(t)$ to the closed-loop algorithm (3.6), is bounded, complete, and unique.*

Proof. Since d and β , defined via (3.3), are constants, and since by Assumption 3.1.1 L is \mathcal{C}^2 , then h in (3.5) and κ in (3.2) are also continuous. Furthermore, since by Assumption 3.1.3 ∇L is Lipschitz continuous, then h in (3.5) and κ in (3.2) are also Lipschitz continuous which, in turn, means the map $z \mapsto F_P(z, \kappa(h(z)))$ is

Lipschitz continuous. Consequently, since the map $z \mapsto F_P(z, \kappa(h(z)))$ is Lipschitz continuous, then by [77, Theorem 3.2], $\dot{z} = F_P(z, \kappa(h(z)))$ has no finite escape time¹ from \mathbb{R}^{2n} and each maximal solution is complete and unique.

To show that each maximal solution is bounded, we first define the Lyapunov function

$$V_1(z) := \frac{1}{2} |a(z_1 - z_1^*) + z_2|^2 + \frac{1}{M} (L(z_1) - L^*) \quad (3.8)$$

where the constant $a > 0$ is defined as

$$a := d + \frac{\beta}{2\kappa_c} = \frac{1}{\kappa_c} - \frac{1}{2\kappa_c} > 0. \quad (3.9)$$

Then, we will show that solutions to $\dot{z} = F_P(z, \kappa(h(z)))$ starting from any c_V -sublevel set

$$W := \left\{ z \in \mathbb{R}^{2n} : V_1(z) \leq c_V \right\} \quad (3.10)$$

with $c_V > 0$, remain in such a set for all time, and then we will show that V_1 is radially unbounded.

To that end, note that V_1 is positive definite with respect to $\{z_1^*\} \times \{0\}$ since, by Assumption 3.1.1, L is \mathcal{C}^2 and strongly convex. Then, letting

$$v_1(z) := z_1 + \beta z_2 \quad (3.11)$$

and since $\nabla V_1(z) = \left[a(a(z_1 - z_1^*) + z_2) + \frac{1}{M} \nabla L(z_1) \quad (a(z_1 - z_1^*) + z_2) \right]$, we evaluate the derivative of V_1 , using the map $z \mapsto F_P(z, \kappa(h(z)))$, where F_P is defined in (3.1), κ is defined in (3.2), and h is defined in (3.5), to yield

$$\dot{V}_1(z) = \langle \nabla V_1(z), F_P(z, \kappa(h(z))) \rangle$$

¹*Finite escape time* describes when there exists a solution $t \mapsto x(t)$ to a continuous-time nonlinear system that satisfies $\lim_{t \nearrow t_e} |x(t)| = \infty$ for some finite time t_e .

$$\begin{aligned}
&= \left\langle \nabla V_1(z), \begin{bmatrix} z_2 \\ -2dz_2 - \frac{1}{M}\nabla L(z_1 + \beta z_2) \end{bmatrix} \right\rangle \\
&= a^2 \langle z_1 - z_1^*, z_2 \rangle + a|z_2|^2 + \frac{1}{M} \langle z_2, \nabla L(z_1) \rangle - 2d|z_2|^2 - 2da \langle z_1 - z_1^*, z_2 \rangle \\
&\quad - \frac{a}{M} \langle z_1 - z_1^*, \nabla L(v_1(z)) \rangle - \frac{1}{M} \langle z_2, \nabla L(v_1(z)) \rangle \\
&= -\frac{a}{M} \langle z_1 - z_1^*, \nabla L(v_1(z)) \rangle + a(a - 2d) \langle z_1 - z_1^*, z_2 \rangle \\
&\quad + (a - 2d)|z_2|^2 - \frac{1}{M} \langle z_2, \nabla L(v_1(z)) - \nabla L(z_1) \rangle \tag{3.12}
\end{aligned}$$

for all $z \in \mathbb{R}^{2n}$.

Since L is \mathcal{C}^2 , strongly convex by Assumption 3.1.1, and ∇L is Lipschitz continuous with constant $M > 0$ by Assumption 3.1.3, then using κ_c in (3.4) and the definition of strong convexity in item (SC2) of Definition 2.2.1 with $u_1 = z_1^*$ and $w_1 = v_1(z)$, where v_1 is defined via (3.11), we get

$$-\langle v_1(z) - z_1^*, \nabla L(v_1(z)) \rangle \leq -(L(v_1(z)) - L^*) - \frac{M}{2\kappa_c} |v_1(z) - z_1^*|^2 \tag{3.13}$$

for each $z \in \mathbb{R}^{2n}$. Using κ in (3.4) and the definition of strong convexity in item (SC2) of Definition 2.2.1 with $u_1 = v_1(z)$, where v_1 is defined via (3.11), and $w_1 = z_1$ yields

$$L(v_1(z)) \geq L(z_1) + \langle \nabla L(z_1), \beta z_2 \rangle + \frac{M}{2\kappa_c} \beta^2 |z_2|^2 \tag{3.14}$$

for each $z \in \mathbb{R}^{2n}$. Combining (3.13) and (3.14) yields

$$\begin{aligned}
-\langle v_1(z) - z_1^*, \nabla L(v_1(z)) \rangle &\leq -(L(z_1) - L^*) - \langle \nabla L(z_1), \beta z_2 \rangle \\
&\quad - \frac{M}{2\kappa_c} (|v_1(z) - z_1^*|^2 + \beta^2 |z_2|^2)
\end{aligned}$$

Then, rearranging terms gives, for all $z \in \mathbb{R}^{2n}$,

$$\begin{aligned} -\langle z_1 - z_1^*, \nabla L(v_1) \rangle &\leq - (L(z_1) - L^*) - \frac{M}{2\kappa_c} (|v_1(z) - z_1^*|^2 + \beta^2 |z_2|^2) \\ &\quad - \langle \beta z_2, \nabla L(v_1(z)) - \nabla L(z_1) \rangle. \end{aligned} \quad (3.15)$$

In addition, note that

$$\begin{aligned} |v_1(z) - z_1^*|^2 &= |z_1|^2 - 2\langle z_1, z_1^* \rangle + |z_1^*|^2 + 2\beta \langle z_1 - z_1^*, z_2 \rangle + \beta^2 |z_2|^2 \\ &= |z_1 - z_1^*|^2 + 2\beta \langle z_1 - z_1^*, z_2 \rangle + \beta^2 |z_2|^2 \end{aligned} \quad (3.16)$$

Substituting the expression for $|v_1(z) - z_1^*|^2$ in (3.16) into the bound in (3.15), then subsequently substituting the bound in (3.15) into (3.12) yields

$$\begin{aligned} \dot{V}_1(z) &\leq -\frac{a}{M} (L(z_1) - L^*) - \frac{a}{2\kappa_c} |z_1 - z_1^*|^2 - \frac{a\beta}{\kappa_c} \langle z_1 - z_1^*, z_2 \rangle - \frac{a\beta^2}{\kappa_c} |z_2|^2 \\ &\quad + a(a - 2d) \langle z_1 - z_1^*, z_2 \rangle + (a - 2d) |z_2|^2 \\ &\quad - \frac{1}{M} (1 - \beta a) \langle z_2, \nabla L(v_1(z)) - \nabla L(z_1) \rangle. \end{aligned} \quad (3.17)$$

for each $z \in \mathbb{R}^{2n}$. Then, noticing that $\frac{a}{2} |a(z_1 - z_1^*) + z_2|^2 = \frac{a^3}{2} |z_1 - z_1^*|^2 + a^2 \langle z_1 - z_1^*, z_2 \rangle + \frac{a}{2} |z_2|^2$, adding it to and subtracting it from (3.17), and rearranging terms, yields

$$\begin{aligned} \dot{V}_1(z) &\leq -aV_1(z) - \frac{a}{2\kappa_c} |z_1 - z_1^*|^2 - \frac{a\beta}{\kappa_c} \langle z_1 - z_1^*, z_2 \rangle - \frac{a\beta^2}{\kappa_c} |z_2|^2 \\ &\quad + a(a - 2d) \langle z_1 - z_1^*, z_2 \rangle + (a - 2d) |z_2|^2 + \frac{a^3}{2} |z_1 - z_1^*|^2 \\ &\quad + a^2 \langle z_1 - z_1^*, z_2 \rangle + \frac{a}{2} |z_2|^2 - \frac{1}{M} (1 - \beta a) \langle z_2, \nabla L(v_1(z)) - \nabla L(z_1) \rangle \\ &\leq -aV_1(z) + \frac{a}{2} \left(a^2 - \frac{1}{\kappa_c} \right) |z_1 - z_1^*|^2 + \left(\frac{3a}{2} - 2d - \frac{a\beta^2}{\kappa_c} \right) |z_2|^2 \end{aligned}$$

$$\begin{aligned}
& + a \left(2a - 2d - \frac{\beta}{\kappa_c} \right) \langle z_1 - z_1^*, z_2 \rangle \\
& - \frac{1}{M} (1 - \beta a) \langle z_2, \nabla L(v_1) - \nabla L(z_1) \rangle.
\end{aligned} \tag{3.18}$$

for each $z \in \mathbb{R}^{2n}$. Due to the definition of a in (3.9), the cross term $\langle z_1 - z_1^*, z_2 \rangle$ vanishes since $\left(2 \left(d + \frac{\beta}{2\kappa_c} \right) - 2d - \frac{\beta}{\kappa_c} \right) = \left(2d + \frac{\beta}{\kappa_c} - 2d - \frac{\beta}{\kappa_c} \right) = 0$.

By Assumption 3.1.1, L is \mathcal{C}^2 and strongly convex with constant $\mu > 0$. An equivalent characterization of the strong convexity of L , from [52, Theorem 2.1.9], for all $w_1, u_1 \in \mathbb{R}^n$, is $\langle \nabla L(w_1) - \nabla L(u_1), w_1 - u_1 \rangle \geq \mu |w_1 - u_1|^2$. Then, using such a bound with $w_1 = v_1(z)$, where v_1 is defined via (3.11), and $u_1 = z_1$, we get, for all $z \in \mathbb{R}^{2n}$,

$$\begin{aligned}
- \langle \nabla L(v_1(z)) - \nabla L(z_1), \beta z_2 \rangle & = - \langle \nabla L(v_1(z)) - \nabla L(z_1), v_1(z) - z_1 \rangle \\
& \leq -\mu \beta^2 |z_2|^2.
\end{aligned} \tag{3.19}$$

Therefore, we use $-\langle z_2, \nabla L(v_1(z)) - \nabla L(z_1) \rangle \leq -\mu \beta |z_2|^2$, the definition of κ_c in (3.4), and the fact that $1 - \beta a \geq 0$ and $\beta \geq 0$, to upper bound the last term in (3.18) as follows:

$$-\frac{1}{M} (1 - \beta a) \langle z_2, \nabla L(v_1) - \nabla L(z_1) \rangle \leq - (1 - \beta a) \frac{\beta}{\kappa_c} |z_2|^2,$$

which implies

$$\dot{V}_1(z) \leq -aV_1(z) + \frac{a}{2} \left(a^2 - \frac{1}{\kappa_c} \right) |z_1 - z_1^*|^2 + \left(\frac{3a}{2} - 2d - \frac{\beta}{\kappa_c} \right) |z_2|^2. \tag{3.20}$$

Using the definitions of a in (3.9), d , and β in (3.3), we show that $a^2 - \frac{1}{\kappa_c} \leq 0$ for

all $\kappa_c \geq 1$, as follows.

$$\begin{aligned} a^2 - \frac{1}{\kappa_c} &\leq 0 \\ \frac{1}{\kappa_c} - \frac{1}{4\kappa_c^2} - \frac{1}{\kappa_c} &\leq 0 \\ -\frac{1}{4\kappa_c^2} &\leq 0. \end{aligned} \tag{3.21}$$

Therefore, since $\kappa_c \geq 1$, as defined in (3.4), then $-\frac{1}{4\kappa_c^2} \leq 0$. We can also show, using the definition of a in (3.9), that $\left(\frac{3a}{2} - 2d - \frac{\beta}{\kappa_c}\right) \leq 0$ for all $\kappa_c \geq 1$. Namely, since $d > 0$, $\beta \geq 0$, and $\kappa_c \geq 1$, then $\frac{3}{2}\left(d + \frac{\beta}{2\kappa_c}\right) - 2d - \frac{\beta}{\kappa_c} = -\frac{d}{2} - \frac{\beta}{4\kappa_c} \leq 0$. Therefore, since $a^2 - \frac{1}{\kappa_c} \leq 0$ and $\left(\frac{3a}{2} - 2d - \frac{\beta}{\kappa_c}\right) \leq 0$ for all $\kappa_c \geq 1$, then (3.20) is upper bounded by

$$\dot{V}_1(z) \leq -aV_1(z) \tag{3.22}$$

for all $z \in \mathbb{R}^{2n}$.

Then, to show that V_1 is radially unbounded, we show that there exist α_1 and α_2 such that, for all $z \in \mathbb{R}^{2n}$, with $z^* := (z_1^*, 0)$,

$$\alpha_1 |z - z^*|^2 \leq \begin{bmatrix} (z_1 - z_1^*)^\top & z_2^\top \end{bmatrix} B \begin{bmatrix} z_1 - z_1^* \\ z_2 \end{bmatrix} \leq V_1(z) \leq \alpha_2 |z - z^*|^2 \tag{3.23}$$

where $\alpha_2 := \left(a^2 + 1 - \frac{1}{2\kappa_c}\right)$ and B is defined as

$$B := \begin{bmatrix} \left(\frac{a^2}{2} + \frac{\alpha}{M}\right) & \frac{a}{2} \\ \frac{a}{2} & \frac{1}{2} \end{bmatrix} \tag{3.24}$$

where $M > 0$ comes from Assumption 3.1.3 and where, since L is strongly convex by Assumption 3.1.1, then this implies that L also satisfies² Definition 2.2.3,

²This is true since quadratic growth is a weaker property than strong convexity; see [67], [19], [68], [69], [70].

namely, L has quadratic growth away from z_1^* with constant $\alpha > 0$. Then, we lower bound V_1 as follows:

$$\begin{aligned}
V_1(z) &= \frac{1}{2} |a(z_1 - z_1^*) + z_2|^2 + \frac{1}{M} (L(z_1) - L^*) & (3.25) \\
&\geq \frac{1}{2} |a(z_1 - z_1^*) + z_2|^2 + \frac{\alpha}{M} |z_1 - z_1^*|^2 \\
&\geq \frac{a^2}{2} |z_1 - z_1^*|^2 + a \langle z_1 - z_1^*, z_2 \rangle + \frac{1}{2} |z_2|^2 + \frac{\alpha}{M} |z_1 - z_1^*|^2 \\
&\geq \left(\frac{a^2}{2} + \frac{\alpha}{M} \right) |z_1 - z_1^*|^2 + \frac{a}{2} \langle z_1 - z_1^*, z_2 \rangle + \frac{a}{2} \langle z_1 - z_1^*, z_2 \rangle + \frac{1}{2} |z_2|^2 \\
&\geq \begin{bmatrix} (z_1 - z_1^*)^\top & z_2^\top \end{bmatrix} B \begin{bmatrix} z_1 - z_1^* \\ z_2 \end{bmatrix}
\end{aligned}$$

for each $z \in \mathbb{R}^{2n}$, where B is defined via (3.24). Next, we show that B is positive definite, such that there exists α_1 such that the first two inequalities in (3.23) hold for each $z \in \mathbb{R}^{2n}$. To that end, we show that the leading principal minors of B are strictly positive, as follows. Since $a > 0$, $\alpha > 0$, and $M > 0$, we have

$$\left(\frac{a^2}{2} + \frac{\alpha}{M} \right) > 0 \tag{3.26a}$$

$$\det(B) = \left(\frac{1}{2} \right) \left(\frac{a^2}{2} + \frac{\alpha}{M} \right) - \left(\frac{a}{2} \right)^2 = \frac{\alpha}{2M} > 0. \tag{3.26b}$$

Therefore, since the leading principal minors of B are strictly positive, then B is positive definite. Hence, there exists α_1 such that the first two inequalities in (3.23) hold for each $z \in \mathbb{R}^{2n}$. The choice of α_2 comes from the following. The first term of V_1 is upper bounded by

$$\frac{1}{2} |a(z_1 - z_1^*) + z_2|^2 \leq a^2 |z_1 - z_1^*|^2 + |z_2|^2 \leq a^2 |z - z^*| \tag{3.27}$$

for each $z \in \mathbb{R}^{2n}$. Then, the second term of V_1 can be bounded as follows. First,

∇L is Lipschitz continuous with constant $M > 0$ due to Assumption 3.1.3. Then, since by Assumption 3.1.1, L is strongly convex with constant $\mu > 0$, then using item (SC2) of Definition 2.2.1 with $u_1 = z_1^*$, $w_1 = z_1$, and $\nabla L^* = 0$ we have

$$\begin{aligned}
L^* &\geq L(z_1) + \langle \nabla L(z_1), z_1^* - z_1 \rangle + \frac{\mu}{2} |z_1^* - z_1|^2 \\
|\langle \nabla L(z_1), z_1^* - z_1 \rangle| &\geq L(z_1) - L^* + \frac{\mu}{2} |z_1^* - z_1|^2 \\
|\nabla L(z_1)| |z_1^* - z_1| &\geq L(z_1) - L^* + \frac{\mu}{2} |z_1^* - z_1|^2 \\
M |z_1^* - z_1| |z_1^* - z_1| &\geq L(z_1) - L^* + \frac{\mu}{2} |z_1^* - z_1|^2 \\
M |z_1^* - z_1|^2 &\geq L(z_1) - L^* + \frac{\mu}{2} |z_1^* - z_1|^2 \\
\left(M - \frac{\mu}{2}\right) |z_1^* - z_1|^2 &\geq L(z_1) - L^*. \tag{3.28}
\end{aligned}$$

for each $z_1 \in \mathbb{R}^n$. Hence, the second term of V_1 can be upper bounded as follows:

$$\begin{aligned}
\frac{1}{M}(L(z_1) - L^*) &\leq \left(\frac{M - \frac{\mu}{2}}{M}\right) |z_1 - z_1^*|^2 = \left(1 - \frac{1}{2\kappa_c}\right) |z_1 - z_1^*|^2 \\
&\leq \left(1 - \frac{1}{2\kappa_c}\right) |z - z^*|. \tag{3.29}
\end{aligned}$$

for each $z \in \mathbb{R}^{2n}$. Therefore, V_1 is upper bounded as follows:

$$\begin{aligned}
V_1(z) &= \frac{1}{2} |a(z_1 - z_1^*) + z_2|^2 + \frac{1}{M} (L(z_1) - L^*) \\
&\leq \left(a^2 + 1 - \frac{1}{\kappa_c}\right) |z - z^*| = \alpha_2 |z - z^*| \tag{3.30}
\end{aligned}$$

for each $z \in \mathbb{R}^{2n}$.

Therefore, since (3.23) is satisfied for V_1 in (3.8) for all $z \in \mathbb{R}^{2n}$, where B is defined via (3.24) and α_2 is defined below (3.23), then V_1 is radially unbounded (in z , relative to $\{z_1^*\} \times \{0\}$). Since L is \mathcal{C}^2 and strongly convex by Assumption 3.1.1, then L is positive definite with respect to z_1^* and, consequently, V_1 is positive

definite with respect to $\{z_1^*\} \times \{0\}$. Therefore, since $a > 0$ by (3.9) and V_1 satisfies (3.22) for all $z \in \mathbb{R}^{2n}$, then solutions to $\dot{z} = F_P(z, \kappa(h(z)))$ starting from any c_V -sublevel set $W := \{z \in \mathbb{R}^{2n} : V_1(z) \leq c_V\}$, $c_V > 0$, remain in such a set for all time. Therefore, W in (3.10) is compact and, due to (3.22), forward invariant for (3.6), that is, any nontrivial solution starting in the subset W is complete and stays in W . Therefore, each maximal solution to (3.6) is bounded. \square

When L satisfies Assumptions 3.1.3 and 3.1.1, the rate of convergence for (3.6) is exponential, as extended from [12, Proposition 3.1] to a generic L^* at a generic z_1^* . It is as follows.

Proposition 3.1.6. *(Convergence rate for (3.6)) Let L satisfy Assumptions 3.1.3 and 3.1.1. Let the functions d and β be defined as in (3.3). Let κ be defined via (3.2). Then, each maximal solution $t \mapsto (z(t))$ to (3.6) satisfies*

$$L(z_1(t)) - L^* \leq (L(z_1(0)) - L^*) \exp(-at) \quad (3.31)$$

for all $t \geq 0$, where $a > 0$ is defined via (3.9).

Proof. By Proposition 3.1.5, V_1 in (3.8) satisfies (3.22) for all $z \in \mathbb{R}^{2n}$. Applying Grönwall's inequality, shows that each maximal solution $t \mapsto (z(t))$ to (3.6) satisfies

$$V_1(z(t)) \leq V_1(z(0)) \exp(-at) \quad (3.32)$$

for all $t \geq 0$. This, in turn, implies that each maximal solution $t \mapsto (z(t))$ to (3.6) satisfies (3.31) for all $t \geq 0$. \square

The following theorem shows that (3.6), when L satisfies Assumptions 3.1.1 and 3.1.3, has the set

$$\mathcal{A} := \left\{ z \in \mathbb{R}^{2n} : \nabla L(z_1) = z_2 = 0 \right\} = \{z_1^*\} \times \{0\}, \quad (3.33)$$

uniformly globally asymptotically stable.

Theorem 3.1.7. *(Uniform global asymptotic stability of \mathcal{A} for (3.6)) Let L satisfy Assumption 3.1.3 and Assumption 3.1.1. Let the functions d and β be defined as in (3.3). Let κ be defined via (3.2). Then, the set \mathcal{A} , defined via (3.33), is uniformly globally asymptotically stable for (3.6).*

Proof. By Proposition 3.1.5, each maximal solution to (3.6), is bounded, complete, and unique. In addition, by Proposition 3.1.5, V_1 is positive definite and radially unbounded (in z , relative to \mathcal{A} in (3.33)). Then, since $a > 0$ by (3.9) and V_1 satisfies (3.22) for each $z \in \mathbb{R}^{2n}$, $\rho(|z|_{\mathcal{A}}) := aV_1(z)$ is positive definite with respect to \mathcal{A} . Therefore, by an application of [21, Theorem 3.18], every complete solution to (3.6) converges to $\{z_1^*\} \times \{0\}$. The arguments above involving the Lyapunov theorem in Theorem A.1.3 yields uniform global pre-asymptotic stability of \mathcal{A} in (3.33) for (3.6). Since by Proposition 3.1.5, each maximal solution to (3.6) is complete, then \mathcal{A} is globally asymptotically stable for (3.6). \square

In Proposition 3.1.6, not only do we recover the convergence rate (3.31) from [12, Proposition 3.1], which assumes $L^* = 0$ and $z_1^* = 0$, but we extend this proof to show that the rate (3.31) also applies for general $L^* \in \mathbb{R}$ and $z_1^* \in \mathbb{R}^n$. Additionally, in Theorem 3.1.7, we establish uniform global asymptotic stability of the set \mathcal{A} in (3.33) for (3.6), which was not established in [12].

3.1.2 Nonstrongly Convex L

For the analysis in this section, we impose Assumption 3.1.3 and the following Assumption on L .

Assumption 3.1.8. *The function L is \mathcal{C}^1 , (nonstrongly) convex, and has a single minimizer z_1^* .*

Remark 3.1.9. *Assumption 3.1.8, which is a common assumption used in the analysis of optimization algorithms [66] [52], ensures that the objective function is continuously differentiable, which is necessary for well-posedness of \mathcal{H} , as was explained in Section 1.3.4. Additionally, the nonstrongly convex property and the restriction that L has a single minimizer z_1^* in Assumption 3.1.8 rules out the possibility of the objective function having a continuum of minimizers or multiple isolated minimizers.*

The control algorithm leading to (1.5) is

$$u = \kappa(h(z, t), t) = -2\bar{d}(t)z_2 - \frac{\zeta^2}{M}\nabla L(z_1 + \bar{\beta}(t)z_2) \quad (3.34)$$

where $M > 0$ is the Lipschitz constant for ∇L , \bar{d} and $\bar{\beta}$ are defined, for all $t \geq 0$, as

$$\bar{d}(t) := \frac{3}{2(t+2)}, \quad \bar{\beta}(t) := \frac{t-1}{t+2}, \quad (3.35)$$

and h is defined as

$$h(z, t) := \begin{bmatrix} z_2 \\ \nabla L(z_1 + \bar{\beta}(t)z_2) \end{bmatrix}. \quad (3.36)$$

Since the ODE in (1.5) is time varying, and since solutions to hybrid systems are parameterized by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, we employ the state τ to capture ordinary time as a state variable, in this way, leading to a time-invariant system. To this end, using the plant in (3.1), we denote the closed-loop system resulting from κ in (3.34) as

$$\dot{z} = \begin{bmatrix} z_2 \\ \kappa(h(z, \tau), \tau) \end{bmatrix}, \quad \dot{\tau} = 1 \quad (z, \tau) \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}. \quad (3.37)$$

Under Assumptions 3.1.8 and 3.1.3, each maximal solution to (3.37) is com-

plete and unique. Such a property is useful since it guarantees that nontrivial solutions to (3.37) exist from each initial point in $\mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$, and that such solutions do not escape $\mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$. When each maximal solution is complete, then uniform global pre-asymptotic stability of $\{z_1^*\} \times \{0\} \times \mathbb{R}_{\geq 0}$ becomes uniform global asymptotic stability.

Proposition 3.1.10. *(Existence of solutions to (3.37)) Let L satisfy Assumptions 3.1.8 and 3.1.3. Let the functions \bar{d} and $\bar{\beta}$ be defined in (3.35). Let κ be defined via (3.34). Then, each maximal solution $t \mapsto (z(t), \tau(t))$ to (3.37) is complete and unique.*

Proof. Since \bar{d} and $\bar{\beta}$, defined via (3.35), are continuous, and since by Assumption 3.1.8, L is \mathcal{C}^1 , then h in (3.36) and κ in (3.34) are also continuous. Furthermore, since by Assumption 3.1.3 ∇L is Lipschitz continuous, then h in (3.36) and κ in (3.34) are Lipschitz continuous which, in turn, means the map $z \mapsto F_P(z, \kappa(h(z, \tau), \tau))$ is Lipschitz continuous. Consequently, since the map $z \mapsto F_P(z, \kappa(h(z, \tau), \tau))$ is Lipschitz continuous and since the solution component τ of (3.37) increases linearly, then by [77, Theorem 3.2], (3.37) has no finite escape time from $\mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$ and each maximal solution to \mathcal{H}_0 is unique. Therefore, each maximal solution to (3.37), is complete and unique. \square

To analyze the convergence and stability properties of (3.37), we use the Lyapunov function

$$V_1(z, \tau) := \frac{1}{2} |\bar{a}(\tau) (z_1 - z_1^*) + z_2|^2 + \frac{\zeta^2}{M} (L(z_1) - L^*) \quad (3.38)$$

defined for each $z \in \mathbb{R}^{2n}$ and each $\tau \geq 0$, where $\zeta > 0$, $M > 0$ is the Lipschitz

constant of ∇L , and the function \bar{a} is defined as

$$\bar{a}(\tau) := \frac{2}{\tau + 2}. \quad (3.39)$$

When L satisfies Assumptions 3.1.8 and 3.1.3, then we can derive an upper bound, for all $t \geq 0$, on the Lyapunov function in (3.38) along solutions to (3.37). To derive such a bound, we extend [12, Proposition 3.2], which assumes $L^* = 0$ and $z_1^* = 0$, to the general case of $L^* \in \mathbb{R}$ and a single minimizer $z_1^* \in \mathbb{R}^n$, in the following proposition.

Proposition 3.1.11. *Let L satisfy Assumptions 3.1.8 and 3.1.3. Then, each maximal solution $t \mapsto (z(t), \tau(t))$ to the closed-loop algorithm (3.37) with $\tau(0) = 0$ satisfies*

$$V_1(z(t), t) \leq \frac{4}{(t + 2)^2} V_1(z(0), 0) \quad (3.40)$$

for all $t \geq 0$, where V_1 is defined via (3.38).

Proof. The Lyapunov function V_1 , defined via (3.38), is positive definite with respect to \mathcal{A}_1 , defined via (3.64), since, by Assumption 3.1.8, L is \mathcal{C}^1 , nonstrongly convex, and has a unique minimizer z_1^* . Then, letting

$$\bar{v}_1(z, \tau) := z_1 + \bar{\beta}(\tau)z_2, \quad (3.41)$$

letting $\varphi(z, \tau) := \bar{a}(\tau) (\bar{a}(\tau) (z_1 - z_1^*) + z_2) + \frac{\xi^2}{M} \nabla L(z_1)$, and since $\nabla V_1(z, \tau) = \left[\varphi(z, \tau) \quad (\bar{a}(\tau) (z_1 - z_1^*) + z_2) \quad \frac{d\bar{a}(\tau)}{d\tau} \langle z_1 - z_1^*, (\bar{a}(\tau) (z_1 - z_1^*) + z_2) \rangle \right]$, we evaluate the derivative of V_1 , using the map $z \mapsto F_P(z, \kappa_1(h(z, \tau), \tau))$, where F_P is

defined in (3.1), κ is defined via (3.34), and h is defined in (3.36), to yield

$$\begin{aligned}
\dot{V}_1(z, \tau) &= \left\langle \nabla V_1(z, \tau), \begin{bmatrix} F_P(z, \kappa_1(h(z, \tau), \tau)) \\ 1 \end{bmatrix} \right\rangle \\
&= \left\langle \nabla V_1(z, \tau), \begin{bmatrix} z_2 \\ -2\bar{d}(\tau)z_2 - \frac{\zeta^2}{M}\nabla L(\bar{v}_1(z, \tau)) \\ 1 \end{bmatrix} \right\rangle \\
&= \bar{a}(\tau) \langle \bar{a}(\tau) (z_1 - z_1^*) + z_2, z_2 \rangle + \frac{\zeta^2}{M} \langle z_2, \nabla L(z_1) \rangle - 2\bar{d}(\tau) |z_2|^2 \\
&\quad - 2\bar{d}(\tau)\bar{a}(\tau) \langle z_1 - z_1^*, z_2 \rangle - \frac{\bar{a}(\tau)\zeta^2}{M} \langle z_1 - z_1^*, \nabla L(\bar{v}_1(z, \tau)) \rangle \\
&\quad - \frac{\zeta^2}{M} \langle z_2, \nabla L(\bar{v}_1(z, \tau)) \rangle + \bar{a}(\tau) \frac{d\bar{a}(\tau)}{d\tau} |z_1 - z_1^*|^2 + \frac{d\bar{a}(\tau)}{d\tau} \langle z_1 - z_1^*, z_2 \rangle \\
&= -\frac{\bar{a}(\tau)\zeta^2}{M} \langle z_1 - z_1^*, \nabla L(\bar{v}_1(z, \tau)) \rangle + \bar{a}(\tau) \frac{d\bar{a}(\tau)}{d\tau} |z_1 - z_1^*|^2 \\
&\quad + \left(\bar{a}(\tau) - 2\bar{d}(\tau) \right) |z_2|^2 + \left(\bar{a}^2(\tau) - 2\bar{d}(\tau)\bar{a}(\tau) + \frac{d\bar{a}(\tau)}{d\tau} \right) \langle z_1 - z_1^*, z_2 \rangle \\
&\quad - \frac{\zeta^2}{M} \langle z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle \tag{3.42}
\end{aligned}$$

for all $(z, \tau) \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$. Since L is \mathcal{C}^1 , nonstrongly convex, and has a unique minimizer by Assumption 3.1.8, then using the definition of nonstrong convexity in Definition 2.2.2 with $u_1 = z_1^*$ and $w_1 = \bar{v}_1(z, \tau)$, where \bar{v}_1 is defined via (3.41), we get

$$-\langle \bar{v}_1(z, \tau) - z_1^*, \nabla L(\bar{v}_1(z, \tau)) \rangle \leq -(L(\bar{v}_1(z, \tau)) - L^*) \tag{3.43}$$

for each $z \in \mathbb{R}^{2n}$ and $\tau \in \mathbb{R}_{\geq 0}$. Using the definition of nonstrong convexity in Definition 2.2.2 with $u_1 = \bar{v}_1(z, \tau)$, where \bar{v}_1 is defined via (3.41), and $w_1 = z_1$

yields

$$\langle \nabla L(z_1), \bar{\beta}(\tau)z_2 \rangle \leq L(\bar{v}_1(z, \tau)) - L(z_1) \quad (3.44)$$

for each $z \in \mathbb{R}^{2n}$ and $\tau \in \mathbb{R}_{\geq 0}$. Combining (3.43) and (3.44) yields

$$-\langle \bar{v}_1(z, \tau) - z_1^*, \nabla L(\bar{v}_1(z, \tau)) \rangle + \langle \nabla L(z_1), \bar{\beta}(\tau)z_2 \rangle \leq -L(\bar{v}_1(z, \tau)) + L(\bar{v}_1(z, \tau)) - L(z_1) + L^*. \text{ Then, rearranging terms gives, for all } z \in \mathbb{R}^{2n} \text{ and } \tau \in \mathbb{R}_{\geq 0},$$

$$\begin{aligned} & -\langle z_1 - z_1^*, \nabla L(\bar{v}_1(z, \tau)) \rangle \\ & \leq -(L(z_1) - L^*) + \langle \bar{\beta}(\tau)z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle. \end{aligned} \quad (3.45)$$

Substituting the bound in (3.45) into (3.42) yields

$$\begin{aligned} \dot{V}_1(z, \tau) & \leq -\frac{\bar{a}(\tau)\zeta^2}{M} \left((L(z_1) - L^*) - \langle \bar{\beta}(\tau)z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle \right) \\ & \quad + \bar{a}(\tau) \frac{d\bar{a}(\tau)}{d\tau} |z_1 - z_1^*|^2 + (\bar{a}(\tau) - 2\bar{d}(\tau)) |z_2|^2 \\ & \quad + \left(\bar{a}^2(\tau) - 2\bar{d}(\tau)\bar{a}(\tau) + \frac{d\bar{a}(\tau)}{d\tau} \right) \langle z_1 - z_1^*, z_2 \rangle \\ & \quad - \frac{\zeta^2}{M} \langle z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle \end{aligned} \quad (3.46)$$

for all $(z, \tau) \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$. Then, noticing that $\frac{\bar{a}(\tau)}{2} |\bar{a}(\tau)(z_1 - z_1^*) + z_2|^2 = \frac{\bar{a}^3(\tau)}{2} |z_1 - z_1^*|^2 + \bar{a}^2(\tau) \langle z_1 - z_1^*, z_2 \rangle + \frac{\bar{a}(\tau)}{2} |z_2|^2$, adding it to and subtracting it from (3.46), and rearranging terms, yields

$$\begin{aligned} \dot{V}_1(z, \tau) & \leq -\bar{a}(\tau)V_1(z, \tau) + \bar{a}(\tau) \frac{d\bar{a}(\tau)}{d\tau} |z_1 - z_1^*|^2 + (\bar{a}(\tau) - 2\bar{d}(\tau)) |z_2|^2 \\ & \quad + \left(\bar{a}^2(\tau) - 2\bar{d}(\tau)\bar{a}(\tau) + \frac{d\bar{a}(\tau)}{d\tau} \right) \langle z_1 - z_1^*, z_2 \rangle + \frac{\bar{a}^3(\tau)}{2} |z_1 - z_1^*|^2 \\ & \quad + \frac{\bar{a}(\tau)}{2} |z_2|^2 + \bar{a}^2(\tau) \langle z_1 - z_1^*, z_2 \rangle \end{aligned}$$

$$\begin{aligned}
& - \frac{\zeta^2}{M} \left(1 - \bar{\beta}(\tau)\bar{a}(\tau)\right) \langle z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle \\
\leq & -\bar{a}(\tau)V_1(z, \tau) + \left(\frac{\bar{a}^3(\tau)}{2} + \bar{a}(\tau)\frac{d\bar{a}(\tau)}{d\tau}\right) |z_1 - z_1^*|^2 \\
& + \left(\frac{3\bar{a}(\tau)}{2} - 2\bar{d}(\tau)\right) |z_2|^2 \\
& + \left(2\bar{a}^2(\tau) - 2\bar{d}(\tau)\bar{a}(\tau) + \frac{d\bar{a}(\tau)}{d\tau}\right) \langle z_1 - z_1^*, z_2 \rangle \\
& - \frac{\zeta^2}{M} \left(1 - \bar{\beta}(\tau)\bar{a}(\tau)\right) \langle z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle
\end{aligned} \tag{3.47}$$

for all $(z, \tau) \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$. Due to the definitions of the functions \bar{a} and \bar{d} , in (3.39) and (3.35), respectively, the cross term $\langle z_1 - z_1^*, z_2 \rangle$ vanishes since $2\bar{a}^2(\tau) - 2\bar{d}(\tau)\bar{a}(\tau) + \frac{d\bar{a}(\tau)}{d\tau} = 2\left(\frac{2}{\tau+2}\right)^2 - 2\left(\frac{3}{2(\tau+2)}\right)\left(\frac{2}{\tau+2}\right) - \frac{2}{(\tau+2)^2} = 0$. Moreover, the definitions of the functions \bar{d} and \bar{a} lead to the $|z_1 - z_1^*|^2$ and $|z_2|^2$ terms in (3.47) vanishing due to $\frac{\bar{a}^3(\tau)}{2} + \bar{a}(\tau)\frac{d\bar{a}(\tau)}{d\tau} = \frac{\left(\frac{2}{\tau+2}\right)^3}{2} + \left(\frac{2}{\tau+2}\right)\left(-\frac{2}{(\tau+2)^2}\right) = 0$ and $\frac{3\bar{a}(\tau)}{2} - 2\bar{d}(\tau) = \frac{3\left(\frac{2}{\tau+2}\right)}{2} - 2\left(\frac{3}{2(\tau+2)}\right) = 0$. The bound in (3.47) reduces to

$$\dot{V}_1(z, \tau) \leq -\bar{a}(\tau)V_1(z, \tau) - \frac{\zeta^2}{M} \left(1 - \bar{\beta}(\tau)\bar{a}(\tau)\right) \langle z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle \tag{3.48}$$

for all $(z, \tau) \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$. By Assumption 3.1.8, L is \mathcal{C}^1 and nonstrongly convex. By [52, Theorem 2.1.3], a function L is \mathcal{C}^1 and nonstrongly convex if and only if, for each $w_1, u_1 \in \mathbb{R}^n$,

$$\langle \nabla L(w_1) - \nabla L(u_1), w_1 - u_1 \rangle \geq 0. \tag{3.49}$$

Then, since $\bar{\beta}(\tau) \geq 0$ for all $t \geq 0$, using the bound in (3.49) with $w_1 = \bar{v}_1(z, \tau)$,

where \bar{v}_1 is defined in (3.41), and $u_1 = z_1$, we get, for all $z \in \mathbb{R}^{2n}$ and $\tau \in \mathbb{R}_{\geq 0}$,

$$\begin{aligned}
& \langle \bar{v}_1(z, \tau) - z_1, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle = \\
& \quad \bar{\beta}(\tau) \langle z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle \geq 0 \\
& \quad -\bar{\beta}(\tau) \langle z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle \leq 0 \\
& \quad - \langle z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle \leq 0
\end{aligned} \tag{3.50}$$

Therefore, since $1 - \bar{\beta}(\tau)\bar{a}(\tau) \geq 0$, due to \bar{a} , defined via (3.39), equaling 1 at $\tau = 0$ and monotonically decreasing toward zero (but being always positive) as τ tends to ∞ , and due to $\bar{\beta}$, defined via (3.35), equaling 0 at $\tau = 0$ and monotonically increasing to 1 as τ tends to ∞ , we use (3.50) to upper bound the last term of (3.48) as follows:

$$-\frac{\zeta^2}{M} \left(1 - \bar{\beta}(\tau)\bar{a}(\tau)\right) \langle z_2, \nabla L(\bar{v}_1(z, \tau)) - \nabla L(z_1) \rangle \leq 0 \tag{3.51}$$

This leads to, $z \in \mathbb{R}^{2n}$ and $\tau \in \mathbb{R}_{\geq 0}$,

$$\dot{V}_1(z, \tau) \leq -\bar{a}(\tau)V_1(z(\tau), \tau). \tag{3.52}$$

Applying Grönwall's Inequality to (3.52), namely,

$$\begin{aligned}
V_1(z(t), t) & \leq V_1(z(0), 0) \exp\left(-\int_0^t \bar{a}(\tau) d\tau\right) \\
& = V_1(z(0), 0) \exp(-2 \ln(t+2) - 2 \ln(2)) \\
& = V_1(z(0), 0) \exp\left(-\ln\left(\frac{t+2}{2}\right)^2\right) \\
& = V_1(z(0), 0) \left(\frac{1}{\exp\left(\ln\left(\frac{t+2}{2}\right)^2\right)}\right)
\end{aligned}$$

$$= \frac{4}{(t+2)^2} V_1(z(0), 0)$$

shows that each maximal solution $t \mapsto (z(t), \tau(t))$ to the closed-loop algorithm \mathcal{H}_1 , such that $\tau(0) = 0$, satisfies (3.40), for all $t \geq 0$.

□

The following proposition establishes that the closed-loop algorithm (3.37) has a convergence rate $\frac{1}{(t+2)^2}$ for all $t \geq 0$. To prove it, we use Proposition 3.1.11. This theorem is a new result, which was not analyzed in [12].

Proposition 3.1.12. *(Convergence rate for (3.37)) Let L satisfy Assumptions 3.1.8 and 3.1.3. Let $\zeta > 0$ and $M > 0$ come from Assumption 3.1.3. Then, for each maximal solution $t \mapsto (z(t), \tau(t))$ to the closed-loop algorithm (3.37) with $\tau(0) = 0$, the following holds:*

$$\begin{aligned} & \frac{\zeta^2}{M} (L(z_1(t)) - L^*) & (3.53) \\ & \leq V_1(z(t), t) \leq \frac{4c}{(t+2)^2} (|z_1(0) - z_1^*|^2 + |z_2(0)|^2) \end{aligned}$$

for all $t \geq 0$, where $c := (1 + \zeta^2) \exp\left(\sqrt{\frac{13}{4} + \frac{\zeta^4}{M}}\right)$.

Proof. The proof consists of the following steps.

- 1) First, we use the definition of nonstrong convexity in Definition 2.2.2 and the Lipschitz continuity of ∇L in Assumption 3.1.3, to show that V_1 satisfies

$$V_1(z, \tau) \leq \alpha_2 |z|_{\mathcal{A}_2}^2 := (1 + \zeta^2) |z|_{\mathcal{A}_2}^2 \quad (3.54)$$

where $1 + \zeta^2 > 0$;

- 2) Then, we use the Lipschitz continuity of ∇L in Assumption 3.1.3 and the comparison principle to show that the bound in step 1) along $t \mapsto z(t)$ satisfies

$V_1(z(t), t) \leq (1 + \zeta^2) \exp\left(2\left(\sqrt{\frac{13}{4} + \frac{\zeta^4}{M}}\right)t\right) (|z_1(0) - z_1^*|^2 + |z_2(0)|^2)$ for all $t \geq 0$;

3) Next, we show that at $t = 0$, $V_1(z(0), 0)$ is upper bounded by

$$c (|z_1(0) - z_1^*|^2 + |z_2(0)|^2), \text{ where } c = (1 + \zeta^2) \exp\left(\sqrt{\frac{13}{4} + \frac{\zeta^4}{M}}\right);$$

4) Finally, we combine the bound in 3) with (3.40) to get (3.53) for all $t \geq 0$.

Proceeding with step 1), the Lyapunov function V_1 , defined via (3.38), can be upper bounded by a class- \mathcal{K}_∞ function, namely, defining the set

$$\mathcal{A}_2 := \{z_1^*\} \times \{0\} \tag{3.55}$$

then, V_1 satisfies

$$V_1(z, \tau) \leq \alpha_2 |z|_{\mathcal{A}_2}^2 \tag{3.56}$$

for all $(z, \tau) \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$, and with α_2 derived as follows. Since \bar{a} , defined via (3.39), equals 1 at $\tau = 0$ and \bar{a} is monotonically decreasing toward zero (but being always positive) as τ tends to ∞ , then \bar{a} is upper bounded by 1, and, consequently, the first term of V_1 can be upper bounded, for all $(z, \tau) \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$, as follows:

$$\frac{1}{2} \left| \frac{2}{(\tau + 2)} (z_1 - z_1^*) + z_2 \right|^2 \leq |z_1 - z_1^*|^2 + |z_2|^2. \tag{3.57}$$

The second term of V_1 can be bounded as follows. Since by Assumption 3.1.8, L is \mathcal{C}^1 , (nonstrongly) convex, and has a single minimizer z_1^* , then, since $\nabla L(z_1^*) = 0$, we can upper bound $L(z_1) - L^*$ in the following manner, using the definition of nonstrong convexity in Definition 2.2.2 and the Lipschitz continuity of ∇L in Assumption 3.1.3, using $u_1 = z_1^*$ and $w_1 = z_1$: $|L(z_1) - L^*| \leq |\langle \nabla L(z_1), z_1^* - z_1 \rangle| \leq |\nabla L(z_1)| |z_1 - z_1^*| \leq M |z_1 - z_1^*|^2$, for each $z_1 \in \mathbb{R}^n$ and each $\tau \in \mathbb{R}_{\geq 0}$. Therefore,

since $L(z_1) \geq L^*$, we can upper bound the second term of V_1 as follows:

$$\frac{\zeta^2}{M}(L(z_1) - L^*) \leq \zeta^2 |z_1 - z_1^*|^2 \leq \zeta^2 (|z_1 - z_1^*|^2 + |z_2|^2) \quad (3.58)$$

for all $z \in \mathbb{R}^{2n}$. Using (3.57) and (3.58), $V_1(z, \tau)$ is upper bounded as in (3.54) for each $z \in \mathbb{R}^{2n}$ and each $\tau \in \mathbb{R}_{\geq 0}$.

Next, for step 2), in order to apply the comparison principle, we define the system

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} z_2 \\ -2\bar{d}(t)z_2 - \frac{\zeta^2}{M}\nabla L(z_1 + \bar{\beta}(t)z_2) \end{bmatrix} =: f(z, t) \quad z \in \mathbb{R}^{2n}. \quad (3.59)$$

Since ∇L is Lipschitz continuous with constant $M > 0$ by Assumption 3.1.3, then using Assumption 3.1.3 with $w_1 = z_1 + \bar{\beta}(t)z_2$ and $u_1 = z_1^*$ yields, for each $z_1, z_2 \in \mathbb{R}^n$ and each $t \in \mathbb{R}_{\geq 0}$,

$$|\nabla L(z_1 + \bar{\beta}(t)z_2)| \leq M |z_1 - z_1^* + \bar{\beta}(t)z_2|. \quad (3.60)$$

Then, since $|\bar{d}(t)| \leq \frac{3}{4}$ and $|\bar{\beta}(t)| \leq 1$ for all $t \geq 0$, we have

$$\begin{aligned} |f(z, t)|^2 &= |z_2|^2 + \left| -2\bar{d}(t)z_2 - \frac{\zeta^2}{M}\nabla L(z_1 + \bar{\beta}(t)z_2) \right|^2 \\ &\leq |z_2|^2 + \frac{9}{4}|z_2|^2 + \frac{\zeta^4}{M^2} |\nabla L(z_1 + \bar{\beta}(t)z_2)|^2 \\ &\leq \frac{13}{4}|z_2|^2 + \frac{\zeta^4}{M} (|z_1 - z_1^*|^2 + |z_2|^2) \\ &= \frac{\zeta^4}{M} |z_1 - z_1^*|^2 + \left(\frac{13}{4} + \frac{\zeta^4}{M} \right) |z_2|^2 \\ &\leq \left(\frac{13}{4} + \frac{\zeta^4}{M} \right) |z|_{\mathcal{A}_2}^2 \end{aligned} \quad (3.61)$$

for all $z \in \mathbb{R}^{2n}$ and all $t \in \mathbb{R}_{\geq 0}$, where \mathcal{A}_2 is defined via (3.55). The second

inequality in (3.61) comes from applying (3.60). The comparison principle [77, Lemma 3.4], leads to the following bound of the norm of the solution to (3.59):

$$|z(t)|_{\mathcal{A}_2} \leq \exp\left(\frac{1}{2}\sqrt{\frac{13}{4} + \frac{\zeta^4}{M}}t\right) |z(0)|_{\mathcal{A}_2} \quad (3.62)$$

for all $t \geq 0$. Then, (3.56) along $t \mapsto (z(t))$ reduces to, for all $t \geq 0$

$$\begin{aligned} V_1(z(t), t) &\leq (1 + \zeta^2) |z(t)|_{\mathcal{A}_2}^2 \\ &\leq (1 + \zeta^2) \exp\left(\sqrt{\frac{13}{4} + \frac{\zeta^4}{M}}t\right) (|z_1(0) - z_1^*|^2 + |z_2(0)|^2). \end{aligned} \quad (3.63)$$

In step 3), we evaluate this bound at $t = 0$. Finally, for step 4), taking $c = (1 + \zeta^2) \exp\left(\sqrt{\frac{13}{4} + \frac{\zeta^4}{M}}\right)$, combining (3.40) with 3) at $t = 0$ yields (3.53) for all $t \geq 0$. \square

The following proposition establishes that the closed-loop system (3.37) has the set

$$\mathcal{A}_1 := \{z_1^*\} \times \{0\} \times \mathbb{R}_{\geq 0} \quad (3.64)$$

uniformly globally asymptotically stable. To prove it, we use Proposition 3.1.12 and [21, Theorem 3.18]. This proposition is a new result, which was not analyzed in [12].

Proposition 3.1.13. *(UGAS of \mathcal{A}_1 in (3.64) for (3.37)) Let L satisfy Assumptions 3.1.8 and 3.1.3. Let $\zeta > 0$ and let $M > 0$ come from Assumption 3.1.3. Then, the set \mathcal{A}_1 in (3.64) is uniformly globally asymptotically stable for (3.37).*

Proof. By Proposition 3.1.10, each maximal solution to \mathcal{H}_1 in (3.37) is complete and unique. Next, since L is \mathcal{C}^1 , nonstrongly convex, and has a unique minimizer by Assumption 3.1.8, then $\mathcal{A}_1 \subset \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$, defined via (3.64), is closed by

construction, satisfying the first assumption of [21, Theorem 3.18]. Then, since by Assumption 3.1.8, L is \mathcal{C}^1 , then V_1 in (3.38) is continuously differentiable and therefore, since $\mathbb{R}^{2n} \times \mathbb{R}_{\geq 0} \subset \text{dom } V_1$, V_1 is a Lyapunov function candidate for \mathcal{H}_1 by [21, Definition 3.16], satisfying the second assumption of [21, Theorem 3.18].

Next, since the distance of the state τ to $\mathbb{R}_{\geq 0}$ is always zero, then we show that V_1 in (3.38) is radially unbounded in z , relative to \mathcal{A}_2 , defined via (3.55). Since L has quadratic growth away from z_1^* with constant $\alpha > 0$, by Assumption 3.2.4, and due to \bar{a} , defined via (3.39), equaling 1 at $\tau = 0$ and monotonically decreasing toward zero (but being always positive) as τ tends to ∞ , then we lower bound V_1 as follows:

$$\begin{aligned}
V_1(z, \tau) &= \frac{1}{2} |\bar{a}(\tau) (z_1 - z_1^*) + z_2|^2 + \frac{\zeta^2}{M} (L(z_1) - L^*) & (3.65) \\
&\geq \frac{1}{2} |\bar{a}(\tau) (z_1 - z_1^*) + z_2|^2 + \frac{\alpha \zeta^2}{M} |z_1 - z_1^*|^2 \\
&\geq \frac{\bar{a}^2(\tau)}{2} |z_1 - z_1^*|^2 + \bar{a}(\tau) \langle z_1 - z_1^*, z_2 \rangle + \frac{1}{2} |z_2|^2 + \frac{\alpha \zeta^2}{M} |z_1 - z_1^*|^2 \\
&\geq \left(\frac{\bar{a}^2(\tau)}{2} + \frac{\alpha \zeta^2}{M} \right) |z_1 - z_1^*|^2 + \frac{\bar{a}(\tau)}{2} \langle z_1 - z_1^*, z_2 \rangle \\
&\quad + \frac{\bar{a}(\tau)}{2} \langle z_1 - z_1^*, z_2 \rangle + \frac{1}{2} |z_2|^2 \\
&\geq \begin{bmatrix} (z_1 - z_1^*)^\top & z_2^\top \end{bmatrix} B \begin{bmatrix} z_1 - z_1^* \\ z_2 \end{bmatrix}
\end{aligned}$$

for each $z \in \mathbb{R}^{2n}$ and each $\tau \in \mathbb{R}_{\geq 0}$, where

$$B := \begin{bmatrix} \left(\frac{\bar{a}^2(\tau)}{2} + \frac{\alpha \zeta^2}{M} \right) & \frac{\bar{a}(\tau)}{2} \\ \frac{\bar{a}(\tau)}{2} & \frac{1}{2} \end{bmatrix}. \quad (3.66)$$

Next, we show that B in (3.66) is positive definite, so that there exists α_1 such

that³

$$\alpha_1 |z|_{\mathcal{A}_2}^2 \leq \begin{bmatrix} (z_1 - z_1^*)^\top & z_2^\top \end{bmatrix} B \begin{bmatrix} z_1 - z_1^* \\ z_2 \end{bmatrix} \leq V_1(z, \tau) \quad (3.67)$$

for each $z \in \mathbb{R}^{2n}$ and each $\tau \in \mathbb{R}_{\geq 0}$. To that end, we show that the leading principal minors of B in (3.66) are strictly positive, as follows. Since $\bar{a}(\tau) \in (0, 1]$ for each $\tau \in \mathbb{R}_{\geq 0}$, $\alpha > 0$ from Assumption 3.2.4, $\zeta > 0$, and $M > 0$ from Assumption 3.1.3, we have:

$$\left(\frac{\bar{a}^2(\tau)}{2} + \frac{\alpha\zeta^2}{M} \right) > 0 \quad (3.68a)$$

$$\det(B) = \left(\frac{1}{2} \right) \left(\frac{\bar{a}^2(\tau)}{2} + \frac{\alpha\zeta^2}{M} \right) - \left(\frac{\bar{a}(\tau)}{2} \right)^2 = \frac{\alpha\zeta^2}{2M} > 0. \quad (3.68b)$$

Therefore, since the leading principal minors of B are strictly positive, then B is positive definite. Hence, there exists α_1 such that (3.67) is true, and V_1 is radially unbounded in z , relative to \mathcal{A}_2 .

By Proposition 3.1.11, V_1 satisfies (3.52) for each $z \in \mathbb{R}^{2n}$ and $\tau \in \mathbb{R}_{\geq 0}$. Since L is \mathcal{C}^1 , nonstrongly convex, and has a unique minimizer by Assumption 3.1.8, then L is positive definite with respect to z_1^* and, consequently, V_1 is positive definite with respect to \mathcal{A}_1 in (3.64). Then, since $\bar{a}(\tau) \in (0, 1]$ for each $\tau \geq 0$, $\rho(|x|_{\mathcal{A}_1}) := \bar{a}(\tau)V_1(z, \tau)$ is positive definite with respect to \mathcal{A}_1 . Therefore, by an application of [21, Theorem 3.18], every complete solution to (3.37) converges to \mathcal{A}_1 in (3.64). The arguments above involving the Lyapunov theorem in [21, Theorem 3.18] yield UGPAS of \mathcal{A}_1 for (3.37). Since by Proposition 3.1.10, each maximal solution to (3.37) is complete, then \mathcal{A}_1 is uniformly globally asymptotically stable for (3.37). \square

³It was already shown that there exists α_2 such that the upper bound on V_1 in (3.54) holds.

3.1.3 Extensions of the Results for Nonstrongly Convex L

Some possible extensions to the results in Section 3.1.2 are as follows.

It is possible to extend the results in Section 3.1.2 to include \mathcal{C}^1 , nonstrongly convex objective functions L with a compact and connected set of minimizers. With such an assumption, it would be straightforward to extend Proposition 3.1.10. Propositions 3.1.11, 3.1.12, and 3.1.13 could be extended via the assumption of a compact and connected set of minimizers and the use of Clarke's generalized derivative in (2.4) with the Lyapunov function

$$V_1(z, \tau) := \frac{1}{2} \left| \bar{a}(\tau) \nabla_{z_1} |z_1|_{\mathcal{A}_1} + z_2 \right|^2 + \frac{\zeta^2}{M} (L(z_1) - L^*) \quad (3.69)$$

where the compact and connected set \mathcal{A}_3 is defined as

$$\mathcal{A}_3 := \{ z_1 \in \mathbb{R}^n : \nabla L(z_1) = 0 \} \quad (3.70)$$

and⁴ $L^* := L(\mathcal{A}_3)$. Such an extension would lead to UGAS of the set \mathcal{A}_1 in (3.64) for (3.37) when $\tau(0) = 0$, for all $t \geq 0$, as well as an exponential convergence rate for all $t \geq 0$, when \mathcal{A}_3 is a compact and connected set of minimizers.

It would be possible to further extend the results in Propositions 3.1.10, 3.1.11, 3.1.12, and 3.1.13 to include \mathcal{C}^1 , nonstrongly convex objective functions L that are also nonsmooth, through the use of Clarke's generalized derivative.

⁴Since the value of L is the same for all $z_1^* \in \mathcal{A}_3$, $L(\mathcal{A}_3)$ is a singleton.

3.2 The Heavy Ball Method Modeled as a Dynamical System

The control algorithm leading to (1.1) is

$$u = \kappa(h(z)) = -\lambda z_2 - \gamma \nabla L(z_1) \quad (3.71)$$

where $\lambda > 0$ and $\gamma > 0$. The function h is defined differently, based on the different algorithms proposed in this dissertation. For some such algorithms, h is defined as

$$h(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1) \end{bmatrix} \quad (3.72)$$

while for others, h is defined as

$$h(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1) \\ L(z_1) \end{bmatrix}. \quad (3.73)$$

Using the plant in (3.1), we denote the closed-loop system resulting from κ in (3.71) as

$$\dot{z} = \begin{bmatrix} z_2 \\ \kappa(h(z)) \end{bmatrix} \quad z \in \mathbb{R}^{2n}. \quad (3.74)$$

3.2.1 Strongly Convex L

For the analysis in this section, we impose Assumptions 3.1.1 and 3.1.3 on L .

The following lemma, from [13], summarizes some of the results of Lyapunov theory from an optimization perspective, for strongly convex L . It employs the

following Lyapunov function, proposed in [13]

$$V(z) := L(z_1) - L^*. \quad (3.75)$$

Lemma 3.2.1. *Let L satisfy Assumption 3.1.3 and assume L is bounded below, namely, $L(z_1) \geq L^*$. Let the function \mathcal{W} be such that $-\dot{V}(z(t)) \geq \mathcal{W}(z(t)) \geq 0$ for all $t \geq 0$. Let $\mathcal{W}(z(t)) \geq \eta V(z(0))$ for all $t \geq 0$, where $\eta > 0$. Then, each maximal solution $t \mapsto z(t)$ to (3.74) satisfies*

$$V(z(t)) \leq V(z(0)) \exp(-\eta t) \quad (3.76)$$

for all $t \geq 0$.

Proof. Since $\mathcal{W}(z(t)) \geq \eta V(z(0))$ for all $t \geq 0$, and since $-\dot{V}(z(t)) \geq \mathcal{W}(z(t))$, this means that $\dot{V}(z(t)) \leq \eta V(z(0))$. Applying Grönwall's inequality to $\dot{V}(z(t)) \leq \eta V(z(0))$, we arrive at (3.76). \square

The following proposition, from [13], gives an exponential convergence rate for functions that satisfy the Polyak-Łojasiewicz condition in Definition 2.2.4, which is a weaker condition than strong convexity [68] [71]. To prove it, we use Lemma 3.2.1.

Proposition 3.2.2. *Let L satisfy the Polyak-Łojasiewicz condition in Definition 2.2.4. Let the function \mathcal{W} be such that $-\dot{V}(z(t)) \geq \mathcal{W}(z(t)) \geq 0$ for all $t \geq 0$. Let $\mathcal{W}(z(t)) \geq \eta V(z(0))$ for all $t \geq 0$, where $\eta > 0$. Then, each maximal solution $t \mapsto (z_1(t), z_2(t))$ to (3.74) satisfies*

$$L(z_1(t)) - L^* \leq (L(z_1(0)) - L^*) \exp(-2\mu t) \quad (3.77)$$

for all $t \geq 0$, where $\mu > 0$ comes from Definitions 2.2.4 and 2.2.1.

Proof. By assumption, L satisfies the Polyak-Łojasiewicz condition in Definition 2.2.4. Then, since V is defined via (3.75), we have $\dot{V}(z(t)) = \nabla L(z_1(t))$. Letting $\mathcal{W}(z) := |\nabla L(z_1)|^2$ and $\eta := 2\mu$, by Lemma 3.2.1 we have

$$\begin{aligned} -\dot{V}(z(t)) &\geq \mathcal{W}(z(t)) \\ \dot{V}(z(t)) &\leq -\mathcal{W}(z(t)) \\ \dot{V}(z(t)) &\leq -\eta V(z(0)) \end{aligned} \tag{3.78}$$

Applying Grönwall's inequality to (3.78), we get that each solution $t \mapsto (z(t))$ of (3.74) also satisfies (3.77) for all $t \geq 0$. \square

When L satisfies Assumption 3.1.1, the following theorem gives an exponential convergence rate for the closed-loop algorithm in (3.74). To prove it, we use Proposition 3.2.2.

Proposition 3.2.3. *(Convergence rate of (3.74)) Let L satisfy Assumption 3.1.1. Then, each maximal solution $t \mapsto z(t)$ to (3.74) satisfies (3.77) for all $t \geq 0$.*

Proof. Since L is strongly convex with $\mu > 0$ by Assumption 3.1.1, then this implies that L also satisfies the Polyak-Łojasiewicz condition in Definition 2.2.4 with $\mu > 0$; see [68, Appendix B]. Therefore, by Proposition 3.2.2, each maximal solution $t \mapsto z(t)$ to (3.74) satisfies (3.77) for all $t \geq 0$. \square

3.2.2 Nonstrongly Convex L

For the analysis in this section, we impose Assumptions 3.1.8 and 3.1.3 on L . We also impose the following assumption on L for the results in this section.

Assumption 3.2.4 (Quadratic growth of L). *The function L has quadratic growth*

away from its minimizer z_1^* ; i.e., there exists $\alpha > 0$ such that

$$L(z_1) - L^* \geq \alpha |z_1 - z_1^*|^2 \quad \forall z_1 \in \mathbb{R}^n. \quad (3.79)$$

Remark 3.2.5. *Assumption 3.2.4 is commonly used in the analysis of convex optimization algorithms; see, e.g., [67], [68]. Such an assumption is employed to establish the convergence rate for (3.74), when L is nonstrongly convex. Additionally, Assumption 3.2.4 will be employed in some of our proposed hybrid algorithms to as a means of determining when the state z is near the minimizer of L , via measurements of ∇L .*

When L satisfies Assumptions 3.1.8 and 3.1.3, the closed-loop system in (3.74) satisfies the hybrid basic conditions, listed in Definition 2.1.1, as demonstrated in the following lemma. A closed-loop system that satisfies the hybrid basic conditions is said to be well-posed in the sense that the limit of a graphically convergent sequence of solutions to (3.74) having a mild boundedness property is also a solution to (3.74) [21].

Lemma 3.2.6. *(Well-posedness of (3.74)) Let the function L satisfy Assumptions 3.1.8 and 3.1.3. Let κ be defined via (3.71). Then, the closed-loop system in (3.74) satisfies the hybrid basic conditions.*

Proof. The set $C := \mathbb{R}^{2n}$ is closed, and the set $D := \emptyset$.

The objective function L is \mathcal{C}^1 , nonstrongly convex, and has a single minimizer by Assumption 3.1.8. Therefore, since ∇L is continuous, then h in both (3.72) and (3.73) and κ in (3.71) are continuous.

In turn, the map $z \mapsto F_P(z, \kappa(h(z)))$ is also continuous since F_P in (3.1) is a \mathcal{C}^1 function of κ in (3.71) and h in either (3.72) or (3.73). The map $G(z) := \emptyset$. \square

Under Assumptions 3.1.8, 3.1.3, and 3.2.4, each maximal solution to (3.74) is bounded, complete, and unique. Such a property is useful since it guarantees that nontrivial solutions to (3.74) exist from each initial point in \mathbb{R}^{2n} , and that such solutions do not escape \mathbb{R}^{2n} . When each maximal solution is complete, then uniform global pre-asymptotic stability of $\{z_1^*\} \times \{0\}$ becomes uniform global asymptotic stability.

Proposition 3.2.7. *(Existence of solutions to (3.74)) Let the function L satisfy Assumptions 3.1.8, 3.1.3, and 3.2.4. Let κ be defined via (3.71). Then, each maximal solution $t \mapsto z(t)$ to (3.74) is bounded, complete, and unique.*

Proof. Since L is \mathcal{C}^1 by Assumption 3.1.8, and ∇L is Lipschitz continuous by Assumption 3.1.3, then h in (3.72) and (3.73) and κ in (3.71) are Lipschitz continuous, which, since F_P is a \mathcal{C}^1 function of h and κ , means the map $z \mapsto F_P(z, \kappa(h(z)))$ is also Lipschitz continuous. Therefore, by [77, Theorem 3.2], $\dot{z} = F_P(z, \kappa(h(z)))$ has no finite time escape and each maximal solution to (3.74) is complete and unique. To show that each maximal solution to (3.74) is bounded, we use the Lyapunov function

$$V_0(z) := \gamma(L(z_1) - L^*) + \frac{1}{2}|z_2|^2 \quad (3.80)$$

defined for each $z \in \mathbb{R}^{2n}$, where $\gamma > 0$. Then, solutions to $\dot{z} = F_P(z, \kappa(h(z)))$ starting from any c_V -sublevel set $W := \{z \in \mathbb{R}^{2n} : V_0(z) \leq c_V\}$, $c_V \geq 0$, remains in such a set for all time since V_0 in (3.80) satisfies

$$\dot{V}_0(z) = \langle \nabla V(z), F_P(z, \kappa(h(z))) \rangle = -\lambda |z_2|^2 \leq 0 \quad (3.81)$$

for each $z \in \mathbb{R}^{2n}$, since λ is positive. Then, to show that V_0 in (3.81) is radially unbounded, we derive class- \mathcal{K}_∞ functions α_1 and α_2 such that, for all $z \in \mathbb{R}^{2n}$,

with $z^* := (z_1^*, 0)$,

$$\begin{aligned} \alpha_1(|z - z^*|) &:= \min \left\{ \alpha\gamma, \frac{1}{2} \right\} |z - z^*|^2 \leq V_0(z) \\ &\leq \alpha_2(|z - z^*|) := \left(M\gamma + \frac{1}{2} \right) |z - z^*|^2. \end{aligned} \quad (3.82)$$

Since L has quadratic growth away from z_1^* with constant $\alpha > 0$ by Assumption 3.2.4, then the choice of α_1 comes from lower bounding V_0 in (3.80) as follows

$$\begin{aligned} V_0(z) &= \gamma(L(z_1) - L^*) + \frac{1}{2}|z_2|^2 \geq \alpha\gamma|z_1 - z_1^*|^2 + \frac{1}{2}|z_2|^2 \\ &\geq \min \left\{ \alpha\gamma, \frac{1}{2} \right\} |z - z^*|^2 = \alpha_1(|z - z^*|) \end{aligned} \quad (3.83)$$

for each $z \in \mathbb{R}^{2n}$. The choice of α_2 comes from the following. Since L is \mathcal{C}^1 , nonstrongly convex, has a single minimizer by Assumption 3.1.8, and since ∇L is Lipschitz continuous with constant $M > 0$ by Assumption 3.1.3, we first upper bound V_0 in (3.80) by using the definition of nonstrong convexity in Definition 2.2.2 to get, for each $z \in \mathbb{R}^{2n}$,

$$V_0(z) = \gamma(L(z_1) - L^*) + \frac{1}{2}|z_2|^2 \leq \gamma|\nabla L(z_1)||z_1 - z_1^*| + \frac{1}{2}|z_2|^2. \quad (3.84)$$

Then, using the Lipschitz bound in Assumption 3.1.3 with $u_1 = z_1^*$ and $w_1 = z_1$, we upper bound (3.84), yielding

$$\begin{aligned} V_0(z) &= \gamma(L(z_1) - L^*) + \frac{1}{2}|z_2|^2 \leq \gamma|\nabla L(z_1)||z_1 - z_1^*| + \frac{1}{2}|z_2|^2 \\ &\leq M\gamma|z_1 - z_1^*|^2 + \frac{1}{2}|z_2|^2 \\ &\leq \left(M\gamma + \frac{1}{2} \right) |z - z^*|^2 = \alpha_2(|z - z^*|) \end{aligned} \quad (3.85)$$

for each $z \in \mathbb{R}^{2n}$. Since (3.82) is satisfied for V_0 in (3.80) for each $z \in \mathbb{R}^{2n}$, then

V_0 is radially unbounded (in z , relative to $\{z_1^*\} \times \{0\}$). Therefore, W is compact and, due to (3.81), forward invariant for (3.74), that is, any nontrivial solution starting in the subset W is complete and stays in W . Therefore, each maximal solution to (3.74), is bounded. \square

The following result establishes that the closed-loop algorithm (3.74) has the set $\{z_1^*\} \times \{0\}$ uniformly globally asymptotically stable. To prove it, we use an invariance principle.

Proposition 3.2.8. *(Uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for (3.74))*
Let L satisfy Assumptions 3.1.8, 3.2.4, and 3.1.3. For each $\lambda > 0$ and $\gamma > 0$, the set $\{z_1^\} \times \{0\}$ is uniformly globally asymptotically stable for the closed-loop algorithm (3.74).*

Proof. By Proposition 3.2.7, each maximal solution to the closed-loop algorithm in (3.74) is bounded, complete, and unique. Recall that, in the proof of Proposition 3.2.7, it was shown that V_0 in (3.80) satisfies (3.81) for all $z \in \mathbb{R}^{2n}$, since λ is positive. Therefore, by an application of Theorem A.1.3, since $\gamma > 0$ and $\lambda > 0$, the set $\{z_1^*\} \times \{0\}$ is stable for the closed-loop algorithm in (3.74). Since by Lemma 3.2.6 the closed-loop algorithm in (3.74) satisfies the hybrid basic conditions, then, using the invariance principle in Theorem A.1.6, each maximal solution that is complete and bounded approaches the largest weakly invariant set for the closed-loop algorithm in (3.74) that is contained in

$$\{z \in \mathbb{R}^{2n} : \dot{V}_0(z) = 0\} \cap \{z \in \mathbb{R}^{2n} : V_0(z) = r\}, \quad r \geq 0. \quad (3.86)$$

Such a set is nonempty only when $r = 0$ and, precisely, is equal to $\{z_1^*\} \times \{0\}$. This property can be seen by noticing that $\{z \in \mathbb{R}^{2n} : \dot{V}_0(z) = 0\} = \{z \in \mathbb{R}^{2n} : z_2 = 0\}$,

and that after setting z_2 to zero in (3.74) we obtain $\begin{bmatrix} \dot{z}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\gamma \nabla L(z_1) \end{bmatrix}$. For any solution to this system, its z_1 component satisfies $0 = \gamma \nabla L(z_1)$, which, since $\gamma > 0$ and since $\nabla L(z_1) = 0$ only when z_1 is the minimizer of L , leads to $z_1 = z_1^*$. Then, the only maximal solution that starts and stays in (3.86) is the solution from $\{z_1^*\} \times \{0\}$, for which $r = 0$. Then, every bounded and complete solution to the closed-loop algorithm in (3.74) converges to $\{z_1^*\} \times \{0\}$. The arguments above involving the Lyapunov theorem in Theorem A.1.3 and the invariance principle in Theorem A.1.6 yield global pre-asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_0 . Since by Proposition 3.2.7, each maximal solution to (3.74) is complete, then $\{z_1^*\} \times \{0\}$ is globally asymptotically stable for the closed-loop algorithm in (3.74). Since (3.74) satisfies the hybrid basic conditions by Lemma 3.2.6, then, by Theorem A.1.4, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for (3.74). \square

Next, we establish the convergence rate of the closed-loop algorithm in (3.74). To do so, we use the following Lyapunov function, proposed in [25, Lemma 4.2], for (3.74):

$$V(z) := \gamma(L(z_1) - L^*) + \frac{1}{2} |\psi(z_1 - z_1^*) + z_2|^2 + \frac{\nu}{2} |z_1 - z_1^*|^2 \quad (3.87)$$

where, given $\lambda > 0$, $\psi > 0$ is chosen such that $\nu := \psi(\psi - \lambda) < 0$. When L satisfies Assumption 3.1.8, the following lemma, which is a version of [25, Lemma 4.2] tailored for the unperturbed heavy ball algorithm in (3.74), gives an upper bound on the change of the Lyapunov function in (3.87).

Lemma 3.2.9. *Let L satisfy Assumption 3.1.8, and let $\lambda > 0$ and $\gamma > 0$, which come from (3.74), be given. For each $\psi > 0$ such that $\nu := \psi(\psi - \lambda) < 0$, the*

following bound is satisfied for each $z \in \mathbb{R}^{2n}$:

$$\dot{V}(z) \leq -\psi(a(z_1) + 2\nu c(z_1)) + 2(\psi - \lambda)b(z) \quad (3.88)$$

where V is defined in (3.87), $a(z_1) := \gamma(L(z_1) - L^*)$, $b(z) := \frac{1}{2}|\psi(z_1 - z_1^*) + z_2|^2$, and $c(z_1) := \frac{1}{2}|z_1 - z_1^*|^2$.

Proof. Since L is \mathcal{C}^1 , nonstrongly convex, and has a single minimizer z_1^* , and since $\nabla V(z) = [\gamma \nabla L(z_1) + \psi(\psi(z_1 - z_1^*) + z_2) + \nu(z_1 - z_1^*) \quad \psi(z_1 - z_1^*) + z_2]$, then we evaluate the derivative of V , defined via (3.87), using the map $z \mapsto F_P(\kappa(h(z)))$, where F_P is defined via (3.1), κ is defined in (3.71), and h is defined via either (3.72) or (3.73). For each $z \in \mathbb{R}^{2n}$, we obtain

$$\begin{aligned} \dot{V}(z) &= \langle \nabla V(z), F_P(\kappa(h(z))) \rangle = \left\langle \nabla V(z), \begin{bmatrix} z_2 \\ \kappa(h(z)) \end{bmatrix} \right\rangle \quad (3.89) \\ &= \gamma \langle \nabla L(z_1), z_2 \rangle + \psi \langle z_2, \psi(z_1 - z_1^*) + z_2 \rangle + \nu \langle z_2, z_1 - z_1^* \rangle \\ &\quad - \lambda \langle z_2, \psi(z_1 - z_1^*) + z_2 \rangle - \gamma \langle \nabla L(z_1), \psi(z_1 - z_1^*) + z_2 \rangle \\ &= -\gamma \psi \langle \nabla L(z_1), z_1 - z_1^* \rangle + (\nu + \psi(\psi - \lambda)) \langle z_2, z_1 - z_1^* \rangle + (\psi - \lambda) |z_2|^2. \end{aligned}$$

Note that $|\psi(z_1 - z_1^*) + z_2|^2 = |z_2|^2 + 2\psi \langle z_2, z_1 - z_1^* \rangle + \psi^2 |z_1 - z_1^*|^2$, from where we obtain $|z_2|^2 = |\psi(z_1 - z_1^*) + z_2|^2 - 2\psi \langle z_2, z_1 - z_1^* \rangle - \psi^2 |z_1 - z_1^*|^2$. Substituting the expression for $|z_2|^2$ into (3.89), we arrive at, for all $z \in \mathbb{R}^{2n}$,

$$\begin{aligned} \dot{V}(z) &= -\gamma \psi \langle \nabla L(z_1), z_1 - z_1^* \rangle + (\psi - \lambda) |\psi(z_1 - z_1^*) + z_2|^2 \\ &\quad + (\nu - \psi(\psi - \lambda)) \langle z_2, z_1 - z_1^* \rangle - \psi^2 (\psi - \lambda) |z_1 - z_1^*|^2 \\ &= -\gamma \psi \langle \nabla L(z_1), z_1 - z_1^* \rangle + 2(\psi - \lambda)b(z) - 2\psi \nu c(z_1) \quad (3.90) \end{aligned}$$

since $\nu = \psi(\psi - \lambda)$, where $b(z) = \frac{1}{2}|\psi(z_1 - z_1^*) + z_2|^2$ and $c(z_1) = \frac{1}{2}|z_1 - z_1^*|^2$.

Since L is \mathcal{C}^1 , nonstrongly convex, and has a unique minimizer by Assumption 3.1.8, then using the definition of nonstrong convexity in Definition 2.2.2 with $u_1 = z_1^*$ and $w_1 = z_1$, we get $-(L(z_1) - L^*) \geq -\langle \nabla L(z_1), z_1 - z_1^* \rangle$. Substituting it into (3.90) yields, for all $z \in \mathbb{R}^{2n}$, $\dot{V}(z) \leq -\psi a(z_1) + 2(\psi - \lambda)b(z) - 2\psi\nu c(z_1)$, where $a(z_1) = \gamma(L(z) - L^*)$, and (3.88) is satisfied. \square

We employ Lemma 5.2 to show that when L satisfies Assumptions 3.1.8 and 3.2.4, the convergence rate of the closed-loop algorithm in (3.74) is exponential. This is supported by the following proposition, which is a version of [25, Theorem 3.2] tailored for the unperturbed heavy ball algorithm in (3.74).

Proposition 3.2.10. *(Convergence rate for (3.74)) Let L satisfy Assumptions 3.1.8 and 3.2.4, let $\alpha > 0$ come from (3.79), and let $\lambda > 0$ and $\gamma > 0$ come from (3.74). For each $m \in (0, 1)$ such that $\psi := \frac{m\alpha\gamma}{\lambda} > 0$ and $\nu := \psi(\psi - \lambda) < 0$, each maximal solution $t \mapsto z(t)$ to the closed-loop algorithm in (3.74) satisfies*

$$L(z_1(t)) - L^* = \mathcal{O}(\exp(-(1-m)\psi t)) \quad \forall t \in \text{dom } z (= \mathbb{R}_{\geq 0}). \quad (3.91)$$

Proof. By Lemma 3.2.9, the bound in (3.88) is satisfied for V in (3.87) for each $z \in \mathbb{R}^{2n}$ since, by Assumption 3.1.8, L is \mathcal{C}^1 , nonstrongly convex, and has a single minimizer z_1^* . Then, since $\psi = \frac{m\alpha\gamma}{\lambda} > 0$ is such that $\nu = \psi(\psi - \lambda) < 0$ and c is nonnegative, this leads to

$$V(z) = a(z_1) + b(z) + \nu c(z_1) \leq a(z_1) + b(z) \quad \forall z \in \mathbb{R}^{2n} \quad (3.92)$$

where a , b , and c are defined below (3.88). By Assumption 3.2.4, L has quadratic growth away from z_1^* . Therefore, we have, for all $z \in \mathbb{R}^{2n}$,

$$a(z_1) + 2\nu c(z_1) = a(z_1) - 2|\nu|c(z_1) = \gamma(L(z_1) - L^*) - |\nu||z_1 - z_1^*|^2 \quad (3.93)$$

$$\geq \gamma (L(z_1) - L^*) - \frac{|\nu| (L(z_1) - L^*)}{\alpha} = \left(1 - \frac{|\nu|}{\alpha\gamma}\right) a(z_1).$$

Observe that, for each $m \in (0, 1)$ such that $\psi = \frac{m\alpha\gamma}{\lambda} > 0$ and $\nu = \psi(\psi - \lambda) < 0$, we have

$$|\nu| = \psi(\lambda - \psi) \leq \lambda\psi = m\alpha\gamma \quad (3.94)$$

It follows from (3.93) and (3.94) that

$$a(z_1) + 2\nu c(z_1) \geq (1 - m)a(z_1) \quad (3.95)$$

for all $z \in \mathbb{R}^{2n}$. Noticing that from (3.92) we have $a(z_1) + \frac{1}{(1-m)}b(z) \geq a(z_1) + b(z) \geq V(z)$, substituting (3.95) into (3.88) we have

$$\begin{aligned} \dot{V}(z) &\leq -(1 - m)\psi a(z_1) + 2(\psi - \lambda)b(z) \\ &\leq -(1 - m)\psi a(z_1) + \psi b(z) + (\psi - 2\lambda)b(z) \\ &\leq -(1 - m)\psi a(z_1) + \psi b(z) \\ &\leq -(1 - m)\psi \left(a(z_1) + \frac{1}{(1 - m)}b(z) \right) \\ &\leq -(1 - m)\psi V(z) \end{aligned} \quad (3.96)$$

for all $z \in \mathbb{R}^{2n}$. The third inequality comes from the fact that we choose $\psi = \frac{m\alpha\gamma}{\lambda} > 0$ such that $\psi - \lambda < 0$ and, consequently, $\psi - 2\lambda < 0$. Applying Grönwall's inequality to (3.96) shows that every maximal solution $t \mapsto z(t)$ to (1.1) satisfies $V(z(t)) \leq V(z(0)) \exp(-(1 - m)\psi t)$ for all $t \in \text{dom } z (= \mathbb{R}_{\geq 0})$. Therefore, each maximal solution $t \mapsto z(t)$ to the closed-loop algorithm in (3.74) satisfies (3.91) for all $t \in \text{dom } z (= \mathbb{R}_{\geq 0})$. \square

3.2.3 Extensions of the Results for Nonstrongly Convex L

Some possible extensions to the results in Section 3.2.2 are as follows.

It is possible to extend the results in Section 3.2.2 to include \mathcal{C}^1 , nonstrongly convex objective functions L with a compact and connected set of minimizers. With such an assumption, it would be straightforward to extend Lemma 3.2.6 and Propositions 3.2.7 and Proposition 3.2.8. Lemma 3.2.9 and Proposition 3.2.10 could be extended via the assumption of a compact and connected set of minimizers and the use of Clarke's generalized derivative in (2.4) with the Lyapunov function

$$V(z) := \gamma(L(z_1) - L^*) + \frac{1}{2} \left| \psi \nabla_{z_1} |z_1|_{\mathcal{A}_3} + z_2 \right|^2 + \frac{\nu}{2} |z_1 - z_1^*|^2 \quad (3.97)$$

where \mathcal{A}_3 is defined in (3.70) and $L^* := L(\mathcal{A}_3)$. Namely, when L is \mathcal{C}^1 , nonstrongly convex, and has a compact and connected set of minimizers, and given $\lambda > 0$ and $\gamma > 0$, it can be shown that, for each $\psi > 0$ such that $\nu := \psi(\psi - \lambda) < 0$, the following bound is satisfied for each $z \in \mathbb{R}^{2n}$:

$$V^\circ(z, F_P(z, \kappa(h(z)))) \leq -\psi(a(z) + 2\nu c(z)) + 2(\psi - \lambda)b(z) \quad (3.98)$$

where V is defined via (3.97), F_P is defined in (3.1), κ is defined via (3.71), h is defined by either (3.72) or (3.73), and a , b , and c are defined under (3.88). When, in addition, L satisfies Assumption 3.2.4, then it can be shown that each maximal solution $t \mapsto z(t)$ to the closed-loop algorithm in (3.74) satisfies (3.91), for all $t \in \text{dom } z (= \mathbb{R}_{\geq 0})$.

It would be possible to further extend the results in Lemmas 3.2.6 and 3.2.9 and Propositions 3.2.7, 3.2.8, and 3.2.10 to include \mathcal{C}^1 , nonstrongly convex objective functions L that are also nonsmooth, through the use of Clarke's generalized

derivative.

3.2.4 Nonconvex L

The set of all local minimizers of L is denoted as

$$\mathcal{A}_{1_{\min}} = \{z_1 \in \mathbb{R} : \nabla L(z_1) = 0, \nabla^2 L(z_1) > 0\}. \quad (3.99)$$

Conversely, the set of all local maximizers of L is denoted as

$$\mathcal{A}_{1_{\max}} = \{z_1 \in \mathbb{R} : \nabla L(z_1) = 0, \nabla^2 L(z_1) < 0\}. \quad (3.100)$$

Then, the set of all critical points of $L : \mathbb{R} \rightarrow \mathbb{R}$ is given as

$$\mathcal{A}_1 = \mathcal{A}_{1_{\min}} \cup \mathcal{A}_{1_{\max}}. \quad (3.101)$$

The following assumptions are required by some of the forthcoming results in this section.

Assumption 3.2.11. (*Properties of the objective function L*)

(M1) L is a Morse function;

(M2) L is \mathcal{C}^2 ;

(M3) There exists $d_0 > 0$ such that each $z^* = (z_1^*, 0) \in \mathcal{A}_1 \times \{0\}$ satisfies

$$(z^* + d_0\mathbb{B}) \cap ((\mathcal{A}_1 \times \{0\}) \setminus \{z^*\}) = \emptyset \quad (3.102)$$

(M4) L is radially unbounded;

(M5) There exists $\alpha \in \mathcal{K}$ such that for each $\varepsilon > 0$ sufficiently small, there exists $\delta \in (0, \alpha(\varepsilon))$ such that if $|\nabla L(z_1)| \leq \varepsilon$ then $|z_1|_{\mathcal{A}_1} \leq \delta$.

Remark 3.2.12. The finite separation $d_0 > 0$ between critical points from (M3) ensures that critical points do not accumulate, which is required for our algorithm to solve (1.7). A similar finite separation assumption can be found in [78]. Additionally, we do not expect that a solution to (1.7) exists without (M3). Item (M4) ensures radial unboundedness of the Lyapunov function used in the attractivity analysis of the proposed algorithm. Item (M5) of Assumption 3.2.11 means that z_1 is suboptimal [66]. Item (M5) is used to ensure that the algorithm can detect when the state z is near a critical point, using only measurements of ∇L .

In this section, we will analyze the properties of the heavy ball algorithm defined in (3.74), when the objective function L is Morse, as described in Definition 2.3.1, Lemma 2.3.2, and Corollary 2.3.3. When L is a Morse function, the heavy ball algorithm cannot converge to a local minimum when $z_1(0)$ is at a local maximum, while $z_2(0) = 0$, as will be shown in Theorem 3.2.14 below. Thus, we define the set of interest (namely, the set of local minimizers, when $z_2 = 0$) as

$$\mathcal{A}_{a_{\min}} := \mathcal{A}_{1_{\min}} \times \{0\}. \quad (3.103)$$

We also define the set of local maximizers, when $z_2 = 0$, as

$$\mathcal{A}_{a_{\max}} := \mathcal{A}_{1_{\max}} \times \{0\}. \quad (3.104)$$

When item (M2) of Assumption 3.2.11 is satisfied, then each maximal solution to (3.74), is complete and bounded, as stated in the following lemma.

Lemma 3.2.13. (Existence of solutions to (3.74) when L is nonconvex and Morse)

Let L satisfy items (M1)-(M4) of Assumption 3.2.11. Let $\lambda > 0$ and $\gamma > 0$ be given. Let κ and h be defined via (3.71) and (3.72), respectively. Then, each maximal solution $t \mapsto (z(t))$ to (3.74) starting from $z(0) \in \mathcal{U} = \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$ is bounded and complete.

Proof. To show that there is no finite time escape from \mathbb{R}^2 , for each solution to (3.74) starting from $z(0) \in \mathcal{U} = \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, we show that each solution to (3.74) starting from $z(0) \in \mathcal{U} = \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$ lies entirely in a compact subset $W \subset \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, as stated in [77, Theorem 3.3]. Since L is \mathcal{C}^2 , by item (M2) of Assumption 3.2.11, then ∇L exists and is \mathcal{C}^1 . This means that h in (3.72) and κ in (3.71) are \mathcal{C}^1 and, in turn, the map $z \mapsto F_P(z, \kappa(h(z)))$ is \mathcal{C}^1 . Therefore, by [77, Lemma 3.2], the map $z \mapsto F_P(z, \kappa(h(z)))$ is locally Lipschitz. Next, due to L being Morse and having isolated critical points by (M1), radially unbounded by (M4), and due to the critical points having a minimum separation $d_0 > 0$ by (M3), L has n minimizers, indexed over $i \in \{1, 2, \dots, n\}$ and $n - 1$ maximizers, indexed over $k \in \{1, 2, \dots, n - 1\}$. Then, we define the Lyapunov function V piecewise, as follows.

$$V(z) := \begin{cases} \gamma \left(L(z_1) - L(z_{1_i}^*) \right) + \frac{1}{2} z_2^2, & \text{if } z_1 < z_{1_k}^*, i = k = 1 \\ \gamma \left(L(z_1) - L(z_{1_i}^*) \right) + \frac{1}{2} z_2^2, & \text{if } z_1 \geq z_{1_{k-1}}^* \text{ and } z_1 < z_{1_k}^*, \\ & i = k, 1 < i \leq n - 1, 1 < k \leq n - 1 \\ \gamma \left(L(z_1) - L(z_{1_n}^*) \right) + \frac{1}{2} z_2^2, & \text{if } z_1 \geq z_{1_k}^*, i = n, k = n - 1 \end{cases} \quad (3.105)$$

where $\gamma > 0$, $z_{1_i}^*, z_{1_n}^* \in \mathcal{A}_{1_{\min}}$, and $z_{1_{k-1}}^*, z_{1_k}^* \in \mathcal{A}_{1_{\max}}$, where $\mathcal{A}_{1_{\min}}$ and $\mathcal{A}_{1_{\max}}$ are defined via (3.99) and (3.100), respectively. Since L is Morse by item (M1), L has isolated critical points with a minimum separation $d_0 > 0$, by item (M3). Moreover, L is radially unbounded by item (M4) of Assumption 3.2.11. Therefore,

there exists a region of each minimizer $z_{1_i}^* \in \mathcal{A}_{1_{\min}}$ where L is positive definite with respect to such a minimizer. Namely:

- 1) L is positive definite with respect to $z_{1_1}^* \in \mathcal{A}_{1_{\min}}$ on the domain $(-\infty, z_{1_k}^*)$, where $k = 1$ and $z_{1_k}^* \in \mathcal{A}_{1_{\max}}$;
- 2) L is positive definite with respect to each $z_{1_i}^*$, $1 < i \leq n - 1$ on each respective domain $[z_{1_{k-1}}^*, z_{1_k}^*)$, where $1 < k \leq n - 1$ and $z_{1_{k-1}}^*, z_{1_k}^* \in \mathcal{A}_{1_{\max}}$;
- 3) L is positive definite with respect to $z_{1_n}^* \in \mathcal{A}_{1_{\min}}$ on the domain $[z_{1_k}^*, \infty)$, where $k = n - 1$ and $z_{1_k}^* \in \mathcal{A}_{1_{\max}}$.

Consequently, a region of each minimizer, $z^* \in \mathcal{A}_{a_{\min}}$, where $\mathcal{A}_{a_{\min}}$ is defined in (3.103), of V is positive definite with respect to each such minimizer. Namely:

- 1) V is positive definite with respect to $(z_{1_i}^* \times \{0\}) \in \mathcal{A}_{a_{\min}}$, $i = 1$, where $\mathcal{A}_{a_{\min}}$ is defined in (3.103), on

$$\mathcal{U}_1 := (-\infty, z_{1_k}^*) \times \mathbb{R} \quad z_{1_k}^* \in \mathcal{A}_{1_{\max}}, k = 1 \quad (3.106)$$

- 2) V is positive definite with respect to each $(z_{1_i}^* \times \{0\}) \in \mathcal{A}_{a_{\min}}$ on

$$\mathcal{U}_i := [z_{1_{k-1}}^*, z_{1_k}^*) \times \mathbb{R} \quad z_{1_{k-1}}^*, z_{1_k}^* \in \mathcal{A}_{1_{\max}}, i = k, 1 < k \leq n - 1, 1 < i \leq n - 1 \quad (3.107)$$

- 3) V is positive definite with respect to $(z_{1_n}^* \times \{0\}) \in \mathcal{A}_{a_{\min}}$, where $i = n$, on

$$\mathcal{U}_n := [z_{1_k}^*, \infty) \times \mathbb{R} \quad z_{1_k}^* \in \mathcal{A}_{1_{\max}}, k = n - 1 \quad (3.108)$$

Furthermore, since L is radially unbounded by item (M4) of Assumption 3.2.11, Then V is also radially unbounded on $\mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$.

Although L is \mathcal{C}^2 by item (M2), V , defined via (3.105), is only piecewise continuous. Namely, V is \mathcal{C}^2 on $\mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, where $\mathcal{A}_{a_{\max}}$ is defined via (3.104), but V is not continuous at each $z_1 = z_{1_k}^* \in \mathcal{A}_{1_{\max}}$. Therefore, since for (3.74) $z_1 \in \mathbb{R}$ and $z_2 \in \mathbb{R}$, and since $\nabla V(z) = [\gamma \nabla L(z_1) \quad z_2]$ exists everywhere except at $z_1 = z_{1_k}^* \in \mathcal{A}_{1_{\max}}$, for each $k \in \{1, 2, \dots, n-1\}$, then we evaluate Clarke's generalized derivative of V in (3.105), using the map $z \mapsto F_P(z, \kappa(h(z)))$, where F_P is defined via (3.1), κ is defined in (3.71), and h is defined via (3.72). For each $z \in \mathcal{U} := \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, we obtain

$$\begin{aligned} V^\circ(z, F_P(z, \kappa(h(z)))) &= \max_{\nabla V(z) \in \partial V(z)} \left\langle \nabla V(z), \begin{bmatrix} z_2 \\ \kappa(h(z)) \end{bmatrix} \right\rangle & (3.109) \\ &= \left\langle [\gamma \nabla L(z_1) \quad z_2] \begin{bmatrix} z_2 \\ -\lambda z_2 - \gamma \nabla L(z_1) \end{bmatrix} \right\rangle \\ &= -\lambda z_2^2 \leq 0 \end{aligned}$$

due to $\lambda > 0$. Therefore, each solution to $\dot{z} = F_P(z, \kappa(h(z)))$ starting from any c_V -sublevel set

$$W := \left\{ z \in \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}} : V(z) \leq c_V \right\} \quad (3.110)$$

remains in such a set for all time. Since V is \mathcal{C}^2 and radially unbounded on $\mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, due to L being \mathcal{C}^2 by item (M2) of Assumption 3.2.11 and radially unbounded by item (M4) of Assumption 3.2.11, and due to $\dot{z} = F_P(z, \kappa(h(z)))$ in (3.74) being locally Lipschitz on $\mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$ by [77, Lemma 3.2], since L is \mathcal{C}^2 by item (M2) of Assumption 3.2.11, then W in (3.110) is compact and, due to (3.109), forward invariant for (3.74), that is, by [77, Theorem 3.3], any nontrivial solution starting in the subset W is complete and stays in W . Therefore, each maximal solution to (3.74) is completed and bounded. \square

The following result shows that, when Assumption 3.2.11 holds, then (3.74) has the set $\mathcal{A}_{a_{\min}}$, defined via (3.103), that is attractive from all points such that $z(0) \in \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$. To establish this result, we use an invariance principle.

Theorem 3.2.14. *(Almost global asymptotic stability of $\mathcal{A}_{a_{\min}}$ for (3.74)) Let L satisfy Assumption 3.2.11. Let $\lambda > 0$ and $\gamma > 0$ be given. Let κ and h be defined via (3.71) and (3.72), respectively. Then, the set $\mathcal{A}_{a_{\min}}$ in (3.103) is almost globally asymptotically stable with basin of attraction given by $\mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, for (3.74), where $\mathcal{A}_{a_{\max}}$ is defined via (3.104).*

Proof. By Lemma 3.2.13, each maximal solution is bounded and complete for (3.74). Recall that, in the proof of Lemma 3.2.13, it was shown that V in (3.105) satisfies (3.109) for each $z \in \mathcal{U} = \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$. Therefore, by an application of Theorem A.1.3, since $\gamma > 0$ and $\lambda > 0$, then $\mathcal{A}_{a_{\min}}$ is stable for each $z \in \mathcal{U} := \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, for the closed-loop algorithm (3.74). Using an invariance principle, each maximal solution that is complete and bounded approaches the largest weakly invariant set for (3.74) that is contained in

$$\left\{ z \in \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}} : V^\circ(z, F_P(z, \kappa(h(z)))) = 0 \right\} \cap \left\{ z \in \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}} : V(z) = r \right\}. \quad (3.111)$$

Such a set is nonempty only when $r = 0$ and, precisely, is equal to $\mathcal{A}_{a_{\min}}$. This property can be seen by noticing that

$\{z \in \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}} : V^\circ(z, F_P(z, \kappa(h(z)))) = 0\} = \{z \in \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}} : z_2 = 0\}$, and that after setting z_2 to zero in (3.74) we obtain $\begin{bmatrix} \dot{z}_1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\gamma \nabla L(z_1) \end{bmatrix}$. For any solution to this system, the z_1 component satisfies $0 = \gamma \nabla L(z_1)$, which since $\gamma > 0$ and the minimizers of L belong to the set $\mathcal{A}_{1_{\min}}$, leads to $z_1 \in \mathcal{A}_{1_{\min}}$. Then, every complete and bounded solution starting from $z(0, 0) \in \mathcal{U} := \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$ to the closed-loop algorithm in (3.74) converges to $\mathcal{A}_{a_{\min}}$.

Although V is not continuous on $\mathcal{A}_{1_{\max}}$, note that each $z_1 \in \mathcal{A}_{1_{\max}}$ satisfies $\nabla L(z_1) = 0$, since L is \mathcal{C}^2 by item (M2). Hence, since the points $z \in \mathcal{A}_{a_{\max}}$ are not in $\mathcal{U} := \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, then such points are simply equilibria, and if the state z starts at such a point, it will be stuck at such a point for all time. Therefore, $\mathcal{A}_{a_{\min}}$ is not attractive from all $z(0,0) \in \mathbb{R}^2$, but instead $\mathcal{A}_{a_{\min}}$ is attractive from all initial points such that $z(0,0) \in \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$.

The arguments above involving the Lyapunov theorem in Theorem A.1.3 and an invariance principle yield almost global pre-asymptotic stability of $\mathcal{A}_{a_{\min}}$, with basin of attraction given by $\mathbb{R}^2 \setminus (\mathcal{A}_{1_{\max}} \times \{0\})$, for (3.74). Since by Lemma 3.2.13, each maximal solution to (3.74) is complete, then $\mathcal{A}_{a_{\min}}$ is almost globally asymptotically stable, with basin of attraction given by $\mathbb{R}^2 \setminus (\mathcal{A}_{1_{\max}} \times \{0\})$, for (3.74). \square

Chapter 4

Uniting Heavy Ball Algorithms

The uniting algorithms proposed in this chapter impose Assumptions 3.1.8, 3.1.3, and 3.2.4 on the objective function L . Namely, L is \mathcal{C}^1 , nonstrongly convex, and has a unique minimizer by Assumption 3.1.8, has a Lipschitz continuous gradient by Assumption 3.1.3, and has quadratic growth away from its minimizer z_1^* by assumption 3.2.4.

4.1 Problem Statement

In Section 1.2.2, the performance of the heavy ball method for finding the minimizer of an objective function is highly dependent on the choice of γ and λ , and the need for a logic-based algorithm to determine which set of parameters to use far from the minimizer and which one to use near the minimizer, was discussed.

The problem to solve is stated as follows:

Problem 4.1.1. *Given a scalar, real-valued, continuously differentiable, and non-strongly convex objective function L with a unique minimizer, design an optimization algorithm that, without knowing the function L or the location of its minimizer, has the minimizer uniformly globally asymptotically stable, with robustness*

to arbitrarily small noise in measurements of ∇L .

4.2 Modeling

In this chapter, we propose an algorithm that solves Problem 4.1.1. As described in Section 3.2, defining z_1 as ξ and z_2 as $\dot{\xi}$, we interpret the heavy ball ODE in (1.1) as a closed-loop system consisting of the plant in (3.1) and control algorithm assigning u to κ in (3.71), where $\gamma > 0$ and $\lambda > 0$.

To cope with the trade-off between damping oscillations and converging fast, we propose a logic-based algorithm that unites two control algorithms, one with small λ used far from the minimizer to quickly get close to the critical point, and one with large λ used near the minimizer to avoid oscillations. The proposed logic-based algorithm “unites” the two optimization algorithms modeled by κ_q , where the logic variable $q \in Q := \{0, 1\}$ indicates which algorithm is currently being used. Such optimization algorithms are designed as static state-feedback laws of the form

$$\kappa_q(h(z)) := -\lambda_q z_2 - \gamma_q \nabla L(z_1) \quad (4.1)$$

for each $z \in \mathbb{R}^{2n}$, where the output h will adopt different forms in the upcoming sections. The parameters $\lambda_q > 0$ and $\gamma_q > 0$ should be designed for each $q \in Q$, so as to achieve fast convergence without oscillations nearby the minimizer. The algorithm for $q = 1$ will be designed to achieve fast convergence and is referred to as global. The algorithm for $q = 0$ will be designed to achieve stable convergence near the minimizer and is referred to as local; see, e.g., [22, Chapter 4]. Due to q jumping between 0 and 1 with hysteresis, we refer to the proposed logic-based algorithms as hybrid algorithms.

The design of the switching rules for uniting algorithm is done using the Lya-

punov function

$$V_q(z) := \gamma_q(L(z_1) - L^*) + \frac{1}{2} |z_2|^2 \quad (4.2)$$

defined for each $q \in Q$ and each $z \in \mathbb{R}^{2n}$, where $L^* := L(z_1^*)$.

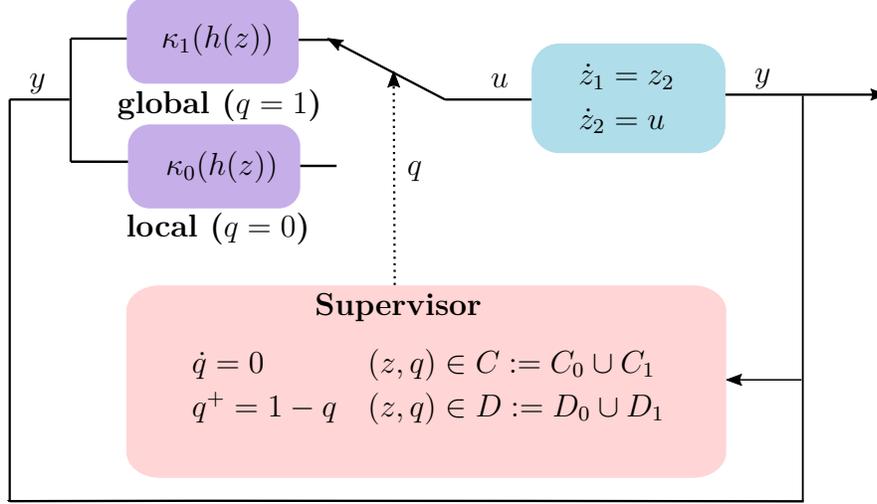


Figure 4.1: Feedback diagram of the hybrid closed-loop system \mathcal{H} , in (4.3), uniting global and local optimization algorithms.

The switch between κ_0 and κ_1 is governed by a *supervisory algorithm* governing switching logic; see Figure 4.1. The supervisor selects between these two autonomous optimization algorithms, based on the plant's output and on the optimization algorithm currently applied. In our simplest algorithm, which is introduced first, the idea is to define sublevel sets of V_q and use hysteresis to switch between the global heavy ball algorithm and the local one. More precisely, when the supervisor is using the global optimization algorithm κ_1 and $V_1(z) \leq c_{1,0}$ with $c_{1,0}$ small, then z_1 is close to the minimum and a switch to the local algorithm is performed to converge without oscillations. When the supervisor is using the local algorithm κ_0 and $V_0(z) \geq c_0$ with $c_0 > c_{1,0}$, then z_1 is too far from its minimum

and the supervisor switches to the global algorithm to converge quickly to the neighborhood of the minimum. These switching events constitute the jumps in the hybrid closed-loop system, and the c_0 - and $c_{1,0}$ -sublevel sets need to be properly tuned to solve Problem 4.1.1. At times other than when these events occur, the hybrid algorithm executes the individual optimization algorithm associated with the current value of q , namely, it applies (4.1) to (3.1).

To capture the mechanism outlined above, encapsulating the plant in (3.1) and static state-feedback laws in (4.1), we define a hybrid closed-loop system \mathcal{H} with state $x := (z, q) \in \mathbb{R}^{2n} \times Q$ as:

$$\left. \begin{array}{l} \dot{z} = \begin{bmatrix} z_2 \\ \kappa_q(h(z)) \end{bmatrix} \\ \dot{q} = 0 \end{array} \right\} =: F(x) \quad x \in C := C_0 \cup C_1 \quad (4.3a)$$

$$\left. \begin{array}{l} z^+ = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \\ q^+ = 1 - q \end{array} \right\} =: G(x) \quad x \in D := D_0 \cup D_1 \quad (4.3b)$$

where C_0 , D_0 , C_1 , and D_1 are defined differently in the subsequent sections of this chapter.

Figure 4.1 shows the feedback diagram of this hybrid closed-loop system \mathcal{H} . We denote the closed-loop systems resulting from using the individual heavy ball algorithms (κ_q) as \mathcal{H}_q , which is given by

$$\dot{z} = \begin{bmatrix} z_2 \\ \kappa_q(h(z)) \end{bmatrix} \quad z \in \mathbb{R}^{2n}. \quad (4.4)$$

for each $q \in Q = \{0, 1\}$; i.e., \mathcal{H}_0 denotes the local heavy ball algorithm that uses

λ_0 and γ_0 , and \mathcal{H}_1 denotes the global heavy ball algorithm that uses λ_1 and γ_1 .

The reader may wonder whether a (nonhybrid) discontinuous algorithm would solve Problem 4.1.1 robustly. Unfortunately, that is not the case since solutions without hysteresis switching may exhibit chatter at the switching surface induced by a discontinuous algorithm. The proposed hybrid systems approach solves the problem with robustness by virtue of hysteresis switching.

4.3 Uniting Heavy Ball Methods Using Measurements of L and ∇L

In this section, we present a uniting optimization algorithm with switching rules derived from sublevel sets of V_0 and V_1 , where V_q is defined via (4.2). This algorithm measures L and ∇L at the current value of z_1 . That is, the output h is defined via (3.73). Although the algorithm requires no knowledge of L , it still requires knowledge of L^* .

For the hybrid closed-loop algorithm \mathcal{H} in (4.3), the sets C_0 , D_0 , C_1 , and D_1 are defined as

$$C_0 := \{z \in \mathbb{R}^{2n} : \mathcal{U}_0\} \times \{0\}, \quad C_1 := \{z \in \mathbb{R}^{2n} : \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}\} \times \{1\} \quad (4.5a)$$

$$D_0 := \{z \in \mathbb{R}^{2n} : \overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0}\} \times \{0\}, \quad D_1 := \{z \in \mathbb{R}^{2n} : \mathcal{T}_{1,0}\} \times \{1\} \quad (4.5b)$$

The sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$ are precisely designed in section 4.3.1, but the idea behind their construction is as follows. When $z \in \mathcal{U}_0$ and $q = 0$ (i.e., $x \in C_0$), due to the design of \mathcal{U}_0 in Section 4.3.1, then the state z is near the minimizer, which is denoted z_1^* , and the supervisor allows flows of (4.3) using κ_0 , with large $\lambda_0 > 0$, to avoid oscillations. Conversely, when $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}$ and $q = 1$ (i.e., $x \in C_1$), due to

the design of $\mathcal{T}_{1,0}$ in Section 4.3.1, then the state z is far from the minimizer and the supervisor allows flows of (4.3) using κ_1 , with large $\lambda_1 > 0$, to converge quickly to the neighborhood of the minimizer. When $z \in \mathcal{T}_{1,0}$ and $q = 1$ (i.e., $x \in D_1$), then this indicates that the state z is near the minimizer, and the supervisor assigns u to κ_0 and resets q to 0. Conversely, when $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0}$ and $q = 0$ (i.e., $x \in D_0$), then this indicates that the state z is far from the minimizer and the supervisor assigns u to κ_1 and resets q to 1. The complete algorithm, defined in (4.3) and (4.5), is summarized in Algorithm 1.

Algorithm 1 Uniting algorithm

- 1: Set $q(0,0)$ to 0 and set $z(0,0)$ as an initial condition with an arbitrary value.
 - 2: **while** true **do**
 - 3: **if** $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0}$ and $q = 0$ **then**
 - 4: Reset q to 1.
 - 5: **else if** $z \in \mathcal{T}_{1,0}$ and $q = 1$ **then**
 - 6: Reset q to 0.
 - 7: **else if** $z \in \mathcal{U}_0$ and $q = 0$ **then**
 - 8: Assign u to $\kappa_0(h(z))$, where κ_0 is defined via (4.1) and h is defined in (3.73), and update z and q according to (4.3a).
 - 9: **else if** $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}$ and $q = 1$ **then**
 - 10: Assign u to $\kappa_1(h(z))$, where κ_1 is defined via (4.1) and h is defined in (3.73), and update z and q according to (4.3a).
 - 11: **end if**
 - 12: **end while**
-

4.3.1 Design of the Sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$

Based on the outline provided in Section 4.2, the supervisor selects κ_0 or κ_1 in (4.1) using sublevel sets of V_q in (4.2). When the system measures L and ∇L , these sets are defined as follows. Let the set \mathcal{U}_0 be defined by the c_0 -sublevel set of V_0 , namely,

$$\mathcal{U}_0 := \{z \in \mathbb{R}^{2n} : V_0(z) \leq c_0\}. \quad (4.6)$$

The parameter $c_0 > 0$, along with λ_0 and γ_0 , are designed so that \mathcal{U}_0 is in the region where κ_0 is to be used. In this design, λ_0 is large to avoid oscillations when converging to the minimum. Since κ_0 in (4.1) is such that the set $\{z_1^*\} \times \{0\}$ is globally asymptotically stable for the closed-loop system resulting from controlling (3.1) by κ_0 , as shown in Proposition 3.2.8, then \mathcal{U}_0 is contained in the basin of attraction induced by κ_0 . Then, roughly speaking, when $q = 0$ and $V_0(z) \leq c_0$, the hybrid closed-loop system will switch to the global algorithm defined by κ_1 . Otherwise, the local algorithm κ_0 is used.

The set $\mathcal{T}_{1,0}$ is defined by a $c_{1,0}$ -sublevel of V_1 with $c_{1,0} \in (0, c_0)$ chosen so that $\mathcal{T}_{1,0}$ is contained in the interior of \mathcal{U}_0

$$\mathcal{T}_{1,0} := \{z \in \mathbb{R}^{2n} : V_1(z) \leq c_{1,0}\} \quad (4.7)$$

This choice of $\mathcal{T}_{1,0}$ is possible since \mathcal{U}_0 and the sublevel sets of V_1 are compact for small enough constants c_0 and $c_{1,0}$. Then, due to the global attractivity guaranteed by κ_0 in Proposition 3.2.8, once z is in $\mathcal{T}_{1,0}$, the boundary of \mathcal{U}_0 will never be reached. When $q = 1$ and $V_1(z) \leq c_{1,0}$, the supervisor will switch from the global algorithm κ_1 to the local algorithm κ_0 . The constants c_0 and $c_{1,0}$ comprise the hysteresis necessary to avoid chattering at the switching boundary. Note that, due to $c_0 > c_{1,0}$, $\overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0} \cap \mathcal{T}_{1,0} = \emptyset$.

4.3.2 Design of the Parameter λ_q

The heavy ball parameter $\lambda_0 > 0$ should be made large enough to avoid oscillations near the minimizer and $\lambda_1 > 0$ should be made small enough to enable fast convergence to the neighborhood of the minimizer, as stated in Section 4.2. To gain some intuition on how to tune λ_q , consider the quadratic objective function $L(z_1) = \frac{1}{2}a_1z_1^2$, $a_1 > 0$, which was analyzed in detail in [14]. For such a case,

solutions to the heavy ball algorithm are overdamped (i.e., converge slowly with no oscillations) when $\lambda > 2\sqrt{a_1}$, critically damped (i.e., the fastest convergence possible with no oscillations) when $\lambda = 2\sqrt{a_1}$, and underdamped (fast convergence with oscillations) when $\lambda < 2\sqrt{a_1}$. Therefore, setting $\lambda_0 \geq 2\sqrt{a_1}$ gives the desired behavior of solutions to \mathcal{H}_0 and setting $\lambda < 2\sqrt{a_1}$ gives the desired behavior of solutions to \mathcal{H}_1 , for such an objective function. More generally, setting λ_0 sufficiently large to avoid oscillations and setting λ_1 sufficiently small for fast convergence suffices, in practice. Numerically, λ_0 can be tuned as follows. Choose an arbitrarily large value of λ_0 . If there is still oscillations or overshoot locally, despite the switch from κ_1 to κ_0 being made near the minimizer, then gradually increase λ_0 until the oscillations and overshoot disappear. See Examples 5.2.5, 5.2.6, and 5.2.7 where λ_0 was tuned in such a way. Numerically, λ_1 can be tuned as follows. Choose an arbitrarily small value of λ_1 (generally, less than 1, to begin with). If the state component z_1 does not get to the desired neighborhood of z_1^* before the switch to κ_0 , then gradually decrease λ_1 until the algorithm switches after z_1 reaches the desired neighborhood of z_1^* . See Examples 4.3.4, 4.4.5, and 4.4.6, where λ_1 was tuned in such a way.

4.3.3 Well-posedness of the Hybrid Closed-Loop System

\mathcal{H}

Under Assumption 3.1.8, the hybrid closed-loop system \mathcal{H} in (4.3), with C and D defined via (4.5), is well-posed as it satisfies the hybrid basic conditions.

Lemma 4.3.1. *(well-posedness of \mathcal{H}) Let L satisfy Assumption 3.1.8. Let the sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$ be defined via (4.6) and (4.7), respectively. Let κ_q be defined via (4.1) and h be defined in (3.73). Then, the hybrid closed-loop system \mathcal{H} in (4.3), with C and D as defined in (4.5), satisfies the hybrid basic conditions, as listed in*

Definition 2.1.1.

Proof. Since by Assumption 3.1.8 L is \mathcal{C}^1 , then for each $q \in Q$, V_q in (4.2), is continuous. Therefore, since V_q is continuous, then the following hold: the set \mathcal{U}_0 , defined via (4.6), is closed since it is a sublevel set of the continuous function V_0 ; the set $\mathcal{T}_{1,0}$, defined in (4.7), is closed since it is a sublevel set of the continuous function V_1 . Therefore, since the sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$ are closed, then the sets C_0 , D_0 , C_1 , and D_1 are closed. Since C and D are both finite unions of finite and closed sets, then C and D are also closed.

Also, by construction, the map $x \mapsto F(x)$ in (4.3a) is continuous, due to the fact that L is \mathcal{C}^1 , since this means that ∇L exists and is continuous, which in turn makes h in (3.73) and κ_q in (4.1) continuous. Similarly, the map $G(x)$ in (4.3b) is continuous by construction, satisfying item (A3) of Definition 2.1.1. \square

In the forthcoming Theorem 4.3.3 we show that \mathcal{H} has a compact pre-asymptotically stable set. In light of this property, Lemma 4.3.1 is key as it leads to pre-asymptotic stability that is robust to small perturbations [21, Theorem 7.21]. In the case of gradient-based algorithms, for instance, such perturbations can take the form of small noise in measurements of the gradient.

4.3.4 Existence of solutions to \mathcal{H}

When, in addition, Assumptions 3.1.3 and 3.2.4 hold, each maximal solution to the hybrid closed-loop system \mathcal{H} is complete and bounded, as shown in the following proposition. Such a property is useful since it guarantees that nontrivial solutions to \mathcal{H} exist from each initial point in $C \cup D$, and that such solutions do not escape $C \cup D$. When every maximal solution is complete, then uniform

global pre-asymptotic stability¹ of the set \mathcal{A} becomes uniform global asymptotic stability.

Proposition 4.3.2. *(Existence of solutions to \mathcal{H}) Let L satisfy Assumptions 3.1.3, 3.1.8, and 3.2.4. Let $\lambda_q > 0$, $\gamma_q > 0$, and $c_{1,0} \in (0, c_0)$. Let κ_q be defined via (4.1) and h be defined in (3.73). Then, $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$, $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$, and each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to the hybrid closed-loop system \mathcal{H} in (4.3), with C and D as defined in (4.5), is bounded and complete.*

Proof. Since L is \mathcal{C}^1 by Assumption 3.1.8, then by Lemma 4.3.1 \mathcal{H} in (4.3), with C and D as defined in (4.5), is well-posed.

By construction, every $z \in \mathcal{U}_0$ belongs to the c_0 -sublevel set of V_0 ; recall that \mathcal{U}_0 is defined in (4.6) and V_0 is defined in (4.2). Additionally, by construction, every $z \in \mathcal{T}_{1,0}$ belongs to the $c_{1,0}$ -sublevel set of V_1 ; recall that $\mathcal{T}_{1,0}$ is defined in (4.7) and V_1 is defined in (4.2). Therefore, since \mathcal{U}_0 is defined in (4.6), and since by the definitions of D_0 and C_0 in (4.5), D_0 is the closed complement of C_0 , then $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$. Furthermore, since $\mathcal{T}_{1,0}$ is defined in (4.7), and since by the definitions of D_1 and C_1 in (4.5), C_1 is the closed complement of D_1 , then $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$.

Due to the definitions of C_0 , D_0 , C_1 , and D_1 in (4.5), \mathcal{U}_0 in (4.6), and $\mathcal{T}_{1,0}$ in (4.7), then $C \setminus D$ is equal to $\text{int}(C)$. Hence, for each point $x \in C \setminus D$, the tangent cone to C at x is

$$T_C(x) := \begin{cases} \mathbb{R}^{2n} \times \{0\} & \text{if } x \in C_0 \setminus D_0, \\ \mathbb{R}^{2n} \times \{1\} & \text{if } x \in C_1 \setminus D_1. \end{cases} \quad (4.8)$$

¹Uniform global pre-asymptotic stability indicates the possibility of a maximal solution that is not complete, even though it may be bounded.

Therefore, $F(x) \cap T_C(x) \neq \emptyset$, satisfying (VC) of Proposition A.1.1 for each point $x \in C \setminus D$, and nontrivial solutions exist for every initial point in $(C_0 \cup C_1) \cup (D_0 \cup D_1)$. To prove that item (c) of Proposition A.1.1 does not hold, we show that $G(D) \subset C \cup D$, as follows. For D as defined in (4.5), $G(D)$ is:

$$G(D) := (\overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0} \times \{1\}) \cup (\mathcal{T}_{1,0} \times \{0\}).$$

Then compare $G(D)$ to the flow set C , defined in (4.5): since $c_{1,0} < c_0$, then $(\overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0} \times \{1\})$ is a subset of C_1 , and $(\mathcal{T}_{1,0} \times \{0\})$ is a subset of C_0 . Since this means that $G(D) \subset C$ and, consequently, $G(D) \subset C \cup D$, then item (c) of Proposition A.1.1 does not hold. The only thing left to prove is that item (b) of Proposition A.1.1 does not hold.

We show that there is no finite time escape from C for \mathcal{H} as follows. First, since L is \mathcal{C}^1 , nonstrongly convex, and has a single minimizer z_1^* by Assumption 3.1.8, since ∇L is Lipschitz continuous by Assumption 3.1.3, and since L has quadratic growth away from z_1^* by Assumption 3.2.4, then each maximal solution to $\dot{z} = F_P(z, \kappa_q(h(z)))$, defined via (4.4) is bounded, complete, and unique by Proposition 3.2.7. Next, for the hybrid closed-loop system \mathcal{H} , since $\dot{z} = F_P(z, \kappa_q(h(z)))$ has no finite time escape from \mathbb{R}^{2n} , this also means that $\dot{x} = F(x)$ has no finite time escape from C , as q does not change in C . Therefore, there is no finite time escape from $C \cup D$, for solutions x to \mathcal{H} in (4.3), with C and D defined via (4.5). Therefore, item (b) from Proposition A.1.1 does not hold. \square

4.3.5 Main Result

In this section, we present a result that establishes UGAS of the set

$$\mathcal{A} := \left\{ z \in \mathbb{R}^{2n} : \nabla L(z_1) = z_2 = 0 \right\} \times \{0\} = \{z_1^*\} \times \{0\} \times \{0\} \quad (4.9)$$

and a hybrid convergence rate that is exponential both globally and locally, for the hybrid closed-loop algorithm \mathcal{H} in (4.3), with C and D as defined in (4.5). Recall that the state $x := (z, q)$. In light of this, the first component of \mathcal{A} , namely, z_1^* , is the minimizer of L . The second component of \mathcal{A} , namely, $\{0\}$, reflects the fact that we need the velocity state z_2 to equal zero in \mathcal{A} so that solutions are not pushed out of such a set. The third component in \mathcal{A} , namely, $\{0\}$, is due to the logic state ending with the value $q = 0$, namely using κ_0 as the state z reaches the minimizer of L .

Theorem 4.3.3. *(Uniform global asymptotic stability of \mathcal{A} for \mathcal{H}) Let L satisfy Assumptions 3.1.8, 3.1.3, and 3.2.4. Let $\alpha > 0$ be generated by Assumption 3.2.4, and let $M > 0$ be generated by Assumption 3.1.3. Let $\lambda_q > 0$, $\gamma_q > 0$, and $c_{1,0} \in (0, c_0)$ be given. Let κ_q be defined via (4.1) and let h be defined in (3.73). Then, the set \mathcal{A} in (4.9) is uniformly globally asymptotically stable for \mathcal{H} in (4.3), with C and D as defined in (4.5). Furthermore, each maximal solution $(t, j) \mapsto x(t, j) = (z(t), q(t))$ to the closed-loop algorithm \mathcal{H} in (4.3) starting from C_1 satisfies the following:*

- 1) *The domain $\text{dom } x$ of the solution x is of the form $\cup_{j=0}^1 (I^j \times \{j\})$, with I^0 of the form $[t_0, t_1]$ and with I^1 of the form $[t_1, \infty)$ for some $t_1 \geq 0$ defining the time of the first jump;*
- 2) *For each² $t \in I^0$*

$$L(z_1(t, 0)) - L^* = \mathcal{O}(\exp(-(1-m)\psi_1 t)) \quad (4.10)$$

where $m \in (0, 1)$ is such that $\psi_1 := \frac{m\alpha\gamma_1}{\lambda_1} > 0$ and $\nu_1 := \psi_1(\psi_1 - \lambda_1) < 0$.

²Note that at each $t \in I^0$, $q(t, 0) = 1$, and at each $t \in I^1$, $q(t, 1) = 0$.

3) For each $t \in I^1$

$$L(z_1(t, 1)) - L^* = \mathcal{O}(\exp(-(1-m)\psi_0 t)) \quad (4.11)$$

where $m \in (0, 1)$ is such that $\psi_0 := \frac{m\alpha\gamma_0}{\lambda_0} > 0$ and $\nu_0 := \psi_0(\psi_0 - \lambda_0) < 0$.

Proof. The hybrid closed-loop algorithm \mathcal{H} in (4.3) and (4.5) satisfies the hybrid basic conditions, by Lemma 4.3.1, satisfying the first assumption of Theorem A.1.3. Furthermore, each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to \mathcal{H} is complete and bounded by Proposition 4.3.2. Since by Assumption 3.1.8, L has a unique minimizer, then \mathcal{A} in (4.9) is compact by construction, and $\mathcal{U} = \mathbb{R}^{2n} \times Q$ contains a nonzero open neighborhood of \mathcal{A} , satisfying the second assumption of Theorem A.1.3.

To prove attractivity of \mathcal{A} , we proceed by contradiction. Suppose there exists a complete solution x to \mathcal{H} such that $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} \neq 0$. Since Proposition 4.3.2 guarantees completeness of maximal solutions, we have the following cases:

- a) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_1 \setminus D_1$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$;
- b) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_0 \setminus (\mathcal{A} \cup D_0)$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$;
- c) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in D$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$.

Case a) contradicts the fact that, by Proposition 3.2.8, the set $\{z_1^*\} \times \{0\}$, is uniformly globally asymptotically stable \mathcal{H}_1 in (4.4). Such uniform asymptotic stability of $\{z_1^*\} \times \{0\}$, guaranteed by Proposition 3.2.8, implies there exist $\tilde{c}_1 \in (0, \tilde{c}_{1,0})$ and $d_1 \in (0, d_{1,0})$ such that the state z reaches $(\{z_1^*\} + \tilde{c}_1 \mathbb{B}) \times (\{0\} + d_1 \mathbb{B}) \subset \mathcal{T}_{1,0}$

as $t \rightarrow \infty$. In turn, due to the construction of C_1 and D_1 in (4.5), with $\mathcal{T}_{1,0}$ defined via (4.7), the solution x must reach D_1 at some $(t, j) \in \text{dom } x, t + j \geq t' + j'$. Therefore, case a) does not happen.

Case b) contradicts the fact that, by Proposition 3.2.8, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_0 in (4.4). In fact, $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} = 0$, and since $\mathcal{A} \subset C_0$, case b) does not happen.

Case c) contradicts the fact that, due to the construction of $\mathcal{T}_{1,0}$ in (4.7) and \mathcal{U}_0 in (4.6), we have

$$\begin{aligned} G(D) \cap D &:= \left((\overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0} \times \{1\}) \cup (\mathcal{T}_{1,0} \times \{0\}) \right) \\ &\quad \cap \left((\overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0} \times \{0\}) \cup (\mathcal{T}_{1,0} \times \{1\}) \right) \\ &= \emptyset \end{aligned} \tag{4.12}$$

where $G(D)$ is defined via (4.9) and D is defined in (4.5). Such an equality holds since $\overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0} \cap \mathcal{T}_{1,0} = \emptyset$; see the end of Section 4.3.1. Therefore, case c) does not happen.

Therefore, cases a)-c) do not happen, and each maximal and complete solution $x = (z, q)$ to \mathcal{H} in (4.3) converges to $\{z_1^*\} \times \{0\}$. Consequently, by the construction of C and D in (4.5), the uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_q in (4.4) established in Proposition 3.2.8, and since each maximal solution to \mathcal{H} is complete by Proposition 4.3.2, the set $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H} .

To show that each maximal and complete solution x to \mathcal{H} jumps no more than twice, we proceed by contradiction. Without loss of generality, suppose there exists a maximal and complete solution that jumps three times. We have the following possible cases:

- i) The solution first jumps at a point in D_0 , then jumps at a point in D_1 , and then jumps at a point in D_0 ; or
- ii) The solution first jumps at a point in D_1 , then jumps at a point in D_0 , and then jumps at a point in D_1 .

Case i) does not hold since, once the jump in D_1 occurs, the solution x is in $(\mathcal{T}_{1,0} \times \{0\}) \subset C_0$. Due to the construction of $\mathcal{T}_{1,0}$ in (4.25) and \mathcal{U}_0 in (4.6) such that $\mathcal{T}_{0,1}$ in (4.30) such that $\overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0} \cap \mathcal{T}_{1,0} = \emptyset$, as described in the contradiction of case c) above, and due to the uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_q in (4.4) by Proposition 3.2.8, the solution x will never return to D_0 . Therefore, case i) does not happen. Case ii) leads to a contradiction for the same reason, and in this case, once the first jumps in D_1 occurs, no more jumps happen. Therefore, since cases i)-ii) do not happen, each maximal and complete solution x to \mathcal{H} in (4.3), with C and D defined via (4.5), has no more than two jumps.

Finally, we prove the hybrid convergence rate of \mathcal{H} . By Proposition 3.2.10, since L satisfies Assumptions 3.1.8 and 3.2.4, then, given $\gamma_q > 0$ and $\lambda_q > 0$, for each $m \in (0, 1)$ such that $\psi_q := \frac{m\alpha\gamma_q}{\lambda_q} > 0$ and $\nu_q := \psi_q(\psi_q - \lambda) < 0$, each maximal solution $t \mapsto z(t)$ to the closed-loop algorithm \mathcal{H}_q in (4.4) satisfies (3.91) for all $t \in \text{dom } z (= \mathbb{R}_{\geq 0})$. Since maximal solutions $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to \mathcal{H} starting from C_1 are guaranteed to jump no more than once, as implied by the contradiction in cases i)-ii) above, then the domain of each maximal solution x to \mathcal{H} starting from C_1 is $\cup_{j=0}^1 (I^j, j)$, with I^0 of the form $[t_0, t_1]$ and with I^1 of the form $[t_1, \infty)$. Therefore, given $\lambda_q > 0$, $\gamma_q > 0$, $c_{1,0} \in (0, c_0)$, $\varepsilon_{1,0} \in (0, \varepsilon_0)$, $\alpha > 0$ from Assumption 3.2.4, and $M > 0$ from Assumption 3.1.3, due to the construction of \mathcal{U}_0 in (4.6) and $\mathcal{T}_{1,0}$ in (4.7) with $c_{1,0} \in (0, c_0)$, and due to the individual convergence rates of \mathcal{H}_q , each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to the hybrid closed-loop algorithm \mathcal{H} in (4.3) that starts in C_1 satisfies (4.10) for

each $t \in I^0$ at which $q(t, 0)$ is equal to 1, and satisfies (4.11) for each $t \in I^1$ at which $q(t, 1)$ is equal to 0. □

4.3.6 Numerical Example

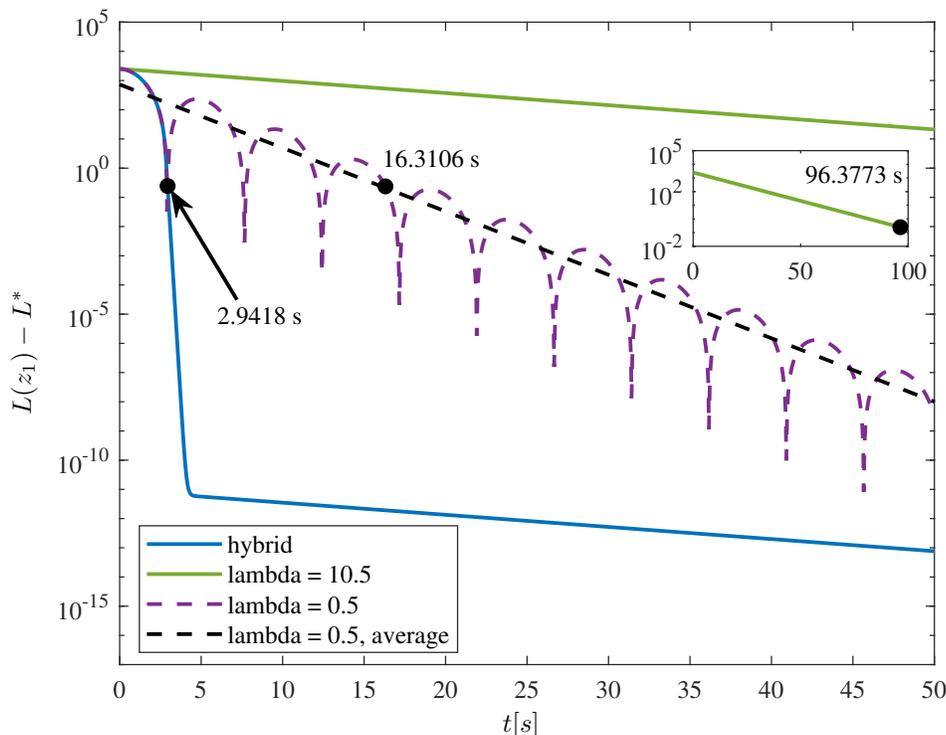


Figure 4.2: A comparison of the evolution of L over time for \mathcal{H}_0 , \mathcal{H}_1 , and \mathcal{H} , for a function $L = \frac{1}{4}z_1^\top Pz_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer at $z_1^* = (0, 0, \dots, 0)$. The heavy ball algorithm \mathcal{H}_1 uses $\lambda_1 = \frac{1}{2}$ (shown in purple) and settles to within 1% of z_1^* in about 16.3 seconds. The heavy ball algorithm \mathcal{H}_0 uses $\lambda_0 = 10.5$ (shown in green) and settles to within 1% of z_1^* in about 96.4 seconds. The hybrid closed-loop system \mathcal{H} (shown in blue) settles to within 1% of z_1^* in about 2.9 seconds.

Example 4.3.4. To show the effectiveness of the hybrid algorithm \mathcal{H} in (4.3), with C and D defined in (4.5), we compared it in simulation with the individual optimization algorithms \mathcal{H}_0 and \mathcal{H}_1 . The local algorithm, \mathcal{H}_0 , uses a large λ value to produce slow convergence with no oscillations. The global algorithm, \mathcal{H}_1 , uses a

small λ value to produce fast convergence with oscillations. The hybrid closed-loop system \mathcal{H} switches between the two lambda values, to ensure fast convergence with no oscillations.

The choice of objective function, parameter values, and initial conditions are as follows. We use the objective function $L = \frac{1}{4}z_1^\top P z_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer at $z_1^* = (0, 0, \dots, 0)$. The Lipschitz constant of ∇L is $M = \frac{1}{2}$. The choice of the objective function is made both to show how the algorithm performs with arguments $z_1 \in \mathbb{R}^n$, with $n > 1$, and was also chosen so that we can easily tune λ_q , as described in Section 5.1.6. Namely, we tuned λ_0 to 10.5 by choosing a value arbitrarily larger than $2\sqrt{a_1}$, where a_1 comes from Section 5.1.6, and gradually increasing it until there is no overshoot in the hybrid algorithm. We also tuned λ_1 to $\frac{1}{2}$ by choosing a value arbitrarily smaller than $2\sqrt{a_1}$ and gradually decreasing until the switch to \mathcal{H}_0 occurs once z_1 reaches the desired neighborhood of z_1^* . We arbitrarily chose $\gamma_0 = \gamma_1 = \frac{1}{2}$. The parameter values for the uniting algorithm are $c_0 = 1200$ and $c_{1,0} \approx 638.370$, which are chosen for proper tuning of the algorithm, in order to get nice performance, and the value of $c_{1,0}$ is chosen to exploit the properties of \mathcal{H}_1 for a longer time, so that the nominal solution gets closer to the minimizer faster. Initial conditions for \mathcal{H} , \mathcal{H}_0 , and \mathcal{H}_1 are $z_1(0, 0) = -10$, $z_2(0, 0) = 0$, and $q(0, 0) = 0$.

Algorithm	Time to converge (s)	% improvement of \mathcal{H}
\mathcal{H}	2.942	–
\mathcal{H}_0	96.377	96.9
\mathcal{H}_1	16.311	82.0

Table 4.1: Average times for which \mathcal{H} , \mathcal{H}_0 , and \mathcal{H}_1 settle to within 1% of z_1^* , and the average percent improvement of \mathcal{H} over each algorithm. Percent improvement is calculated via (4.13). The objective function used for this table is $L = \frac{1}{4}z_1^\top P z_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer at $z_1^* = (0, 0, \dots, 0)$.

Table 4.1 shows the time that each algorithm takes to settle within³ 1% of z_1^* , and the percent improvement of \mathcal{H} over \mathcal{H}_0 , \mathcal{H}_1 , and HAND-1, which is calculated using the following formula

$$\left(\frac{\text{Time of } \mathcal{H}_0 \text{ or } \mathcal{H}_1 - \text{Time of } \mathcal{H}}{\text{Time of } \mathcal{H}_0 \text{ or } \mathcal{H}_1} \right) \times 100\%. \quad (4.13)$$

As can be seen in Figure 4.2 and Table 4.1, \mathcal{H} converges faster than the other algorithms, and the percent improvement of \mathcal{H} over each of the other algorithms in Table 4.1 is 96.9% over \mathcal{H}_0 and 82.0% over \mathcal{H}_1 .

4.4 Uniting Heavy Ball Methods Using Measurements of ∇L

In this section, we propose a switching rule for the uniting algorithm that exploits measurements of ∇L , which in practice are typically approximated using measurements of L . Namely, the algorithm in this section has output h defined in (3.72). Unlike the switching rule in Section 4.3, the switching rule proposed in this section does not require prior knowledge of L or of its minimizer.

For the hybrid closed-loop algorithm \mathcal{H} in (4.3), the sets C_0 , C_1 , D_0 , and D_1 in this section are defined as

$$C_0 := \mathcal{U}_0 \times \{0\}, \quad C_1 := \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}} \times \{1\} \quad (4.14a)$$

$$D_0 := \mathcal{T}_{0,1} \times \{0\}, \quad D_1 := \mathcal{T}_{1,0} \times \{1\}. \quad (4.14b)$$

The sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ are precisely defined in Sections 4.4.1-4.4.3, but the idea behind their construction is as follows. As in Section 4.2, the switch between

³Code at github.com/HybridSystemsLab/UnitingLevelSetsHBF.

κ_0 and κ_1 is governed by a supervisory algorithm implementing switching logic; see Figure 4.1. When $z \in \mathcal{U}_0$ and $q = 0$ (i.e., $x \in C_0$), due to the design of \mathcal{U}_0 in Section 4.4.1, then the state z is near the minimizer, which is denoted z_1^* , and the supervisor allows flows of (4.3) using κ_0 , with large $\lambda_0 > 0$, to avoid oscillations. Conversely, when $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}$ and $q = 1$ (i.e., $x \in C_1$), due to the design of $\mathcal{T}_{1,0}$ in Section 4.4.2, then the state z is far from the minimizer and the supervisor allows flows of (4.3) using κ_1 , with large $\lambda_1 > 0$, to converge quickly to the neighborhood of the minimizer. When $z \in \mathcal{T}_{1,0}$ and $q = 1$ (i.e., $x \in D_1$), then this indicates that the state z is near the minimizer, and the supervisor assigns u to κ_0 and resets q to 0. Conversely, when $z \in \mathcal{T}_{0,1}$ and $q = 0$ (i.e., $x \in D_0$), due to the design of $\mathcal{T}_{0,1}$ in Section 4.4.3, then this indicates that the state z is far from the minimizer and the supervisor assigns u to κ_1 and resets q to 1. The complete algorithm, defined in (4.3) and (4.14), is summarized in Algorithm 2.

Algorithm 2 Uniting algorithm

- 1: Set $q(0,0)$ to 0, $\tau(0,0)$ to 0, and set $z(0,0)$ as an initial condition with an arbitrary value.
 - 2: **while** true **do**
 - 3: **if** $z \in \mathcal{T}_{0,1}$ and $q = 0$ **then**
 - 4: Reset q to 1.
 - 5: **else if** $z \in \mathcal{T}_{1,0}$ and $q = 1$ **then**
 - 6: Reset q to 0.
 - 7: **else if** $z \in \mathcal{U}_0$ and $q = 0$ **then**
 - 8: Assign u to $\kappa_0(h(z))$, where κ_0 is defined via (4.1) and h is defined in (3.72), and update z and q according to (4.3a).
 - 9: **else if** $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}$ and $q = 1$ **then**
 - 10: Assign u to $\kappa_1(h(z))$, where κ_1 is defined via (4.1) and h is defined in (3.72), and update z and q according to (4.3a).
 - 11: **end if**
 - 12: **end while**
-

4.4.1 Design of \mathcal{U}_0

In order for the supervisor to determine when the state component z_1 is close to the minimizer of L , denoted z_1^* , without knowledge of z_1^* or $L^* := L(z_1)$, we will use Assumptions 3.1.8 and 3.2.4. Under such assumptions, the following lemma, used in some of the results to follow, relates the size of the gradient at a point to the distance from the point to z_1^* .

Lemma 4.4.1. *(Suboptimality) Let L satisfy Assumptions 3.1.8 and 3.2.4, and let $\alpha > 0$ come from Assumption 3.2.4. For some $\varepsilon > 0$, if $z_1 \in \mathbb{R}^n$ is such that $|\nabla L(z_1)| \leq \varepsilon\alpha$, then $|z_1 - z_1^*| \leq \varepsilon$.*

Proof. Combining Assumption 3.1.8 and (3.79) from Assumption 3.2.4 with $u_1 = z_1^*$ and $w_1 = z_1$ yields

$$\alpha |z_1 - z_1^*|^2 \leq |L(z_1) - L^*| \leq |\langle \nabla L(z_1), z_1^* - z_1 \rangle| \leq |\nabla L(z_1)| |z_1 - z_1^*| \quad (4.15)$$

where the first inequality holds since $L(z_1) \geq L^*$. Then,

$$|z_1 - z_1^*| \leq \frac{1}{\alpha} |\nabla L(z_1)|. \quad (4.16)$$

From (4.16), we can deduce that $|\nabla L(z_1)| \leq \varepsilon\alpha$ implies $|z_1 - z_1^*| \leq \frac{1}{\alpha} (\varepsilon\alpha) = \varepsilon$. \square

The suboptimality condition from Lemma 4.4.1 is typically used as a stopping condition for optimization, as it indicates that the argument of L is close enough to the minimizer z_1^* [66]. We exploit Lemma 4.4.1 to determine when the state component z_1 of the hybrid closed-loop system \mathcal{H} is close enough to the minimizer z_1^* so as to switch to the local optimization algorithm, κ_0 , in this way activating \mathcal{H}_0 ; see Figure 4.1.

Recall from lines 7-8 of Algorithm 2 that the objective is to design \mathcal{U}_0 such

that when $z \in \mathcal{U}_0$ and $q = 0$, the state component z_1 is near z_1^* and the uniting algorithm allows flows of (4.3) with κ_0 in (4.1) and $q = 0$. For such a design, we use Assumptions 3.1.8 and 3.2.4 and the Lyapunov function V_0 in (4.2). Given $\varepsilon_0 > 0$, $c_0 > 0$, and $\gamma_0 > 0$ from κ_0 in (4.1), let $\alpha > 0$ come from Assumption 3.2.4 such that

$$\tilde{c}_0 := \varepsilon_0 \alpha > 0, \quad d_0 := c_0 - \gamma_0 \left(\frac{\tilde{c}_0^2}{\alpha} \right) > 0. \quad (4.17)$$

Then, V_0 in (4.2) can be upper bounded, using Assumption 3.1.8 as done to arrive to (4.15), as follows: for each $z \in \mathbb{R}^{2n}$

$$V_0(z) = \gamma_0 (L(z_1) - L^*) + \frac{1}{2} |z_2|^2 \leq \gamma_0 |\nabla L(z_1)| |z_1 - z_1^*| + \frac{1}{2} |z_2|^2. \quad (4.18)$$

Then, due to L being \mathcal{C}^1 , nonstrongly convex, and having a single minimizer z_1^* by Assumption 3.1.8, and due to L having quadratic growth away from z_1^* by Assumption 3.2.4, when $|\nabla L(z_1)| \leq \tilde{c}_0$, the suboptimality condition in Lemma 4.4.1 implies $|z_1 - z_1^*| \leq \frac{\tilde{c}_0}{\alpha}$, from where we get

$$V_0(z) \leq \gamma_0 \left(\frac{\tilde{c}_0^2}{\alpha} \right) + \frac{1}{2} |z_2|^2 \quad (4.19)$$

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

$$\mathcal{U}_0 := \left\{ z \in \mathbb{R}^{2n} : |\nabla L(z_1)| \leq \tilde{c}_0, \frac{1}{2} |z_2|^2 \leq d_0 \right\}, \quad (4.20)$$

every $z \in \mathcal{U}_0$ belongs to the c_0 -sublevel set of V_0 . In fact, using the conditions in (4.17) and (4.19), we have that for each $z \in \mathcal{U}_0$,

$$V_0(z) \leq \gamma_0 \left(\frac{\tilde{c}_0^2}{\alpha} \right) + \frac{1}{2} |z_2|^2 \leq c_0. \quad (4.21)$$

Since κ_0 in (4.1) is such that the set $\{z_1^*\} \times \{0\}$ is globally asymptotically stable for the closed-loop system resulting from controlling (3.1) by κ_0 , as was shown in the Proposition 3.2.8, the set \mathcal{U}_0 is contained in the basin of attraction induced by κ_0 .

4.4.2 Design of $\mathcal{T}_{1,0}$

Recall from lines 5-6 of Algorithm 2 that the objective is to design $\mathcal{T}_{1,0}$ such that when $z \in \mathcal{T}_{1,0}$ and $q = 1$, the state component z_1 is near z_1^* and the supervisor resets q to 0 and assigns u to $\kappa_0(h_0(z))$, where κ_0 is defined via (4.1) and h is defined in (3.72). For such a design, we use Assumptions 3.1.8 and 3.2.4 and the Lyapunov function V_1 in (4.2). Given $c_{1,0} \in (0, c_0)$ and $\varepsilon_{1,0} \in (0, \varepsilon_0)$, where $c_0 > 0$ and $\varepsilon_0 > 0$ come from Section 4.4.1, let \tilde{c}_0 and d_0 be given in (4.17), and let $\alpha > 0$ come from Assumption 3.2.4 such that

$$\tilde{c}_{1,0} := \varepsilon_{1,0}\alpha \in (0, \tilde{c}_0) \quad (4.22a)$$

$$d_{1,0} := c_{1,0} - \gamma_1 \left(\frac{\tilde{c}_{1,0}^2}{\alpha} \right) \in (0, d_0) \quad (4.22b)$$

Then, with V_1 given in (4.2) and using Assumption 3.1.8 with $u_1 = z_1^*$ and $w_1 = z_1$,

$$V_1(z) \leq \gamma_1 |\nabla L(z_1)| |z_1 - z_1^*| + \frac{1}{2} |z_2|^2. \quad (4.23)$$

Then, due to L being \mathcal{C}^1 , nonstrongly convex, and having a single minimizer z_1^* by Assumption 3.1.8, and due to L having quadratic growth away from z_1^* by Assumption 3.2.4, when $|\nabla L(z_1)| \leq \tilde{c}_{1,0}$, the suboptimality condition in Lemma 4.4.1 implies $|z_1 - z_1^*| \leq \frac{\tilde{c}_{1,0}}{\alpha}$, from where we get

$$V_1(z) \leq \gamma_1 \left(\frac{\tilde{c}_{1,0}}{\alpha} \right)^2 + \frac{1}{2} |z_2|^2. \quad (4.24)$$

Then, by defining $\mathcal{T}_{1,0}$ as

$$\mathcal{T}_{1,0} := \left\{ z \in \mathbb{R}^{2n} : |\nabla L(z_1)| \leq \tilde{c}_{1,0}, \frac{1}{2} |z_2|^2 \leq d_{1,0} \right\} \quad (4.25)$$

which, by construction, is contained in the interior of \mathcal{U}_0 defined in (4.20), every $z \in \mathcal{T}_{1,0}$ belongs to the $c_{1,0}$ -sublevel set of V_1 . In fact, using the conditions in (4.22) and (4.24), we have for each $z \in \mathcal{T}_{1,0}$,

$$V_1(z) \leq \gamma_1 \left(\frac{\tilde{c}_{1,0}}{\alpha} \right)^2 + \frac{1}{2} |z_2|^2 \leq c_{1,0}. \quad (4.26)$$

The constants \tilde{c}_0 , $\tilde{c}_{1,0}$, d_0 , and $d_{1,0}$ in (4.17) and (4.22) comprise the hysteresis necessary to avoid chattering at the switching boundary. The idea behind these hysteresis boundaries is as follows. When $z \in \mathcal{U}_0$ and $q = 1$, we have that $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}$, and it is not yet time to switch to κ_0 but to continue to flow using κ_1 . But once $z \in \mathcal{T}_{1,0}$ then z is close enough to $\{z_1^*\} \times \{0\}$, and the supervisor switches to κ_0 . Figure 4.3 illustrates the hysteresis mechanism in the design of \mathcal{U}_0 and $\mathcal{T}_{1,0}$.

4.4.3 Design of $\mathcal{T}_{0,1}$

To make the switch back to κ_1 in (4.1), we utilize Assumption 3.1.3. Recall from lines 3-4 of Algorithm 2 that the objective is to design $\mathcal{T}_{0,1}$ such that when $z \in \mathcal{T}_{0,1}$ and $q = 0$, the state component z_1 is far from z_1^* and the supervisor resets q to 1 and assigns u to $\kappa_1(h(z))$, h is defined in (3.72), so that κ_1 steers z_1 back to nearby z_1^* . Given $c_0 > 0$, let $\alpha > 0$ come from Assumption 3.2.4, and let $M > 0$ come from Assumption 3.1.3. Then, using Assumption 3.1.3 with $u_1 = z_1^*$ and $w_1 = z_1$ yields

$$|\nabla L(z_1)| \leq M |z_1 - z_1^*| \quad (4.27)$$

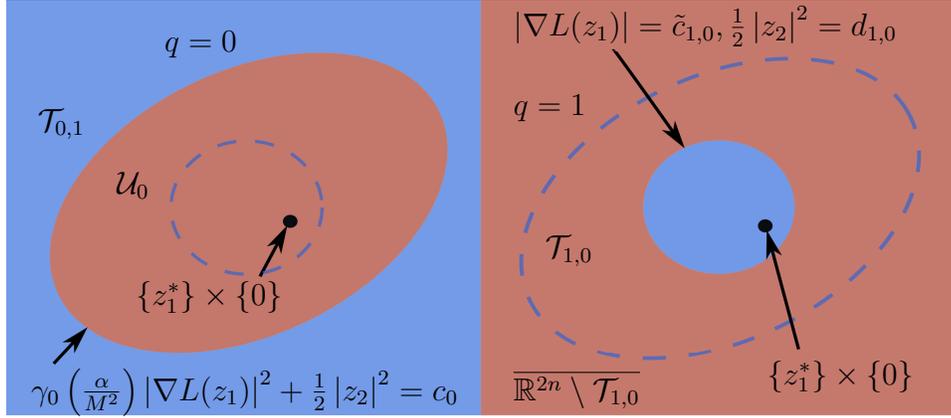


Figure 4.3: An illustration of hysteresis in the design of the sets \mathcal{U}_0 , \mathcal{T}_0 , and $\mathcal{T}_{0,1}$ on \mathbb{R}^{2n} , via the constants $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$, $d_{1,0} \in (0, d_0)$, and $c_0 > 0$. Left: due to the design of \mathcal{U}_0 in (4.20), every $z \in \mathcal{U}_0$ belongs to the c_0 -sublevel set of the Lyapunov function V_0 , where V_0 is defined via (4.2). Hence, the same value of $c_0 > 0$ is also used to define $\mathcal{T}_{0,1}$ as the closed complement of a sublevel set of V_0 with level equal to c_0 . Right: the constants $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ and $d_{1,0} \in (0, d_0)$, defined via (4.22), are chosen such that the set $\mathcal{T}_{1,0}$ in (4.25) is contained in the interior of \mathcal{U}_0 .

for all $z_1 \in \mathbb{R}^n$. Since L has quadratic growth away from z_1^* by Assumption 3.2.4, then dividing both sides of (4.27) by M and substituting into (3.79) leads to

$$L(z_1) - L^* \geq \frac{\alpha}{M^2} |\nabla L(z_1)|^2 \quad (4.28)$$

where $\alpha > 0$ comes from Assumption 3.2.4. Then, V_0 in (4.2) is lower bounded as follows: for each $z \in \mathbb{R}^{2n}$,

$$V_0(z) = \gamma_0 (L(z_1) - L^*) + \frac{1}{2} |z_2|^2 \geq \gamma_0 \left(\frac{\alpha}{M^2} \right) |\nabla L(z_1)|^2 + \frac{1}{2} |z_2|^2. \quad (4.29)$$

Using the right-hand side of (4.29) and the same $c_0 > 0$ as in Section 4.4.1, we define the set

$$\mathcal{T}_{0,1} := \left\{ z \in \mathbb{R}^{2n} : \gamma_0 \left(\frac{\alpha}{M^2} \right) |\nabla L(z_1)|^2 + \frac{1}{2} |z_2|^2 \geq c_0 \right\}. \quad (4.30)$$

The set in (4.30) defines the (closed) complement of a sublevel set of the Lyapunov function V_0 in (4.2) with level equal to c_0 . The constant c_0 is also a part of the hysteresis mechanism, as shown in Figure 4.3. When $z \in \mathcal{U}_0$ and $q = 0$, then the supervisor does not need to switch to κ_1 , as the state component z is close enough to the minimizer to keep using κ_0 . But if $z \in \mathcal{T}_{0,1}$ while $q = 0$, then z is far enough from the minimizer, and the supervisor then switches to κ_1 . Note that $\mathcal{T}_{0,1} \cap \mathcal{T}_{1,0} = \emptyset$.

4.4.4 Well-posedness of the Hybrid Closed-Loop System

\mathcal{H}

When L satisfies Assumptions 3.1.3, 3.1.8, and 3.2.4, the hybrid closed-loop system \mathcal{H} in (4.3) satisfies the hybrid basic conditions, listed in Definition 2.1.1, as demonstrated in the following lemma.

Lemma 4.4.2. (*Well-posedness of \mathcal{H}*) *Let the function L satisfy Assumptions 3.1.3, 3.1.8, and 3.2.4. Let the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ be defined via (4.20) (4.25), and (4.30), respectively. Let κ_q be defined via (4.1) and let h be defined in (3.72). Let $\lambda_q > 0$ and $\gamma_q > 0$. Then, the hybrid closed-loop system \mathcal{H} in (4.3), with C and D defined via (4.14), satisfies the hybrid basic conditions.*

Proof. The objective function L is \mathcal{C}^1 , nonstrongly convex, and has a single minimizer by Assumption 3.1.8. Therefore, since ∇L is continuous, the following hold: the set \mathcal{U}_0 , defined via (4.20), is closed since it is a sublevel set of the continuous function V_0 in (4.2); the set $\mathcal{T}_{1,0}$, defined via (4.25), is closed since it is a sublevel set of the continuous function V_1 in (4.2); the set $\mathcal{T}_{0,1}$, defined via (4.30), is closed since it is the closed complement of a set. Therefore, since the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ are closed, then the sets D_0 , D_1 , C_0 , and C_1 are closed.

Since C and D are both finite unions of finite and closed sets, then C and D are also closed.

Since by Assumption 3.1.8, L is \mathcal{C}^1 , then h in (3.72) and κ_q in (4.1) are continuous. In turn, the map $z \mapsto F_P(z, \kappa_q(h(z)))$ is also continuous since F_P in (3.1) is a \mathcal{C}^1 function of κ_q and h . Therefore, $x \mapsto F(x)$ in (4.3a) is continuous. The map G in (4.3b) satisfies (A3) by construction since it is continuous. \square

4.4.5 Existence of solutions to \mathcal{H}

Under Assumptions 3.1.3, 3.1.8, and 3.2.4, each maximal solution to \mathcal{H} is complete and bounded, as stated in the following lemma. The following lemma also states that $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$ and $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$. Such a property ensures that nontrivial solutions to \mathcal{H} in (4.3), which exist from each initial point in $C \cup D$, also exist from any initial point in $\mathbb{R}^{2n} \times Q$.

Proposition 4.4.3. *(Existence of solutions to \mathcal{H}) Let the function L satisfy Assumptions 3.1.3, 3.1.8, and 3.2.4. Let the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ be defined via (4.20), (4.25), and (4.30), respectively. Let κ_q be defined via (4.1) and let h be defined in (3.72). Let $\lambda_q > 0$ and $\gamma_q > 0$. Then, $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$, $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$, and each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to \mathcal{H} in (4.3), with C and D defined via (4.14), is bounded and complete.*

Proof. Since Assumptions 3.1.3, 3.1.8, and 3.2.4 hold, then \mathcal{H} satisfies the hybrid basic conditions by Lemma 4.4.2. With $\tilde{c}_0 > 0$ and $d_0 > 0$ defined via (4.17), since L is \mathcal{C}^1 , nonstrongly convex, has a single minimizer by Assumption 3.1.8, and has quadratic growth away from z_1^* by Assumption 3.2.4, from the arguments below (4.18), every $z \in \mathcal{U}_0$ belongs to the c_0 -sublevel set of V_0 ; recall that \mathcal{U}_0 is defined in (4.20) and that V_0 is defined via (4.2). Additionally, since by Assumption 3.2.4

L has quadratic growth away from z_1^* and since ∇L is Lipschitz continuous by Assumption 3.1.3, then $\mathcal{T}_{0,1}$ in (4.30) defines the closed complement of a sublevel set of V_0 with level equal to c_0 . Therefore, due to the definitions of \mathcal{U}_0 in (4.20) and $\mathcal{T}_{0,1}$ in (4.30), $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$. Furthermore, since $\mathcal{T}_{1,0}$ is defined via (4.25), and since by the definitions of C_1 and D_1 in (4.14), C_1 is the closed complement of D_1 , then $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$.

Due to the definitions of C_0 , D_0 , C_1 , and D_1 in (4.14), \mathcal{U}_0 in (4.20), $\mathcal{T}_{1,0}$ in (4.25), and $\mathcal{T}_{0,1}$ in (4.30), then $C \setminus D$ is equal to $\text{int}(C)$. Hence, for each point $x \in C \setminus D$, the tangent cone to C at x is defined in (4.8). Therefore, $F(x) \cap T_C(x) \neq \emptyset$, satisfying (VC) of Proposition A.1.1 for each point $x \in C \setminus D$, and nontrivial solutions exist for every initial point in $(C_0 \cup C_1) \cup (D_0 \cup D_1)$, where $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$ and $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$. To prove that item (c) of Proposition A.1.1 does not hold, we need to show that $G(D) \subset C \cup D$. With D defined in (4.14),

$$G(D) = (\mathcal{T}_{0,1} \times \{1\}) \cup (\mathcal{T}_{1,0} \times \{0\}). \quad (4.31)$$

Notice that $\mathcal{T}_{1,0} \times \{0\} \subset C_0$ and $\mathcal{T}_{0,1} \times \{1\} \subset C_1$. Therefore, $G(D) \subset C$; hence $G(D) \subset C \cup D$. Therefore, item (c) of Proposition A.1.1 does not hold. Then it remains to prove that item (b) does not happen.

We show that there is no finite time escape from C for \mathcal{H} as follows. First, since L is \mathcal{C}^1 , nonstrongly convex, and has a single minimizer z_1^* by Assumption 3.1.8, since ∇L is Lipschitz continuous by Assumption 3.1.3, and since L has quadratic growth away from z_1^* by Assumption 3.2.4, then each maximal solution to $\dot{z} = F_P(z, \kappa_q(h(z)))$, defined via (4.4) with κ_q in (4.1) and h in (3.72), is bounded, complete, and unique by Proposition 3.2.7. Next, for the hybrid closed-loop system \mathcal{H} in (4.3), with C and D defined via (4.14), since \mathcal{H}_q has no finite time escape from \mathbb{R}^{2n} , then this means $\dot{x} = F(x)$ has no finite time escape from

C for \mathcal{H} , as q does not change in C . Therefore, there is no finite time escape from $C \cup D$, for solutions x to \mathcal{H} in (4.3), with C and D defined via (4.14) and κ_q defined in (4.1). Therefore, item (b) from Proposition A.1.1 does not hold. \square

4.4.6 Main Result

In this section, we present a result that establishes UGAS of the set \mathcal{A} in (4.9), and a hybrid convergence rate that is exponential both globally and locally, for the hybrid closed-loop algorithm \mathcal{H} in (4.3), with C and D as defined in (4.14).

Theorem 4.4.4. *(Uniform global asymptotic stability of \mathcal{A} in (4.9) for \mathcal{H}) Let the function L satisfy Assumptions 3.1.3, 3.1.8, and 3.2.4. Let $\alpha > 0$ be generated by Assumption 3.2.4 and let $M > 0$ be generated by Assumption 3.1.3. Let $\lambda_q > 0$, $\gamma_q > 0$, $c_{1,0} \in (0, c_0)$, $\varepsilon_{1,0} \in (0, \varepsilon_0)$, be given. Let the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ be defined via (4.20), (4.25), and (4.30), respectively. Let κ_q be defined via (4.1) and let h be defined in (3.72). Then, the set \mathcal{A} , defined via (4.9), is uniformly globally asymptotically stable for \mathcal{H} given in (4.3), with C and D defined via (4.14). Furthermore, each maximal solution $(t, j) \mapsto x(t, j) = (z(t), q(t))$ to the closed-loop algorithm \mathcal{H} in (4.3) starting from C_1 satisfies items 1)-3) in Theorem 4.3.3*

Proof. The hybrid closed-loop algorithm \mathcal{H} in (4.3) – with \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ defined via (4.20), (4.25), and (4.30), respectively – satisfies the hybrid basic conditions by Lemma 4.4.2, satisfying the first assumption of Theorem A.1.3. Furthermore, $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$, $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$, and each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to \mathcal{H} is complete and bounded by Proposition 4.4.3. Since by Assumption 3.1.8, L has a unique minimizer z_1^* , then \mathcal{A} , defined via (4.9), is compact by construction, and $\mathcal{U} = \mathbb{R}^{2n} \times Q$ contains a nonzero open neighborhood of \mathcal{A} , satisfying the second assumption of Theorem A.1.3.

To prove attractivity of \mathcal{A} , we proceed by contradiction. Suppose there exists a complete solution x to \mathcal{H} such that $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} \neq 0$. Since Proposition 4.4.3 guarantees completeness of maximal solutions, we have the following cases:

- a) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_1 \setminus D_1$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$;
- b) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_0 \setminus (\mathcal{A} \cup D_0)$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$;
- c) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in D$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$.

Case a) contradicts the fact that, by Proposition 3.2.8, the set $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_1 in (4.4). Such uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$, guaranteed by Proposition 3.2.8, implies there exist $\tilde{c}_1 \in (0, \tilde{c}_{1,0})$ and $d_1 \in (0, d_{1,0})$ such that the state z reaches $(\{z_1^*\} + \tilde{c}_1 \mathbb{B}) \times (\{0\} + d_1 \mathbb{B}) \subset \mathcal{T}_{1,0}$ as $t \rightarrow \infty$. In turn, due to the construction of C_1 and D_1 in (4.14), with $\mathcal{T}_{1,0}$ defined via (4.25), the solution x must reach D_1 at some $(t, j) \in \text{dom } x, t + j \geq t' + j'$. Therefore, case a) does not happen.

Case b) contradicts the fact that, by Proposition 3.2.8, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_0 in (4.4). In fact, $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} = 0$, and since $\mathcal{A} \subset C_0$, case b) does not happen.

Case c) contradicts the fact that, due to the construction of $\mathcal{T}_{1,0}$ in (4.25) and $\mathcal{T}_{0,1}$ in (4.30), we have

$$G(D) \cap D := ((\mathcal{T}_{0,1} \times \{1\}) \cup (\mathcal{T}_{1,0} \times \{0\})) \cap ((\mathcal{T}_{0,1} \times \{0\}) \cup (\mathcal{T}_{1,0} \times \{1\})) = \emptyset \quad (4.32)$$

where $G(D)$ is defined via (4.31) and D is defined in (4.14). Such an equality

holds since $\mathcal{T}_{1,0} \cap \mathcal{T}_{0,1} = \emptyset$; see the end of Section 4.4.3. Therefore, case c) does not happen.

Therefore, cases a)-c) do not happen, and each maximal and complete solution $x = (z, q)$ to \mathcal{H} converges to \mathcal{A} . Consequently, by the construction of C and D in (4.14), the uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_q established in Proposition 3.2.8, and since each maximal solution to \mathcal{H} is complete by Proposition 4.4.3, the set \mathcal{A} in (4.9) is uniformly globally asymptotically stable for \mathcal{H} in (4.3), with C and D defined via (4.14).

To show that each maximal and complete solution x to \mathcal{H} jumps no more than twice, we proceed by contradiction. Without loss of generality, suppose there exists a maximal and complete solution that jumps three times. We have the following possible cases:

- i) The solution first jumps at a point in D_0 , then jumps at a point in D_1 , and then jumps at a point in D_0 ; or
- ii) The solution first jumps at a point in D_1 , then jumps at a point in D_0 , and then jumps at a point in D_1 .

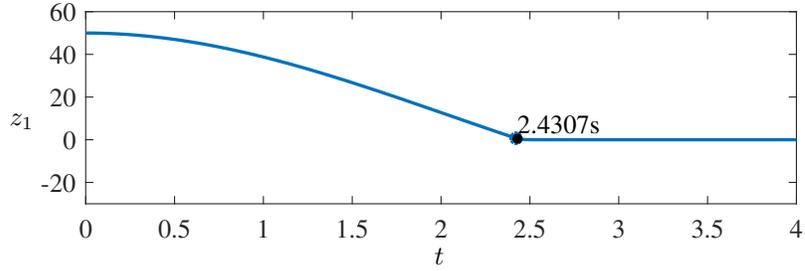
Case i) does not hold since, once the jump in D_1 occurs, the solution x is in $(\mathcal{T}_{1,0} \times \{0\}) \subset C_0$. Due to the construction of $\mathcal{T}_{1,0}$ in (4.25) and $\mathcal{T}_{0,1}$ in (4.30) such that $\mathcal{T}_{1,0} \cap \mathcal{T}_{0,1} = \emptyset$, as described in the contradiction of case c) above, and due to the uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_q in (4.4) by Proposition 3.2.8, the solution x will never return to D_0 . Therefore, case i) does not happen. Case ii) leads to a contradiction for the same reason, and in this case, once the first jumps in D_1 occurs, no more jumps happen. Therefore, since cases i)-ii) do not happen, each maximal and complete solution x to \mathcal{H} in (4.3), with C and D defined via (4.14), has no more than two jumps.

Finally, we prove the hybrid convergence rate of \mathcal{H} . By Proposition 3.2.10, since L satisfies Assumptions 3.1.8 and 3.2.4, then, given $\gamma_q > 0$ and $\lambda_q > 0$, for each $m \in (0, 1)$ such that $\psi_q := \frac{m\alpha\gamma_q}{\lambda_q} > 0$ and $\nu_q := \psi_q(\psi_q - \lambda_q) < 0$, each maximal solution $t \mapsto z(t)$ to the closed-loop algorithm \mathcal{H}_q in (4.4) satisfies (3.91) for all $t \in \text{dom } z (= \mathbb{R}_{\geq 0})$. Since maximal solutions $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to \mathcal{H} starting from C_1 are guaranteed to jump no more than once, as implied by the contradiction in cases i)-ii) above, then the domain of each maximal solution x to \mathcal{H} starting from C_1 is $\cup_{j=0}^1 (I^j, j)$, with I^0 of the form $[t_0, t_1]$ and with I^1 of the form $[t_1, \infty)$. Therefore, given $\lambda_q > 0$, $\gamma_q > 0$, $c_{1,0} \in (0, c_0)$, $\varepsilon_{1,0} \in (0, \varepsilon_0)$, $\alpha > 0$ from Assumption 3.2.4, and $M > 0$ from Assumption 3.1.3, due to the construction of \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ in (4.20), (4.25), and (4.30), with $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ and $d_{1,0} \in (0, d_0)$ defined via (4.17) and (4.22), and due to the individual convergence rates of \mathcal{H}_q , each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to the hybrid closed-loop algorithm \mathcal{H} in (4.3) that starts in C_1 satisfies (4.10) for each $t \in I^0$ at which $q(t, 0)$ is equal to 1, and satisfies (4.11) for each $t \in I^1$ at which $q(t, 1)$ is equal to 0. □

4.4.7 Numerical Examples

In this section, we present multiple numerical examples to illustrate the hybrid closed-loop algorithm \mathcal{H} in (4.3), with C and D defined in (4.14). Example 4.4.5 first illustrates the operation of the nominal hybrid closed-loop algorithm \mathcal{H} , and then demonstrates the robustness of \mathcal{H} to different amounts of noise in measurements of ∇L . Example 4.4.6 compares solutions to the hybrid closed-loop algorithm with solutions to \mathcal{H}_0 and \mathcal{H}_1 .

Example 4.4.5. *In this example, we simulate a solution to the nominal hybrid closed-loop system \mathcal{H} to illustrate how the uniting algorithm works. Then, we*



σ	$\limsup_{t+j \rightarrow \infty} z_1(t, j) - z_1^* $	$\limsup_{t+j \rightarrow \infty} L(z_1(t, j)) - L^* $
0.01	1.621×10^{-4}	1.314×10^{-8}
0.1	3.048×10^{-3}	4.645×10^{-6}
0.5	8.400×10^{-3}	3.528×10^{-5}
1	1.748×10^{-3}	1.528×10^{-6}
5	3.387×10^{-2}	5.736×10^{-4}
10	6.666×10^{-2}	2.222×10^{-3}
15	1.332×10^{-1}	8.871×10^{-3}
20	1.114×10^{-1}	6.209×10^{-3}
25	5.588×10^{-1}	1.931×10^{-3}

Figure 4.4: Top: The evolution over time of z_1 , for the nominal hybrid closed-loop algorithm \mathcal{H} in (4.3), with C and D defined in (4.14), for a function $L(z_1) := \frac{1}{2}z_1^2$ with a single minimizer at $z_1^* = 0$. The time at which the solution settles to within 1% of z_1^* is marked with a dot and labeled in seconds. The jump is labeled with an asterisk. Bottom: Simulation results for perturbed solutions using zero mean Gaussian noise, with each simulation using a different value of the standard deviation σ . Results listed are for a large value of $t + j$.

compare that same solution to solutions with different amounts of noise in measurements of ∇L . For both the nominal system and the perturbed system, the choice of objective function, parameter values, and initial conditions are as follows. We use the objective function $L(z_1) := \frac{1}{2}z_1^2$, which is nonstrongly convex with a Lipschitz continuous gradient, with constant $M = 1$, and which has a single minimizer at $z_1^* = 0$. This choice of objective function is made so that we can easily tune λ , as described in Section 5.1.6. We arbitrarily chose the heavy ball parameter values $\gamma_0 = \frac{2}{3}$ and $\gamma_1 = \frac{1}{2}$. We tuned λ_0 to 10.5 by choosing a value arbitrarily larger than $2\sqrt{a_1}$, where a_1 comes from Section 5.1.6, and gradually increasing it until there is no overshoot in the hybrid algorithm. We tuned λ_1 to $\frac{1}{5}$ by choosing a value arbitrarily smaller than $2\sqrt{a_1}$ and gradually decreasing until the switch to \mathcal{H}_0 occurs once z_1 reaches the desired neighborhood of z_1^* .

The parameter values for the uniting algorithm are $c_0 = 9000$, $c_{1,0} \approx 499.38$, $\varepsilon_0 = 20$, $\varepsilon_{1,0} = 15$, and $\alpha = \frac{1}{2}$, which yield the values $\tilde{c}_0 = 20$, $\tilde{c}_{1,0} = 15$, $d_0 \approx 8733.3$, and $d_{1,0} \approx 386.88$, which are calculated via (4.17) and (4.22). These values are chosen for proper tuning of the algorithm, in order to get nice performance, and the value of $c_{1,0}$ is chosen to exploit the properties of \mathcal{H}_1 for a longer time, so that the nominal solution gets closer to the minimizer faster. Initial conditions for \mathcal{H} are $z_1(0,0) = 50$, $z_2(0,0) = 0$, and $q(0,0) = 1$. The plot on the top in Figure 4.4 shows the solution to the nominal hybrid closed-loop algorithm \mathcal{H} , namely, the value of z_1 over time, with the time it takes for the solution to settle to within 1% of z_1^* marked with a black dot and labeled in seconds. The jump at which the switch from \mathcal{H}_1 to \mathcal{H}_0 occurs is labeled with an asterisk. The solution converges quickly, without oscillations near the minimizer.

To show that the uniform global asymptotic stability of \mathcal{A} , established in Theorem 4.4.4, is robust to small perturbations, due to the hybrid closed-loop system

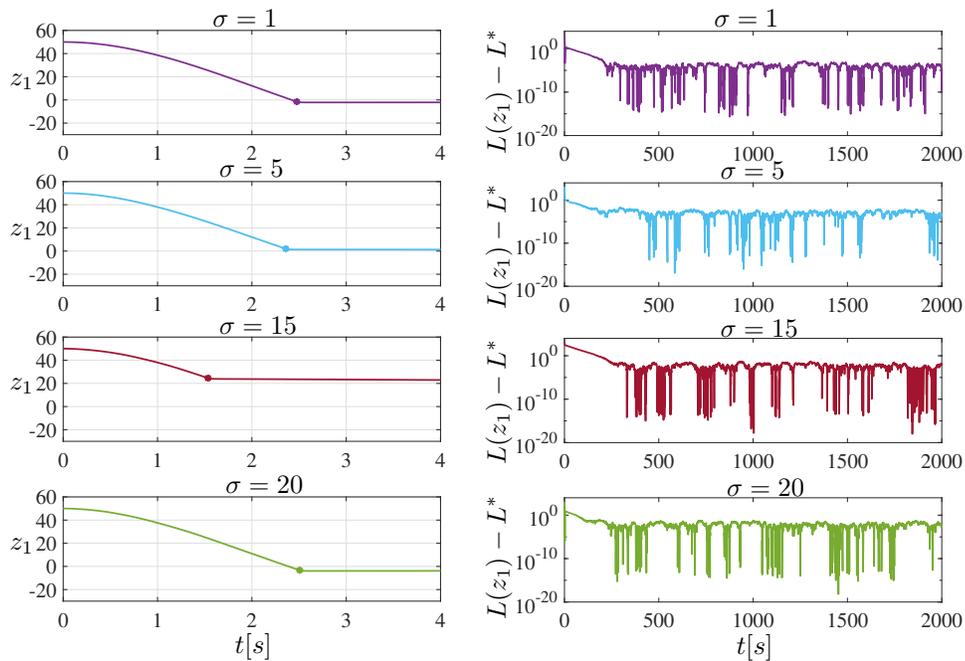


Figure 4.5: Simulation results for hybrid closed-loop algorithm \mathcal{H} in (4.3), with C and D defined in (4.14), for a function $L(z_1) := \frac{1}{2}z_1^2$ with a single minimizer at $z_1^* = 0$, with zero-mean Gaussian noise added to measurements of the gradient. Each subplot is labeled with the standard deviation used. Left subplots: the value of z_1 over time for each perturbed solution, with the jump in each solution labeled by an asterisk. Right subplots: the corresponding value of L over time for each perturbed solution.

\mathcal{H} satisfying the hybrid basic conditions by Lemma 4.4.2. we simulate the hybrid algorithm, using the objective function, parameter values, and initial conditions listed in the first paragraph of this example, with zero-mean Gaussian noise added to measurements of the gradient. Separate simulations were run for each of the following standard deviations: $\sigma \in \{0.01, 0.1, 0.5, 1, 5, 10, 15, 20, 25\}$. Figure 4.5 shows some of these perturbed solutions⁴, with each subplot labeled with the corresponding standard deviation used. The subplots on the left side of Figure 5.2 show the value of z_1 over time for different standard deviations, and the subplots on the right side of Figure 4.5 show the corresponding value of L over time for such standard deviations. Note that, while most perturbed solutions shown in Figure 4.5 get close to the minimizer quickly, the solution with $\sigma = 15$ does not, as the random noise causes the solution to jump too early. However, the solution with $\sigma = 15$ still gets close to the minimizer eventually, as seen in Figure 4.5 and the table in the bottom of Figure 4.4. Overall, the perturbed solutions in Figure 4.5 do not get as close to the minimizer as the solution to the nominal algorithm does; see the plot on the top in Figure 4.4. Also note that, in general, as the standard deviation gets larger, the corresponding perturbed solution stays slightly farther away from the minimizer. The results for all standard deviations are listed in the table in Figure 4.4, showing the neighborhood of z_1^* that each solution settles to, for a large value of $t + j$, along with the corresponding value of L .

Example 4.4.6. As was done in Example 4.3.4, in this example we show the effectiveness of the hybrid algorithm \mathcal{H} in (4.3), with C and D defined via (4.14), by comparing it in simulation with the individual optimization algorithms \mathcal{H}_0 and \mathcal{H}_1 in (4.4). We use the same objective function as in Example 4.3.4, namely, $L = \frac{1}{4}z_1^\top Pz_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer

⁴Code at github.com/HybridSystemsLab/UnitingRobustnessHBF.

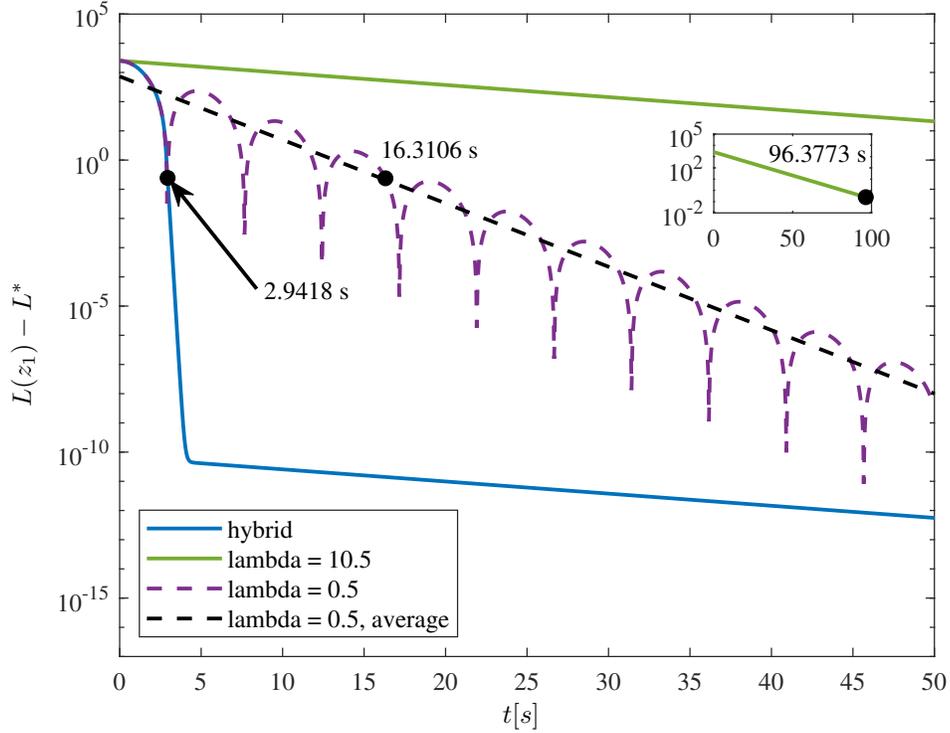


Figure 4.6: A comparison of the evolution of L over time for \mathcal{H}_0 , \mathcal{H}_1 , and \mathcal{H} in (4.3), with C and D defined in (4.14), for a function $L = \frac{1}{4}z_1^\top Pz_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer at $z_1^* = (0, 0, \dots, 0)$. The heavy ball algorithm \mathcal{H}_1 uses $\lambda_1 = \frac{1}{2}$ (shown in purple) and settles to within 1% of z_1^* in about 16.3 seconds. The heavy ball algorithm \mathcal{H}_0 uses $\lambda_0 = 10.5$ (shown in green) and settles to within 1% of z_1^* in about 96.4 seconds. The hybrid closed-loop system \mathcal{H} (shown in blue) settles to within 1% of z_1^* in about 2.9 seconds.

at $z_1^* = (0, 0, \dots, 0)$. The Lipschitz constant of L is $M = \frac{1}{2}$. We use the same approach, described in Example 4.3.4, to tuning λ_q , which leads to $\lambda_0 = 10.5$ and $\lambda_1 = \frac{1}{2}$. Additionally, we use the same arbitrarily chosen $\gamma_0 = \gamma_1 = \frac{1}{2}$ as in Example 4.3.4. The parameter values for the uniting algorithm are $c_0 = 2000$, $c_{1,0} \approx 1351.95$, $\varepsilon_0 = 35$, $\varepsilon_{1,0} = 25$, and $\alpha = \frac{1}{4}$, which yield the values $\tilde{c}_0 = 8.75$, $\tilde{c}_{1,0} = 6.25$, $d_0 \approx 1846.9$, and $d_{1,0} \approx 1273.8$, which are calculated via (4.17) and (4.22). These values are chosen for proper tuning of the algorithm, in order to get nice performance, and the value of $c_{1,0}$ is chosen to exploit the properties of \mathcal{H}_1 for a longer time, so that the solution gets closer to the minimizer faster. Initial conditions for \mathcal{H} , \mathcal{H}_0 , and \mathcal{H}_1 are $z_1(0, 0) = -10$, $z_2(0, 0) = 0$, and $q(0, 0) = 0$.

Algorithm	Time to converge (s)	% improvement of \mathcal{H}
\mathcal{H}	2.942	–
\mathcal{H}_0	96.377	96.9
\mathcal{H}_1	16.311	82.0

Table 4.2: Average times for which \mathcal{H} in (4.3), with C and D defined in (4.14), \mathcal{H}_0 , and \mathcal{H}_1 settle to within 1% of z_1^* , and the average percent improvement of \mathcal{H} over each algorithm. Percent improvement is calculated via (4.13). The objective function used for this table is $L = \frac{1}{4}z_1^\top Pz_1$, where $z_1 \in \mathbb{R}^{100}$ and $P = I_{100 \times 100}$, which has a single minimizer at $z_1^* = (0, 0, \dots, 0)$.

Table 4.2 shows the time that each algorithm takes to settle within⁵ 1% of z_1^* , and the percent improvement of \mathcal{H} over \mathcal{H}_0 , \mathcal{H}_1 , and *HAND-1*, which is calculated using the formula in (4.13). As can be seen in Figure 4.6 and Table 4.2, \mathcal{H} converges faster than the other algorithms, and the average percent improvement of \mathcal{H} over each of the other algorithms in Table 5.2 is 96.9% over \mathcal{H}_0 and 82.0% over \mathcal{H}_1 .

⁵Code at github.com/HybridSystemsLab/UnitingGradientsHBF.

4.5 Extensions

Some possible extensions to the results in Section 4.3 are as follows.

It is possible to extend the results in Section 4.3 to include \mathcal{C}^1 , nonstrongly convex objective functions L with a compact and connected set of minimizers. With such an assumption, it would be straightforward to extend Lemma 4.3.1, Proposition 4.3.2, and the UGAS result in Theorem 4.3.3. The exponential convergence rate result in Theorem 4.3.3 can be extended via the assumption of a compact and connected set of minimizers and the use of Clarke's generalized derivative in (2.4) with the Lyapunov function in (3.97), as described in Section 3.2.3. In particular, with the assumption that L has a compact and connected set of minimizers, it can be shown that \mathcal{A} in (4.9) is UGAS for \mathcal{H} in (4.3), with C and D defined via (4.5), and it can be shown that each maximal solution to \mathcal{H} that starts in C_1 satisfies (4.10) for each $t \in I^0$ at which $q(t, 0)$ is equal to 1 and $t \geq 1$, and satisfies (4.11) for each $t \in I^1$ at which $q(t, 1)$ is equal to 0.

It would be possible to further extend the results in Lemma 4.3.1, Proposition 4.3.2, and Theorem 4.3.3 to nonstrongly convex objective functions L that are also nonsmooth, through the use of Clarke's generalized derivative.

It would be possible to extend the results in Section 4.4 in the following ways.

With the extra assumption that the \mathcal{C}^1 , nonstrongly convex objective functions L has a compact and connected set of minimizers, it would be straightforward to extend Lemmas 4.4.1 and 4.4.2, Proposition 4.4.3, and the UGAS result in Theorem 4.3.3. The exponential convergence rate result in Theorem 4.4.4 can be extended via the assumption of a compact and connected set of minimizers and the use of Clarke's generalized derivative in (2.4) with the Lyapunov function in (3.97), as described in Section 3.2.3. In particular, with the assumption that L has a compact and connected set of minimizers, it can be shown that \mathcal{A} in (4.9) is

UGAS for \mathcal{H} in (4.3), with C and D defined via (4.14), and it can be shown that each maximal solution to \mathcal{H} that starts in C_1 satisfies (4.10) for each $t \in I^0$ at which $q(t, 0)$ is equal to 1 and $t \geq 1$, and satisfies (4.11) for each $t \in I^1$ at which $q(t, 1)$ is equal to 0.

It would be possible to further extend the results in Lemmas 4.4.1 and 4.4.2, Proposition 4.4.3, and Theorem 4.4.4 to include \mathcal{C}^1 , nonstrongly convex objective functions L that are also nonsmooth, through the use of Clarke's generalized derivative.

Chapter 5

Uniting Nesterov's Method and the Heavy Ball Method

In this chapter, we propose algorithms uniting Nesterov's algorithm globally and the heavy ball algorithm locally, in order to achieve UGAS of the unique minimizer of L , while taking advantage of the convergence rates of Nesterov's algorithm and heavy ball in order to achieve a fast (hybrid) convergence rate. We first present such an algorithm for strongly convex L , and then we present an algorithm nonstrongly convex L .

5.1 Strongly Convex L

For the algorithm we present in this section, we impose Assumption 3.1.1 on L and 3.1.3 on ∇L . Namely, L is \mathcal{C}^2 , strongly convex, and ∇L is Lipschitz continuous with constant $M > 0$.

5.1.1 Problem Statement

As illustrated in Figure 1.2, 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 1.3 we discussed how Nesterov’s algorithm guarantees an exponential convergence rate for strongly convex L . We also discussed how the heavy ball algorithm guarantees an exponential convergence rate for strongly convex L . We desire to attain such rates globally and locally, while avoiding oscillations via the heavy ball algorithm with large λ . We state the problem to solve as follows:

Problem 5.1.1. *Given a scalar, real-valued, continuously differentiable, and strongly convex objective function L , design an optimization algorithm that, without knowing the function L or the location of its minimizer, has the minimizer uniformly globally asymptotically stable, with an exponential convergence rate globally and locally, and with robustness to arbitrarily small noise in measurements of ∇L .*

5.1.2 Modeling

In this section, we present an algorithm that solves Problem 5.1.1. As in Chapter 4, defining z_1 as ξ and z_2 as $\dot{\xi}$, we interpret the heavy ball ODE in (1.1) as a closed-loop system consisting of the plant in (3.1) and a control algorithm assigning u to κ in (3.71). We interpret the Nesterov ODE in (1.2) as a closed-loop system consisting of the plant in (3.1) and a control algorithm assigning u to κ in (3.2).

The proposed logic-based algorithm “unites” the two optimization algorithms modeled by κ_q , where the logic variable $q \in Q := \{0, 1\}$ indicates which algorithm

is currently being used. The local and global algorithms, respectively, are defined as

$$\kappa_0(h_0(z)) = -\lambda z_2 - \gamma \nabla L(z_1) \quad (5.1a)$$

$$\kappa_1(h_1(z)) = -2dz_2 - \frac{1}{M} \nabla L(z_1 + \beta z_2) \quad (5.1b)$$

where $\gamma > 0$, $\lambda > 0$, $M > 0$ is the Lipschitz constant for ∇L , and d and β are defined in (3.3). Note that, as in [12], we have set $\zeta = 1$ in (1.2) and, consequently, in (5.1b), for simplicity of analysis. The algorithm defined by κ_1 plays the role of the global algorithm in uniting control (see, e.g., [22, Chapter 4]), while the algorithm defined by κ_0 plays the role of the local algorithm. The outputs h_q are defined as

$$h_0(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1) \end{bmatrix}, \quad h_1(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1 + \beta z_2) \end{bmatrix}. \quad (5.2a)$$

where β is defined via (3.3). Namely, the algorithm exploits measurements of ∇L , which in practice are typically approximated using measurements of L . The parameters $\lambda > 0$ and $\gamma > 0$ should be designed to achieve convergence without oscillations nearby the minimizer.

To encapsulate the plant and static state-feedback laws, the hybrid closed-loop system \mathcal{H} with state $x := (z, q) \in \mathbb{R}^{2n} \times Q$ is defined as

$$\left. \begin{array}{l} \dot{z} = \begin{bmatrix} z_2 \\ \kappa_q(h_q(z)) \end{bmatrix} \\ \dot{q} = 0 \end{array} \right\} =: F(x) \quad x \in C := C_0 \cup C_1 \quad (5.3a)$$

$$\left. \begin{aligned} z^+ &= \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \\ q^+ &= 1 - q \end{aligned} \right\} =: G(x) \quad x \in D := D_0 \cup D_1 \quad (5.3b)$$

with the sets C_0 , C_1 , D_0 , and D_1 defined in (4.14). We denote, for each $q \in Q := \{0, 1\}$, the closed-loop systems resulting from the individual optimization algorithms as \mathcal{H}_q , which is defined as

$$\dot{z} = \begin{bmatrix} z_2 \\ \kappa_q(h_q(z)) \end{bmatrix} \quad z \in \mathbb{R}^{2n}. \quad (5.4)$$

Namely, the closed-loop resulting from using κ_1 (Nesterov's algorithm) is denoted as \mathcal{H}_1 , and the closed-loop resulting from using κ_0 (heavy ball) is denoted as \mathcal{H}_0 .

The sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ are precisely defined in Sections 5.1.3-5.1.5, but the idea behind their construction is as follows. The switch between κ_0 and κ_1 is governed by a *supervisory algorithm* implementing switching logic; see Figure 4.1. The supervisor selects between these two optimization algorithms, based on the output of the plant and the optimization algorithm currently applied. When $z \in \mathcal{U}_0$ and $q = 0$ (i.e., $x \in C_0$), due to the design of \mathcal{U}_0 in Section 5.1.3, then the state z is near the minimizer, which is denoted z_1^* , and the supervisor allows flows of (5.3) using κ_0 in (5.1a) (heavy ball, with large λ) to avoid oscillations. Conversely, when $z \in \overline{\mathbb{R}^{2n}} \setminus \overline{\mathcal{T}_{1,0}}$ and $q = 1$ (i.e., $x \in C_1$), due to the design of $\mathcal{T}_{1,0}$ in Section 5.1.4, then the state z is far from the minimizer and the supervisor allows flows of (5.3) using κ_1 in (5.1b) (Nesterov's algorithm) to converge quickly to the neighborhood of the minimizer. When $z \in \mathcal{T}_{1,0}$ and $q = 1$ (i.e., $x \in D_1$), then this indicates that the state z is near the minimizer, and the supervisor assigns u to κ_0 in (5.1a) and resets q to 0. Conversely, when $z \in \mathcal{T}_{0,1}$ and $q = 0$ (i.e., $x \in D_0$), due to the design of $\mathcal{T}_{0,1}$ in Section 4.4.3, then this indicates that the state z is far

from the minimizer and the supervisor assigns u to κ_1 in (5.1b) and resets q to 1. The complete algorithm, defined in (5.3) and (4.14), is summarized in Algorithm 2.

5.1.3 Design of \mathcal{U}_0

Recall from lines 7-8 of Algorithm 2 that the objective is to design \mathcal{U}_0 such that when $z \in \mathcal{U}_0$ and $q = 0$, the state component z_1 is near z_1^* and the uniting algorithm allows flows of (5.3) with κ_0 in (5.1a) and $q = 0$. For such a design, we use Definition 2.2.2 and¹ 2.2.3 and the Lyapunov function V_0 in (3.80), where $\gamma > 0$. Given $\varepsilon_0 > 0$, $c_0 > 0$, and $\gamma > 0$ from κ_0 in (5.1a), let $\alpha > 0$ come from Definition 2.2.3 such that

$$\tilde{c}_0 := \varepsilon_0 \alpha > 0, \quad d_0 := c_0 - \gamma \left(\frac{\tilde{c}_0^2}{\alpha} \right) > 0. \quad (5.5)$$

Since the strong convexity of L in Assumption 3.1.1 implies that L also satisfies Definition 2.2.2, then V_0 in (3.80) can be upper bounded, using Definition 2.2.2 as done to arrive to (4.15), as follows: for each $z \in \mathbb{R}^{2n}$

$$V_0(z) = \gamma (L(z_1) - L^*) + \frac{1}{2} |z_2|^2 \leq \gamma |\nabla L(z_1)| |z_1 - z_1^*| + \frac{1}{2} |z_2|^2. \quad (5.6)$$

Then, due to L being \mathcal{C}^2 , strongly convex, and having a single minimizer z_1^* by Assumption 3.1.1 – which implies that L also satisfies Definition 2.2.2 and has quadratic growth away from z_1^* as in Definition 2.2.3, when $|\nabla L(z_1)| \leq \tilde{c}_0$, the

¹The strong convexity of L in Assumption 3.1.1 implies that L also satisfies Definition 2.2.2; see [66]. The strong convexity of L in Assumption 3.1.1 also implies quadratic growth of L in Definition 2.2.3, as quadratic growth is a weaker property than strong convexity; see [67], [19], [68], [69], [70].

suboptimality condition in Lemma 4.4.1 implies $|z_1 - z_1^*| \leq \frac{\tilde{c}_0}{\alpha}$, from where we get

$$V_0(z) \leq \gamma \left(\frac{\tilde{c}_0^2}{\alpha} \right) + \frac{1}{2} |z_2|^2 \quad (5.7)$$

Then, by defining the set \mathcal{U}_0 as in (4.20), every $z \in \mathcal{U}_0$ belongs to the c_0 -sublevel set of V_0 , with V_0 defined via (3.80). In fact, using the conditions in (5.5) and (5.7), we have that for each $z \in \mathcal{U}_0$,

$$V_0(z) \leq \gamma \left(\frac{\tilde{c}_0^2}{\alpha} \right) + \frac{1}{2} |z_2|^2 \leq c_0. \quad (5.8)$$

Since κ_0 in (5.1a) is such that the set $\{z_1^*\} \times \{0\}$ is globally asymptotically stable for the closed-loop system resulting from controlling (3.1) by κ_0 , as was shown in the Proposition 3.2.8, the set \mathcal{U}_0 is contained in the basin of attraction induced by κ_0 .

5.1.4 Design of $\mathcal{T}_{1,0}$

Recall from lines 5-6 of Algorithm 2 that the objective is to design $\mathcal{T}_{1,0}$ such that when $z \in \mathcal{T}_{1,0}$ and $q = 1$, the state component z_1 is near z_1^* and the supervisor resets q to 0, resets τ to 0, and assigns u to $\kappa_0(h_0(z))$, where κ_0 is defined in (5.1a) and h_0 is defined via (5.2). For such a design, we use Assumption 3.1.1 – which implies that L also satisfies Definitions 2.2.2 and 2.2.3, the suboptimality condition from Lemma 4.4.1, and the Lyapunov function in (3.8), defined for each $z \in \mathbb{R}^{2n}$, where $M > 0$ is the Lipschitz constant of ∇L , the constants d and β are defined via (3.3), and the constant a is defined in (3.9).

Given $c_{1,0} \in (0, c_0)$ and $\varepsilon_{1,0} \in (0, \varepsilon_0)$, where $c_0 > 0$ and $\varepsilon_0 > 0$ come from Section 5.1.3, let \tilde{c}_0 and d_0 be given in (5.5), and let $\alpha > 0$ come from Definition

2.2.3 such that

$$\tilde{c}_{1,0} := \varepsilon_{1,0}\alpha \in (0, \tilde{c}_0) \quad (5.9a)$$

$$d_{1,0} := c_{1,0} - a^2 \left(\frac{\tilde{c}_{1,0}}{\alpha} \right)^2 - \frac{1}{M} \left(\frac{\tilde{c}_{1,0}^2}{\alpha} \right) \in (0, d_0) \quad (5.9b)$$

Then, with V_1 given in (3.8) and using Definition 2.2.2 with $u_1 = z_1^*$ and $w_1 = z_1$,

$$V_1(z) \leq a^2 |z_1 - z_1^*|^2 + |z_2|^2 + \frac{1}{M} |\nabla L(z_1)| |z_1 - z_1^*| \quad (5.10)$$

Then, due to L being \mathcal{C}^2 , strongly convex, and having a single minimizer z_1^* by Assumption 3.1.1 – which implies that L also satisfies Definition 2.2.2 and has quadratic growth away from z_1^* as in Definition 2.2.3, when $|\nabla L(z_1)| \leq \tilde{c}_{1,0}$, the suboptimality condition in Lemma 4.4.1 implies $|z_1 - z_1^*| \leq \frac{\tilde{c}_{1,0}}{\alpha}$, from where we get

$$V_1(z) \leq a^2 \left(\frac{\tilde{c}_{1,0}}{\alpha} \right)^2 + |z_2|^2 + \frac{1}{M} \left(\frac{\tilde{c}_{1,0}^2}{\alpha} \right). \quad (5.11)$$

Then, by defining $\mathcal{T}_{1,0}$ as

$$\mathcal{T}_{1,0} := \left\{ z \in \mathbb{R}^{2n} : |\nabla L(z_1)| \leq \tilde{c}_{1,0}, |z_2|^2 \leq d_{1,0} \right\} \quad (5.12)$$

which, by construction, is contained in the interior of \mathcal{U}_0 defined in (4.20), every $z \in \mathcal{T}_{1,0}$ belongs to the $c_{1,0}$ -sublevel set of V_1 . In fact, using the conditions in (5.9) and (5.11), we have for each $z \in \mathcal{T}_{1,0}$,

$$V_1(z) \leq a^2 \left(\frac{\tilde{c}_{1,0}}{\alpha} \right)^2 + |z_2|^2 + \frac{1}{M} \left(\frac{\tilde{c}_{1,0}^2}{\alpha} \right) \leq c_{1,0}. \quad (5.13)$$

As was described below (4.26), the constants \tilde{c}_0 , $\tilde{c}_{1,0}$, d_0 , and $d_{1,0}$ in (5.5) and (5.9) comprise the hysteresis necessary to avoid chattering at the switching boundary;

see Figure 4.3.

5.1.5 Design of $\mathcal{T}_{0,1}$

To make the switch back to κ_1 , we utilize Assumption 3.1.3. Recall from lines 3-4 of Algorithm 2 that the objective is to design $\mathcal{T}_{0,1}$ such that when $z \in \mathcal{T}_{0,1}$ and $q = 0$, the state component z_1 is far from z_1^* and the supervisor resets q to 1 and assigns u to $\kappa_1(h(z))$, where κ_1 is defined via (5.1b) and h_1 is defined in (5.2), so that κ_1 steers z_1 back to nearby z_1^* . Given $c_0 > 0$, let $\alpha > 0$ come from Definition 2.2.3, and let $M > 0$ come from Assumption 3.1.3. Then, using Assumption 3.1.3 with $u_1 = z_1^*$ and $w_1 = z_1$ yields (4.27) for all $z_1 \in \mathbb{R}^n$. Since L is \mathcal{C}^2 , strongly convex, and has a single minimizer z_1^* by Assumption 3.1.1 – which implies that L also has quadratic growth away from z_1^* as in Definition 2.2.3, then dividing both sides of (4.27) by M and substituting into (3.79) leads to (4.28), where $\alpha > 0$ comes from Definition 2.2.3.

Then, V_0 in (3.80) is lower bounded as follows: for each $z \in \mathbb{R}^{2n}$,

$$V_0(z) = \gamma(L(z_1) - L^*) + \frac{1}{2}|z_2|^2 \geq \gamma\left(\frac{\alpha}{M^2}\right)|\nabla L(z_1)|^2 + \frac{1}{2}|z_2|^2. \quad (5.14)$$

Using the right-hand side of (5.14) and the same $c_0 > 0$ as in Section 4.4.1, we define the set

$$\mathcal{T}_{0,1} := \left\{ z \in \mathbb{R}^{2n} : \gamma\left(\frac{\alpha}{M^2}\right)|\nabla L(z_1)|^2 + \frac{1}{2}|z_2|^2 \geq c_0 \right\}. \quad (5.15)$$

The set in (5.15) defines the (closed) complement of a sublevel set of the Lyapunov function V_0 in (3.80) with level equal to c_0 . The constant c_0 is also a part of the hysteresis mechanism, as shown in Figure 4.3 and as described below (4.30). Note that, as in Section 4.4.3, $\mathcal{T}_{0,1} \cap \mathcal{T}_{1,0} = \emptyset$.

5.1.6 Design of the Parameter λ

The heavy ball parameter $\lambda > 0$ should be made large enough to avoid oscillations near the minimizer, as stated in Sections 1.3.1, 1.3.3, and 5.1.1. The intuition on how to tune λ is the same as the intuition behind how to tune λ_0 for the the uniting algorithms in Chapter 4, which was described in detail in Section 4.3.2. See Examples 5.2.5, 5.2.6, and 5.2.7 where λ was tuned in such a way.

5.1.7 Well-posedness of the Hybrid Closed-Loop System

\mathcal{H}

When L satisfies Assumptions 3.1.1 and 3.1.3, the hybrid closed-loop system \mathcal{H} in (5.3), with \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ defined via (4.20), (5.12), and (5.15), satisfies the hybrid basic conditions, listed in Definition 2.1.1, as demonstrated in the following lemma.

Lemma 5.1.2. *(Well-posedness of \mathcal{H}) Let the function L satisfy Assumptions 3.1.1 and 3.1.3. Let the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ be defined via (4.20) (5.12), and (5.15), respectively. Let the functions d and β be defined in (3.3). Let κ_0 and κ_1 be defined via (5.1). Then, the hybrid closed-loop system \mathcal{H} in (5.3), with C and D defined via (4.14), satisfies the hybrid basic conditions.*

Proof. The objective function L is \mathcal{C}^2 and strongly convex by Assumption 3.1.1. Therefore, since ∇L is continuous, the following hold: the set \mathcal{U}_0 , defined via (4.20), is closed since it is a sublevel set of the continuous function V_0 ; due to the definition of a in (3.9), the set $\mathcal{T}_{1,0}$, defined via (5.12), is closed since it is a sublevel set of the continuous function V_1 ; the set $\mathcal{T}_{0,1}$, defined via (5.15), is closed since it is the closed complement of a set. Therefore, since the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ are closed, then the sets D_0 , D_1 , C_0 , and C_1 are closed. Since C and D are

both finite unions of finite and closed sets, then C and D are also closed.

Due to the definition of d and β in (3.3), and since by Assumption 3.1.1, L is \mathcal{C}^2 , then h_q in (5.2) and κ_q in (5.1) are continuous. In turn, the map $z \mapsto F_P(z, \kappa_q(h_q(z)))$ is also continuous since F_P in (3.1) is a \mathcal{C}^2 function of κ_q and h_q . Therefore, $x \mapsto F(x)$ is continuous. The map G satisfies (A3) by construction since it is continuous. \square

5.1.8 Existence of Solutions to \mathcal{H}

Under Assumptions 3.1.1 and 3.1.3, $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$ and $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$ and each maximal solution to \mathcal{H} in (5.3), with \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ defined via (4.20), (5.12), and (5.15), is complete and bounded, as stated in the following lemma.

Proposition 5.1.3. *(Existence of solutions to \mathcal{H}) Let the function L satisfy Assumptions 3.1.1 and 3.1.3. Let $M > 0$ come from 3.1.3. Let the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ be defined via (4.20), (5.12), and (5.15), respectively. Let κ_q be defined via (5.1). Let $\lambda > 0$ and $\gamma > 0$. Let the functions d and β be defined in (3.3). Then, $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$, $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$, and each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to \mathcal{H} in (5.3), with C and D defined via (4.14), is bounded and complete.*

Proof. Since Assumptions 3.1.8 and 3.1.3 hold, then \mathcal{H} satisfies the hybrid basic conditions by Lemma 5.1.2. With $\tilde{c}_0 > 0$ and $d_0 > 0$ defined via (5.5), since L is \mathcal{C}^2 and strongly convex by Assumption 3.1.8 – which implies that L also satisfies Definition 2.2.2 and has quadratic growth away from z_1^* as in Definition 2.2.3 – from the arguments below (5.6), every $z \in \mathcal{U}_0$ belongs to the c_0 -sublevel set of V_0 ; recall that \mathcal{U}_0 is defined in (4.20) and that V_0 is defined via (3.80). Additionally,

since by Definition 2.2.3 L has quadratic growth away from z_1^* and since ∇L is Lipschitz continuous by Assumption 3.1.3, then $\mathcal{T}_{0,1}$ in (5.15) defines the closed complement of a sublevel set of V_0 with level equal to c_0 . Therefore, due to the definitions of \mathcal{U}_0 in (4.20) and $\mathcal{T}_{0,1}$ in (5.15), $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$. Furthermore, since $\mathcal{T}_{1,0}$ is defined via (5.12), and since by the definitions of C_1 and D_1 in (4.14), C_1 is the closed complement of D_1 , then $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$.

Due to the definitions of C_0 , D_0 , C_1 , and D_1 in (4.14), \mathcal{U}_0 in (4.20), $\mathcal{T}_{1,0}$ in (5.12), and $\mathcal{T}_{0,1}$ in (5.15), then $C \setminus D$ is equal to $\text{int}(C)$. Hence, for each point $x \in C \setminus D$, the tangent cone to C at x is defined in (4.8). Therefore, $F(x) \cap T_C(x) \neq \emptyset$, satisfying (VC) of Proposition A.1.1 for each point $x \in C \setminus D$, and nontrivial solutions exist for every initial point in $(C_0 \cup C_1) \cup (D_0 \cup D_1)$, where $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$ and $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$. To prove that item (c) of Proposition A.1.1 does not hold, we need to show that $G(D) \subset C \cup D$. With D defined in (4.14), $G(D)$ is as shown in (4.31). Notice that $\mathcal{T}_{1,0} \times \{0\} \subset C_0$ and $\mathcal{T}_{0,1} \times \{1\} \subset C_1$. Therefore, $G(D) \subset C$; hence $G(D) \subset C \cup D$. Therefore, item (c) of Proposition A.1.1 does not hold. Then it remains to prove that item (b) does not happen.

We show that there is no finite time escape from C for \mathcal{H} as follows. First, since ∇L is Lipschitz continuous by Assumption 3.1.3 and since L is \mathcal{C}^2 and strongly convex by Assumption 3.1.1 – which implies that L also satisfies Definition 2.2.2 and has quadratic growth away from z_1^* as in Definition 2.2.3, then each maximal solution to $\dot{z} = F_P(z, \kappa_0(h_0(z)))$, defined via (5.4) – with κ_0 in (5.1a) and h_0 in (5.2) – is bounded, complete, and unique by Proposition 3.2.7. Next, since ∇L is Lipschitz continuous by Assumption 3.1.3 and since L is \mathcal{C}^2 and strongly convex by Assumption 3.1.1 – which implies that L also satisfies Definition 2.2.2 and has quadratic growth away from z_1^* as in Definition 2.2.3, then $\dot{z} = F_P(z, \kappa_1(h_1(z)))$ –

with κ_1 in (5.1b) and h_1 in (5.2) – is bounded, complete, and unique by Proposition 3.1.5. Finally, for the hybrid closed-loop system \mathcal{H} in (5.3), with C and D defined via (4.14) and κ_q defined in (5.1), since \mathcal{H}_q in (5.4) has no finite time escape from \mathbb{R}^{2n} , then this means $\dot{x} = F(x)$ has no finite time escape from C for \mathcal{H} , as q does not change in C . Therefore, there is no finite time escape from $C \cup D$, for solutions x to \mathcal{H} in (5.3), with C and D defined via (4.14) and κ_q defined in (5.1). Therefore, item (b) from Proposition A.1.1 does not hold. \square

5.1.9 Main Result

In this section, we present a result that establishes UGAS of the set \mathcal{A} in (4.9), and a hybrid convergence rate that is exponential both globally and locally, for the hybrid closed-loop algorithm \mathcal{H} in (5.3), with C and D defined in (4.14) and with the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ defined via (4.20), (5.12), and (5.15), respectively.

Theorem 5.1.4. *(Global asymptotic stability of \mathcal{A} and convergence rate for \mathcal{H})*

Let L satisfy Assumptions 3.1.1 and 3.1.3, and let $M > 0$ come from Assumption 3.1.3. Additionally, let $\lambda > 0$, $\gamma > 0$, $\varepsilon_{1,0} \in (0, \varepsilon_0)$, $c_{1,0} \in (0, c_0)$, $c_0 > 0$, $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ come from (5.5) and (5.9a), $d_{1,0} \in (0, d_0)$ come from (5.5) and (5.9b). Let the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ be defined via (4.20), (5.12), and (5.15), respectively. Let the constants d and β be defined in (3.3). Let κ_q be defined via (5.1). Then, the set \mathcal{A} , defined in (4.9), is globally asymptotically stable for \mathcal{H} in (5.3), with C and D defined via (4.14). Furthermore, each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ of the hybrid closed-loop algorithm \mathcal{H} starting from C_1 satisfies the following:

- 1) *The domain $\text{dom } x$ of the solution x is of the form $\cup_{j=0}^1(I^j \times \{j\})$, with I^0 of the form $[t_0, t_1]$ and with I^1 of the form $[t_1, \infty)$ for some $t_1 \geq 0$ defining the time of the first jump;*

2) For each² $t \in I^0$

$$L(z_1(t, 0)) - L^* \leq (L(z_1(0, 0)) - L^*) \exp(-at) \quad (5.16)$$

3) For each $t \in I^1$

$$L(z_1(t, 1)) - L^* \leq (L(z_1(t_1, 1)) - L^*) \exp(-2\mu t) \quad (5.17)$$

where $\mu > 0$ comes from Definition 2.2.1 and where $a > 0$ is defined in (3.9), with κ_c defined via (3.4).

Proof. The hybrid closed-loop algorithm \mathcal{H} in (5.3) – with \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ defined via (4.20), (5.12), and (5.15), respectively – satisfies the hybrid basic conditions by Lemma 5.1.2, satisfying the first assumption of Theorem A.1.3. Furthermore, $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$, $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$, and each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to \mathcal{H} is complete and bounded by Proposition 5.1.3. Since by Assumption 3.1.1, L is strongly convex, then L has a unique minimizer z_1^* . Hence, \mathcal{A} , defined via (4.9), is compact by construction, and $\mathcal{U} = \mathbb{R}^{2n} \times Q$ contains a nonzero open neighborhood of \mathcal{A} , satisfying the second assumption of Theorem A.1.3.

To prove attractivity of \mathcal{A} in (4.9), we proceed by contradiction. Suppose there exists a complete solution x to \mathcal{H} such that $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} \neq 0$. Since Proposition 5.1.3 guarantees completeness of maximal solutions, we have the following cases:

- a) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_1 \setminus D_1$ for all $(t, j) \in \text{dom } x$, $t + j \geq t' + j'$;
- b) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_0 \setminus (\mathcal{A} \cup D_0)$ for all $(t, j) \in$

²Note that at each $t \in I^0$, $q(t, 0) = 1$, and at each $t \in I^1$, $q(t, 1) = 0$.

$$\text{dom } x, t + j \geq t' + j';$$

- c) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in D$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$.

Case a) contradicts the fact that, by Theorem 3.1.7, the set $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_1 in (5.4). Such uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$, guaranteed by Theorem 3.1.7, implies there exist $\tilde{c}_1 \in (0, \tilde{c}_{1,0})$ and $d_1 \in (0, d_{1,0})$ such that the state z reaches $(\{z_1^*\} + \tilde{c}_1\mathbb{B}) \times (\{0\} + d_1\mathbb{B}) \subset \mathcal{T}_{1,0}$ as $t \rightarrow \infty$. In turn, due to the construction of C_1 and D_1 in (4.14), with $\mathcal{T}_{1,0}$ defined via (5.12), the solution x must reach D_1 at some $(t, j) \in \text{dom } x, t + j \geq t' + j'$. Therefore, case a) does not happen.

Case b) contradicts the fact that, by Proposition 3.2.8, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_0 in (5.4). In fact, $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} = 0$, and since $\mathcal{A} \subset C_0$, case b) does not happen.

Case c) contradicts the fact that, due to the construction of $\mathcal{T}_{1,0}$ in (5.12) and $\mathcal{T}_{0,1}$ in (5.15), we have $G(D) \cap D$ as defined in (4.32), where $G(D)$ is defined via (4.31) and D is defined in (4.14). Such an equality holds since $\mathcal{T}_{0,1} \cap \mathcal{T}_{1,0} = \emptyset$; see the end of Section 5.1.5. Therefore, case c) does not happen.

Therefore, cases a)-c) do not happen, and each maximal and complete solution $x = (z, q)$ to \mathcal{H} converges to \mathcal{A} in (4.9). Consequently, by the construction of C and D in (4.14), the definition of the sets \mathcal{U}_0 in (4.20), $\mathcal{T}_{1,0}$ in (5.12), and $\mathcal{T}_{0,1}$ in (5.15), the uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_0 established in Proposition 3.2.8, the uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_1 established in Theorem 3.1.7, and since each maximal solution to \mathcal{H} is complete by Proposition 5.1.3, the set \mathcal{A} in (4.9) is uniformly globally asymptotically stable for \mathcal{H} in (5.3), with C and D defined via (4.14) and with \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ defined via (4.20), (5.12), and (5.15).

To show that each maximal and complete solution x to \mathcal{H} jumps no more than twice, we proceed by contradiction. Without loss of generality, suppose there exists a maximal and complete solution that jumps three times. We have the following possible cases:

- i) The solution first jumps at a point in D_0 , then jumps at a point in D_1 , and then jumps at a point in D_0 ; or
- ii) The solution first jumps at a point in D_1 , then jumps at a point in D_0 , and then jumps at a point in D_1 .

Case i) does not hold since, once the jump in D_1 occurs, the solution x is in $(\mathcal{T}_{1,0} \times \{0\}) \subset C_0$. Due to the construction of $\mathcal{T}_{1,0}$ in (5.12) and $\mathcal{T}_{0,1}$ in (5.15) such that $\mathcal{T}_{1,0} \cap \mathcal{T}_{0,1} = \emptyset$, as described in the contradiction of case c) above, and due to the uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_0 in (5.4) by Proposition 3.2.8, the solution x will never return to D_0 . Therefore, case i) does not happen. Case ii) leads to a contradiction for the same reason, and in this case, once the first jump in D_1 occurs, no more jumps happen. Therefore, since cases i)-ii) do not happen, each maximal and complete solution x to \mathcal{H} in (5.3), with C and D defined via (4.14) and with \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ defined via (4.20), (5.12), and (5.15), has no more than two jumps.

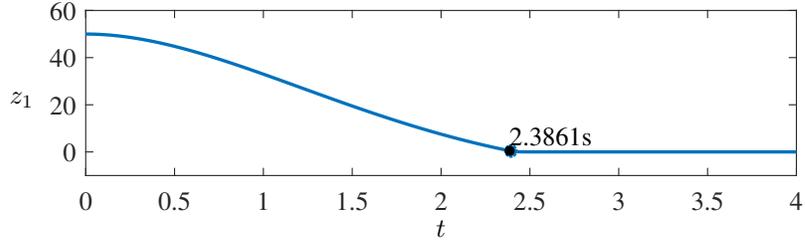
Finally, we prove the hybrid convergence rate of \mathcal{H} . By Proposition 3.2.3, since L is \mathcal{C}^2 and strongly convex by Assumption 3.1.1, then, given $\mu > 0$ from Definition 2.2.1, each maximal solution $t \mapsto z(t)$ to the closed-loop algorithm \mathcal{H}_0 in (5.4) satisfies (3.77) for all $t \in \text{dom } z (= \mathbb{R}_{\geq 0})$. By Proposition 3.1.6, L is \mathcal{C}^2 and strongly convex by Assumption 3.1.1, and ∇L is Lipschitz continuous with constant $M > 0$ by 3.1.1, then, given $\mu > 0$ from Definition 2.2.1 and $a > 0$ from (3.9), each maximal solution $t \mapsto z(t)$ to the closed-loop algorithm \mathcal{H}_1 in (5.4) satisfies (3.31) for all $t \in \text{dom } z (= \mathbb{R}_{\geq 0})$. Since maximal solutions

$(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to \mathcal{H} starting from C_1 are guaranteed to jump no more than once, as implied by the contradiction in cases i)-ii) above, then the domain of each maximal solution x to \mathcal{H} starting from C_1 is $\cup_{j=0}^1(I^j, j)$, with I^0 of the form $[t_0, t_1]$ and with I^1 of the form $[t_1, \infty)$. Therefore, given $\lambda > 0$, $\gamma > 0$, $c_{1,0} \in (0, c_0)$, $\varepsilon_{1,0} \in (0, \varepsilon_0)$, $\alpha > 0$ from Definition 2.2.3, and $M > 0$ from Assumption 3.1.3, due to the construction of \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ in (4.20), (5.12), and (5.15), with $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ and $d_{1,0} \in (0, d_0)$ defined via (5.5) and (5.9), and due to the individual convergence rates of \mathcal{H}_q , each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j))$ to the hybrid closed-loop algorithm \mathcal{H} in (5.3) that starts in C_1 satisfies (5.16) for each $t \in I^0$ at which $q(t, 0)$ is equal to 1, and satisfies (5.17) for each $t \in I^1$ at which $q(t, 1)$ is equal to 0. \square

5.1.10 Numerical Examples

In this section, we present multiple numerical examples to illustrate the hybrid closed-loop algorithm \mathcal{H} in Sections 5.1.2-5.1.5. Example 5.1.5 first illustrates the operation of the nominal hybrid closed-loop system \mathcal{H} , and then demonstrates the robustness of \mathcal{H} to different amounts of noise in measurements of ∇L . Example 5.1.6 compares solutions to the hybrid closed-loop algorithm in (5.22) and (4.14) – with \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ defined via (4.20), (5.12), and (5.15), respectively, – with solutions to \mathcal{H}_0 , \mathcal{H}_1 , and HAND-2 from [32], with parameters chosen such that HAND-2 and \mathcal{H} are compared on equal footing.

Example 5.1.5. *In this example, we simulate a solution to the nominal hybrid closed-loop system \mathcal{H} to illustrate how the uniting algorithm works. Then, we compare that same solution to solutions with different amounts of noise in measurements of ∇L . For both the nominal system and the perturbed system, the choice of objective function, parameter values, and initial conditions are as fol-*



σ	$\lim_{t+j \rightarrow \infty} \sup z_1(t, j) - z_1^* $	$\lim_{t+j \rightarrow \infty} \sup L(z_1(t, j)) - L^* $
0.01	5.480×10^{-5}	3.003×10^{-9}
0.1	1.855×10^{-4}	3.441×10^{-8}
0.5	1.119×10^{-2}	1.252×10^{-4}
1	3.406×10^{-2}	1.160×10^{-3}
5	2.069×10^{-2}	4.281×10^{-4}
10	1.434×10^{-2}	2.056×10^{-4}
15	4.693×10^{-2}	2.202×10^{-3}
20	2.494×10^{-1}	6.220×10^{-2}
25	5.100×10^{-2}	2.601×10^{-3}

Figure 5.1: Top: The evolution over time of z_1 , for the nominal hybrid closed-loop algorithm \mathcal{H} in Sections 5.1.2-5.1.5, for a function $L(z_1) := z_1^2$ with a single minimizer at $z_1^* = 0$. The time at which the solution settles to within 1% of z_1^* is marked with a dot and labeled in seconds. The jump is labeled with an asterisk. Bottom: Simulation results for perturbed solutions using zero mean Gaussian noise, with each simulation using a different value of the standard deviation σ . Results listed are for a large value of $t + j$.

lows. We use the objective function $L(z_1) := z_1^2$, which is strongly convex with $\mu = 2$, the gradient of which is Lipschitz continuous with $M = 2$, which has a single minimizer at $z_1^* = 0$. This choice of objective function is made so that we can easily tune λ , as described in Section 5.1.6. Since $\kappa_c = \frac{M}{\mu} = \frac{2}{2} = 1$ for such an objective function, then $d = \frac{1}{2}$ and $\beta = 0$, as calculated via (3.3). We arbitrarily chose the heavy ball parameter value $\gamma = \frac{2}{3}$ and we tuned λ to 40 by choosing a value arbitrarily larger than $2\sqrt{a_1}$, where a_1 comes from Section 5.1.6, and gradually increasing it until there is no overshoot in the hybrid algorithm. The parameter values for the uniting algorithm are $c_0 = 3000$, $c_{1,0} \approx 402.83$, $\varepsilon_0 = 20$, $\varepsilon_{1,0} = 15$, and $\alpha = 1$, which yield the values $\tilde{c}_0 = 20$, $\tilde{c}_{1,0} = 15$, $d_0 \approx 2733.3$, and $d_{1,0} \approx 234.08$, which are calculated via (5.5) and (5.9). These values are chosen for proper tuning of the algorithm, in order to get nice performance, and the value of $c_{1,0}$ is chosen to exploit the properties of Nesterov's method for a longer time, so that the nominal solution gets closer to the minimizer faster. Initial conditions for \mathcal{H} are $z_1(0,0) = 50$, $z_2(0,0) = 0$, and $q(0,0) = 1$. The plot on the top in Figure 5.1 shows the solution to the nominal hybrid closed-loop algorithm \mathcal{H} , namely, the value of z_1 over time, with the time it takes for the solution to settle to within 1% of z_1^* marked with a black dot and labeled in seconds³. The jump at which the switch from \mathcal{H}_1 to \mathcal{H}_0 occurs is labeled with an asterisk. The solution converges quickly, without oscillations near the minimizer.

To show that the uniform global asymptotic stability of \mathcal{A} , established in Theorem 5.1.4, is robust to small perturbations, due to the hybrid closed-loop system \mathcal{H} satisfying the hybrid basic conditions by Lemma 5.1.2. we simulate the hybrid algorithm, using the objective function, parameter values, and initial conditions listed in the first paragraph of this example, with zero-mean Gaussian noise added to measurements of the gradient. Separate simulations were run for each of the

³Code at github.com/HybridSystemsLab/UnitingRobustnessSC.

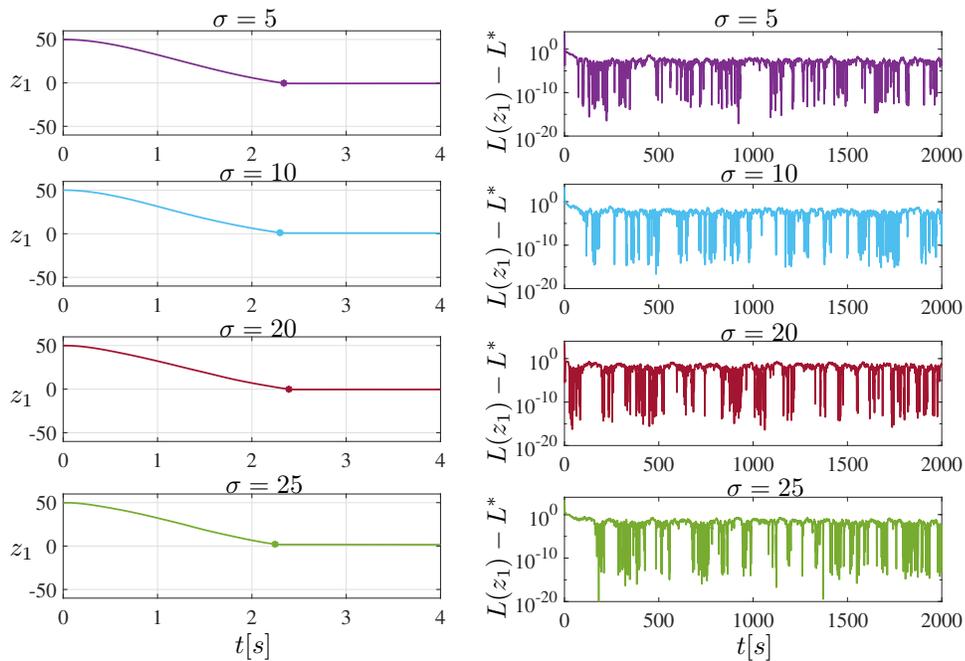


Figure 5.2: Simulation results for hybrid closed-loop algorithm \mathcal{H} in Sections 5.1.2-5.1.5, for a function $L(z_1) := z_1^2$ with a single minimizer at $z_1^* = 0$, with zero-mean Gaussian noise added to measurements of the gradient. Each subplot is labeled with the standard deviation used. Left subplots: the value of z_1 over time for each perturbed solution, with the jump in each solution labeled by an asterisk. Right subplots: the corresponding value of L over time for each perturbed solution.

following standard deviations: $\sigma \in \{0.01, 0.1, 0.5, 1, 5, 10, 15, 20, 25\}$. Figure 5.2 shows some of these perturbed solutions, with each subplot labeled with the corresponding standard deviation used. The subplots on the left side of Figure 5.2 show the value of z_1 over time for different standard deviations, and the subplots on the right side of Figure 5.2 show the corresponding value of L over time for such standard deviations. Note that, while all perturbed solutions shown in Figure 5.2 get close to the minimizer quickly, such perturbed solutions do not get as close to the minimizer as the solution to the nominal algorithm does; see the plot on the top in Figure 5.1. Also note that, in general, as the standard deviation gets larger, the corresponding perturbed solution stays slightly farther away from the minimizer. The results for all standard deviations are listed in the table in Figure 5.1, showing the neighborhood of z_1^* that each solution settles to, for a large value of $t + j$, along with the corresponding value of L .

Example 5.1.6. In this example, to show the effectiveness of the uniting algorithm, we compare the hybrid closed-loop algorithm \mathcal{H} , defined via (5.3) and (4.14) – with \mathcal{U} defined in (4.20), $\mathcal{T}_{1,0}$ defined via (5.12), and $\mathcal{T}_{0,1}$ defined in (5.15) – with the individual closed-loop optimization algorithms \mathcal{H}_0 and \mathcal{H}_1 and with the HAND-2 algorithm from [32] which, in [32], is designed and analyzed for strongly convex functions L satisfying Assumptions 3.1.1 and 3.1.3. 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)

$$F(z, \tau) := \begin{bmatrix} \frac{2}{\tau}(z_2 - z_1) \\ -2c_1\tau\nabla L(z_1) \\ 1 \end{bmatrix} \quad (z, \tau) \in C \quad (5.18a)$$

$$G(z, \tau) := \begin{bmatrix} z_1 \\ z_1 \\ T_{\min} \end{bmatrix} \quad (z, \tau) \in D \quad (5.18b)$$

where $c_1 > 0$ and the flow and jump sets are $C := \{(z, \tau) \in \mathbb{R}^{2n+1} : \tau \in [T_{\min}, T_{\max}]\}$, and $D := \{(z, \tau) \in \mathbb{R}^{2n+1} : \tau \geq T_{\max}\}$. It is shown in [32] that, letting $0 < T_{\min} < T_{\max} < \infty$ and $\frac{1}{c_1\mu} < T_{\max}^2 - T_{\min}^2$, each maximal solution $(t, j) \mapsto (z(t, j), \tau(t, j))$ to the HAND-2 algorithm satisfies

$$L(z_1(t, j)) - L^* \leq k_a |\tilde{z}_1(0, 0)|^2 \exp(-\tilde{k}_b \tilde{\alpha}(t + j)) \quad (5.19)$$

for all $(t, j) \in \text{dom}(z, \tau)$ such that $z_1(0, 0) = z_2(0, 0)$ and $\tau(0, 0) = T_{\min}$, where $k_a := 0.5k_1M$, $M > 0$, $k_1 := \frac{(c_1\mu)^{-1} + T_{\min}^2}{\Delta T^2}$, $\Delta T := T_{\max} - T_{\min}$, $\tilde{k}_b := 1 - k_0$, $k_0 := \frac{(c_1\mu)^{-1} + T_{\min}^2}{T_{\max}^2}$, $j \geq \tilde{\alpha}(t + j) := \frac{\max\{t + j - \Delta T, 0\}}{\Delta T + 1}$, and $|\tilde{z}_1(0, 0)| := |z_1(0, 0) - z_1^*|$.

To compare \mathcal{H}_0 , \mathcal{H}_1 , \mathcal{H} , and HAND-2 in simulation, we use the objective function $L(z_1) := z_1^2$, which is strongly convex with $\mu = 2$, and the gradient of which is Lipschitz continuous with $M = 2$, leading to $\kappa = \frac{M}{\mu} = 1$. Therefore, $d = \frac{1}{2}$ and $\beta = 0$, calculated via (3.3). Such an L has a single minimizer at $z_1^* = 0$. This choice of objective function is made so that we can easily tune λ , as described in Section 5.1.6. Namely, we tuned λ to 40 by choosing a value arbitrarily larger than $2\sqrt{a_1}$, where a_1 comes from Section 5.1.6, and gradually increasing it until there is no overshoot in the hybrid algorithm. We arbitrarily chose the heavy ball parameter value $\gamma = \frac{2}{3}$.

The HAND-2 parameters $c_1 = \frac{1}{8}$ and $T_{\min} = 3$ are chosen such that the resulting gain coefficients for z_1 and z_2 are the same for both \mathcal{H} and HAND-2, so that these algorithms are compared on equal footing. The remaining HAND-2 parameter, $T_{\max} = 5.1$, is chosen to satisfy $\frac{1}{c_1\mu} < T_{\max}^2 - T_{\min}^2$ and to get nice

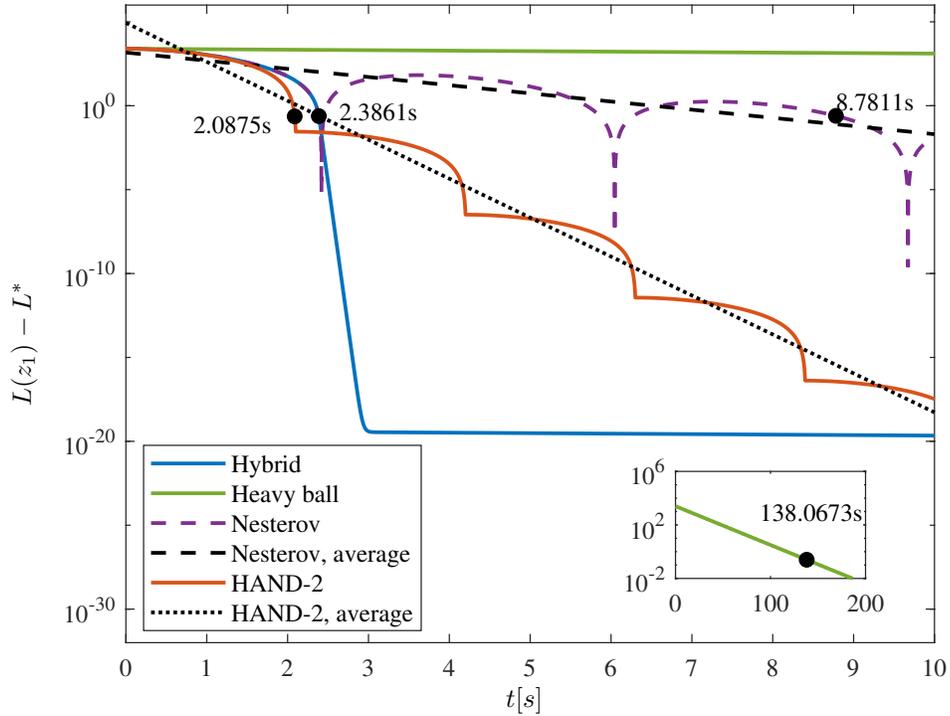


Figure 5.3: A comparison of the evolution of L over time for \mathcal{H}_0 , \mathcal{H}_1 (both in (5.4)), HAND-2, and \mathcal{H} in (5.3) – with \mathcal{U} in (4.20), $\mathcal{T}_{1,0}$ in (5.12), and $\mathcal{T}_{0,1}$ in (5.15) – for a function $L(z_1) := z_1^2$, with a single minimizer at $z_1^* = 0$. Nesterov’s method, shown in purple, settles to within 1% of z_1^* in about 8.8 seconds. The heavy ball algorithm, shown in green, settles to within 1% of z_1^* in about 138.1 seconds. HAND-2, shown in orange, settles to within 1% of z^* in about 2.1 seconds. The hybrid closed-loop system \mathcal{H} , shown in blue, settles to within 1% of z_1^* in about 2.4 seconds.

performance. The parameter values for the uniting algorithm are $c_0 = 3000$, $c_{1,0} \approx 402.83$, $\varepsilon_0 = 20$, $\varepsilon_{1,0} = 15$, and $\alpha = 1$, which yield the values $\tilde{c}_0 = 20$, $\tilde{c}_{1,0} = 15$, $d_0 \approx 2733.3$, and $d_{1,0} \approx 234.08$, which are calculated via (5.5) and (5.9). These values are chosen for proper tuning of the algorithm, in order to get nice performance, and the value of $c_{1,0}$ is chosen to exploit the properties of Nesterov’s method for a longer time, so that the nominal solution gets closer to the minimizer faster. The initial conditions for \mathcal{H}_0 and \mathcal{H}_1 are $z_1(0,0) = 50$ and $z_2(0,0) = 0$. Initial conditions for \mathcal{H} are $z_1(0,0) = 50$, $z_2(0,0) = 0$, and $q = 1$. Initial conditions for HAND-2 are $z_1(0,0) = z_2(0,0) = 50$ and $\tau(0,0) = T_{\min}$.

Figure 5.3 compares the evolution of L over time^A, for \mathcal{H}_0 , \mathcal{H}_1 , \mathcal{H} , and HAND-2. Table 5.1 shows the time that each algorithm takes to settle within 1% of z_1^* , and the percent improvement of \mathcal{H} over \mathcal{H}_0 and \mathcal{H}_1 , which is calculated via the formula in (4.13). Note that HAND-2 settles within 1% of z_1^* slightly faster than \mathcal{H} does, but HAND-2 and \mathcal{H} have similarly good performance.

Algorithm	Time to converge (s)	% improvement of \mathcal{H}
\mathcal{H}	2.386	–
\mathcal{H}_0	138.067	98.3
\mathcal{H}_1	8.781	72.8
HAND-2	2.088	–

Table 5.1: Average times for which \mathcal{H} , \mathcal{H}_0 , and \mathcal{H}_1 settle to within 1% of z_1^* , and the average percent improvement of \mathcal{H} over each algorithm. Percent improvement is calculated via (4.13). The objective function used for this table is $L(z_1) := z_1^2$.

5.2 Nonstrongly Convex L

The uniting algorithm proposed in this section imposes Assumptions 3.1.8, 3.1.3, and 3.2.4 on the objective function L . Namely, L is \mathcal{C}^1 , nonstrongly convex,

^ACode at github.com/HybridSystemsLab/UnitingSC.

and has a unique minimizer by Assumption 3.1.8, has a Lipschitz continuous gradient by Assumption 3.1.3, and has quadratic growth away from its minimizer z_1^* by assumption 3.2.4.

5.2.1 Problem Statement

In Section 5.1.1, we discussed how we desired to design a uniting algorithm which attains an exponential rate of convergence both globally (via Nesterov’s algorithm) and locally (via the heavy ball algorithm, with large $\lambda > 0$), for strongly convex L . In this section, we desire to relax the algorithm’s requirement to allow L to be nonstrongly convex. In Section 1.3 we discussed how Nesterov’s algorithm guarantees a rate of $\frac{1}{(t+2)^2}$ for nonstrongly convex L . We also discussed how the heavy ball algorithm guarantees a rate of $\frac{1}{t}$ for nonstrongly convex L , although it was demonstrated in [25] that the heavy ball algorithm converges exponentially for nonstrongly convex L when such an objective function also has the property of quadratic growth away from its minimizer. We desire to attain the rate $\frac{1}{(t+2)^2}$ globally and an exponential rate locally, while avoiding oscillations via the heavy ball algorithm with large λ . We state the problem to solve as follows:

Problem 5.2.1. *Given a scalar, real-valued, continuously differentiable, and non-strongly convex objective function L with a unique minimizer, design an optimization algorithm that, without knowing the function L or the location of its minimizer, has the minimizer uniformly globally asymptotically stable, with a convergence rate of $\frac{1}{(t+2)^2}$ globally and an exponential convergence rate locally, and with robustness to arbitrarily small noise in measurements of ∇L .*

5.2.2 Modeling

In this section, we present an algorithm that solves Problem 5.2.1. As described in Section 3.2, defining z_1 as ξ and z_2 as $\dot{\xi}$, we interpret the heavy ball ODE in (1.1) and the Nesterov ODE in (1.5) as a closed-loop system consisting of the plant in (3.1). With this model, the optimization algorithms that we consider assign u to a function of the state that involves the cost function, and such a function of the state may be time dependent. The control algorithm leading to (1.1) assigns u to κ in (3.71), where $\gamma > 0$ and $\lambda > 0$, and the control algorithm leading to (1.5) assigns u to κ in (3.34), where $\zeta > 0$, $M > 0$ is the Lipschitz constant for ∇L , and where \bar{d} and $\bar{\beta}$ are defined, for all $t \geq 0$, in (3.35).

The proposed logic-based algorithm “unites” the two optimization algorithms modeled by κ_q , where the logic variable $q \in Q := \{0, 1\}$ indicates which algorithm is currently being used. The local algorithm κ_0 is defined in (5.1a) and the global algorithm is defined as

$$\kappa_1(h_1(z, t), t) = -2\bar{d}(t)z_2 - \frac{\zeta^2}{M}\nabla L(z_1 + \bar{\beta}(t)z_2) \quad (5.20)$$

where the algorithm defined by κ_1 plays the role of the global algorithm in uniting control (see, e.g., [22, Chapter 4]), while the algorithm defined by κ_0 plays the role of the local algorithm. The outputs h_0 corresponding to the output for the heavy ball algorithm and h_1 corresponding to the output for Nesterov’s algorithm are defined as

$$h_0(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1) \end{bmatrix}, h_1(z, t) := \begin{bmatrix} z_2 \\ \nabla L(z_1 + \bar{\beta}(t)z_2) \end{bmatrix}. \quad (5.21)$$

Namely, the algorithm exploits measurements of ∇L , which in practice are typi-

cally approximated using measurements of L . The parameters $\lambda > 0$ and $\gamma > 0$ should be designed to achieve convergence without oscillations nearby the minimizer.

Since the ODE in (1.5) is time varying, and since solutions to hybrid systems are parameterized by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, we employ the state τ to capture ordinary time as a state variable, in this way, leading to a time-invariant hybrid system. To encapsulate the plant, static state-feedback laws, and the time-varying nature of the ODE in (1.5), we define a hybrid closed-loop system \mathcal{H} with state $x := (z, q, \tau) \in \mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$ as

$$\left. \begin{array}{l} \dot{z} = \begin{bmatrix} z_2 \\ \kappa_q(h_q(z, \tau), \tau) \end{bmatrix} \\ \dot{q} = 0 \\ \dot{\tau} = q \end{array} \right\} =: F(x) \quad x \in C := C_0 \cup C_1 \quad (5.22a)$$

$$\left. \begin{array}{l} z^+ = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \\ q^+ = 1 - q \\ \tau^+ = 0 \end{array} \right\} =: G(x) \quad x \in D := D_0 \cup D_1 \quad (5.22b)$$

The sets C_0 , C_1 , D_0 , and D_1 are defined as

$$C_0 := \mathcal{U}_0 \times \{0\} \times \{0\}, \quad C_1 := \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}} \times \{1\} \times \mathbb{R}_{\geq 0} \quad (5.23a)$$

$$D_0 := \mathcal{T}_{0,1} \times \{0\} \times \{0\}, \quad D_1 := \mathcal{T}_{1,0} \times \{1\} \times \mathbb{R}_{\geq 0}. \quad (5.23b)$$

The sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ are precisely defined in Sections 5.2.3-5.2.5, but the idea behind their construction is as follows. The switch between κ_0 and κ_1 is

governed by a *supervisory algorithm* implementing switching logic; see Figure 5.4. The supervisor selects between these two optimization algorithms, based on the output of the plant and the optimization algorithm currently applied. When $z \in \mathcal{U}_0$, $q = 0$, and $\tau = 0$ (i.e., $x \in C_0$), due to the design of \mathcal{U}_0 in Section 5.2.3, then the state z is near the minimizer, which is denoted z_1^* , and the supervisor allows flows of (5.22) using κ_0 and $\dot{\tau} = q = 0$ to avoid oscillations. Conversely, when $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}$ and $q = 1$ (i.e., $x \in C_1$), due to the design of $\mathcal{T}_{1,0}$ in Section 5.2.4, then the state z is far from the minimizer and the supervisor allows flows of (5.22) using κ_1 and $\dot{\tau} = q = 1$ to converge quickly to the neighborhood of the minimizer. When $z \in \mathcal{T}_{1,0}$ and $q = 1$ (i.e., $x \in D_1$), then this indicates that the state z is near the minimizer, and the supervisor assigns u to κ_0 , resets q to 0, and resets τ to 0. Conversely, when $z \in \mathcal{T}_{0,1}$, $q = 0$, and $\tau = 0$ (i.e., $x \in D_0$), due to the design of $\mathcal{T}_{0,1}$ in Section 5.2.5, then this indicates that the state z is far from the minimizer and the supervisor assigns u to κ_1 and resets q to 1. The complete algorithm, defined in (5.22)-(5.23), is summarized in Algorithm 3.

Algorithm 3 Uniting algorithm

- 1: Set $q(0,0)$ to 0, $\tau(0,0)$ to 0, and set $z(0,0)$ as an initial condition with an arbitrary value.
 - 2: **while** true **do**
 - 3: **if** $z \in \mathcal{T}_{0,1}$, $q = 0$, and $\tau = 0$ **then**
 - 4: Reset q to 1.
 - 5: **else if** $z \in \mathcal{T}_{1,0}$ and $q = 1$ **then**
 - 6: Reset q to 0 and τ to 0.
 - 7: **else if** $z \in \mathcal{U}_0$, $q = 0$, and $\tau = 0$ **then**
 - 8: Assign u to $\kappa_0(h_0(z))$ and update z , q , and τ according to (5.22a).
 - 9: **else if** $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}$ and $q = 1$ **then**
 - 10: Assign u to $\kappa_1(h_1(z, \tau), \tau)$ and update z , q , and τ according to (5.22a).
 - 11: **end if**
 - 12: **end while**
-

The reason that the state τ in (5.22) changes at the rate q during flows and is reset to 0 at jumps is that when the state x is in C_1 , then $\dot{\tau} = q = 1$, which

implies that τ behaves as ordinary time, so it is used to represent time in the time-varying algorithm κ_1 . On the other hand, when the state x is in C_0 , then $\dot{\tau} = q = 0$ causes the state τ to stay at zero, which is an appropriate value for τ as it is not required by the time-invariant algorithm κ_0 . Such an evolution ensures that the set to asymptotically stabilize is compact.

Figure 5.4 shows the feedback diagram of this hybrid closed-loop system \mathcal{H} . We denote the closed-loop system resulting from κ_0 as \mathcal{H}_0 , which is given by

$$\dot{z} = \begin{bmatrix} z_2 \\ \kappa_0(h_0(z)) \end{bmatrix} \quad z \in \mathbb{R}^{2n} \quad (5.24)$$

and we denote the closed-loop system resulting from κ_1 as \mathcal{H}_1 , which is given by

$$\dot{z} = \begin{bmatrix} z_2 \\ \kappa_1(h_1(z, \tau), \tau) \end{bmatrix}, \quad \dot{\tau} = 1 \quad (z, \tau) \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}. \quad (5.25)$$

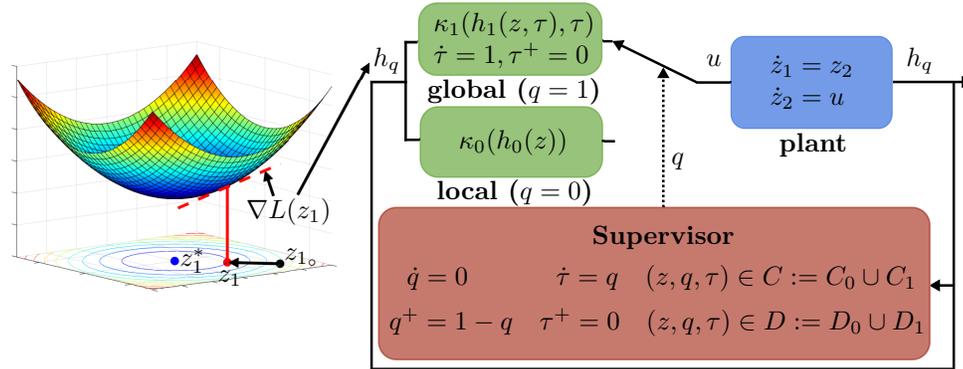


Figure 5.4: Feedback diagram of the hybrid closed-loop system \mathcal{H} (on the right), in (5.22), uniting global and local optimization algorithms. An example optimization problem to solve is shown on the left and, for this example optimization problem, measurements of the gradient are used for the input of κ_q .

5.2.3 Design of the Set \mathcal{U}_0

In order for the supervisor to determine when the state component z_1 is close to the minimizer of L , denoted z_1^* , without knowledge of z_1^* or $L^* := L(z_1)$, we will use Assumptions 3.1.8 and 3.2.4, and the suboptimality condition from Lemma 4.4.1.

Recall from lines 7-8 of Algorithm 3 that the objective is to design \mathcal{U}_0 such that when $z \in \mathcal{U}_0$, $q = 0$, and $\tau = 0$, the state component z_1 is near z_1^* and the uniting algorithm allows flows of (5.22) with κ_0 and $q = 0$. For such a design, we use Assumptions 3.1.8 and 3.2.4 and the Lyapunov function V_0 in (3.80), defined for each $z \in \mathbb{R}^{2n}$, where $\gamma > 0$. Given $\varepsilon_0 > 0$, $c_0 > 0$, and $\gamma > 0$ from κ_0 in (5.1a), let $\alpha > 0$ come from Assumption 3.2.4 such that $\tilde{c}_0 > 0$ and $d_0 > 0$ are defined in (5.5). Then, V_0 in (3.80) can be upper bounded, using Assumption 3.1.8 as done to arrive to (4.15), as follows: for each $z \in \mathbb{R}^{2n}$, V_0 in (3.80) is upper bounded as in (5.6). Then, due to L being \mathcal{C}^1 , nonstrongly convex, and having a single minimizer z_1^* by Assumption 3.1.8, and due to L having quadratic growth away from z_1^* by Assumption 3.2.4, when $|\nabla L(z_1)| \leq \tilde{c}_0$, the suboptimality condition in Lemma 4.4.1 implies $|z_1 - z_1^*| \leq \frac{\tilde{c}_0}{\alpha}$, from where we get the bound on V_0 shown in (5.7), where V_0 is defined via (3.80). Then, by defining the set \mathcal{U}_0 as in (4.20), every $z \in \mathcal{U}_0$ belongs to the c_0 -sublevel set of V_0 , where V_0 is defined in (3.80). In fact, using the conditions in (5.5) and (5.7), we have that for each $z \in \mathcal{U}_0$, the bound in (5.8) is satisfied. Since κ_0 in (5.1a) is such that the set $\{z_1^*\} \times \{0\}$ is globally asymptotically stable for the closed-loop system resulting from controlling (3.1) by κ_0 in (5.1a), as was shown in the Proposition 3.2.8, the set \mathcal{U}_0 in (4.20) is contained in the basin of attraction induced by κ_0 .

5.2.4 Design of the Set $\mathcal{T}_{1,0}$

Recall from lines 5-6 of Algorithm 3 that the objective is to design $\mathcal{T}_{1,0}$ such that when $z \in \mathcal{T}_{1,0}$ and $q = 1$, the state component z_1 is near z_1^* and the supervisor resets q to 0, resets τ to 0, and assigns u to $\kappa_0(h_0(z))$, where κ_0 is defined via (5.1a). For such a design, we use Assumptions 3.1.8 and 3.2.4 and the Lyapunov function in (3.38), defined for each $z \in \mathbb{R}^{2n}$ and each $\tau \geq 0$, where $\zeta > 0$, $M > 0$ is the Lipschitz constant of ∇L , and the function \bar{a} is defined in (3.39). Given $c_{1,0} \in (0, c_0)$ and $\varepsilon_{1,0} \in (0, \varepsilon_0)$, where $c_0 > 0$ and $\varepsilon_0 > 0$ come from Section 5.2.3, let \tilde{c}_0 and d_0 be given in (5.5), and let $\alpha > 0$ come from Assumption 3.2.4 such that

$$\tilde{c}_{1,0} := \varepsilon_{1,0}\alpha \in (0, \tilde{c}_0) \quad (5.26a)$$

$$d_{1,0} := c_{1,0} - \left(\frac{\tilde{c}_{1,0}}{\alpha}\right)^2 - \frac{\zeta^2}{M} \left(\frac{\tilde{c}_{1,0}^2}{\alpha}\right) \in (0, d_0) \quad (5.26b)$$

where $\zeta > 0$ comes from (1.5). Note that \bar{a} , defined via (3.39), which is in V_1 , equals 1 when $\tau = 0$ and monotonically decreases toward zero (but being always positive) as τ tends to ∞ . Namely, \bar{a} is upper bounded by 1. Then, with V_1 given in (3.38) and using Assumption 3.1.8 with $u_1 = z_1^*$ and $w_1 = z_1$,

$$V_1(z, \tau) \leq |z_1 - z_1^*|^2 + |z_2|^2 + \frac{\zeta^2}{M} |\nabla L(z_1)| |z_1 - z_1^*|. \quad (5.27)$$

Then, due to L being \mathcal{C}^1 , nonstrongly convex, and having a single minimizer z_1^* by Assumption 3.1.8, and due to L having quadratic growth away from z_1^* by Assumption 3.2.4, when $|\nabla L(z_1)| \leq \tilde{c}_{1,0}$, the suboptimality condition in Lemma

4.4.1 implies $|z_1 - z_1^*| \leq \frac{\tilde{c}_{1,0}}{\alpha}$, from where we get

$$V_1(z, \tau) \leq \left(\frac{\tilde{c}_{1,0}}{\alpha}\right)^2 + |z_2|^2 + \frac{\zeta^2}{M} \left(\frac{\tilde{c}_{1,0}^2}{\alpha}\right). \quad (5.28)$$

Then, by defining $\mathcal{T}_{1,0}$ as in (5.12) which, by construction, is contained in the interior of \mathcal{U}_0 defined in (4.20), every $z \in \mathcal{T}_{1,0}$ belongs to the $c_{1,0}$ -sublevel set of V_1 . In fact, using the conditions in (5.26) and (5.28), we have for each $z \in \mathcal{T}_{1,0}$,

$$V_1(z, \tau) \leq \left(\frac{\tilde{c}_{1,0}}{\alpha}\right)^2 + |z_2|^2 + \frac{\zeta^2}{M} \left(\frac{\tilde{c}_{1,0}^2}{\alpha}\right) \leq c_{1,0}. \quad (5.29)$$

As was described below (4.26), the constants \tilde{c}_0 , $\tilde{c}_{1,0}$, d_0 , and $d_{1,0}$ in (5.5) and (5.9) comprise the hysteresis necessary to avoid chattering at the switching boundary; see Figure 4.3.

5.2.5 Design of the Set $\mathcal{T}_{0,1}$

Recall from lines 3-4 of Algorithm 3 that the objective is to design $\mathcal{T}_{0,1}$ such that when $z \in \mathcal{T}_{0,1}$, $q = 0$, and $\tau = 0$, the state component z_1 is far from z_1^* and the supervisor resets q to 1 and assigns u to $\kappa_1(h_1(z, \tau), \tau)$ so that κ_1 steers z_1 back to nearby z_1^* . Given $c_0 > 0$, let $\alpha > 0$ come from Assumption 3.2.4, and let $M > 0$ come from Assumption 3.1.3. Then, using Assumption 3.1.3 with $u_1 = z_1^*$ and $w_1 = z_1$ yields (4.27) for all $z_1 \in \mathbb{R}^n$. Since L has quadratic growth away from z_1^* by Assumption 3.2.4, then dividing both sides of (4.27) by M and substituting into (3.79) leads to (4.28), where $\alpha > 0$ comes from Assumption 3.2.4. Then, V_0 in (3.80) satisfies the bound in (5.14) for each $z \in \mathbb{R}^{2n}$. Using the right-hand side of (5.14) and the same $c_0 > 0$ as in Section 5.2.3, the set $\mathcal{T}_{0,1}$ is defined as in (5.15). The set in (5.15) defines the (closed) complement of a sublevel set of the Lyapunov function V_0 in (3.80) with level equal to c_0 . The constant c_0 is also a

part of the hysteresis mechanism, as shown in Figure 4.3. When $z \in \mathcal{U}_0$ – where \mathcal{U}_0 is defined in (4.20), $q = 0$, and $\tau = 0$, then the supervisor does not need to switch to κ_1 , as the state component z is close enough to the minimizer to keep using κ_0 in (5.1a). But if $z \in \mathcal{T}_{0,1}$ while $q = 0$ and $\tau = 0$, then z is far enough from the minimizer, and the supervisor then switches to κ_1 in (5.20). Note that $\mathcal{T}_{0,1} \cap \mathcal{T}_{1,0} = \emptyset$.

5.2.6 Well-posedness of the Hybrid Closed-Loop System

\mathcal{H}

When L satisfies Assumptions 3.1.8, 3.2.4, and 3.1.3, the hybrid closed-loop system \mathcal{H} in (5.22), with the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ defined via (4.20) (5.12), and (5.15), respectively, satisfies the hybrid basic conditions, listed in Definition 2.1.1, as demonstrated in the following lemma. A hybrid closed-loop system \mathcal{H} that satisfies the hybrid basic conditions is said to be well-posed in the sense that the limit of a graphically convergent sequence of solutions to \mathcal{H} having a mild boundedness property is also a solution to \mathcal{H} [21].

Lemma 5.2.2. (*Well-posedness of \mathcal{H}*) *Let the function L satisfy Assumptions 3.1.8, 3.2.4, and 3.1.3. Let the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ be defined via (4.20), (5.12), and (5.15), respectively. Let the functions \bar{d} and $\bar{\beta}$ be defined as in (3.35). Let κ_0 and κ_1 be defined via (5.1a) and (5.20), respectively. Then, the hybrid closed-loop system \mathcal{H} in (5.22) satisfies the hybrid basic conditions.*

Proof. The objective function L is \mathcal{C}^1 , nonstrongly convex, and has a single minimizer by Assumption 3.1.8. Therefore, since ∇L is continuous, the following hold: the set \mathcal{U}_0 , defined via (4.20), is closed since it is a sublevel set of the continuous function V_0 ; due to \bar{a} in (3.39) being continuous, the set $\mathcal{T}_{1,0}$, defined

via (5.12), is closed since it is a sublevel set of the continuous function V_1 ; the set $\mathcal{T}_{0,1}$, defined via (5.15), is closed since it is the closed complement of a set. Therefore, since the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ are closed, then the sets D_0 , D_1 , C_0 , and C_1 are closed. Since C and D are both finite unions of finite and closed sets, then C and D are also closed.

Since \bar{d} and $\bar{\beta}$, defined via (3.35), are continuous, and since by Assumption 3.1.8, L is \mathcal{C}^1 , then h_q in (5.21), κ_0 in (5.1a), and κ_1 in (5.20) are continuous. In turn, the map $z \mapsto F_P(z, \kappa_q(h_q(z, \tau), \tau))$ is also continuous since F_P in (3.1) is a \mathcal{C}^1 function of κ_q and h_q . Therefore, $x \mapsto F(x)$ is continuous. The map G satisfies (A3) by construction since it is continuous. \square

5.2.7 Existence of Solutions to \mathcal{H}

Under Assumptions 3.1.8, 3.2.4, and 3.1.3, each maximal solution to \mathcal{H} is complete and bounded, as stated in the following lemma. Such a property is useful since it guarantees that nontrivial solutions to \mathcal{H} exist from each initial point in $C \cup D$, and that such solutions do not escape $C \cup D$. When every maximal solution is complete, then uniform global pre-asymptotic stability⁵ of the set \mathcal{A} becomes uniform global asymptotic stability. The following lemma also states that $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$ and $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$. Such a property ensures that nontrivial solutions to \mathcal{H} , which exist from each initial point in $C \cup D$, also exist from any initial point in $\mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$.

Proposition 5.2.3. *(Existence of solutions to \mathcal{H}) Let the function L satisfy Assumptions 3.1.8, 3.2.4, and 3.1.3. Let the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ be defined via (4.20), (5.12), and (5.15), respectively. Let the functions \bar{d} and $\bar{\beta}$ be defined as in (3.35). Let κ_0 and κ_1 be defined via (5.1a) and (5.20), respectively.*

⁵Uniform global pre-asymptotic stability indicates the possibility of a maximal solution that is not complete, even though it may be bounded.

Then, $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$, $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$, and each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j), \tau(t, j))$ to \mathcal{H} in (5.22) is bounded and complete.

Proof. Since Assumptions 3.1.8, 3.2.4, and 3.1.3 hold, then \mathcal{H} satisfies the hybrid basic conditions by Lemma 5.2.2. With $\tilde{c}_0 > 0$ and $d_0 > 0$ defined via (5.5), since L is \mathcal{C}^1 , nonstrongly convex, has a single minimizer by Assumption 3.1.8, and has quadratic growth away from z_1^* by Assumption 3.2.4, from the arguments below (3.84), every $z \in \mathcal{U}_0$ belongs to the c_0 -sublevel set of V_0 ; recall that \mathcal{U}_0 is defined in (4.20) and that V_0 is defined via (3.80). Additionally, since by Assumption 3.2.4 L has quadratic growth away from z_1^* and since ∇L is Lipschitz continuous by Assumption 3.1.3, then $\mathcal{T}_{0,1}$ in (5.15), defines the closed complement of a sublevel set of V_0 with level equal to c_0 . Therefore, due to the definitions of \mathcal{U}_0 in (4.20) and $\mathcal{T}_{0,1}$ in (5.15), $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$. Furthermore, since $\mathcal{T}_{1,0}$ is defined via (5.12), and since by the definitions of C_1 and D_1 in (5.23), C_1 is the closed complement of D_1 , then $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$.

Due to the definitions of C_0 , D_0 , C_1 , and D_1 in (5.23), \mathcal{U}_0 in (4.20), $\mathcal{T}_{1,0}$ in (5.12), and $\mathcal{T}_{0,1}$ in (5.15), then $C \setminus D$ is equal to $\text{int}(C)$. Hence, for each point $x \in C \setminus D$, the tangent cone to C at x is

$$T_C(x) := \begin{cases} \mathbb{R}^{2n} \times \{0\} \times \{0\} & \text{if } x \in C_0 \setminus D_0, \\ \mathbb{R}^{2n} \times \{1\} \times \mathbb{R}_{\geq 0} & \text{if } x \in C_1 \setminus D_1. \end{cases} \quad (5.30)$$

Therefore, $F(x) \cap T_C(x) \neq \emptyset$, satisfying (VC) of Proposition A.1.1 for each point $x \in C \setminus D$, and nontrivial solutions exist for every initial point in $(C_0 \cup C_1) \cup (D_0 \cup D_1)$, where $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$ and $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$. To prove that item (c) of Proposition A.1.1 does not hold, we need to show that $G(D) \subset C \cup D$.

With D defined in (5.23),

$$G(D) = (\mathcal{T}_{0,1} \times \{1\} \times \{0\}) \cup (\mathcal{T}_{1,0} \times \{0\} \times \{0\}). \quad (5.31)$$

Notice that $\mathcal{T}_{1,0} \times \{0\} \times \{0\} \subset C_0$ and $\mathcal{T}_{0,1} \times \{1\} \times \{0\} \subset C_1$. Therefore, $G(D) \subset C$; hence $G(D) \subset C \cup D$. Therefore, item (c) of Proposition A.1.1 does not hold. Then it remains to prove that item (b) does not happen.

To this end, since L is \mathcal{C}^1 , nonstrongly convex, and has a single minimizer z_1^* by Assumption 3.1.8, since ∇L is Lipschitz continuous by Assumption 3.1.3, and since L has quadratic growth away from z_1^* by Assumption 3.2.4, then each maximal solution to \mathcal{H}_0 in (5.24), with κ_1 defined via (5.1a) and h_0 defined via (5.21), is bounded, complete, and unique by Proposition 3.2.7. Furthermore, since \bar{d} and $\bar{\beta}$, defined via (3.35), are continuous, since by Assumption 3.1.8, L is \mathcal{C}^1 , and since by Assumption 3.1.3 ∇L is Lipschitz continuous, then each maximal solution to \mathcal{H}_1 , defined via (5.25), is complete and unique by Proposition 3.1.10.

Since \mathcal{H}_0 has no finite time escape from \mathbb{R}^{2n} and \mathcal{H}_1 has no finite time escape from $\mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$, then this means $\dot{x} = F(x)$ has no finite time escape from C for \mathcal{H} , as q does not change in C and as the state component τ is bounded in C , namely, the state component τ – which is always reset to 0 in D – increases linearly in C_1 and remains at 0 in C_0 . Therefore, there is no finite time escape from $C \cup D$, for solutions x to \mathcal{H} . Therefore, item (b) from Proposition A.1.1 does not hold. \square

5.2.8 Main Result

In this section, we present a result that establishes UGAS of the set

$$\mathcal{A} := \{z \in \mathbb{R}^{2n} : \nabla L(z_1) = z_2 = 0\} \times \{0\} \times \{0\} = \{z_1^*\} \times \{0\} \times \{0\} \times \{0\} \quad (5.32)$$

and a hybrid convergence rate that, globally, is equal to $\frac{1}{(t+2)^2}$ while locally, is exponential, for the hybrid closed loop algorithm \mathcal{H} in (5.22) and (5.23). Recall that the state $x := (z, q, \tau) \in \mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$. In light of this, the first component of \mathcal{A} , namely, $\{z_1^*\}$, is the minimizer of L . The second component of \mathcal{A} , namely, $\{0\}$, reflects the fact that we need the velocity state z_2 to equal zero in \mathcal{A} so that solutions are not pushed out of such a set. The third component in \mathcal{A} , namely, $\{0\}$, is due to the logic state ending with the value $q = 0$, namely using κ_0 as the state z reaches the set of minimizers of L . The last component in \mathcal{A} is due to τ being set to, and then staying at, zero when the supervisor switches to κ_0 .

Theorem 5.2.4. *(Uniform global asymptotic stability of \mathcal{A} for \mathcal{H}) Let the function L satisfy Assumptions 3.1.8, 3.2.4, and 3.1.3. Let $\zeta > 0$, $\lambda > 0$, $\gamma > 0$, $c_{1,0} \in (0, c_0)$, and $\varepsilon_{1,0} \in (0, \varepsilon_0)$ be given. Let $\alpha > 0$ be generated by Assumption 3.2.4, and let $M > 0$ be generated by Assumption 3.1.3. Let $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ and $d_{1,0} \in (0, d_0)$ be defined via (5.5) and (5.26). Let the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ be defined via (4.20) (5.12), and (5.15), respectively. Let the functions \bar{d} and $\bar{\beta}$ be defined as in (3.35), and let κ_0 and κ_1 be defined via (5.1a) and (5.20), respectively. Then, the set \mathcal{A} , defined via (5.32), is uniformly globally asymptotically stable for \mathcal{H} given in (5.22)-(5.23). Furthermore, each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j), \tau(t, j))$ of the hybrid closed-loop algorithm \mathcal{H} starting from C_1 with $\tau(0, 0) = 0$ satisfies the following:*

- 1) *The domain $\text{dom } x$ of the solution x is of the form $\cup_{j=0}^1 (I^j \times \{j\})$, with I^0 of the form $[t_0, t_1]$ and with I^1 of the form $[t_1, \infty)$ for some $t_1 \geq 0$ defining the time of the first jump;*

2) For each $t \in I^0$ such that⁶ $t \geq 0$

$$L(z_1(t, 0)) - L^* \leq \frac{4cM}{\zeta^2(t+2)^2} (|z_1(0, 0) - z_1^*|^2 + |z_2(0, 0)|^2) \quad (5.33)$$

where $c := (1 + \zeta^2) \exp\left(\sqrt{\frac{13}{4} + \frac{\zeta^4}{M}}\right)$. Namely, $L(z_1(t, 0)) - L^*$ is $\mathcal{O}\left(\frac{4cM}{\zeta^2(t+2)^2}\right)$;

3) For each $t \in I^1$

$$L(z_1(t, 1)) - L^* = \mathcal{O}(\exp(-(1-m)\psi t)) \quad (5.34)$$

where $m \in (0, 1)$ is such that $\psi := \frac{m\alpha\gamma}{\lambda} > 0$ and $\nu := \psi(\psi - \lambda) < 0$.

Proof. The hybrid closed-loop algorithm \mathcal{H} satisfies the hybrid basic conditions by Lemma 5.2.2, satisfying the first assumption of Theorem A.1.3. Furthermore, $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$, $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$, and each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j), \tau(t, j))$ to \mathcal{H} in (5.22)-(5.23) is complete and bounded by Proposition 5.2.3. Since by Assumption 3.1.8, L has a unique minimizer z_1^* , then \mathcal{A} , defined via (5.32), is compact by construction, and $\mathcal{U} = \mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$ contains a nonzero open neighborhood of \mathcal{A} , satisfying the second assumption of Theorem A.1.3.

To prove attractivity of \mathcal{A} in (5.32), we proceed by contradiction. Suppose there exists a complete solution x to \mathcal{H} such that $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} \neq 0$. Since Proposition 5.2.3 guarantees completeness of maximal solutions, we have the following cases:

- a) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_1 \setminus D_1$ for all $(t, j) \in \text{dom } x$, $t + j \geq t' + j'$;

⁶Note that at each $t \in I^0$, $q(t, 0) = 1$, and at each $t \in I^1$, $q(t, 1) = 0$.

- b) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_0 \setminus (\mathcal{A} \cup D_0)$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$;
- c) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in D$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$.

Case a) contradicts the fact that, by Proposition 3.1.13, the set \mathcal{A}_1 , defined via (3.64), is uniformly globally asymptotically stable for \mathcal{H}_1 in (5.25). Such uniform global attractivity of \mathcal{A} , guaranteed by Proposition 3.1.13, implies there exist $\tilde{c}_1 \in (0, \tilde{c}_{1,0})$ and $d_1 \in (0, d_{1,0})$ such that the state z reaches $(\{z_1^*\} + \tilde{c}_1\mathbb{B}) \times (\{0\} + d_1\mathbb{B}) \subset \mathcal{T}_{1,0}$ at some finite flow time $t \geq 0$ or as $t \rightarrow \infty$. In turn, due to the construction of C_1 and D_1 in (5.23), with $\mathcal{T}_{1,0}$ defined via (5.12), the solution x must reach D_1 at some $(t, j) \in \text{dom } x, t + j \geq t' + j'$. Therefore, case a) does not happen.

Case b) contradicts the fact that, by Proposition 3.2.8, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_0 in (5.24). In fact, $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} = 0$, and since $\mathcal{A} \subset C_0$, case b) does not happen.

Case c) contradicts the fact that, due to the construction of $\mathcal{T}_{1,0}$ in (5.12) and $\mathcal{T}_{0,1}$ in (5.15), we have

$$\begin{aligned}
G(D) \cap D &:= ((\mathcal{T}_{0,1} \times \{1\} \times \{0\}) \cup (\mathcal{T}_{1,0} \times \{0\} \times \{0\})) \\
&\quad \cap ((\mathcal{T}_{0,1} \times \{0\} \times \{0\}) \cup (\mathcal{T}_{1,0} \times \{1\} \times \mathbb{R}_{\geq 0})) \\
&= \emptyset
\end{aligned} \tag{5.35}$$

where $G(D)$ is defined via (5.31) and D is defined in (5.23). Such an equality holds since $\mathcal{T}_{1,0} \cap \mathcal{T}_{0,1} = \emptyset$; see the end of Section 5.2.5. Therefore, case c) does not happen.

Therefore, cases a)-c) do not happen, and each maximal and complete solution

$x = (z, q, \tau)$ to \mathcal{H} with $\tau(0, 0) = 0$ converges to \mathcal{A} . Consequently, by the construction of C and D in (5.23), the uniform global asymptotic stability of \mathcal{A}_1 (defined via (3.64)) for \mathcal{H}_1 in (5.25) established in Proposition 3.1.13, the uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_0 in (5.24) established in Proposition 3.2.8, and since each maximal solution to \mathcal{H} is complete by Proposition 5.2.3, the set \mathcal{A} in (5.32) is uniformly globally asymptotically stable for \mathcal{H} .

To show that each maximal and complete solution x to \mathcal{H} jumps no more than twice, we proceed by contradiction. Without loss of generality, suppose there exists a maximal and complete solution that jumps three times. We have the following possible cases:

- i) The solution first jumps at a point in D_0 , then jumps at a point in D_1 , and then jumps at a point in D_0 ; or
- ii) The solution first jumps at a point in D_1 , then jumps at a point in D_0 , and then jumps at a point in D_1 .

Case i) does not hold since, once the jump in D_1 occurs, the solution x is in $(\mathcal{T}_{1,0} \times \{0\} \times \{0\}) \subset C_0$. Due to the construction of $\mathcal{T}_{1,0}$ in (5.12) and $\mathcal{T}_{0,1}$ in (5.15) such that $\mathcal{T}_{1,0} \cap \mathcal{T}_{0,1} = \emptyset$, as described in the contradiction of case c) above, and due to the uniform global asymptotic stability of $\{z_1^*\} \times \{0\}$ for \mathcal{H}_0 by Proposition 3.2.8, the solution x will never return to D_0 . Therefore, case i) does not happen. Case ii) leads to a contradiction for the same reason, and in this case, once the first jumps in D_1 occurs, no more jumps happen. Therefore, since cases i)-ii) do not happen, each maximal and complete solution x to \mathcal{H} with $\tau(0, 0) = 0$ has no more than two jumps.

Finally, we prove the hybrid convergence rate of \mathcal{H} . Letting $\zeta > 0$ and letting $M > 0$ come from Assumption 3.1.3, then by Proposition 3.1.12, since L satisfies

Assumptions 3.1.8 and 3.1.3, each maximal solution $t \mapsto (z(t), \tau(t))$ to the closed-loop algorithm \mathcal{H}_1 with $\tau(0, 0) = 0$ satisfies (3.53), for all $t \geq 0$, where $c := (1 + \zeta^2) \exp\left(\sqrt{\frac{13}{4} + \frac{\zeta^4}{M}}\right)$. By Proposition 3.2.10, since L satisfies Assumptions 3.1.8 and 3.2.4, then, given $\gamma > 0$ and $\lambda > 0$, for each $m \in (0, 1)$ such that $\psi := \frac{m\alpha\gamma}{\lambda} > 0$ and $\nu := \psi(\psi - \lambda) < 0$, each maximal solution $t \mapsto z(t)$ to the closed-loop algorithm \mathcal{H}_0 satisfies (3.91) for all $t \in \text{dom } z (= \mathbb{R}_{\geq 0})$. Since maximal solutions $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j), \tau(t, j))$ to \mathcal{H} starting from C_1 are guaranteed to jump no more than once, as implied by the contradiction in cases i)-ii) above, then the domain of each maximal solution x to \mathcal{H} starting from C_1 is $\cup_{j=0}^1 (I^j, j)$, with I^0 of the form $[t_0, t_1]$ and with I^1 of the form $[t_1, \infty)$. Therefore, given $\zeta > 0$, $\lambda > 0$, $\gamma > 0$, $c_{1,0} \in (0, c_0)$, $\varepsilon_{1,0} \in (0, \varepsilon_0)$, $\alpha > 0$ from Assumption 3.2.4, and $M > 0$ from Assumption 3.1.3, due to the construction of \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ in (4.20), (5.12), and (5.15), with $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ and $d_{1,0} \in (0, d_0)$ defined via (5.5) and (5.26), and due to the individual convergence rates of \mathcal{H}_1 and \mathcal{H}_0 , each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j), \tau(t, j))$ to the hybrid closed-loop algorithm \mathcal{H} that starts in C_1 , such that $\tau(0, 0) = 0$, satisfies (5.33) for each $t \in I^0$ at which $q(t, 0)$ is equal to 1 and $t \geq 0$, and satisfies (5.34) for each $t \in I^1$ at which $q(t, 1)$ is equal to 0. \square

5.2.9 Numerical Examples

In this section, we present multiple numerical examples to illustrate the hybrid closed-loop algorithm in (5.22) and (5.23). Example 5.2.5 first illustrates the operation of the nominal hybrid closed-loop system \mathcal{H} , and then demonstrates the robustness of \mathcal{H} to different amounts of noise in measurements of ∇L . Example 5.2.6 compares solutions to the hybrid closed-loop algorithm in (5.22) and (5.23) with solutions to \mathcal{H}_0 , \mathcal{H}_1 , and HAND-1 from [32], with parameters chosen such

that HAND-1 and \mathcal{H} are compared on equal footing. Example 5.2.6 then compares multiple solutions of \mathcal{H} , starting from different initial values of z_1 , to multiple solutions of HAND-1 from such initial values of z_1 , to show that \mathcal{H} has a consistent percentage of improvement over HAND-1 for different solutions. Example 5.2.7 illustrates the trade-off between speed of convergence and the resulting values of parameters for the uniting algorithm \mathcal{H} , for different tunings of $\zeta > 0$. As in Example 5.2.6, the parameter values for Example 5.2.7 are chosen such that HAND-1 and \mathcal{H} are compared on equal footing.

Example 5.2.5. *In this example, we simulate a solution to the nominal hybrid closed-loop system \mathcal{H} to illustrate how the uniting algorithm works. Then, we compare that same solution to solutions with different amounts of noise in measurements of ∇L . For both the nominal system and the perturbed system, the choice of objective function, parameter values, and initial conditions are as follows. We use the objective function $L(z_1) := z_1^2$, the gradient of which is Lipschitz continuous with $M = 2$, and which has a single minimizer at $z_1^* = 0$. This choice of objective function is made so that we can easily tune λ , as described in Section 5.1.6. We arbitrarily chose the heavy ball parameter value $\gamma = \frac{2}{3}$ and we tuned λ to 200 by choosing a value arbitrarily larger than $2\sqrt{a_1}$, where a_1 comes from Section 5.1.6, and gradually increasing it until there is no overshoot in the hybrid algorithm. The parameter values for the uniting algorithm are $c_0 = 7000$, $c_{1,0} \approx 6819.68$, $\varepsilon_0 = 10$, $\varepsilon_{1,0} = 5$, and $\alpha = 1$, which yield the values $\tilde{c}_0 = 10$, $\tilde{c}_{1,0} = 5$, $d_0 = 6933$, and $d_{1,0} = 6744$, which are calculated via (5.5) and (5.26). These values are chosen for proper tuning of the algorithm, in order to get nice performance, and the value of $c_{1,0}$ is chosen to exploit the properties of Nesterov's method for a longer time, so that the nominal solution gets closer to the minimizer faster. Initial conditions for \mathcal{H} are $z_1(0,0) = 50$, $z_2(0,0) = 0$, $q(0,0) = 1$, and*

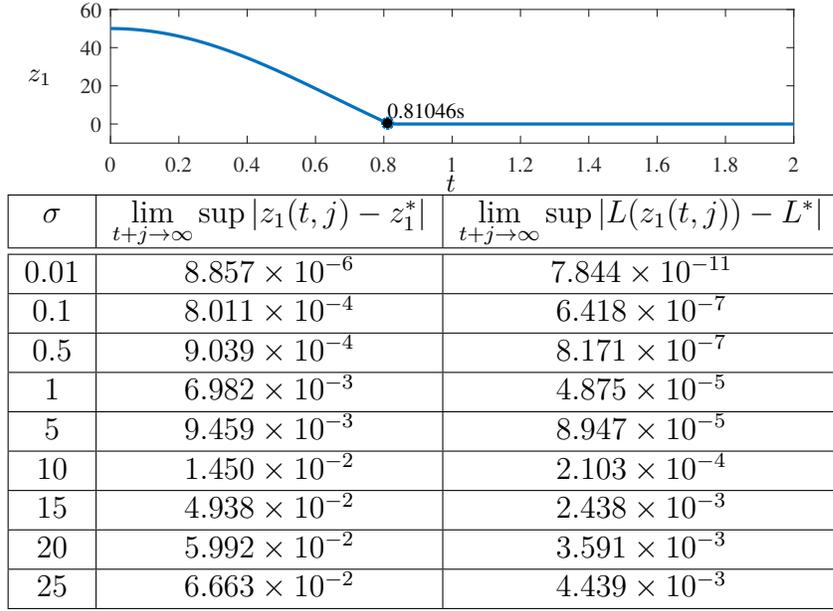


Figure 5.5: Top: The evolution over time of z_1 , for the nominal hybrid closed-loop algorithm \mathcal{H} , for a function $L(z_1) := z_1^2$ with a single minimizer at $z_1^* = 0$. The time at which the solution settles to within 1% of z_1^* is marked with a dot and labeled in seconds. The jump is labeled with an asterisk. Bottom: Simulation results for perturbed solutions using zero mean Gaussian noise, with each simulation using a different value of the standard deviation σ . Results listed are for a large value of $t + j$.

$\tau(0, 0) = 0$. The plot on the top in Figure 5.5 shows the solution to the nominal hybrid closed-loop algorithm⁷ \mathcal{H} , namely, the value of z_1 over time, with the time it takes for the solution to settle to within 1% of z_1^* marked with a black dot and labeled in seconds. The jump at which the switch from \mathcal{H}_1 to \mathcal{H}_0 occurs is labeled with an asterisk. The solution converges quickly, without oscillations near the minimizer.

To show that the uniform global asymptotic stability of \mathcal{A} , established in Theorem 5.2.4, is robust to small perturbations, due to the hybrid closed-loop system \mathcal{H} satisfying the hybrid basic conditions by Lemma 5.2.2. we simulate the hybrid algorithm, using the objective function, parameter values, and initial conditions

⁷Code at github.com/HybridSystemsLab/UnitingRobustness.

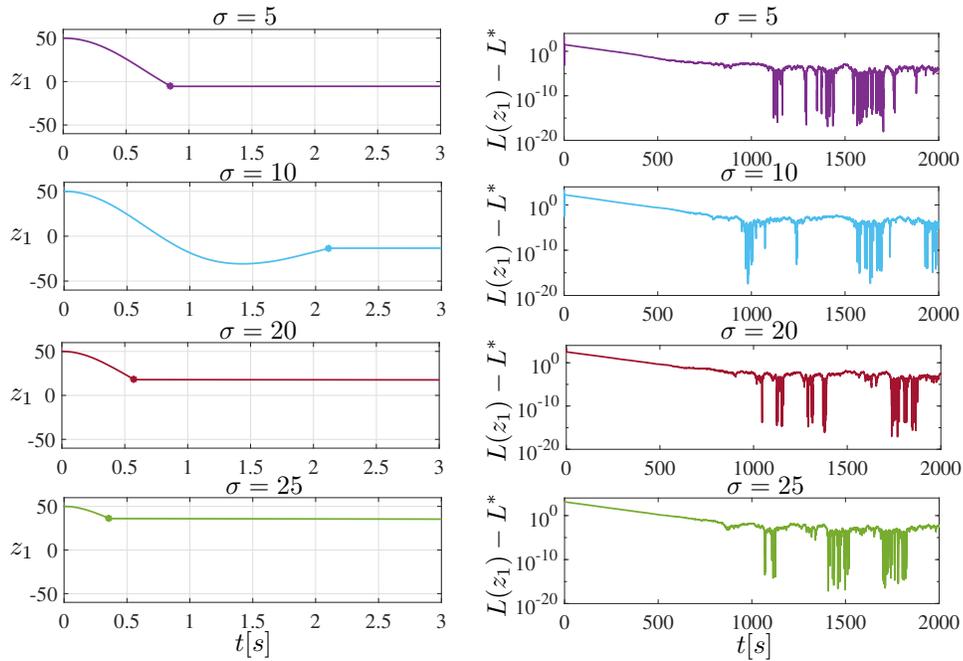


Figure 5.6: Simulation results for hybrid closed-loop algorithm \mathcal{H} , for a function $L(z_1) := z_1^2$ with a single minimizer at $z_1^* = 0$, with zero-mean Gaussian noise added to measurements of the gradient. Each subplot is labeled with the standard deviation used. Left subplots: the value of z_1 over time for each perturbed solution, with the jump in each solution labeled by an asterisk. Right subplots: the corresponding value of L over time for each perturbed solution.

listed in the first paragraph of this example, with zero-mean Gaussian noise added to measurements of the gradient. Separate simulations were run for each of the following standard deviations: $\sigma \in \{0.01, 0.1, 0.5, 1, 5, 10, 15, 20, 25\}$. Figure 5.6 shows some of these perturbed solutions, with each subplot labeled with the corresponding standard deviation used⁸. The subplots on the left side of Figure 5.6 show the value of z_1 over time for different standard deviations, and the subplots on the right side of Figure 5.6 show the corresponding value of L over time for such standard deviations. Note that, while all perturbed solutions shown in Figure 5.6 get close to the minimizer quickly, such perturbed solutions do not get as close to the minimizer as the solution to the nominal algorithm does; see the plot on the top in Figure 5.5. Also note that as the standard deviation gets larger, the corresponding perturbed solution stays slightly farther away from the minimizer. The results for all standard deviations are listed in the table in Figure 5.5, showing the neighborhood of z_1^* that each solution settles to, for a large value of $t + j$, along with the corresponding value of L .

Example 5.2.6. In this example, to show the effectiveness of the uniting algorithm, we compare the hybrid closed-loop algorithm \mathcal{H} , defined via (5.22) and (5.23), with the individual closed-loop optimization algorithms \mathcal{H}_0 and \mathcal{H}_1 and with the HAND-1 algorithm from [32] which, in [32], is designed and analyzed for nonstrongly convex functions L satisfying Assumptions 3.1.8 and 3.1.3. First, we compare the convergence rates of \mathcal{H} and HAND-1 analytically. Using an alternate state space representation, namely, $z_1 := \xi$ and $z_2 := \xi + \frac{\tau}{2}\dot{\xi}$, the HAND-1

⁸Code found at same link as in Footnote 7.

algorithm has state $(z, \tau) \in \mathbb{R}^{2n+1}$ and data (C, F, D, G)

$$F(z, \tau) := \begin{bmatrix} \frac{2}{\tau}(z_2 - z_1) \\ -2c_1\tau\nabla L(z_1) \\ 1 \end{bmatrix} \quad (z, \tau) \in C, \quad G(z, \tau) := \begin{bmatrix} z \\ T_{\min} \end{bmatrix} \quad (z, \tau) \in D \quad (5.36)$$

where $c_1 > 0$ and the flow and jump sets are $C := \{(z, \tau) \in \mathbb{R}^{2n+1} : \tau \in [T_{\min}, T_{\max}]\}$ and $D := \{(z, \tau) \in \mathbb{R}^{2n+1} : \tau \in [T_{\min}, T_{\max}]\}$, with $0 < T_{\min} < T_{\text{med}} < T_{\max} < \infty$, and $T_{\text{med}} \geq \sqrt{\frac{B}{\delta_{\text{med}}}} + T_{\min} > 0$, $\delta_{\text{med}} > 0$. It is shown in [32] that each maximal solution $(t, j) \mapsto (z(t, j), \tau(t, j))$ to the HAND-1 algorithm satisfies

$$L(z_1(t, 0)) - L^* \leq \frac{B}{t^2} \quad (5.37)$$

for all $(t, j) \in \text{dom}(z, \tau)$ such that $j = 0$, $z_1(0, 0) = z_2(0, 0)$, $\tau(0, 0) = T_{\min}$, $z_1(0, 0) \in K_0 := \{z_1^*\} + r\mathbb{B}$, where $B := \frac{r^2}{2c_1} + T_{\min}^2 (L(z_1(0, 0)) - L^*) > 0$, $r \in \mathbb{R}_{>0}$, $c_1 > 0$.

For the hybrid closed-loop algorithm \mathcal{H} , the coefficient of the bound on \mathcal{H}_1 from (5.33), namely,

$$L(z_1(t, 0)) - L^* \leq \frac{4cM}{\zeta^2(t+2)^2} \left(|z_1(0, 0) - z_1^*|^2 + |z_2(0, 0)|^2 \right) \quad (5.38)$$

for each $t \in I^0$, $t \geq 0$, at which $q(t, 0) = 1$, and for each $\zeta > 0$, and $M > 0$, is $\frac{4cM}{\zeta^2} \left(|z_1(0, 0) - z_1^*|^2 + |z_2(0, 0)|^2 \right)$, where $c := (1 + \zeta^2) \exp\left(\sqrt{\frac{13}{4} + \frac{\zeta^4}{M}}\right)$. The coefficient of the bound in HAND-1 is $B := \frac{r^2}{2c_1} + T_{\min}^2 (L(z_1(0, 0)) - L^*)$. Since, as $t \rightarrow \infty$, $\frac{1}{(t+2)^2} \rightarrow \frac{1}{t^2}$, then, comparing the coefficients of the bounds, the bound in (5.38) is slightly better than (5.37) since $\frac{r^2}{2c_1}$ is very large for small t . Neglecting the $\frac{r^2}{2c_1}$ term, however, the bound on \mathcal{H}_1 (5.38) matches (5.37). The rate for HAND-1, nevertheless, is only guaranteed until the first jump. After this, there

is no characterized bound for HAND-1. In contrast, \mathcal{H} has a characterized bound for the domain of every solution such that $t \geq 0$. Namely, it has rate $\frac{1}{(t+2)^2}$ until the state z is within a small neighborhood of the minimizer – where the rate then switches to $\exp(-(1-m)\psi t)$, where, given $\gamma > 0$ and $\lambda > 0$, $m \in (0, 1)$ is such that $\psi = \frac{m\alpha\gamma}{\lambda} > 0$ and $\nu = \psi(\psi - \lambda) < 0$.

Next, we compare \mathcal{H}_0 , \mathcal{H}_1 , \mathcal{H} , and HAND-1 in simulation. To compare these algorithms, we use the same objective function L , heavy ball parameter values λ and γ , Lipschitz parameter M , Nesterov parameter ζ , and uniting algorithm parameter values c_0 , $c_{1,0}$, ε_0 , $\varepsilon_{1,0}$, α , \tilde{c}_0 , $\tilde{c}_{1,0}$, d_0 , and $d_{1,0}$ as in Example 5.2.5. Given $\zeta = 2$, the HAND-1 parameters $c_1 = 0.5$ and $T_{\min} = \frac{1+\sqrt{7}}{2}$ are chosen such that the resulting gain coefficients for z_1 and z_2 are the same for both \mathcal{H} and HAND-1, so that these algorithms are compared on equal footing⁹. The remaining HAND-1 parameters, r and δ_{med} , have different values depending on the initial conditions $z_1(0, 0) = z_2(0, 0)$, listed in Table 5.3, which leads to different values of T_{med} and T_{\max} , for each solution. Such values are chosen such that $T_{med} \geq \sqrt{\frac{B}{\delta_{med}}} + T_{\min} > 0$. Additionally, we choose $T_{\max} = T_{med} + 1$. The parameter values for the uniting algorithm are $\varepsilon_0 = 10$, $\varepsilon_{1,0} = 5$, and $\alpha = 1$. The remaining parameter values c_0 and $c_{1,0}$ are different depending on the initial condition $z_1(0, 0)$ and are listed in Table 5.3, which leads to different values of which leads to different values of d_0 , calculated via (5.5), and $d_{1,0}$ calculated via (5.26). These values are chosen for proper tuning of the algorithm, in order to get nice performance, and for exploiting the properties of Nesterov’s method as long as we want. Initial conditions for all solutions to \mathcal{H} are $z_2(0, 0) = 0$, $q(0, 0) = 1$, and $\tau(0, 0) = 0$, with values of $z_1(0, 0)$ listed in Table 5.3. Initial conditions for all solutions to HAND-1 are $\tau(0, 0) = T_{\min}$, with values of $z_1(0, 0) = z_2(0, 0)$ listed in Table 5.3.

⁹Although there exist parameter values for which HAND-1 has faster, oscillation-free performance, due to the way \mathcal{H} and HAND-1 relate to each other, they are compared fairly for a particular set of parameters.

Table 5.2 shows the time that each algorithm takes to settle within¹⁰ 1% of z_1^* ,

Algorithm	Average time to converge (s)	Average % improvement of \mathcal{H}
\mathcal{H}	0.811	–
\mathcal{H}_0	690.759	99.9
\mathcal{H}_1	4.409	81.6
HAND-1	8.649	90.6

Table 5.2: Average times for which \mathcal{H} , \mathcal{H}_0 , \mathcal{H}_1 , and HAND-1 settle to within 1% of z_1^* , and the average percent improvement of \mathcal{H} over each algorithm. Percent improvement is calculated via (5.39). The objective function used for this table is $L(z_1) := z_1^2$.

averaged over solutions starting from ten different values¹¹ of $z_1(0,0)$ (listed in the first column of Table 5.3), and the average percent improvement of \mathcal{H} over \mathcal{H}_0 , \mathcal{H}_1 , and HAND-1, which is calculated using the following formula

$$\left(\frac{\text{Time of } \mathcal{H}_0, \mathcal{H}_1, \text{ or HAND-1} - \text{Time of } \mathcal{H}}{\text{Time of } \mathcal{H}_0, \mathcal{H}_1, \text{ or HAND-1}} \right) \times 100\%. \quad (5.39)$$

As can be seen in Table 5.2, \mathcal{H} converges faster than the other algorithms, and the average percent improvement of \mathcal{H} over each of the other algorithms in Table 5.2 is 99.9% over \mathcal{H}_0 , 81.6% over \mathcal{H}_1 , and 90.6% over HAND-1.

Figure 5.7 compares different solutions for \mathcal{H} and HAND-1, from different values of $z_1(0,0)$, for the objective function¹² $L(z_1) := z_1^2$. Table 5.3 lists the times for which each solution settles to within 1% of z_1^* for both \mathcal{H} and HAND-1, and shows the percent improvement of \mathcal{H} over HAND-1. As can be seen in Figure 5.7 and in Table 5.3, the percent improvement of \mathcal{H} over HAND-1 for all solutions is 90.6%, which shows consistency in the performance of \mathcal{H} versus HAND-1.

¹⁰Code at github.com/HybridSystemsLab/UnitingNSC.

¹¹Code at github.com/HybridSystemsLab/UnitingDifferentICs

¹²Code found at same link as in Footnote 11.

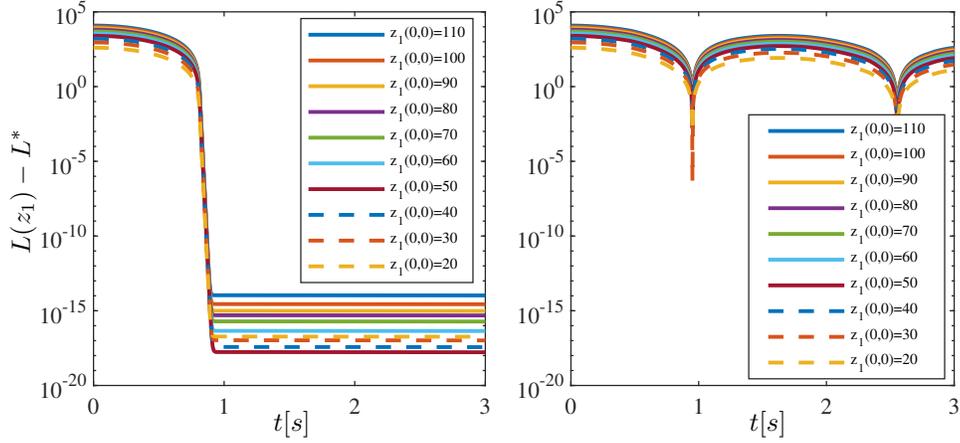


Figure 5.7: The evolution of L over time, from different initial conditions, for \mathcal{H} (left) and HAND-1 (right). All solutions are for the objective function $L(z_1) := z_1^2$, and the parameters used for HAND-1 and \mathcal{H} are listed in Table 5.3, with different values of c_0 and $c_{1,0}$ for each solution of \mathcal{H} , leading to different values of d_0 calculated via (5.5) and $d_{1,0}$ calculated via (5.26), and different values of r and δ_{med} for each solution of HAND-1, leading to different values of T_{med} and T_{max} .

$z_1(0, 0)$	c_0	$c_{1,0}$	r	δ_{med}	Time to converge (s)		% Improvement
					\mathcal{H}	HAND-1	
110	34000	32719.231	111	240700	0.811	8.649	90.6
100	28000	27053.704	101	199000	0.811	8.65	90.6
90	23000	21927.75	91	161300	0.811	8.648	90.6
80	18000	17341.37	81	127550	0.811	8.65	90.6
70	14000	13294.565	71	97700	0.811	8.649	90.6
60	10500	9787.333	61	71875	0.811	8.648	90.6
50	7000	6819.676	51	50000	0.810	8.65	90.6
40	5000	4391.593	41	32075	0.811	8.65	90.6
30	3000	2503.083	31	18110	0.811	8.648	90.6
20	2000	1154.148	21	8112	0.811	8.648	90.6

Table 5.3: Times for which \mathcal{H} and HAND-1 settle to within 1% of z_1^* , and percent improvement of \mathcal{H} over HAND-1, for solutions from different initial conditions, shown in Figure 5.7. The objective function used for this table is $L(z_1) := z_1^2$.

The bound for HAND-1, shown in (5.37) and which holds only until the first reset, is only guaranteed when $z_1(0,0) = z_2(0,0)$. This leads to a required nonzero velocity for HAND-1 in most scenarios, which leads to overshoot. In contrast, \mathcal{H} has no such constraint on $z_2(0,0)$, which can be set to zero in all scenarios. The lack of such a constraint on the initial condition $z_2(0,0)$ for the hybrid closed-loop algorithm \mathcal{H} is essential to its improved performance over HAND-1, as the overshoot in solutions to HAND-1 due to $z_1(0,0) = z_2(0,0)$ leads to a slower convergence time than for \mathcal{H} , as seen in Table 5.2. Moreover, as described previously in this example, no bound for HAND-1 is characterized after the first reset, whereas the (hybrid) convergence bound characterized for \mathcal{H} holds for the domain of every solution such that $t \geq 1$.

Example 5.2.7. This example explores the trade-off that results from using different values of $\zeta > 0$ for the uniting algorithm. Particularly, for $\zeta = 1$, we first compare the uniting algorithm in simulation with the individual optimization algorithms \mathcal{H}_0 , \mathcal{H}_1 , and the HAND-1 algorithm from [32], using the same objective function as in Example 5.2.6, and next we compare the resulting solutions with those in Table 5.2. Recall that the objective function in Example 5.2.6 is $L(z_1) := z_1^2$, the gradient of which is Lipschitz continuous with $M = 2$, and which has a single minimizer at $z_1^* = 0$. Since the gain coefficient of ∇L is proportional to ζ^2 , we choose different parameters for the HAND-1 algorithm for the simulation depicted in¹³ Figure 1.4, so that the gain coefficients of z_1 and z_2 are the same for HAND-1 and \mathcal{H} in this simulation. Namely, given $\zeta = 1$, for HAND-1 we choose $T_{\min} = 3$ and $c_1 = 0.25$. For the other HAND-1 parameters, we choose $r = 51$ and $\delta_{med} = 4010$ such that $T_{med} \geq \sqrt{\frac{B}{\delta_{med}}} + T_{\min} > 0$, and we again choose $T_{\max} = T_{med} + 1$ to ensure resets happen at the proper times. We arbitrarily choose

¹³Code found at same link as in Footnote 4

$\gamma = \frac{2}{3}$, and we tuned λ to 40 by choosing a value arbitrarily larger than $2\sqrt{a_1}$ and gradually increasing until there was no overshoot in the hybrid algorithm. The uniting algorithm parameters are $c_0 = 320$, $c_{1,0} \approx 271.584$, $\varepsilon_0 = 10$, $\varepsilon_{1,0} = 5$, and $\alpha = 1$, which yield the values $\tilde{c}_0 = 10$, $\tilde{c}_{1,0} = 5$, $d_0 \approx 253.333$, and $d_{1,0} \approx 234.084$, which are calculated via (5.5) and (5.26). These values are chosen for proper tuning of the algorithm, in order to get nice performance, and for exploiting the properties of Nesterov's method as long as we want. Initial conditions for \mathcal{H} are $z_1(0,0) = 50$, $z_2(0,0) = 0$, $q(0,0) = 1$, and $\tau(0,0) = 0$, and for HAND-1 are $z_1(0,0) = z_2(0,0) = 50$ and $\tau(0,0) = T_{\min}$.

First, we compare solutions to each algorithm within Figure 1.4 itself. Table 5.4 shows the time that each algorithm takes to settle within 1% of z_1^* , averaged over solutions starting from ten different values of $z_1(0,0)$ (listed in the first column of Table 5.3), and the percent improvement of \mathcal{H} over \mathcal{H}_0 , \mathcal{H}_1 , and HAND-1, which is calculated using (5.39). While the closed-loop algorithm \mathcal{H} still converges faster than all the other algorithms in Figure 1.4 and Table 5.4, the improvement over \mathcal{H}_0 , \mathcal{H}_1 , and HAND-1 is smaller than it is in Table 5.2.

Algorithm	Average time to converge (s)	Average % improvement
\mathcal{H}	2.387	–
\mathcal{H}_0	138.066	98.3
\mathcal{H}_1	8.782	72.8
HAND-1	14.343	83.4

Table 5.4: Times for which \mathcal{H} , \mathcal{H}_0 , \mathcal{H}_1 , and HAND-1 settle to within 1% of z_1^* , and percent improvement of \mathcal{H} over each algorithm, as shown in Figure 1.4. Percent improvement is calculated via (5.39). The objective function used for this table is $L(z_1) := z_1^2$.

Next, we compare solutions using $\zeta = 1$, in Figure 1.4, with solutions using $\zeta = 2$, in Table 5.2. Since $\zeta > 0$ scales time in solutions to (1.5), Then smaller values of ζ result in slower settling to within 1% of z_1^* for \mathcal{H}_1 with less frequent

oscillations, as seen in Figure 1.4 with $\zeta = 1$ (about 8.8 seconds), while larger values of ζ result in settling to within 1% of z_1^* for \mathcal{H}_1 faster, with more frequent oscillations, as seen in Figure 1.3 and Table 5.2 with $\zeta = 2$ (about 4.5 seconds). For the uniting algorithm, this translates to faster settling to within 1% of z_1^* with $\zeta = 2$ (about 0.8 seconds), in Figure 1.3 and Table 5.2, compared with slower settling to within 1% of z_1^* with $\zeta = 1$ (about 2.4 seconds), in Figure 1.4, but with no oscillations, in both cases, due to the switch to \mathcal{H}_0 . In both Figure 1.4, and Table 5.2, the uniting algorithm converges more quickly than the HAND-1 algorithm, when both algorithms are tuned to have the same gain coefficients for the z_1 and z_2 terms. Although larger ζ results in faster convergence, the trade-off is that even though the z_2 (velocity) term generally reduces quickly as it approaches the neighborhood of the minimizer for any size of ζ , the z_2 still ends up relatively larger near the minimizer than it is when ζ is smaller. The consequence is that, when ζ is larger, $d_{1,0}$ needs to be set much larger so that the uniting algorithm can still make the switch to \mathcal{H}_0 at the proper time. This also means that $c_{1,0}$ needs to be set much larger, due to the definition of $d_{1,0}$ in (5.26). Additionally, c_0 and d_0 , also need to be set larger to ensure the algorithm still has adequate hysteresis. Recall that, in Example 5.2.6, for $\zeta = 2$, we have the parameter values $c_0 = 7000$, $c_{1,0} \approx 6819.676$, $d_0 = 6933$, and $d_{1,0} = 6744$, which are quite large, while for the simulation shown in Figure 1.4 these same parameters have much smaller values, as listed in the second paragraph of this example.

5.2.10 Extensions

Some possible extensions to the results in Section 5.2 are as follows.

It is possible to extend the results in Section 5.2 to include \mathcal{C}^1 , nonstrongly convex objective functions L with a compact and connected set of minimizers.

With such an assumption, it would be straightforward to extend Lemma 5.2.2 and Proposition 5.2.3. Theorem 5.2.4 can be extended via the assumption of a compact and connected set of minimizers and the use of Clarke's generalized derivative in (2.4), with the Lyapunov function V in (3.97), as described in Section 3.2.3, and with the Lyapunov function V_1 in (3.69), as described in Section 3.2.3. With such an extension, it can be shown that \mathcal{A} in (5.32) is UGAS for \mathcal{H} in (5.22)-(5.23) with $\tau(0,0) = 0$, and that each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j), \tau(t, j))$ to the hybrid closed-loop algorithm \mathcal{H} that starts in C_1 , such that $\tau(0,0) = 0$, satisfies (5.33) for each $t \in I^0$ at which $q(t,0)$ is equal to 1 and $t \geq 1$, and satisfies (5.34) for each $t \in I^1$ at which $q(t,1)$ is equal to 0.

It would be possible to further extend the results in Lemma 5.2.2, Proposition 5.2.3, and Theorem 5.2.4 to nonstrongly convex objective functions L that are also nonsmooth, through the use of Clarke's generalized derivative.

Chapter 6

Uniting Framework for Accelerated Optimization

In this chapter, we propose a framework for logic-based algorithms, which unite any two continuous-time, gradient-based optimization algorithms κ_0 and κ_1 to solve Problem 6.1.1. The central idea is that the global optimization algorithm κ_1 provides fast convergence to the neighborhood of the set of minimizers and the local optimization algorithm κ_0 provides stable convergence in the neighborhood of the set of minimizers, without oscillations. A logic variable is used to indicate which algorithm – either κ_0 or κ_1 – is currently in use, and the switch between local and global algorithms is based on sublevel sets of the Lyapunov functions of κ_0 and κ_1 . One difficulty in designing such a uniting framework 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.

6.1 Problem Statement

As illustrated in Figure 1.3, 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 1.3 we discussed how the heavy ball algorithm guarantees an exponential rate for strongly convex L and a rate of $\frac{1}{t}$ for nonstrongly convex L , although it was demonstrated in [79] that the heavy ball algorithm converges exponentially for nonstrongly convex L when such an objective function also has the property of quadratic growth away from its minimizer. We also discussed how Nesterov’s algorithm guarantees an exponential convergence rate for strongly convex L and a rate of $\frac{1}{(t+2)^2}$ for nonstrongly convex L . We desire to attain such rates, while avoiding oscillations via the heavy ball algorithm with large λ . We state the following general problem to solve as follows:

Problem 6.1.1. *Given a scalar, real-valued, continuously differentiable objective function L with a unique minimizer, design a unifying optimization framework that, without knowing the function L or the location of its minimizer, has the minimizer uniformly globally asymptotically stable, with a convergence rate that preserves the convergence rates of the local and global algorithms, and with robustness to arbitrarily small noise in measurements of ∇L .*

6.2 Hybrid Uniting Framework for Accelerated Gradient Methods

6.2.1 Modeling

We interpret the ODEs in (1.1), (1.2), and (1.5) as control systems consisting of a plant and a control algorithm [34] [22]. Then, defining z_1 as ξ and z_2 as $\dot{\xi}$, the plant for these ODEs is given by the double integrator in (3.1). With this model, the class of optimization algorithms that we consider assign u to a function of the state that involves the cost function, and such a function of the state may be time dependent. For instance, if the ODE in (1.1) is used, then the algorithm assigns u to $-\lambda_q z_2 - \gamma_q \nabla L(z_1)$, with tunable parameters $\lambda > 0$ and $\gamma > 0$. If the ODE in (1.2) is used, then the algorithm assigns u to $-2dz_2 - \frac{1}{M} \nabla L(z_1 + \beta z_2)$, where $M > 0$ is the Lipschitz constant of ∇L and where d and β are defined via (3.3). If the ODE in (1.5) is used, then the algorithm assigns u to $-2\bar{d}(t)z_2 - \frac{\zeta^2}{M} \nabla L(z_1 + \bar{\beta}(t)z_2)$, where $\zeta > 0$ and where \bar{d} and $\bar{\beta}$ are defined via (3.35).

For the framework presented in this Chapter, we cope with the trade-off between damping oscillations and converging fast by uniting two control algorithms κ_q , where the logic variable $q \in Q := \{0, 1\}$ indicates which algorithm is currently being used. As was discussed in Chapters 4 and 5, the algorithm defined by κ_1 , which plays the role of the global algorithm in uniting control (see, e.g., [22]), is used far from the minimizer and is designed to quickly get close to the critical point. The algorithm defined by κ_0 , which plays the role of the local algorithm, is used near the minimizer and is designed to avoid oscillations. 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. The design

of the logic and parameters of the individual algorithms is done using Lyapunov functions V_q , which take different forms depending on the specific optimization methods used for κ_0 and κ_1 . Since the ODE in (1.5) is time varying, and since solutions to hybrid systems are parameterized by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, we employ the state τ to capture ordinary time as a state variable, in this way, leading to a time-invariant hybrid system.

To encapsulate the plant, static state-feedback laws, and the time-varying nature of the ODE in (1.5), we define a hybrid closed-loop system \mathcal{H} with state $x := (z, q, \tau) \in \mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$ as follows:

$$\left. \begin{aligned} \dot{z} &= \begin{bmatrix} z_2 \\ \kappa_q(h_q(z, \tau), \tau) \end{bmatrix} \\ \dot{q} &= 0 \\ \dot{\tau} &= q \end{aligned} \right\} =: F(x) \quad x \in C := C_0 \cup C_1 \quad (6.1a)$$

$$\left. \begin{aligned} z^+ &= \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \\ q^+ &= 1 - q \\ \tau^+ &= 0 \end{aligned} \right\} =: G(x) \quad x \in D := D_0 \cup D_1 \quad (6.1b)$$

The outputs h_q are defined differently, based on the specific optimization methods used. The sets C_0 , C_1 , D_0 , and D_1 are defined as

$$C_0 := \mathcal{U}_0 \times \{0\} \times \{0\}, \quad C_1 := \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}} \times \{1\} \times \mathbb{R}_{\geq 0} \quad (6.2a)$$

$$D_0 := \mathcal{T}_{0,1} \times \{0\} \times \{0\}, \quad D_1 := \mathcal{T}_{1,0} \times \{1\} \times \mathbb{R}_{\geq 0}, \quad (6.2b)$$

and where \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ are defined differently, depending on the specific

optimization algorithms employed for κ_0 and κ_1 . However, the idea behind their construction is as follows. The switch between κ_0 and κ_1 is governed by a *supervisory algorithm* implementing switching logic; see Figure 5.4. The supervisor selects between these two optimization algorithms, based on the output of the plant and the optimization algorithm currently applied. When $z \in \mathcal{U}_0$, $q = 0$, and $\tau = 0$ (i.e., $x \in C_0$), due to the design of \mathcal{U}_0 , then the state z is near the minimizer, which is denoted z_1^* , and the supervisor allows flows of (6.1) using κ_0 and $\dot{\tau} = q = 0$ to avoid oscillations. Conversely, when $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}$ and $q = 1$ (i.e., $x \in C_1$), due to the design of $\mathcal{T}_{1,0}$, then the state z is far from the minimizer and the supervisor allows flows of (6.1) using κ_1 and $\dot{\tau} = q = 1$ to converge quickly to the neighborhood of the minimizer. When $z \in \mathcal{T}_{1,0}$ and $q = 1$ (i.e., $x \in D_1$), then this indicates that the state z is near the minimizer, and the supervisor assigns u to κ_0 , resets q to 0, and resets τ to 0. Conversely, when $z \in \mathcal{T}_{0,1}$, $q = 0$, and $\tau = 0$ (i.e., $x \in D_0$), due to the design of $\mathcal{T}_{0,1}$, then this indicates that the state z is far from the minimizer and the supervisor assigns u to κ_1 and resets q to 1. The complete algorithm, defined in (6.1)-(6.2), is summarized in Algorithm 4.

Algorithm 4 Uniting algorithm

- 1: Set $q(0,0)$ to 0, $\tau(0,0)$ to 0, and set $z(0,0)$ as an initial condition with an arbitrary value.
 - 2:
 - 3: **while** true **do**
 - 4: **if** $z \in \mathcal{T}_{0,1}$, $q = 0$, and $\tau = 0$ **then**
 - 5: Reset q to 1.
 - 6: **else if** $z \in \mathcal{T}_{1,0}$ and $q = 1$ **then**
 - 7: Reset q to 0 and τ to 0.
 - 8: **else if** $z \in \mathcal{U}_0$, $q = 0$, and $\tau = 0$ **then**
 - 9: Assign u to $\kappa_0(h_0(z, \tau), \tau)$ and update z , q , and τ according to (6.1a).
 - 10: **else if** $z \in \overline{\mathbb{R}^{2n} \setminus \mathcal{T}_{1,0}}$ and $q = 1$ **then**
 - 11: Assign u to $\kappa_1(h_1(z, \tau), \tau)$ and update z , q , and τ according to (6.1a).
 - 12: **end if**
 - 13: **end while**
-

The reason that the state τ in (6.1) changes at the rate q during flows and is reset to 0 at jumps is that when the state x is in C_1 , then $\dot{\tau} = q = 1$, which implies that τ behaves as ordinary time, so it is used to represent time in the potentially time-varying algorithm κ_1 . On the other hand, when the state x is in C_0 , then $\dot{\tau} = q = 0$ causes the state τ to stay at zero. Such an evolution ensures that the set to asymptotically stabilize is compact.

Figure 5.4 shows the feedback diagram of this hybrid closed-loop system \mathcal{H} . We denote, for each $q \in Q = \{0, 1\}$, the closed-loop systems resulting from the individual optimization algorithms as \mathcal{H}_q with state (z, τ) , which are given by

$$\left. \begin{aligned} \dot{z} &= \begin{bmatrix} z_2 \\ \kappa_q(h_q(z, \tau), \tau) \end{bmatrix} \\ \dot{\tau} &= 1 \end{aligned} \right\} (z, \tau) \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}. \quad (6.3)$$

6.2.2 Design

The sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$ need to be designed such that the supervisor can determine when the state component z_1 is close to the set of minimizers of L , denoted z_1^* , without knowledge of z_1^* or L^* . To facilitate such a design, for this chapter, we impose¹ Assumptions 3.1.8 and 3.2.4 on the objective function L . Namely, L is \mathcal{C}^1 , nonstrongly convex, and has a unique minimizer by Assumption 3.1.8, has a Lipschitz continuous gradient by Assumption 3.1.3, and has quadratic growth away from its minimizer z_1^* by Assumption 3.2.4.

The set \mathcal{U}_0 is defined, via Definition 2.2.2, and Lemma 4.4.1, as follows:

$$\mathcal{U}_0 := \left\{ z \in \mathbb{R}^{2n} : |\nabla L(z_1)| \leq \tilde{c}_0, \frac{1}{2} |z_2|^2 \leq d_0 \right\} \quad (6.4)$$

¹When L is strongly convex, then L also satisfies the definition of nonstrong convexity (Definition 2.2.2) and quadratic growth in Definition 2.2.3, since these are weaker properties than strong convexity; see [67], [69], [19], [68], [70].

where the parameters $\tilde{c}_0 > 0$ and $d_0 > 0$ are designed so that \mathcal{U}_0 is in the region where κ_0 is used.

The set $\mathcal{T}_{1,0}$ is defined, via Definition 2.2.2, and Lemma 4.4.1, as follows:

$$\mathcal{T}_{1,0} := \left\{ z \in \mathbb{R}^{2n} : |\nabla L(z_1)| \leq \tilde{c}_{1,0}, |z_2|^2 \leq d_{1,0} \right\} \quad (6.5)$$

where the parameters $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ and $d_{1,0} \in (0, d_0)$ are designed such that $\mathcal{T}_{1,0}$ is contained in the interior of \mathcal{U}_0 . 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 \tilde{c}_0 , $\tilde{c}_{1,0}$, d_0 , and $d_{1,0}$ comprise the hysteresis necessary to avoid chattering at the switching boundary; see Figure 4.3. Examples illustrating the design of \mathcal{U}_0 and $\mathcal{T}_{1,0}$ for specific cases of κ_0 and κ_1 were presented in Chapters 4 and 5.

The set $\mathcal{T}_{0,1}$ should be designed such that the supervisor can determine that the state z is far from the minimizer, when $z \in \mathcal{T}_{0,1}$ and $q = 0$, and make the switch back to κ_1 to ensure that the state z reaches the neighborhood of the minimizer in finite time. Additionally, the set $\mathcal{T}_{0,1}$ should be designed such that, when used in combination with \mathcal{U}_0 and $\mathcal{T}_{1,0}$ to define (6.2), each solution x to the hybrid closed-loop algorithm \mathcal{H} jumps no more than twice, and the set of interest is guaranteed to be at least weakly forward invariant.

To ensure that the hybrid closed-loop system \mathcal{H} in (6.1), with C and D defined via (6.2), is well-posed, and to ensure that every maximal solution to the hybrid closed-loop system \mathcal{H} is complete, we impose the following assumptions on the set $\mathcal{T}_{0,1}$ and the static state-feedback laws $\kappa_q(h_q(z, \tau))$.

Assumption 6.2.1 (Closed sets and continuous static state-feedback laws).

(C1) The set $\mathcal{T}_{0,1}$ is the closed complement of \mathcal{U}_0 , namely,

$$\mathcal{T}_{0,1} := \overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0}; \quad (6.6)$$

(C2) The map $z \mapsto F_P(z, \kappa_q(h_q(z, \tau)))$, in (3.1) with $u = \kappa(h(z, \tau), \tau)$, is Lipschitz continuous with constant $M_\kappa > 0$, namely,

$$|F_P(z, \kappa_q(h_q(z, \tau), \tau)) - F_P(v, \kappa_q(h_q(v, \tau), \tau))| \leq M_\kappa |z - v| \quad (6.7)$$

for all $z, v \in \mathbb{R}^{2n}$ and all $\tau \in \mathbb{R}_{\geq 0}$.

Remark 6.2.2. Whereas C is closed by construction, item (C1) is needed to ensure that the set D is closed. Additionally, (6.6) item (C1) is needed to ensure that nontrivial solutions to \mathcal{H} in (6.1) exist from any initial point in $\mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$. Item (C2) is used to ensure that the map $x \mapsto F(x)$ is continuous. The closure of C and D and the continuity of F and G are required for \mathcal{H} to be well-posed which, in turn, leads to robustness of the asymptotic stability our framework guarantees, as stated in results to follow. Additionally, item (C2) is an assumption commonly used in nonlinear analysis to ensure that the closed-loop systems \mathcal{H}_q in (6.3) resulting from the individual optimization algorithms do not have solutions that escape in finite time, which is used to guarantee the existence of solutions to \mathcal{H}_q [77, Theorem 3.2].

6.2.3 Basic Properties of \mathcal{H}

Under Assumptions 3.1.8, 3.2.4, and 6.2.1, the hybrid closed-loop system \mathcal{H} in (6.1), satisfies the hybrid basic conditions, listed in Definition 2.1.1, as demonstrated in the following lemma.

Lemma 6.2.3. (*Well-posedness of \mathcal{H}*) *Let L satisfy Assumptions 3.1.8 and 3.2.4. Let the set $\mathcal{T}_{0,1}$ satisfy item (C1) of Assumption 6.2.1, and let the map $z \mapsto F_P(z, \kappa_q(h_q(z, \tau)), \tau)$ satisfy item (C2) of Assumption 6.2.1. Let the sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$ be defined via (6.4) and (6.5), respectively. Then, the hybrid closed-loop system \mathcal{H} in (6.1) satisfies the hybrid basic conditions, as listed in Definition 2.1.1.*

Proof. The sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$ are closed by construction via Assumptions 3.1.8 and 3.2.4, and $\mathcal{T}_{0,1}$ is closed by item (C1) of Assumption 6.2.1. Therefore, the sets D_0 , D_1 , C_0 , and C_1 are closed. Since D and C are finite unions of finite and closed sets, then D and C are closed.

By construction the map $x \mapsto F(x)$ is continuous since, by item (C2), $z \mapsto F_P(z, \kappa_q(h_q(z, \tau)), \tau)$ is Lipschitz continuous. The map G satisfies item (A3) by construction. \square

In Theorem 6.2.9 we show that \mathcal{H} has a compact pre-asymptotically stable set. In light of this property, Lemma 6.2.3 is key as it leads to pre-asymptotic stability that is robust to small perturbations [21, Theorem 7.21]. In the case of gradient-based algorithms, for instance, such perturbations can take the form of small noise in measurements of the gradient.

To ensure that each maximal solution to the hybrid closed-loop system \mathcal{H} is bounded, we make the following assumption.

Assumption 6.2.4 (Bounded local algorithm). *Each maximal solution $(t, j) \mapsto (z(t, j), \tau(t, j))$ to \mathcal{H}_0 is bounded.*

Remark 6.2.5. *For specific optimization algorithms, some examples of assumptions under which \mathcal{H}_0 is bounded are as follows. When \mathcal{H}_0 is the heavy ball algorithm in (1.1), then boundedness is established when L is \mathcal{C}^1 , nonstrongly convex,*

has a unique minimizer by Assumption 3.1.8, has quadratic growth away from z_1^* by Assumption 3.2.4, and ∇L is Lipschitz continuous by Assumption 3.1.3; see Proposition 3.1.10. When \mathcal{H}_0 is Nesterov's algorithm in (1.2), then boundedness is established when L is C^2 and strongly convex by Assumption 3.1.1 and ∇L is Lipschitz continuous by Assumption 3.1.3; see Proposition 3.1.5. Boundedness can not be established for Nesterov's algorithm in (1.5), however, since the set \mathcal{A}_1 in (3.64) for such an algorithm is not compact.

When Assumptions 3.1.8, 3.2.4, 6.2.1, and 6.2.4 hold, then each maximal solution to \mathcal{H} is complete and bounded, as stated in the following lemma. Such a property is useful since it guarantees that nontrivial solutions to \mathcal{H} exist from each initial point in $C \cup D$, and that such solutions do not escape $C \cup D$. When every maximal solution is complete, then uniform global pre-asymptotic stability² of the set \mathcal{A} becomes uniform global asymptotic stability. The following lemma also states that $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$ and $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$. Such a property ensures that nontrivial solutions to \mathcal{H} , which exist from each initial point in $C \cup D$, also exist from any initial point in $\mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$.

Proposition 6.2.6. *(Existence of solutions to \mathcal{H}) Let L satisfy Assumptions 3.1.8, 3.2.4, and 6.2.1. Let \mathcal{H}_0 satisfy Assumption 6.2.4. Let the set $\mathcal{T}_{0,1}$ satisfy item (C1) of Assumption 6.2.1 and let the map $z \mapsto F_P(z, \kappa_q(h_q(z, \tau), \tau))$, for each $q \in Q$ and each $\tau \in \mathbb{R}$, satisfy item (C2) of Assumption 6.2.1. Furthermore, let \mathcal{U}_0 and $\mathcal{T}_{1,0}$ be defined via (6.4) and (6.5), respectively. Then, $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$, $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$, and each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j), \tau(t, j))$ to \mathcal{H} in (6.1) is bounded and complete.*

Proof. Since Assumptions 3.1.8 and 6.2.1 hold, then \mathcal{H} satisfies the hybrid basic

²Uniform global pre-asymptotic stability indicates the possibility of a maximal solution that is not complete, even though it may be bounded.

conditions by Lemma 6.2.3. Since L is \mathcal{C}^1 , nonstrongly convex, has a single minimizer by Assumption 3.1.8, and has quadratic growth away from z_1^* by Assumption 3.2.4, since \mathcal{U}_0 is defined via (6.4), and since $\mathcal{T}_{0,1}$ is the closed complement of \mathcal{U}_0 by item (C1) of Assumption 6.2.1, then $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$. (5.12), and since by the definitions of C_1 and D_1 in (6.2), C_1 is the closed complement of D_1 , then $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$.

Due to the definitions of C_0 , D_0 , C_1 , and D_1 in (6.2), \mathcal{U}_0 in (6.4), and $\mathcal{T}_{1,0}$ in (6.5), and due to $\mathcal{T}_{0,1}$ being the closed complement of \mathcal{U}_0 by item (C1) of Assumption 6.2.1, then $C \setminus D$ is equal to $\text{int}(C)$. Hence, for each point $x \in C \setminus D$, the tangent cone to C at x is

$$T_C(x) := \begin{cases} \mathbb{R}^{2n} \times \{0\} \times \{0\} & \text{if } x \in C_0 \setminus D_0, \\ \mathbb{R}^{2n} \times \{1\} \times \mathbb{R}_{\geq 0} & \text{if } x \in C_1 \setminus D_1. \end{cases} \quad (6.8)$$

Therefore, $F(x) \cap T_C(x) \neq \emptyset$, satisfying (VC) of Proposition A.1.1 for each point $x \in C \setminus D$, and nontrivial solutions exist for every initial point in $(C_0 \cup C_1) \cup (D_0 \cup D_1)$, where $\Pi(C_0) \cup \Pi(D_0) = \mathbb{R}^{2n}$ and $\Pi(C_1) \cup \Pi(D_1) = \mathbb{R}^{2n}$. To prove that item (c) of Proposition A.1.1 does not hold, we need to show that $G(D) \subset C \cup D$. With D defined in (6.2),

$$G(D) = (\mathcal{T}_{0,1} \times \{1\} \times \{0\}) \cup (\mathcal{T}_{1,0} \times \{0\} \times \{0\}) \quad (6.9)$$

Notice that $\mathcal{T}_{1,0} \times \{0\} \times \{0\} \subset C_0$ and $\mathcal{T}_{0,1} \times \{1\} \times \{0\} \subset C_1$. Therefore, $G(D) \subset C$; hence $G(D) \subset C \cup D$. Therefore, item (c) of Proposition A.1.1 does not hold. Then it remains to prove that item (b) does happen.

Since by item (C2), $z \mapsto F_P(z, \kappa_q(h_q(z, \tau), \tau))$ is Lipschitz continuous, and since the solution component τ increases linearly, then by [77, Theorem 3.2], \mathcal{H}_q ,

defined via (6.3), has no finite time escape from $\mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}$. Therefore, each maximal solution to \mathcal{H}_q is complete and unique. Therefore, this means $\dot{x} = F(x)$ has no finite time escape from C for \mathcal{H} , as q does not change in C and as τ is bounded in C , namely, τ – which is always reset to 0 in D – increases linearly in C_1 and remains at 0 in C_0 . Moreover, since each maximal solution $(t, j) \mapsto (z(t, j), \tau(t, j))$ to \mathcal{H}_0 is bounded, by Assumption 6.2.4, then each maximal solution x to \mathcal{H} is also bounded. Therefore, there is no finite time escape from $C \cup D$, for solutions x to \mathcal{H} . Therefore, item (b) from Proposition A.1.1 does not hold. This means only item (a) is true, and every maximal solution x to \mathcal{H} is bounded and complete. \square

To establish uniform global asymptotic stability of the set of interest for the hybrid closed-loop algorithm \mathcal{H} , we impose the following assumptions on the closed-loop algorithms \mathcal{H}_q in (6.3). We also make assumptions about the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$, to ensure that each solution x to the hybrid closed-loop algorithm \mathcal{H} jumps no more than twice, and ensure that the set of interest is guaranteed to be at least weakly forward invariant for \mathcal{H} .

Assumption 6.2.7 (Assumptions on asymptotic stability and attractivity). *Given the set*

$$\mathcal{A}_1 := \{z_1^*\} \times \{0\} \times \mathbb{R}_{\geq 0} \in \mathbb{R}^{2n} \times \mathbb{R}_{\geq 0}, \quad (6.10)$$

given a plant defined via (3.1) with $y = h_q(z)$, given $\tilde{c}_0 > 0$ and $d_0 > 0$ defining the closed set $\mathcal{U}_0 \in \mathbb{R}^{2n}$ via (6.4), the interior of which contains an open neighborhood of \mathcal{A}_1 in (6.10), and given $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ and $d_{1,0} \in (0, d_0)$ defining the closed set $\mathcal{T}_{1,0} \subset \mathcal{U}_0$ via (6.5), the following conditions hold:

(UC1) *The closed-loop algorithm \mathcal{H}_1 , resulting from κ_1 , has the set \mathcal{A}_1 uniformly globally attractive, namely, each maximal solution $\chi := (z, \tau)$ to \mathcal{H}_1 , given by (6.3) with $u = \kappa_1(h_1(z, \tau), \tau)$, is complete and for each $\varepsilon > 0$ and $r > 0$*

there exists $T > 0$ such that, for any solution χ to \mathcal{H}_1 with $|\chi(0)|_{\mathcal{A}_1} \leq r$, $t \in \text{dom } \chi$ and $t \geq T$ imply $|\chi(t)|_{\mathcal{A}_1} \leq \varepsilon$;

(UC2) There exist positive constants $\tilde{c}_1 \in (0, \tilde{c}_{1,0})$ and $d_1 \in (0, d_{1,0})$ such that the closed set $\mathcal{T}_{1,0}$ satisfies

$$(z_1^* + \tilde{c}_1 \mathbb{B}) \times (\{0\} + d_1 \mathbb{B}) \subset \mathcal{T}_{1,0} \quad (6.11)$$

and each solution to \mathcal{H}_0 in (6.3) with initial condition in $\mathcal{T}_{1,0} \times \mathbb{R}_{\geq 0}$, resulting from applying κ_0 , remains in \mathcal{U}_0 ;

(UC3) The closed-loop algorithm \mathcal{H}_0 , resulting from κ_0 , has a set \mathcal{A}_1 uniformly globally asymptotically stable, namely, each maximal solution χ to \mathcal{H}_0 , given by (3.1) with $u = \kappa_0(h_0(z, \tau), \tau)$, is complete, \mathcal{A}_1 is uniformly globally attractive for \mathcal{H}_0 , and there exists a class- \mathcal{K}_∞ function α such that any solution χ to \mathcal{H}_0 satisfies $|\chi(t)|_{\mathcal{A}_1} \leq \alpha(|\chi(0)|_{\mathcal{A}_1})$ for all $t \in \text{dom } \chi$.

(UC4) When $z \in \mathcal{T}_{0,1}$, where $\mathcal{T}_{0,1}$ comes from item (C1) of Assumption 6.2.1, and κ_0 is currently being used ($q = 0$), the hybrid closed-loop algorithm \mathcal{H} assigns u to κ_1 .

Remark 6.2.8. Items (UC1) and (UC3) of Assumption 6.2.7 ensure that \mathcal{H}_1 and \mathcal{H}_0 have the desired attractivity and stability properties, respectively, to establish the stability of the hybrid closed-loop algorithm \mathcal{H} . Such assumptions are reasonable in light of the numerous results in the literature for gradient-based optimization algorithms, cited in Sections 1.2.1 and 1.3.1. Furthermore, the conditions in Assumption 6.2.7 are similar to conditions imposed in [22] for general uniting control algorithms. Item (UC1) ensures that solutions (z, τ) starting with z in \mathcal{U}_0 , with $q = 0$ and $\tau \in \mathbb{R}_{\geq 0}$, stay in \mathcal{U}_0 and converge to \mathcal{A}_1 in (6.10) under the affect

of κ_0 . Item (UC4) guarantees that solutions starting with the state $z \in T_{0,1}$ and with $q = 0$ triggers a jump resetting q to 1. Item (UC3) guarantees that, after such a jump, z reaches $\mathcal{T}_{1,0}$ in finite time with κ_1 applied. Item (UC2) ensures that solutions from $\mathcal{T}_{1,0}$ under the effect of κ_0 cannot reach the boundary of \mathcal{U}_0 .

The following theorem establishes that the hybrid closed-loop system \mathcal{H} in (6.1)-(6.2) has the set

$$\mathcal{A} := \left\{ z \in \mathbb{R}^{2n} : \nabla L(z_1) = z_2 = 0 \right\} \times \{0\} \times \{0\} = \{z_1^*\} \times \{0\} \times \{0\} \times \{0\} \quad (6.12)$$

uniformly globally asymptotically stable. Recall that the state $x := (z, q, \tau) \in \mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$. In light of this, the first component of \mathcal{A} , namely, $\{z_1^*\}$, is the minimizer of L . The second component of \mathcal{A} , namely, $\{0\}$, reflects the fact that we need the velocity state z_2 to equal zero in \mathcal{A} so that solutions are not pushed out of such a set. The third component in \mathcal{A} , namely, $\{0\}$, is due to the logic state ending with the value $q = 0$, namely using κ_0 as the state z reaches the set of minimizers of L . The last component in \mathcal{A} is due to τ being set to, and then staying at, zero when the supervisor switches to κ_0 .

Theorem 6.2.9. *(Uniform global asymptotic stability of \mathcal{A} for \mathcal{H}) Let L satisfy Assumptions 3.1.8 3.2.4. Let the map $z \mapsto F_P(z, \kappa_q(h_q(z, \tau), \tau))$, for each $q \in Q$ and each $\tau \in \mathbb{R}$, in (3.1) satisfy item (C2) of Assumption 6.2.1. Let the set $\mathcal{T}_{0,1}$ satisfy item (C1) of Assumption 6.2.1. Let \mathcal{H}_0 satisfy Assumption 6.2.4. Let the closed-loop optimization algorithms \mathcal{H}_q in (6.3) and the sets \mathcal{U}_0 , $\mathcal{T}_{1,0}$, and $\mathcal{T}_{0,1}$ satisfy Assumption 6.2.7. Additionally, let $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$, $d_{1,0} \in (0, d_0)$, $\tilde{c}_1 \in (0, \tilde{c}_{1,0})$, and $d_1 \in (0, d_{1,0})$. Then, the set \mathcal{A} , defined via (6.12), is uniformly globally asymptotically stable for \mathcal{H} given in (6.1)-(6.2).*

Proof. The hybrid closed-loop algorithm \mathcal{H} satisfies the hybrid basic conditions

by Lemma 6.2.3, satisfying the first assumption of Theorem A.1.3. Furthermore, each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j), \tau(t, j))$ to \mathcal{H} for (6.1)-(6.2) is complete and bounded by Proposition 6.2.6. Since by Assumption 3.1.8, L has a unique minimizer z_1^* , then \mathcal{A} , defined via (6.12), is compact by construction, and $\mathcal{U} = \mathbb{R}^{2n} \times Q \times \mathbb{R}_{\geq 0}$ contains a nonzero open neighborhood of \mathcal{A} , satisfying the second assumption of Theorem A.1.3.

To prove attractivity of \mathcal{A} , we proceed by contradiction. Suppose there exists a complete solution x to \mathcal{H} such that $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} \neq 0$. Since Proposition 6.2.6 guarantees completeness of maximal solutions, we have the following cases:

- a) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_1 \setminus D_1$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$;
- b) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in C_0 \setminus (\mathcal{A} \cup D_0)$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$;
- c) There exists $(t', j') \in \text{dom } x$ such that $x(t, j) \in D$ for all $(t, j) \in \text{dom } x, t + j \geq t' + j'$.

Case a) contradicts the fact that, by item (UC1) of Assumption 6.2.7, the set \mathcal{A}_1 , defined via (6.10), is uniformly globally attractive for \mathcal{H}_1 . Such uniform global attractivity of \mathcal{A} , guaranteed by item (UC1) of Assumption 6.2.7, implies that the state z reaches $(\{z_1^*\} + \tilde{c}_1 \mathbb{B}) \times (\{0\} + d_1 \mathbb{B}) \subset \mathcal{T}_{1,0}$ in item (UC2) at some finite flow time $t \geq T$ or as t approaches ∞ . In turn, due to the construction of C_1 and D_1 in (6.2), with $\mathcal{T}_{1,0}$ defined via (5.12), $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$, and $d_{1,0} \in (0, d_0)$, the solution x must reach D_1 at some $(t, j) \in \text{dom } x, t + j \geq t' + j'$. Therefore, case a) does not happen.

Case b) contradicts the fact that, by item (UC3) of Assumption 6.2.7, \mathcal{A}_1 is uniformly globally asymptotically stable for \mathcal{H}_0 . In fact, $\lim_{t+j \rightarrow \infty} |x(t, j)|_{\mathcal{A}} = 0$, and

since $\mathcal{A} \subset C_0$, case b) does not happen.

Case c) contradicts the fact that, due to $\tilde{c}_{1,0} \in (0, \tilde{c}_0)$ and $d_{1,0} \in (0, d_0)$, due to the construction of $\mathcal{T}_{1,0}$ in (5.12), and due to $\mathcal{T}_{0,1} = \overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0}$ by item (C1) of Assumption 6.2.1, $\mathcal{T}_{1,0} \cap \mathcal{T}_{0,1} = \emptyset$ and hence we have

$$\begin{aligned} G(D) \cap D &:= ((\mathcal{T}_{0,1} \times \{1\} \times \{0\}) \cup (\mathcal{T}_{1,0} \times \{0\} \times \{0\})) \\ &\quad \cap ((\mathcal{T}_{0,1} \times \{0\} \times \{0\}) \cup (\mathcal{T}_{1,0} \times \{1\} \times \mathbb{R}_{\geq 0})) \\ &= \emptyset \end{aligned}$$

where $G(D)$ is defined via 6.9 and D is defined in 6.2. Therefore, case c) does not happen.

Therefore, cases a)-c) do not happen, and each maximal and complete solution $x = (z, q, \tau)$ to \mathcal{H} with $\tau(0, 0) = 0$ converges to \mathcal{A} . As a consequence, by the construction of C and D in (6.2), the uniform global attractivity of \mathcal{A}_1 (defined via (6.10)) for \mathcal{H}_1 in item (UC1) of Assumption 6.2.7, the uniform global asymptotic stability of \mathcal{A}_1 for \mathcal{H}_0 in item (UC3) of Assumption 6.2.7, and since each maximal solution to \mathcal{H} is complete by Proposition 6.2.6, the set \mathcal{A} is uniformly globally asymptotically stable for \mathcal{H} .

To show that each maximal and complete solution x to \mathcal{H} jumps no more than twice, we proceed by contradiction. Without loss of generality, suppose there exists a maximal and complete solution that jumps three times. We have the following possible cases:

- i) The solution first jumps at a point in D_0 , then jumps at a point in D_1 , and then jumps at a point in D_0 ; or
- ii) The solution first jumps at a point in D_1 , then jumps at a point in D_0 , and then jumps at a point in D_1 .

Case i) does not hold since, once the jump in D_1 occurs, the solution x is in $(\mathcal{T}_{1,0} \times \{0\} \times \{0\}) \subset C_0$. Due to the construction of $\mathcal{T}_{1,0}$ in (5.12) and due to $\mathcal{T}_{0,1} = \overline{\mathbb{R}^{2n} \setminus \mathcal{U}_0}$ by item (C1) of Assumption 6.2.1 such that $\mathcal{T}_{1,0} \cap \mathcal{T}_{0,1} = \emptyset$, as described in the contradiction of case c) above, and due to the uniform global asymptotic stability of \mathcal{A}_1 for \mathcal{H}_0 by item (UC3) of Assumption 6.2.7, the solution x will never return to D_0 . Therefore, case i) does not happen. Case ii) leads to a contradiction for the same reason, and in this case, once the first jump in D_1 occurs, no more jumps happen. Therefore, since cases i)-ii) do not happen, each maximal and complete solution x to \mathcal{H} with $\tau(0,0) = 0$ has no more than two jumps. \square

The framework defined in (6.1)-(6.2) allows for combinations of different methods, including Nesterov's algorithm, the heavy ball method, and the triple momentum method [38] [39], to name a few examples.

6.3 Examples for Applying the framework

In this section, we show how the framework in Section 6.2.1 applies to some of the algorithms proposed in Chapters 4 and 5.

6.3.1 Uniting Heavy Ball Algorithms

In Section 4.4, we proposed an algorithm uniting two heavy ball algorithms in (1.1) with properly designed parameters λ_q and γ_q , which uses measurements of ∇L . Namely, the hybrid closed-loop system \mathcal{H} is defined in (4.3) with C and D defined via (4.14), \mathcal{U}_0 defined in (4.20) (and is defined the same way in (6.4)), $\mathcal{T}_{1,0}$ defined via (4.25), and $\mathcal{T}_{0,1}$ defined in (4.30). Note that $\mathcal{T}_{1,0}$ in (4.25) is defined slightly differently than $\mathcal{T}_{1,0}$ in (6.5). To modify $\mathcal{T}_{1,0}$ in (4.25) to fit the design in

Section 6.2.2, use the same arguments and assumptions as in Section 4.4.2, but with $d_{1,0}$ defined as

$$d_{1,0} := 2c_{1,0} - 2\gamma_1 \left(\frac{\tilde{c}_{1,0}^2}{\alpha} \right) \in (0, d_0) \quad (6.13)$$

to yield the definition of $\mathcal{T}_{1,0}$ in (6.5). Note also that the state τ is omitted from the framework in this special case, since the heavy ball algorithm in (1.1) is time-invariant.

We can apply our results from section 6.2.3 to the hybrid closed-loop algorithm \mathcal{H} in Section 4.4, with $d_{1,0}$ defined via (6.13), as follows. Lemma 6.2.3 holds since

1. Due to L being \mathcal{C}^1 , nonstrongly convex, and having a single minimizer by Assumption 3.1.8, ∇L being Lipschitz continuous by 3.1.3, and L having quadratic growth away from z_1^* by 3.2.4, and with $d_{1,0}$ in (6.13), the sets \mathcal{U}_0 in (6.4), $\mathcal{T}_{1,0}$ in (6.5), and $\mathcal{T}_{0,1}$ in (4.30) are closed by construction and by the arguments in Sections 4.4.1, 4.4.2, and 4.4.3. Moreover, every $z \in \mathcal{U}_0$ is also in the V_0 sublevel set with level $c_0 = 0$, for which $\mathcal{T}_{0,1}$ in (4.30) is the closed complement, by the arguments in Sections 4.4.1 and 4.4.3; recall that V_0 is defined in (4.2). This means that $\mathcal{T}_{0,1}$ satisfies item (C1) of Assumption 6.2.1;
2. Due to L being \mathcal{C}^1 by Assumption 3.1.8 and due to ∇L being Lipschitz continuous by Assumption 3.1.3, the map $z \mapsto F_P(z, \kappa_q(h(z)))$, for \mathcal{H}_q in (4.4), is also Lipschitz continuous since F_P in (3.1) is a \mathcal{C}^1 , Lipschitz continuous function of κ_q in (4.1) and h in (3.72). Therefore, item (C2) of Assumption 6.2.1 also holds.

In addition, Proposition 6.2.6 holds for the hybrid closed-loop algorithm \mathcal{H} in Section 4.4, with $d_{1,0}$ defined via (6.13), since

1. Due to L being \mathcal{C}^1 , nonstrongly convex, and having a single minimizer by Assumption 3.1.8, ∇L being Lipschitz continuous by 3.1.3, and L having quadratic growth away from z_1^* by 3.2.4, then with $d_{1,0}$ in (6.13), by the arguments in Sections 4.4.1 and 4.4.2, the sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$ are defined in (6.4) and (6.5), respectively;
2. Moreover, due to L being \mathcal{C}^1 , nonstrongly convex, having a single minimizer by Assumptions 3.1.8, and having a Lipschitz continuous gradient by 3.1.3, then by the arguments in Sections 4.4.1 and 4.4.3, $\mathcal{T}_{0,1}$ in (4.30) satisfies item (C1) of Assumption 6.2.1;
3. Due to L being \mathcal{C}^1 , nonstrongly convex, having a single minimizer by Assumptions 3.1.8, and having a Lipschitz continuous gradient by 3.1.3, the map $z \mapsto F_P(z, \kappa_q(h(z)))$, for \mathcal{H}_q in (4.4), is also Lipschitz continuous since F_P in (3.1) is a \mathcal{C}^1 function of κ_q and h . Therefore, item (C2) of Assumption 6.2.1 also holds;
4. Since by Proposition 3.2.7, each maximal solution to \mathcal{H}_0 in (4.4) is bounded, then Assumption 6.2.4 holds.

Finally, Theorem 6.2.9 holds for the hybrid closed-loop algorithm \mathcal{H} in Section 4.4, with $d_{1,0}$ defined via (6.13), since

1. By Lemma 6.2.3, \mathcal{H} is well-posed;
2. By Proposition 6.2.6, each maximal solution to \mathcal{H} is complete and bounded;
3. By Proposition 3.2.8, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_1 in (4.4), resulting from κ_1 in (4.1). This is a stronger assumption than item (UC1) of Assumption 6.2.7;

4. By Proposition 3.2.8, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_0 in (4.4), resulting from κ_0 in (4.1). Therefore, item (UC3) of Assumption 6.2.7 is satisfied;
5. By the arguments in Section 4.4.2, with $d_{1,0}$ in (6.13), every $z \in \mathcal{T}_{1,0}$ is in a $c_{1,0}$ -sublevel set of V_1 , which is contained in the interior of \mathcal{U}_0 ; recall that $\mathcal{T}_{1,0}$ is defined in (6.5), \mathcal{U}_0 is defined in (6.4), and V_1 is defined in (4.2). This implies that there exist $\tilde{c}_1 \in (0, \tilde{c}_{1,0})$ and $d_1 \in (0, d_{1,0})$ such that (6.11) holds. Moreover, since item (UC3) holds, then each solution z to \mathcal{H}_0 with initial condition in $\mathcal{T}_{1,0}$ in (6.5), resulting from applying κ_0 in (4.1), remains in \mathcal{U}_0 . Therefore, item (UC2) holds.
6. Due to the construction of D_0 in (6.2) and the jump map G in (6.1b), item (UC4) of Assumption 6.2.7 is satisfied.

6.3.2 Uniting Nesterov's Method and the Heavy Ball Method for Strongly Convex L

In Section 5.1, we proposed an algorithm for strongly convex L uniting Nesterov's method in (1.2) globally and the heavy ball algorithm in (1.1) with large $\lambda > 0$ locally, which uses measurements of ∇L . Namely, the hybrid closed-loop system \mathcal{H} is defined in (4.3) with C and D defined via (4.14), \mathcal{U}_0 defined in (4.20) (and is defined the same way in (6.4)), $\mathcal{T}_{1,0}$ defined via (5.12) (and defined the same way in (6.5)), and $\mathcal{T}_{0,1}$ defined in (5.15). Note that the state τ is omitted from the framework in this special case, since Nesterov's algorithm in (1.2) is time-invariant.

We can apply our results from section 6.2.3 to the hybrid closed-loop algorithm \mathcal{H} in Section 5.1 as follows. Lemma 6.2.3 holds since

1. Due to L being \mathcal{C}^2 and strongly convex by Assumption³ 3.1.1 and ∇L being Lipschitz continuous by 3.1.3, the sets \mathcal{U}_0 in (6.4), $\mathcal{T}_{1,0}$ in (6.5), and $\mathcal{T}_{0,1}$ in (5.15) are closed by construction and by the arguments in Sections 5.1.3, 5.1.4, and 5.1.5. Moreover, every $z \in \mathcal{U}_0$ is also in the V_0 sublevel set with level $c_0 = 0$, for which $\mathcal{T}_{0,1}$ in (5.15) is the closed complement, by the arguments in Sections 5.1.3 and 5.1.5; recall that V_0 is defined in (3.80). This means that $\mathcal{T}_{0,1}$ satisfies item (C1) of Assumption 6.2.1;
2. Due to L being \mathcal{C}^2 by Assumption 3.1.1, due to ∇L being Lipschitz continuous by Assumption 3.1.3, and due to d and β being defined via (3.3), the map $z \mapsto F_P(z, \kappa_q(h(z)))$, for \mathcal{H}_q in (4.4), is also Lipschitz continuous since F_P in (3.1) is a \mathcal{C}^2 , Lipschitz continuous function of κ_q in (5.1) and h in (5.2). Therefore, item (C2) of Assumption 6.2.1 also holds.

In addition, Proposition 6.2.6 holds for the hybrid closed-loop algorithm \mathcal{H} in Section 5.1 since

1. Due to L being \mathcal{C}^2 and strongly convex by Assumption 3.1.1 – which implies L satisfies Definition 2.2.3 – and ∇L being Lipschitz continuous by 3.1.3, then by the arguments in Sections 5.1.3 and 5.1.4, the sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$ are defined in (6.4) and (6.5), respectively;
2. Moreover, due to L being \mathcal{C}^2 , strongly convex by Assumption 3.1.1, and having a Lipschitz continuous gradient by 3.1.3, then by the arguments in Sections 5.1.3 and 5.1.5, $\mathcal{T}_{0,1}$ in (5.15) satisfies item (C1) of Assumption 6.2.1;
3. Due to L being \mathcal{C}^2 and strongly convex, by Assumption 3.1.1, and having a Lipschitz continuous gradient by 3.1.3, the map $z \mapsto F_P(z, \kappa_q(h(z)))$, for

³This implies that L also satisfies Definition 2.2.3.

\mathcal{H}_q in (5.4), is also Lipschitz continuous since F_P in (3.1) is a \mathcal{C}^2 , Lipschitz continuous function of κ_q in (5.1) and h in (5.2). Therefore, item (C2) of Assumption 6.2.1 also holds;

4. Since by Proposition 3.2.7, each maximal solution to \mathcal{H}_0 in (5.4) is bounded, then Assumption 6.2.4 holds.

Finally, Theorem 6.2.9 holds for the hybrid closed-loop algorithm \mathcal{H} in Section 5.1 since

1. By Lemma 6.2.3, \mathcal{H} is well-posed;
2. By Proposition 6.2.6, each maximal solution to \mathcal{H} is complete and bounded;
3. By Theorem 3.1.7, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_1 in (5.4), resulting from κ_1 in (5.1b). This is a stronger assumption than item (UC1) of Assumption 6.2.7;
4. By Proposition 3.2.8, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_0 in (5.4), resulting from κ_0 in (5.1a). Therefore, item (UC3) of Assumption 6.2.7 is satisfied;
5. By the arguments in Section 5.1.4, every $z \in \mathcal{T}_{1,0}$ is in a $c_{1,0}$ -sublevel set of V_1 , which is contained in the interior of \mathcal{U}_0 ; recall that $\mathcal{T}_{1,0}$ is defined in (6.5), \mathcal{U}_0 is defined in (6.4), and V_1 is defined in (3.8). This implies that there exist $\tilde{c}_1 \in (0, \tilde{c}_{1,0})$ and $d_1 \in (0, d_{1,0})$ such that (6.11) holds. Moreover, since item (UC3) holds, then each solution z to \mathcal{H}_0 with initial condition in $\mathcal{T}_{1,0}$ in (6.5), resulting from applying κ_0 in (4.1), remains in \mathcal{U}_0 . Therefore, item (UC2) holds.
6. Due to the construction of D_0 in (6.2) and the jump map G in (6.1b), item (UC4) of Assumption 6.2.7 is satisfied.

6.3.3 Uniting Nesterov's Method and the Heavy Ball Method for Nonstrongly Convex L

In Section 5.1, we proposed an algorithm for nonstrongly convex L uniting Nesterov's method in (1.5) globally and the heavy ball algorithm in (1.1) with large $\lambda > 0$ locally, which uses measurements of ∇L . Namely, the hybrid closed-loop system \mathcal{H} is defined in (4.3) with C and D defined via (5.23) (defined the same way in (6.2)), \mathcal{U}_0 defined in (4.20) (and defined the same way in (6.4)), $\mathcal{T}_{1,0}$ defined via (5.12) (and defined the same way in (6.5)), and $\mathcal{T}_{0,1}$ defined in (5.15).

We can apply our results from section 6.2.3 to the hybrid closed-loop algorithm \mathcal{H} in Section 5.2 as follows. Lemma 6.2.3 holds since

1. Due to L being \mathcal{C}^1 , nonstrongly convex, and having a single minimizer by Assumption 3.1.8, ∇L being Lipschitz continuous by 3.1.3, and L having quadratic growth away from z_1^* by 3.2.4 the sets \mathcal{U}_0 in (6.4), $\mathcal{T}_{1,0}$ in (6.5), and $\mathcal{T}_{0,1}$ in (5.15) are closed by construction and by the arguments in Sections 5.2.3, 5.2.4, and 5.2.5. Moreover, every $z \in \mathcal{U}_0$ is also in the V_0 sublevel set with level $c_0 = 0$, for which $\mathcal{T}_{0,1}$ in (5.15) is the closed complement, by the arguments in Sections 4.4.1 and 4.4.3; recall that V_0 is defined in (3.80). This means that $\mathcal{T}_{0,1}$ satisfies item (C1) of Assumption 6.2.1;
2. Since \bar{d} and $\bar{\beta}$, defined via (3.35), are continuous, since L is \mathcal{C}^1 by Assumption 3.1.8, and since ∇L is Lipschitz continuous by Assumption 3.1.3, then h_q in (5.21), κ_0 in (5.1a), and κ_1 in (5.20) are continuous. In turn, the map $z \mapsto F_P(z, \kappa_q(h_q(z, \tau), \tau))$ is also continuous since F_P in (3.1) is a \mathcal{C}^1 , Lipschitz continuous function of κ_q and h_q . Therefore, item (C2) of Assumption 6.2.1 also holds.

In addition, Proposition 6.2.6 holds for the hybrid closed-loop algorithm \mathcal{H} in

Section 5.2, since

1. Due to L being \mathcal{C}^1 , nonstrongly convex, and having a single minimizer by Assumption 3.1.8, ∇L being Lipschitz continuous by 3.1.3, and L having quadratic growth away from z_1^* by 3.2.4, then by the arguments in Sections 5.2.3 and 5.2.4, the sets \mathcal{U}_0 and $\mathcal{T}_{1,0}$ are defined in (6.4) and (6.5), respectively;
2. Moreover, due to L being \mathcal{C}^1 , nonstrongly convex, having a single minimizer by Assumptions 3.1.8, and having a Lipschitz continuous gradient by 3.1.3, then by the arguments in Sections 5.2.3 and 5.2.5, $\mathcal{T}_{0,1}$ in (5.15) satisfies item (C1) of Assumption 6.2.1;
3. Since \bar{d} and $\bar{\beta}$, defined via (3.35), are continuous, since L is \mathcal{C}^1 by Assumption 3.1.8, and since ∇L is Lipschitz continuous by Assumption 3.1.3, then h_q in (5.21), κ_0 in (5.1a), and κ_1 in (5.20) are continuous. In turn, the map $z \mapsto F_P(z, \kappa_q(h_q(z, \tau), \tau))$ is also continuous since F_P in (3.1) is a \mathcal{C}^1 , Lipschitz continuous function of κ_q and h_q . Therefore, item (C2) of Assumption 6.2.1 also holds;
4. Since by Proposition 3.2.7, each maximal solution to \mathcal{H}_0 in (4.4) is bounded, then Assumption 6.2.4 holds.

Finally, Theorem 6.2.9 holds for the hybrid closed-loop algorithm \mathcal{H} in Section 5.1 since

1. By Lemma 6.2.3, \mathcal{H} is well-posed;
2. By Proposition 6.2.6, each maximal solution to \mathcal{H} is complete and bounded;
3. By Proposition 3.1.13, \mathcal{A}_1 in (6.10) is uniformly globally asymptotically stable for \mathcal{H}_1 in (5.25), resulting from κ_1 in (5.20). This is a stronger assumption

than item (UC1) of Assumption 6.2.7;

4. By Proposition 3.2.8, $\{z_1^*\} \times \{0\}$ is uniformly globally asymptotically stable for \mathcal{H}_0 in (5.24), resulting from κ_0 in (5.1a). Therefore, item (UC3) of Assumption 6.2.7 is satisfied;
5. By the arguments in Section 5.2.4, every $z \in \mathcal{T}_{1,0}$ is in a $c_{1,0}$ -sublevel set of V_1 , which is contained in the interior of \mathcal{U}_0 ; recall that $\mathcal{T}_{1,0}$ is defined in (6.5), \mathcal{U}_0 is defined in (6.4), and V_1 is defined in (3.38). This implies that there exist $\tilde{c}_1 \in (0, \tilde{c}_{1,0})$ and $d_1 \in (0, d_{1,0})$ such that (6.11) holds. Moreover, since item (UC3) holds, then each solution z to \mathcal{H}_0 with initial condition in $\mathcal{T}_{1,0}$ in (6.5), resulting from applying κ_0 in (5.1a), remains in \mathcal{U}_0 . Therefore, item (UC2) holds.
6. Due to the construction of D_0 in (6.2) and the jump map G in (6.1b), item (UC4) of Assumption 6.2.7 is satisfied.

6.4 Uniting Other Gradient Algorithms

In Section 6.3, we illustrated the framework in Section 6.2.1 by applying it to specific cases involving different combinations of the heavy ball algorithm and Nesterov's algorithm, for κ_0 and κ_1 . Other gradient-based algorithms, however, could be used as κ_0 and κ_1 for the general framework.

One example includes the triple momentum method. The triple momentum method was first proposed in [38] as a discrete-time accelerated gradient method. A characterization of the continuous-time, high-resolution dynamical system, derived in [39], is

$$\ddot{\xi} + 2\sqrt{\eta(\gamma, \lambda)}\dot{\xi} + \left(1 + \sqrt{\eta(\gamma, \lambda)}\gamma\right)\nabla L(w) = 0 \quad (6.14a)$$

$$w := \xi + \sqrt{\gamma}\sigma\dot{\xi} \quad (6.14b)$$

$$y := \xi + \sqrt{\gamma}\delta\dot{\xi} \quad (6.14c)$$

where $y \in \mathbb{R}^n$ is the output, where the gradient is applied to $w \in \mathbb{R}^n$, where $\gamma > 0$, $\lambda > 0$, $\sigma > 0$, and $\delta > 0$ are tunable parameters, and where $\eta(\gamma, \lambda)$ is

$$\eta(\gamma, \lambda) := \left(\frac{1 - \lambda}{\sqrt{\gamma}(1 + \lambda)} \right)^2 \in (0, M]. \quad (6.15)$$

where $M > 0$ is the Lipschitz constant of ∇L . The authors in [38] and [39] also give an ideal tuning of the parameters as follows:

$$(\gamma, \lambda, \sigma, \delta) := \left(\frac{1 + \rho}{M}, \frac{\rho^2}{2 - \rho}, \frac{\rho^2}{(1 + \rho)(2 - \rho)}, \frac{\rho^2}{1 - \rho^2} \right) \quad (6.16)$$

where

$$\rho := 1 - \frac{1}{\kappa} \quad (6.17)$$

where $\kappa := \frac{M}{\mu}$, $\mu > 0$, is the condition number of L . The authors in [39] characterize the convergence rate for (6.14) to be exponential, both for the general parameters and for the optimal tuning in (6.16), and numerically found (6.14) to converge more quickly than Nesterov's algorithm, for given values of the condition number κ . Since (6.14) has an exponential convergence rate, it would be ideal for use as the global optimization algorithm κ_1 in the general framework, defined in Section 6.2.1, with heavy ball with large λ as κ_0 .

Another example includes classic gradient descent, which has the ODE

$$\dot{\xi} + \gamma \nabla L(\xi) = 0 \quad (6.18)$$

where $\gamma > 0$ is tunable, and which is commonly known to have a convergence rate

of $\frac{1}{t}$. As gradient descent does not have a “velocity” term, it tends to converge slowly, without oscillations near the minimizer. Such behavior is similar to the behavior of heavy ball with large λ . Due to such behavior, gradient descent could be used as the local algorithm κ_0 in the general framework in 6.2.1, with either heavy ball with small λ , Nesterov’s algorithm, or the triple momentum method as κ_1 .

Chapter 7

Hybrid Accelerated Optimization for Nonconvexity

In this Chapter, we present a logic-based algorithm for Morse functions that uses the heavy ball algorithm when the state z is far from a critical point and that uses linear feedback when the state z is near a critical point, to push z away from such a critical point. For the algorithm in this chapter, impose Assumption 3.2.11 on the objective function L .

7.1 Problem Statement

The problem addressed in this Chapter is as follows.

Problem 7.1.1. *Given a continuously differentiable Morse objective function $L : \mathbb{R} \rightarrow \mathbb{R}$, which may have multiple isolated minimizers and maximizers, design an optimization algorithm that guarantees practical convergence to a local minimizer from all initial conditions – including local maximizers – using measurements of ∇L .*

We emphasize that, to solve Problem 7.1.1, the algorithm has no knowledge of the particular objective function L or of its critical points.

7.2 Design

In this section, we present a logic-based algorithm for Morse functions that uses the heavy ball algorithm when the state z is far from a critical point and that uses linear feedback when the state z is near a critical point, to push z away from such a critical point.

Our proposed algorithm has a state $z := (z_1, z_2) \in \mathbb{R}^2$, where z_1 represents the argument of L and z_2 represents the “velocity” variable. The state z remains unchanged at jumps, but updates during flows according to

$$\dot{z}_1 = z_2, \quad \dot{z}_2 = u \tag{7.1}$$

where u takes different forms depending on whether the state z is close to or far from a critical point. Our algorithm uses a logic variable, $q \in Q := \{0, 1\}$, to indicate when to push the state z_1 away from a critical point. The logic value $q = 0$ leads to the algorithm using the heavy ball method to converge to the neighborhood of a critical point, and $q = 1$ leads to the algorithm using linear feedback to push z_1 away from a critical point. In addition, our algorithm has a state σ to determine the magnitude and direction to push the state z_1 when close to a critical point. To trigger jumps, hysteresis parameters $0 < \varepsilon_1 < \varepsilon_2$ and $0 < \rho_1 < \rho_2$ are used. These parameters are small enough to ensure convergence to a neighborhood of a local minimum without overshooting to a neighboring maximum. The algorithm uses a parameter $\varsigma > 0$, to tune the speed of convergence.

A high-level description of the proposed algorithm is as follows. When the state

z is near a critical point with small velocity, as determined by $|\nabla L(z_1)| \leq \varepsilon_1$ and $|z_2| \leq \rho_1$, the algorithm resets the logic variable q to 1 and assigns u to σ . Then, z moves away from the critical point according to $u = \sigma$, where $\sigma := \varsigma \text{sign}(z_2)$. The feedback $\varsigma \text{sign}(z_2)$ causes the state z_2 to change linearly and z_1 to change quadratically, thus eventually pushing the state z away from a critical point. When the state z is far away from the critical point and the velocity is larger, as determined by $|\nabla L(z_1)| \geq \varepsilon_2$ and $|z_2| \geq \rho_2$, the algorithm resets the logic variable q to 0 and assigns u to

$$\kappa(h(z)) := -\lambda z_2 - \gamma \nabla L(z_1), \quad (7.2)$$

which is defined for all $z \in \mathbb{R}^2$, where $\lambda > 0$ represents friction, $\gamma > 0$ represents gravity, and h is given by

$$h(z) := \begin{bmatrix} z_2 \\ \nabla L(z_1) \end{bmatrix}. \quad (7.3)$$

The function h characterizes the measurements used by the algorithm. With the proposed logic, the state z converges a nearby local minimizer, contained in D_0 , with zero velocity via $u = \kappa(h(z))$. From such a point, although the state z is pushed to D_1 , the state z will again converge to nearby the same local minimizer as before, via $u = \kappa(h(z))$, and this process repeats for all time. The positive parameters $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ need to be properly tuned to keep z in a small neighborhood of a local minimizer. The complete algorithm is summarized in Algorithm 1.

The rest of this section is organized as follows. Section 7.3 introduces the hybrid system model for the proposed algorithm. Finally, Section 7.4 contains the main results, which reveal the nominal properties of the proposed algorithm.

Algorithm 5 Hybrid Algorithm for Morse Functions

```
1: Set  $q(0, 0)$  to 0, and set  $z(0, 0)$  and  $\sigma(0, 0)$  as initial conditions with arbitrary
   values.
2: while true do
3:   if  $|\nabla L(z_1)| \leq \varepsilon_1$  and  $|z_2| \leq \rho_1$  and  $q = 0$  then
4:     Update  $q$  to 1;
5:     Update  $\sigma$  to  $\varsigma \text{sign}(z_2)$  and assign  $u$  to  $\sigma$ ;
6:   else if  $|\nabla L(z_1)| \geq \varepsilon_2$  and  $|z_2| \geq \rho_2$  and  $q = 1$  then
7:     Update  $q$  to 0;
8:     Assign  $u$  to  $\kappa(h(z))$ , defined via (7.2).
9:   else
10:    Allow flows of (7.1) with  $u = \kappa(h(z))$  if  $q = 0$  and with  $u = \sigma$  if  $q = 1$ .
11:   end if
12: end while
```

7.3 Hybrid System Model of the Proposed Algorithm

The proposed algorithm is modeled as a hybrid system \mathcal{H} with parameter $\varsigma > 0$, state $x := (z, q, \sigma) \in \mathbb{R}^2 \times Q \times \{-\varsigma, \varsigma\}$, and data (C, F, D, G) defined as follows:

$$F(x) := \begin{bmatrix} z_2 \\ \tilde{\kappa}(x) \\ 0 \\ 0 \end{bmatrix} \quad x \in C := \overline{(\mathbb{R}^2 \times Q \times \{-\varsigma, \varsigma\})} \setminus D \quad (7.4a)$$

$$G(x) := \begin{bmatrix} z_1 \\ z_2 \\ 1 - q \\ \varsigma \text{sign}(z_2) \end{bmatrix} \quad x \in D := D_0 \cup D_1 \quad (7.4b)$$

where $\varsigma > 0$ is properly tuned, $\text{sign}(z_2)$ is defined as the set-valued map

$$\text{sign}(z_2) = \begin{cases} 1 & \text{if } z_2 > 0 \\ \{-1, 1\} & \text{if } z_2 = 0 \\ -1 & \text{if } z_2 < 0 \end{cases} \quad (7.5)$$

and $\tilde{\kappa}$ is defined as

$$\tilde{\kappa}(x) = \begin{cases} \kappa(h(z)) & \text{if } q = 0 \\ \sigma & \text{if } q = 1 \end{cases} \quad (7.6)$$

where $\kappa(h(z))$ is defined via (7.2). The sets D_0 , and D_1 are defined below. As was outlined above and in Algorithm 1, the algorithm jumps when the state z is near a critical point with small velocity, as determined by $|\nabla L(z_1)| \leq \varepsilon_1$ and $|z_2| \leq \rho_1$, when $q = 0$. The algorithm also jumps when the state z is far from a critical point with larger velocity, as determined by $|\nabla L(z_1)| \geq \varepsilon_2$ and $|z_2| \geq \rho_2$, when $q = 1$, and when $\sigma \in \{-\varsigma, \varsigma\}$. To this end, the sets D_0 and D_1 are defined as

$$D_0 := \{z \in \mathbb{R}^2 : |\nabla L(z_1)| \leq \varepsilon_1, |z_2| \leq \rho_1\} \times \{0\} \times \{-\varsigma, \varsigma\} \quad (7.7a)$$

$$D_1 := \{z \in \mathbb{R}^2 : |\nabla L(z_1)| \geq \varepsilon_2, |z_2| \geq \rho_2\} \times \{1\} \times \{-\varsigma, \varsigma\} \quad (7.7b)$$

where $\varepsilon_2 > \varepsilon_1 > 0$ and $\rho_2 > \rho_1 > 0$ are the inner and outer hysteresis bounds, used to determine whether the system is near a critical point – and needs to be pushed away from such a point using the feedback σ – or far enough away from a critical point to use the feedback $\kappa(h(z))$.

Remark 7.3.1. *Our approach to tuning ε_2 and ε_1 uses the minimum separation $d_0 > 0$ between critical points, from item (M3) of Assumption 3.2.11. If for a given z_1 , $|\nabla L(z_1)| \geq \varepsilon_2$, then the following relation can be derived: $0 < \varepsilon_1 < \varepsilon_2 <$*

$\min \{\nabla L(z_1) : z_1 \in \{z'_1 \in \mathbb{R} : \nabla^2 L(z'_1) = 0\}\}$. Since $\nabla^2 L(z'_1) = 0$ occurs midway between critical points¹, then such a tuning ensures that when the state z is near a local maximizer, it converges to the nearest local minimizer without overshooting to the next local maximizer. Such a tuning also ensures that if the state z is near a local minimizer, it stays near that same local minimizer. The function L , however, is not always known, and the hybrid closed-loop system in (7.4) assumes no knowledge of L . In practice, choosing $0 < \varepsilon_1 < \varepsilon_2$ small enough is sufficient.

7.4 Main Result

In this section, we show that the hybrid closed-loop system \mathcal{H} with data (C, F, D, G) defined in (7.4) has the set

$$\mathcal{A} := \mathcal{A}_{\text{I}_{\min}} \times \{0\} \times Q \times \{-\varsigma, \varsigma\} \quad (7.8)$$

practically globally attractive in the positive parameters $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$, with basin of attraction that has a z_1 component equal to \mathbb{R} . Practical global attractivity of \mathcal{A} means that, for each $\zeta > 0$ and for every solution x to \mathcal{H} , there exists $(t', j') \in \text{dom } x$ such that $|x(t, j)|_{\mathcal{A}} \leq \zeta$ for all $(t, j) \in \text{dom } x$ such that it is satisfying $t + j \geq t' + j'$.

Under item (M2) of Assumption 3.2.11, the hybrid closed-loop system \mathcal{H} , described in (7.4), is well-posed, as it meets the hybrid basic conditions.

Lemma 7.4.1. *(Well posedness of \mathcal{H}) Let L satisfy item (M2) of Assumption 3.2.11. Then, the hybrid closed-loop system \mathcal{H} in (7.4) satisfies the hybrid basic conditions in Definition 2.1.1.*

¹Note that such a point is not itself a critical point, as it is not a stationary point, since L is a Morse function.

Proof. Since $|\nabla L(z_1)|$ is continuous, due to L being \mathcal{C}^2 , by (M2), then the sets D_0 and D_1 are closed. Consequently, D is closed since it is composed of the union of closed sets. Since C is the closed complement of D , then C is also closed.

By construction, the single-valued map $x \mapsto F(x)$ is continuous, since L is \mathcal{C}^2 , which makes $\tilde{\kappa}$ in (7.6) continuous as a function of x – this follows since, for each $q \in Q$, $(z, a) \mapsto \tilde{\kappa}(z, q, a)$ is continuous on $\mathbb{R}^2 \times \{-\varsigma, \varsigma\}$. The map G satisfies (A3) by construction. \square

When Assumption 3.2.11 holds, every maximal solution to the hybrid closed-loop system \mathcal{H} is complete and bounded, as stated in the following lemma.

Lemma 7.4.2. *(Existence of solutions for \mathcal{H}) Let L satisfy items (M1)-(M4) of Assumption 3.2.11. Let $\lambda > 0$, $\gamma > 0$, and $\varsigma > 0$. Let $\tilde{\kappa}$ and sign be defined via (7.6) and (7.5), respectively. Then, each maximal solution $(t, j) \mapsto x(t, j) = (z(t, j), q(t, j), \sigma(t, j))$ to the closed-loop system \mathcal{H} in (7.4) is bounded and complete.*

Proof. Since item (M2) holds, then \mathcal{H} is well-posed by Lemma 7.4.1. Due to the definitions of C in (7.4) and D in (7.7), then $C \setminus D$ is equal to $f(C)$. Hence, for each point $x \in C \setminus D$, the tangent cone to C at x is $T_C(z)$ is $\mathbb{R}^2 \times Q \times \{-\varsigma, \varsigma\}$. Therefore, $F(x) \cap T_C(z) \neq \emptyset$, satisfying (VC) of Proposition A.1.1 for each point $x \in C \setminus D$, and nontrivial solutions exist for every initial point in $C \cup D$. To prove that item (c) of Proposition A.1.1 does not hold, we need to show that $G(D) \subset C \cup D$. Note that $G(D)$ is defined, for D via (7.7), as

$$G(D) := \left(\left\{ z \in \mathbb{R}^2 : |\nabla L(z_1)| \leq \varepsilon_1, |z_2| \leq \rho_1 \right\} \times \{1\} \times \{-\varsigma, \varsigma\} \right) \\ \cup \left(\left\{ z \in \mathbb{R}^2 : |\nabla L(z_1)| \geq \varepsilon_2, |z_2| \geq \rho_2 \right\} \times \{0\} \times \{-\varsigma, \varsigma\} \right).$$

Notice that, since C , as defined in (7.4), is the closed complement of D , then

$G(D) \subset C$ and this means also that $G(D) \subset C \cup D$. Therefore, item (c) of Proposition A.1.1 does not hold. It remains to be proved that item (b) of Proposition A.1.1 does not hold.

By Lemma 3.2.13, (3.74) has no finite time escape from $\mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, where $\mathcal{A}_{a_{\max}}$ is defined via (3.104). Moreover, since $\tilde{\kappa}(h(z)) = \sigma$ is linear, then $\dot{x} = F(x)$ has no finite time escape from C for \mathcal{H} in (7.4), as q and σ do not change in C . Therefore, there is no finite time escape from $C \cup D$ for solutions x to \mathcal{H} . Therefore, item (b) from Proposition A.1.1 does not hold. \square

The following lemma describes the behavior of solutions with $u = \sigma$, when the system state is in the set $\{z \in \mathbb{R}^2 : |\nabla L(z_1)| \leq \varepsilon_1, |z_2| \leq \rho_1\}$. For this lemma, consider the continuous-time system

$$\dot{z}_1 = z_2, \quad \dot{z}_2 \in \varsigma \text{sign}(z_2) \quad (7.9)$$

with $\varsigma > 0$ and $\text{sign}(z_2)$ defined via (7.5), and consider the sets

$$S_1 := \{z \in \mathbb{R}^2 : |\nabla L(z_1)| \leq \varepsilon_1, |z_2| \leq \rho_1\} \quad (7.10a)$$

$$S_2 := \{z \in \mathbb{R}^2 : |\nabla L(z_1)| \geq \varepsilon_2, |z_2| \geq \rho_2\} \quad (7.10b)$$

where $\varepsilon_1 \in (0, \varepsilon_2)$ and $\rho_1 \in (0, \rho_2)$.

Lemma 7.4.3. *Let L satisfy items (M1), (M3) and (M5) of Assumption 3.2.11.*

Given the system in (7.9), the sets in (7.10), $\varepsilon > 0$, $\varepsilon_2 > 0$, and $\rho_2 > 0$, where (ε_2, ρ_2) is sufficiently small defining S_2 , there exists $\delta \in (0, \min\{\frac{d_0}{2}, \varepsilon\})$, where $d_0 > 0$ comes from (M3), such that, for each $\varepsilon_1 \in (0, \varepsilon_2)$ and $\rho_1 \in (0, \rho_2)$ sufficiently small defining S_1 , there exists $T > 0$ such that, for all $z_\circ \in S_1$, each solution $t \mapsto z(t)$ to (7.9) from z_\circ satisfies $z(T) \in S_2$ and $\max\{|z_1(t)|_{\mathcal{A}_1}, |z_2(t)|\} \leq \delta$ for all $t \in [0, T]$.

Proof. By (M1) and (M3), L is a Morse function with a finite number of isolated critical points, and there exists a minimum distance $d_0 > 0$ between any two critical points.

Let $\delta_1 \in (0, \frac{d_0}{16}]$, $\delta_2 \in (\delta_1, \frac{d_0}{8}]$. Then, given $\varepsilon > 0$ and $\varepsilon_2 > 0$, let $\delta \in (0, \min\{\frac{d_0}{2}, \varepsilon\})$. Given $\varepsilon_2 > 0$ sufficiently small, we invoke (M5) to get that, for each $\varepsilon_1 \in (0, \varepsilon_2)$, $|\nabla L(z_{1_o})| \leq \varepsilon_1$ implies that $|z_{1_o}|_{\mathcal{A}_1} \leq \delta_1$. Then, we pick a solution $t \mapsto (z_1(t), z_2(t))$ to (7.9) such that $z_o \in S_1$. Since z_2 changes linearly and z_1 changes quadratically, then at some time $T > 0$ the state z arrives at S_2 . Such a time $T > 0$ must satisfy the property

$$\delta_2 + 2\delta_1 \geq |z_1(T) - z_{1_o}| > \delta_2 + \delta_1. \quad (7.11)$$

To find $T > 0$, we solve the following equation using the solutions to the differential equations in (7.9):

$$|z_1(T) - z_{1_o}| = \left| \frac{1}{2}T^2 + z_{2_o}T \right| = \delta_2 + \frac{4}{3}\delta_1. \quad (7.12)$$

Hence, since for all $z_o \in S_1$, $z_{2_o} = 0$, we obtain

$$T = \sqrt{2\delta_2 + \frac{8}{3}\delta_1}. \quad (7.13)$$

Then, given $\varepsilon_2 > 0$ and since previously we defined $\delta_2 \in (\delta_1, \frac{d_0}{8}]$, we invoke (M5) to get that $|\nabla L(z_1(T))| \leq \varepsilon_2$ implies that $|z_1(T)|_{\mathcal{A}_1} \leq \delta_2$.

Since z_2 changes linearly and z_1 changes quadratically with respect to time, and since δ_1 and δ_2 are made small by definition, we conclude that T in (7.12) is always finite.

Now, T in (7.13) implies $|z_1(T) - z_{1_o}| \in (\delta_2 + \delta_1, \delta_2 + 2\delta_1)$ which, in turn, by

(M5), implies that $|\nabla L(z_1(T))| > \varepsilon_2$. Furthermore, given $\rho_2 > 0$ sufficiently small, for any $t \in [0, T]$,

$$\begin{aligned} |z_1(t) - z_{1\circ}| &\leq \delta_2 + \frac{4}{3}\delta_1 + \rho_2 \sqrt{2\delta_2 + \frac{8}{3}\delta_1} \\ \implies |z_1|_{\mathcal{A}_1} &\leq \delta_2 + \frac{4}{3}\delta_1 + \rho_2 \sqrt{2\delta_2 + \frac{8}{3}\delta_1} =: \delta_1^*. \end{aligned} \quad (7.14)$$

Next, given $\rho_2 > 0$ sufficiently small, it is easy to show, via (7.9), that $|z_2(T)| \geq \rho_2$ and, for all $t \in [0, T]$,

$$|z_2(t)| \leq T + z_{2\circ} \leq \sqrt{2\delta_2 + \frac{8}{3}\delta_1} + \rho_2 =: \delta_2^*. \quad (7.15)$$

Then, substituting (7.14) and (7.15) into $\max\{|z_1(t)|_{\mathcal{A}_1}, |z_2(t)|\} \leq \delta$ yields

$$\delta := \max\{\delta_1^*, \delta_2^*\}. \quad (7.16)$$

As mentioned previously, δ_1 and δ_2 are made small by definition. Therefore, given $\varepsilon_2 > 0$ and $\rho_2 > 0$ sufficiently small, δ_1^* and δ_2^* are made sufficiently small. Hence, given $\varepsilon > 0$, we have $\delta \leq \varepsilon$ which finishes the proof. \square

The following result shows that the hybrid closed-loop system \mathcal{H} has the set \mathcal{A} in (7.8) practically globally attractive. To establish it, we employ the results from Theorem 3.2.14 and Lemma 7.4.3.

Theorem 7.4.4. *(Practical global attractivity of \mathcal{A} in (7.8) for \mathcal{H}) Let L satisfy Assumption 3.2.11. Let $\lambda > 0$, $\gamma > 0$, and $\varsigma > 0$. Consider the hybrid closed-loop system \mathcal{H} with data (C, F, D, G) defined in (7.4), with the functions $\tilde{\kappa}$ in (7.6) and sign in (7.5), and consider the positive parameters $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$. Then, the set \mathcal{A} in (7.8) is practically globally attractive for \mathcal{H} in the sufficiently small parameters $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$; that is, for each $\varepsilon_{\min} > 0$ sufficiently small, there exist*

parameters $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ with $\varepsilon_1 \in (0, \varepsilon_2)$ and $\rho_1 \in (0, \rho_2)$ such that, for each maximal solution x to \mathcal{H} , there exists $(t', j') \in \text{dom } x$ such that

$$|x(t, j)|_{\mathcal{A}} \leq \varepsilon_{\min} \quad \forall (t, j) \in \text{dom } x : t + j \geq t' + j' \quad (7.17)$$

Remark 7.4.5. *The size of $\varepsilon_{\min} > 0$ needs to be sufficiently small to keep the state z_1 in a small neighborhood of a local minimizer. To give some insight into its size, letting $\delta \in (0, \max\{\frac{d_0}{2}, \varepsilon\})$, where $d_0 > 0$ and where $\varepsilon > 0$ is sufficiently small as in (M5), then we need z such that $\max\{|z_1|_{\mathcal{A}_1}, |z_2|\} \leq \delta$. Moreover, since $q \in \{0, 1\}$ and $\sigma \in \{-\varsigma, \varsigma\}$ always holds, then we need $\varepsilon_{\min} \leq \sqrt{2\delta^2 + 1^2 + \varsigma^2}$ for practical global attractivity. Furthermore, we observe in simulation that solutions z to \mathcal{H} converge to a neighborhood of \mathcal{A} in the presence of small noise in measurements of the gradient.*

Proof. An outline of the proof is as follows:

1. We show that the hybrid closed-loop system \mathcal{H} is well posed and that each maximal solution to \mathcal{H} is bounded and complete;
2. We use a trajectory-based approach to show that each solution to the hybrid closed-loop system \mathcal{H} does the following:
 - Converges to D_0 , which contains a local minimizer, in finite time, due to Lemma 7.4.3;
 - Then, is pushed to D_1 in finite time by Lemma 7.4.3;
 - Then, converges to D_0 , in finite time, containing the same local minimizer.

This process repeats for all time, for all solutions;

3. Since the process in 3) happens for all solutions, then to establish practical global attractivity, we derive the smallest sublevel set of each basin of attraction \mathcal{U}_i , defined via (3.106)-(3.108), of V in (3.105) that contains

$$\widehat{\mathcal{U}}_0 := \left\{ z \in \mathbb{R}^2 : |\nabla L(z_1)| \leq \varepsilon_2, |z_2| \leq \rho_2 \right\} \quad (7.18)$$

Such a sublevel set – containing \mathcal{A} – of each basin of attraction \mathcal{U}_i of V represents the smallest sublevel set to which each solution converges and in which each solution remains for all time, giving practical attractivity.

For Step 1, By Lemma 7.4.1, the hybrid closed-loop system \mathcal{H} meets the hybrid basic conditions. In addition, by Lemma 7.4.2, each maximal solution of the hybrid closed-loop system \mathcal{H} is bounded and complete.

Next, for Step 2, We employ a trajectory-based approach, using the properties of the heavy ball algorithm and the properties of $\tilde{\kappa}(x) = \sigma$, to show practical global attractivity of the set \mathcal{A} in (7.8). The types of solutions possible for the hybrid closed-loop system, defined via (7.4), are as follows.

1. Solutions that start in the interior of C_0 ;
2. Solutions that start in the interior of C_1 ;
3. Solutions that start in the interior of D_0 ;
4. Solutions that start in the interior of D_1 ;
5. Solutions that start on the boundary of D_0 , namely, $|\nabla L(z_1(0,0))| = \varepsilon_1$, $|z_2(0,0)| = \rho_1$, $q(0,0) = 0$, and $\sigma \in \{-\varsigma, \varsigma\}$;
6. Solutions that start on the boundary of D_1 , namely, $|\nabla L(z_1(0,0))| = \varepsilon_2$, $|z_2(0,0)| = \rho_2$, $q(0,0) = 1$, and $\sigma \in \{-\varsigma, \varsigma\}$.

We start by picking a solution of type 1). For such a solution, $q(0,0) = 0$. Due to the construction of F and C , defined via (7.4a), respectively, such a solution will flow, using $u = \tilde{\kappa}(x) = \kappa(h(z))$, where κ is defined via (7.2) and h is defined in (7.3). By Theorem 3.2.14, where it was proved that $\mathcal{A}_{a_{\min}}$ in (3.103) is almost globally asymptotically stable with basin of attraction given by $\mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, where $\mathcal{A}_{a_{\max}}$ is defined in (3.104), for the heavy ball algorithm, defined via (3.74), the solution flows until $|\nabla L(z_1)| < \varepsilon_1$ and $|z_2| < \rho_1$. Note that when the state z arrives at the boundary of D_0 – namely, when $|\nabla L(z_1)| = \varepsilon_2$, $|z_2| = \rho_2$, $q = 1$, and $\sigma \in \{-\varsigma, \varsigma\}$ – the algorithm can either flow or jump, due to the fact that V , defined via (3.105), satisfies $V^\circ(z, F_P(z, \kappa(h(z)))) \leq 0$ for all $z \in \mathcal{U} := \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, as shown in (3.109). But due to the construction of D_0 , there exists some time $(t_1, 0)$ in the domain of the solution at which flow is no longer possible. To prove this, we show that $f(z) \cap T_C(z) = \emptyset$, where f is the heavy ball algorithm and $T_C(z)$ is \mathbb{R}^2 , is eventually empty, along solutions. At the boundary of D_0 , since $V^\circ(z, F_P(z, \kappa(h(z)))) \leq 0$, we can still have flow of $u = \kappa(h(z))$, so we have $f(z) \cap T_C(z) \neq \emptyset$. We denote the start of this time as $(0, 0)$. Furthermore, either when $|\nabla L(z_1)| = \varepsilon_1$ and $|z_2| < \rho_1$ or when $|\nabla L(z_1)| < \varepsilon_1$ and $|z_2| = \rho_1$, then we can still have flow, due to $V^\circ(z, F_P(z, \kappa(h(z)))) \leq 0$, so $f(z) \cap T_C(z) \neq \emptyset$. This happens during time $(t, 0)$, where $t < t_1$. Finally, when both $|\nabla L(z_1)| < \varepsilon_1$ and $|z_2| < \rho_1$, we can no longer flow, but can only jump, and we have $f(z) \cap T_C(z) = \emptyset$, which happens at time $(t_1, 0)$. Therefore, the solution will eventually jump when the state z is in D_0 which, due to V satisfying (3.109) for all $z \in \mathcal{U} := \mathbb{R}^2 \setminus \mathcal{A}_{a_{\max}}$, must contain a local minimizer instead of a local maximizer. When this jump occurs, the logic variable q is reset to 1, and u is assigned to $\varsigma \text{sign}(z_2)$, where sign is defined via (7.5), while z remains the same. Hence, the solution is in C_1 . Since, by Lemma 7.4.3, solutions to (7.9) flow from the set S_1 to S_2 in (7.10),

with $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ sufficiently small, in finite time $T = \sqrt{2\delta_2 + \frac{8}{3}\delta_1} > 0$, then due to the construction of D_0 and D_1 in (7.7), with $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ sufficiently small, solutions to the hybrid closed-loop system \mathcal{H} flow from D_0 to D_1 via C_1 using $u = \varsigma \text{sign}(z_2)$ in finite time $T = \sqrt{2\delta_2 + \frac{8}{3}\delta_1} > 0$. As a consequence of Lemma 7.4.3 and due to items (M3) and (M5), the state z will not overshoot to the nearest local maximizer. When the state reaches the boundary of D_1 – namely, when $|\nabla L(z_1)| = \varepsilon_2$, $|z_2| = \rho_2$, $q = 1$, and $\sigma \in \{-\varsigma, \varsigma\}$ – then such a solution can no longer flow, since doing so would cause the state z to leave the set $C \cup D$, due to $u = \varsigma \text{sign}(z_2)$ which causes both $|\nabla L(z_1)|$ and $|z_2|$ to increase. Therefore, the solution only jumps when the state z is on the boundary of D_1 , and the algorithm resets the logic variable q to 0 and assigns u to $\kappa(h(z))$ while z remains the same. Consequently, the solution is once again in C_0 and, by Theorem 3.2.14, must flow using $u = \kappa(h(z))$ until the state z reaches D_0 , which is a local minimizer due to the fact that V satisfies (3.109) for all $z \in \mathcal{U} := \mathbb{R}^2 \setminus (\mathcal{A}_{1_{\max}} \times \{0\})$. This process repeats for all time.

Next, we look at a solution of type 2), namely, a solution that starts in the interior of C_1 . Due to the construction of F and C , such a solution flows, using $u = \varsigma \text{sign}(z_2)$, to push z away from a critical point, which could be either a local maximizer or a local minimizer. Due to Lemma 7.4.3, and the construction of D_0 and D_1 , with $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ sufficiently small, the solution arrives at D_1 in finite time less than or equal to $T = \sqrt{2\delta_2 + \frac{8}{3}\delta_1} > 0$. Once the state z reaches D_1 , the logic variable q is reset to 0 and u is assigned to $\kappa(h(z))$, and the state z remains the same. Hence, the solution is in C_0 , and from here follows the trajectory of type 1) solutions, described above.

Then, we pick a solution of type 3), namely, a solution that starts in the interior of D_0 . Due to the construction of D_0 , in (7.9), and since both local maximizers and

local minimizers both satisfy $\nabla L(z_1) = 0$, then such a solution could start near either a local maximizer or a local minimizer. Such a solution jumps, resetting the logic variable q to 1 and assigning u to $\varsigma \text{sign}(z_2)$. Consequently, the state z is in C_1 , and from here the solution follows the trajectory of type 2) solutions, described above.

Then, we pick a solution of type 4), which starts in the interior of D_1 . Due to the construction of D_1 , in (7.9), such a solution will always start far from any local minimizer or local maximizer. In D_1 , the logic variable q is reset to 0 and u is assigned to $\kappa(h(z))$ while z does not change. Hence, the state z is in C_0 , and from here follows the trajectory of type 1) solutions, described above.

Next, we pick a solution of type 5), which starts on the boundary of D_0 . Since $\dot{V}(z) \leq 0$ for all $z \in \mathcal{U} := \mathbb{R}^2 \setminus (\mathcal{A}_{1\max} \times \{0\})$ by Theorem 3.2.14, then the solution could either flow or jump first. However, since by the construction of D_0 , there exists some time $(t_1, 0)$ in the domain of the solution at which flow is no longer possible, as described for solutions for type 1), then the solution eventually jumps, and the algorithm resets q to 1 and assigns u to $\varsigma \text{sign}(z_2)$ while z does not change. Consequently, the state z is in C_1 , from which the solution follows the trajectory of type 2) solutions, described above.

Then, we pick a solution of type 6), which starts on the boundary of D_1 . Such a solution could not flow, since doing so would cause the state z to leave the set $C \cup D$, due to $u = \varsigma \text{sign}(z_2)$, which causes both $|\nabla L(z_1)|$ and $|z_2|$ to increase. Therefore, the solution always jumps, and the algorithm resets q to 0 and assigns u to $\kappa(h(z))$, while z remains unchanged. Hence, the state z is in C_0 , from which the solution follows the trajectory of type 1) solutions, described above.

Finally, for Step 3, to establish practical global attractivity of \mathcal{A} (7.8) for solutions of types 1)-6) above, we derive the smallest sublevel set of each basin of

attraction \mathcal{U}_i , defined via (3.106)-(3.108), of V , defined via (3.105), that contains $\{z \in \mathbb{R}^2 : |\nabla L(z_1)| = \varepsilon_2, |z_2| = \rho_2\}$ which, when $q = 1$, occurs on the boundary of D_1 . Namely, given $\rho_2 > 0$, $\varepsilon_2 > 0$, and $\alpha \in \mathcal{K}$ from (M5) such that $\delta_2 \in (0, \alpha(\varepsilon_2))$ and such that if $|\nabla L(z_1)| \leq \varepsilon_2$ then $|z_1|_{\mathcal{A}_1} \leq \delta_2$, and defining $\widehat{\mathcal{U}}_0$ as in (7.18), then there exists some $\hat{c}_0 > 0$ such that $V(z) \leq \hat{c}_0$ contains $\widehat{\mathcal{U}}_0$.

As shown in the proof of Theorem 3.2.14, V is positive definite on each basin of attraction \mathcal{U}_i in (3.106)-(3.108) with respect to $(z_{1_i}^* \times \{0\}) \in (\mathcal{A}_{1_{\min}} \times \mathbb{R})$, where $i \in \{1, 2, \dots, n\}$. Furthermore, V is radially unbounded since L is radially unbounded by item (M4) of Assumption 3.2.11. This implies that each piece of V is convex on its respective \mathcal{U}_i , for $i \in \{1, 2, \dots, n\}$, with respect to each $(z_{1_i}^* \times \{0\}) \in (\mathcal{A}_{1_{\min}} \times \mathbb{R})$, which, in turn, means that

$$V(z_{1_i}^*) \geq V(z_1) + \langle \nabla V(z_1), z_{1_i}^* - z_1 \rangle \quad (7.19)$$

for all $z_1, z_{1_i}^* \in \mathcal{U}_i$. Then, each piece of V can be upper bounded on its respective basin of attraction \mathcal{U}_i as follows.

$$\begin{aligned} V(z) &= \gamma \left(L(z_1) - L(z_{1_i}^*) \right) + \frac{1}{2} z_2^2 \\ &\leq \gamma |\nabla L(z_1)| |z_1|_{\mathcal{A}_{1_{\min}}} + \frac{1}{2} |z_2|^2. \end{aligned} \quad (7.20)$$

Then, given $\varepsilon_2 > 0$ and given $\alpha \in \mathcal{K}$ from item (M5) such that $\delta_2 \in (0, \alpha(\varepsilon_2))$ and such that $|\nabla L(z_1)| \leq \varepsilon_2$ implies $|z_1|_{\mathcal{A}_{1_{\min}}} \leq \delta_2$, we have, for each piece of V on its respective basin of attraction \mathcal{U}_i

$$V(z) \leq \gamma(\varepsilon_2 \delta_2) + \frac{1}{2} |z_2|^2 \quad (7.21)$$

Then, given $\rho_2 > 0$ and since we define $\widehat{\mathcal{U}}_0$ via (7.18), then we can further upper

bound V by

$$V(z) \leq \gamma(\varepsilon_2 \delta_2) + \frac{1}{2} \rho_2^2 =: c_0. \quad (7.22)$$

Therefore, every $z \in \widehat{\mathcal{U}}_0$ belongs to the c_0 -sublevel set of V in (7.22). Then, taking $\hat{c}_0 > c_0$, the smallest sublevel set of V that contains $\widehat{\mathcal{U}}_0$ is

$$V(z) \leq \hat{c}_0. \quad (7.23)$$

Therefore, the set \mathcal{A} is practically globally attractive for \mathcal{H} in the parameters $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$.

□

7.5 Numerical Example

Example 7.5.1. *This example compares multiple solutions to demonstrate the effectiveness of the hybrid algorithm \mathcal{H} , both when escaping from local maxima, and when converging from initial points that are not maxima. The algorithm has no knowledge of L , or the location of its critical points, but it uses measurements of ∇L at the current value of z_1 . The values of the heavy ball parameters are $\lambda = 145$, and $\gamma = \frac{3}{4}$, and the hybrid algorithm parameter values are $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.06$, $\rho_1 = 0.05$, $\rho_2 = 0.06$, and $\varsigma = 1$. The objective function is $L(z_1) = \frac{z_1^2(z_1-10)^2(z_1-20)^2(z_1-30)^2}{10,000}$, which has local minima at $\mathcal{A}_{1\min} = \{0, 10, 20, 30\}$ and local maxima at $\mathcal{A}_{1\max} = \{5(3 - \sqrt{5}), 15, 5(3 + \sqrt{5})\}$.*

We show that the objective function L in this example satisfies Assumption 3.2.11 as follows. To show it satisfies item (M2), we take the gradient, namely,

$$\nabla L(z_1) = 0.0002z_1(z_1 - 10)^2(z_1 - 20)^2(z_1 - 30)^2$$

$$\begin{aligned}
&+ 0.0002z_1^2(z_1 - 10)(z_1 - 20)^2(z_1 - 30)^2 \\
&+ 0.0002z_1^2(z_1 - 10)^2(z_1 - 20)(z_1 - 30)^2 \\
&+ 0.0002z_1^2(z_1 - 10)^2(z_1 - 20)^2(z_1 - 30). \tag{7.24}
\end{aligned}$$

By inspection, (7.24) is continuous. Now we take the Hessian:

$$\begin{aligned}
\nabla^2 L(z_1) &= 0.0002z_1^2(z_1 - 10)^2(z_1 - 20)^2 + 0.0002z_1^2(z_1 - 10)^2(z_1 - 30)^2 \\
&+ 0.0002z_1^2(z_1 - 20)^2(z_1 - 30)^2 \\
&+ 0.0002(z_1 - 10)^2(z_1 - 20)^2(z_1 - 30)^2 \\
&+ 0.0008z_1(z_1 - 10)(z_1 - 20)^2(z_1 - 30)^2 \\
&+ 0.0008z_1(z_1 - 10)^2(z_1 - 20)(z_1 - 30)^2 \\
&+ 0.0008z_1(z_1 - 10)^2(z_1 - 20)^2(z_1 - 30) \\
&+ 0.0008z_1^2(z_1 - 10)(z_1 - 20)(z_1 - 30)^2 \\
&+ 0.0008z_1^2(z_1 - 10)(z_1 - 20)^2(z_1 - 30) \\
&+ 0.0008z_1^2(z_1 - 10)^2(z_1 - 20)(z_1 - 30) \tag{7.25}
\end{aligned}$$

By inspection, (7.25) is also continuous. Therefore, L is \mathcal{C}^2 , satisfying (M2). To show L satisfies (M1), we evaluate L at all of the critical points, as follows:

$$\begin{aligned}
\nabla^2 L(0) &= 7200 \\
\nabla^2 L(10) &= 800 \\
\nabla^2 L(20) &= 800 \\
\nabla^2 L(30) &= 7200 \\
\nabla^2 L(5(3 - \sqrt{5})) &\approx -2000
\end{aligned}$$

$$\begin{aligned}\nabla^2 L(15) &\approx -562.5 \\ \nabla^2 L(5(3 + \sqrt{5})) &\approx -2000\end{aligned}\tag{7.26}$$

Since the Hessian of L does not equal zero at any of the critical points, then none of these critical points are degenerate, and therefore L is Morse, satisfying (M1). L satisfies (M3) because there exists a separation of at least $d_0 = 5(3 - \sqrt{5}) \approx 3.28$ between all critical points. Furthermore, it is easy to see that L satisfies (M4). That is, $L(z_1) \rightarrow \infty$ as $|z_1| \rightarrow \infty$. Finally, L satisfies (M5) when choosing $\alpha(\varepsilon) = |\varepsilon|$.

The hand-tuning of the parameters $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ can be approached as follows:

1. Start with small values, e.g., for the function in this example, we started with $\varepsilon_1 = 0.1$, $\varepsilon_2 = 0.11$, and $\rho_1 = 0.1$, $\rho_2 = 0.11$.
2. Then, to speed up the convergence times, gradually reduce these four parameter values. For this example, we decreased by about 0.01 at a time.
3. Eventually, a point is reached where decreasing the parameter values results in convergence times starting to increase again. At this point, increase the parameters by at most the last decrement (in our case, 0.01) and stop.

Initial conditions for the simulations are $z_1(0, 0) = \{-1, 5(3 - \sqrt{5}), 6, 15, 24.5, 5(3 + \sqrt{5}), 31\}$, $z_2(0, 0) = 0$, and $q(0, 0) = 0$. Note that the function L and parameter values are the same as those used in Figure 1.5, with the exception of $\varsigma = 10^7$, $\varepsilon_2 = 10$, and $\rho_2 = 10$ in Figure 1.5. The reason ς , ε_2 , and ρ_2 are set differently in Figure 7.1 is that this example includes no noise in measurements of the gradient. Such noise can cause jump times to be different, and so such parameters needed to be tuned accordingly in Figure 1.5. Recall that Figure 1.5 shows that the state z_1 converges with our algorithm under

arbitrarily small noise in the gradient measurements, when starting close to a local maximum at $z_1 = 15$, whereas for simulated annealing the state z_1 remains stuck at this same local maximum. Although noise in the gradient measurements is not present in Figure 7.1, it would be easy to see that simulated annealing – which still contains a noise signal – would get stuck when starting at the local maxima at $\mathcal{A}_{1\max} = \{5(3 - \sqrt{5}), 15, 5(3 + \sqrt{5})\}$. In contrast, Figure 7.1 shows that the hybrid algorithm \mathcal{H} converges to a local minimum from such initial conditions.

Figure 7.1 shows the evolution of z_1 and z_2 over time for multiple solutions with different initial conditions. Black dots with times labeled in seconds denote when each simulation converges to within 0.01 of \mathcal{A}_1^2 . Conversely, the solutions which start in a small neighborhood of local maxima begin with a jump, followed by a switch to $u = \sigma$, then jump again, switching to the heavy ball algorithm, before such solutions converge to a neighborhood of a local minimum. The solutions which do not start at critical points start with the heavy ball algorithm. Although there are jumps near the local minimum to which such solutions converge, these solutions also do not leave the neighborhood of a local minimum, determined by the values chosen for $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$. The times it takes for different solutions to converge to within 0.01 of \mathcal{A}_1 are listed in Table 7.1.

$z_1(0, 0)$	Time to converge
-1	0.0369
$5(3 - \sqrt{5})$	0.862
6	0.829
15	3.51
24.5	0.868
$5(3 + \sqrt{5})$	0.862
31	0.0373

Table 7.1: Times in which different solutions converge to within 0.01 of \mathcal{A}_1 .

²Code at github.com/HybridSystemsLab/PGASHeavyBall

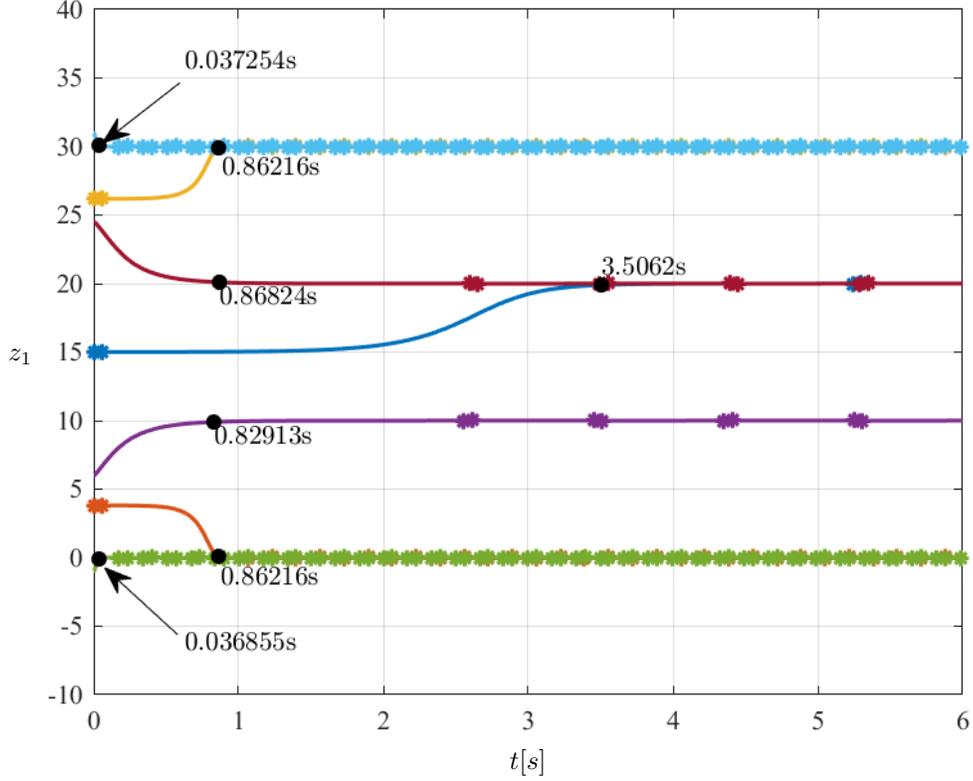


Figure 7.1: The evolution of z_1 over time for the hybrid system \mathcal{H} , for the objective function $L(z_1) = \frac{z_1^2(z_1-10)^2(z_1-20)^2(z_1-30)^2}{10,000}$, with $\mathcal{A}_{1,\min} = \{0, 10, 20, 30\}$, $\mathcal{A}_{1,\max} = \{5(3-\sqrt{5}), 15, 5(3+\sqrt{5})\}$, and $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.06$, $\rho_1 = 0.05$, $\rho_2 = 0.06$, $\varsigma = 1$, $\lambda = 145$, and $\gamma = \frac{3}{4}$. This plot shows different solutions, starting from different initial conditions. Solutions start at local maxima at $z_1(0,0) = 15$, $z_1(0,0) = 5(3-\sqrt{5})$, and $z_1(0,0) = 5(3+\sqrt{5})$, as well as at the points $z_1(0,0) = 6$, $z_1(0,0) = 24.5$, $z_1(0,0) = -1$, and $z_1(0,0) = 31$, which are neither maxima nor minima. All solutions start with $z_2(0,0) = 0$ and $q(0,0) = 0$. Times when each solution converges to within 0.01 of $\mathcal{A}_{1,\min}$ are marked with black dots and labeled in seconds. Jumps are labeled with asterisks.

The particular values chosen for $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ keeps solutions within a neighborhood of size 0.01 around $\mathcal{A}_{1_{\min}}$. We conjecture that different tunings of $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ would change the size of such a neighborhood, with larger $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ yielding a larger neighborhood of $\mathcal{A}_{1_{\min}}$, and smaller $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ resulting in a smaller neighborhood of $\mathcal{A}_{1_{\min}}$.

Chapter 8

Accelerated Multiagent Optimization

8.1 Problem Statement

We propose an algorithm that solves the following optimization problem.

Problem 8.1.1. *Given $f : \mathbb{R}^n \rightarrow \mathbb{R}$ and $X \subset \mathbb{R}^n$, design an optimization algorithm using accelerated methods that asynchronously solves (1.9). Such an algorithm must have the minimizer exponentially stable, an exponential convergence rate, robustness, and show improvement over gradient descent in simulation.*

Our approach to solving Problem 8.1.1 is as follows:

- 1) In Section 8.2, we design a synchronous algorithm that computes a single constrained heavy ball update and communicates with its neighbors once every iteration. Then, we establish that the agents' updates over two iterations are equivalent to a contractive mapping applied to their decision variables;
- 2) In Section 8.3, we design a synchronous algorithm that, during each iteration,

computes two constrained heavy ball updates and communicates once with its neighbors. Then, we establish an exponential convergence rate for the agents’ decision variables based on the contractive property established in Section 8.2, and we show that it satisfies the Synchronous Convergence and Box Conditions in [59];

- 3) In Section 8.4, we design a totally asynchronous version of the algorithm in Section 8.3 and use the forthcoming Proposition B.1.2 to establish that, since the synchronous algorithm has exponential convergence and satisfies the Synchronous Convergence and Box Conditions, then the totally asynchronous algorithm also has an exponential convergence rate.

8.2 Synchronous Heavy Ball

In this section, we present Step 1) in our approach to solving Problem 8.1.1. Namely, as an intermediary step in developing an asynchronous multiagent algorithm, we first design a synchronous algorithm that computes a single constrained heavy ball update and communicates with its neighbors once every iteration. Then, we establish that the agents’ updates over two iterations are equivalent to a contractive mapping applied to their decision variables.

8.2.1 Modeling

To extend (1.11)-(1.12) to a multiagent setting, allowing arbitrary asynchrony for the agents, we start by extending techniques used in [50], for parallelized, multiagent gradient descent. The term “parallelized” means that each decision variable is updated only by a single agent, with each decision variable assigned to each agent referred to as a “block.” Such block-based algorithms scale better with

large learning problems, and have been shown in some cases to tolerate arbitrarily long delays in both communication and computation of unconstrained problems, which eliminates the need to enforce and verify delay boundedness assumptions [60], [61], [62].

We consider $N \in \mathbb{N}_{>0}$ agents indexed over $i \in \mathcal{V} := \{1, 2, \dots, N\}$ that exchange information over an undirected graph $\Gamma = (\mathcal{V}, \mathcal{E})$, where edges are pairs in the set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ which directly link two different nodes that are essential neighbors. We use $\mathcal{N}_i \subset \mathcal{V}$ to denote the set of essential neighbors of i , namely, \mathcal{N}_i is the set of indices of agents $\ell \neq i$ whose decision variables are needed for agent i 's computations. More formally, agent $\ell \in \mathcal{N}_i$ is an essential neighbor of agent i if $\nabla_i f$ explicitly depends upon agent ℓ 's decision variable. Only the essential neighbors $\ell \in \mathcal{N}_i$ need to communicate with agent i to ensure it has the information necessary to compute gradients. In particular, an edge from agent ℓ to agent i , denoted (ℓ, i) , implies that ℓ and i are essential neighbors and agent ℓ can send information to agent i . Since the graph Γ is undirected, agent i is an essential neighbor of agent ℓ if and only if agent ℓ is an essential neighbor of agent i . Conversely, the lack of an edge from agent ℓ to agent i (and, hence, i to ℓ) implies that ℓ and i are not essential neighbors and do not communicate with each other.

We use superscripts to denote ownership by an agent, and use subscripts for indexing. For instance, we denote the vector containing agent i 's local copy of all decision variables as z^i , and we denote the constraint set corresponding to agent i 's decision variable (denoted $(z_{1,i}^i, z_{2,i}^i)$) as $X_i \times X_i$; see definitions below. With this in mind, we impose the following assumption on the constraint set X .

Assumption 8.2.1 (Properties of the constraint set). *The constraint set $X \subset \mathbb{R}^N$*

is nonempty, compact, and convex. The constraint set can be decomposed as

$$X = X_1 \times X_2 \times \dots \times X_N \quad (8.1)$$

where, for each $i \in \mathcal{V}$, $X_i \subset \mathbb{R}$.

Remark 8.2.2. Assumption 8.2.1 permits the use of box constraints, which often arise in multiagent optimization.

For each $i \in \mathcal{V}$, agent i stores its own decision variable and a local copy of the decision variables of all other agents for use in local computations. We denote agent i 's value for its own decision variable as

$$(z_{1,i}^i, z_{2,i}^i) \in X_i \times X_i. \quad (8.2)$$

We denote agent i 's local copy of the decision variable for agent ℓ as

$$(z_{1,\ell}^i, z_{2,\ell}^i) \in X_\ell \times X_\ell. \quad (8.3)$$

Thus, the full state of agent i is

$$z^i := (z_1^i, z_2^i) \in X \times X \quad (8.4)$$

where

$$z_1^i := (z_{1,1}^i, z_{1,2}^i, \dots, z_{1,i}^i, \dots, z_{1,N}^i) \quad (8.5a)$$

$$z_2^i := (z_{2,1}^i, z_{2,2}^i, \dots, z_{2,i}^i, \dots, z_{2,N}^i). \quad (8.5b)$$

The synchronous algorithm for the constrained heavy ball method is given in

Algorithm 6. In this algorithm, at each discrete time instant $k \in \mathbb{N}$, each agent $i \in \mathcal{V}$ computes the new value of $(z_{1,i}^i, z_{2,i}^i)$. In particular, $z_{1,i}^i$ is updated to $\kappa^i(z^i)$ and $z_{2,i}^i$ is updated to $z_{1,i}^i$, where

$$\kappa^i(z^i) := \Pi_{X_i} \left[z_{1,i}^i - \gamma \nabla_i f(z_1^i) + \lambda(z_{1,i}^i - z_{2,i}^i) \right] \quad (8.6)$$

where $\lambda > 0$ and $\gamma > 0$ come from (1.10); see lines 3-5 of Algorithm 6. Then, starting at line 6 of Algorithm 6, each agent $i \in \mathcal{V}$ receives information from its neighbors and updates the entries $(z_{1,\ell}^i, z_{2,\ell}^i)$ of its state z^i , where $\ell \in \mathcal{N}_i$, in the following manner. For each of agent i 's essential neighbors, namely, for each $\ell \in \mathcal{N}_i$, $(z_{1,\ell}^i, z_{2,\ell}^i)$ is updated to $(z_{1,\ell}^\ell, z_{2,\ell}^\ell)$; see lines 7-9 of Algorithm 6. For each ℓ which is not an essential neighbor of i , $(z_{1,\ell}^i, z_{2,\ell}^i)$ is left unchanged; see lines 10-12 of Algorithm 6.

Algorithm 6 Constrained, Synchronous, Multiagent Heavy Ball

- 1: For each $i \in \mathcal{V}$, set the initial state z_0^i to an arbitrary value in $X \times X$.
 - 2: **for** each $k \in \mathbb{N}$ **do**
 - 3: **for** each $i \in \mathcal{V}$ **do**
 - 4: Update $(z_{1,i}^i, z_{2,i}^i)$ in (8.2) to $(\kappa^i(z^i), z_{1,i}^i)$, with κ^i defined via (8.6).
 - 5: **end for**
 - 6: **for** each $i \in \mathcal{V}$ **do**
 - 7: **for** each $\ell \in \mathcal{N}_i$ **do**
 - 8: Update $(z_{1,\ell}^i, z_{2,\ell}^i)$ in (8.3) to $(z_{1,\ell}^\ell, z_{2,\ell}^\ell)$.
 - 9: **end for**
 - 10: **for** each $\ell \notin \mathcal{N}_i$ **do**
 - 11: Keep $(z_{1,\ell}^i, z_{2,\ell}^i)$ constant.
 - 12: **end for**
 - 13: **end for**
 - 14: **end for**
-

Now, we model Algorithm 6 mathematically. From Algorithm 6, for each $i \in \mathcal{V}$

agent i has its state z^i in (8.4) updated via the following difference equation:

$$(z^i)^+ = G_{\text{sync}}^i(z^1, z^2, \dots, z^N) \quad z^i \in D_{\text{sync}}^i := X \times X, \quad (8.7)$$

where we define G_{sync}^i as

$$G_{\text{sync}}^i(z^1, z^2, \dots, z^N) := (g_1^i(z^1, z^2, \dots, z^N), g_2^i(z^1, z^2, \dots, z^N)) \quad (8.8)$$

where, for each $p \in \{1, 2\}$, g_p^i is defined as

$$g_p^i(z^1, z^2, \dots, z^N) := \left(g_{p,1}^i(z^1, z^2, \dots, z^N), g_{p,2}^i(z^1, z^2, \dots, z^N), \dots, g_{p,N}^i(z^1, z^2, \dots, z^N) \right) \quad (8.9)$$

and for each $s \in \mathcal{V}$, $g_{p,s}^i$ is defined as

$$g_{1,s}^i(z^1, z^2, \dots, z^N) := \begin{cases} \kappa^i(z^i) & \text{if } s = i \\ z_{1,s}^s & \text{if } s \in \mathcal{N}_i \\ z_{1,s}^i & \text{if } s \notin \mathcal{N}_i \end{cases} \quad (8.10)$$

if $p = 1$ and as

$$g_{2,s}^i(z^1, z^2, \dots, z^N) := \begin{cases} z_{1,i}^i & \text{if } s = i \\ z_{2,s}^s & \text{if } s \in \mathcal{N}_i \\ z_{2,s}^i & \text{if } s \notin \mathcal{N}_i \end{cases} \quad (8.11)$$

if $p = 2$. The choice of keeping $(z_{1,s}^i, z_{2,s}^i)$ constant for the case of $s \notin \mathcal{N}_i$ in (8.10)-(8.11) – see line 11 in Algorithm 6 – is arbitrary, since the neighbors $s \notin \mathcal{N}_i$ are not essential and, hence, the values of their decision variables do not enter agent i 's computations.

Then, the full multiagent system is denoted as

$$\begin{aligned} (z^1, z^2, \dots, z^N)^+ &= G_{\text{sync}}(z^1, z^2, \dots, z^N) \\ (z^1, z^2, \dots, z^N) &\in D_{\text{sync}} := (X \times X)^N \end{aligned} \quad (8.12)$$

where G_{sync} is defined as

$$\begin{aligned} G_{\text{sync}}(z^1, z^2, \dots, z^N) &:= \\ (G_{\text{sync}}^1(z^1, z^2, \dots, z^N), G_{\text{sync}}^2(z^1, z^2, \dots, z^N), \dots, G_{\text{sync}}^N(z^1, z^2, \dots, z^N)). \end{aligned} \quad (8.13)$$

8.2.2 Results for Algorithm 6

In the results to follow, we impose the following assumption on the objective function f .

Assumption 8.2.3. *The function f is \mathcal{C}^2 and convex.*

Remark 8.2.4. *Assumption 8.2.3, which is a common assumption used in the analysis of optimization algorithms [66], ensures that the Hessian matrix exists. Additionally, the convex property ensures that every diagonal entry of the Hessian is positive [66, Section 3.1.4].*

Additionally, we impose the following diagonal dominance assumption on the objective function f .

Assumption 8.2.5 (Diagonal dominance). *The $N \times N$ Hessian matrix $d \mapsto H(d) = \nabla^2 f(d)$ is μ -diagonally dominant on $X \subset \mathbb{R}^n$ for some $\mu > 0$. That is, for each $i \in \{1, 2, \dots, N\}$,*

$$|H_{ii}(d)| - \mu \geq \sum_{j=1, j \neq i}^N |H_{ij}(d)| \quad \forall d \in X. \quad (8.14)$$

Remark 8.2.6. *Assumption 8.2.5, which is used to show that, for (D_{sync}, G_{sync}) in (8.12), the agents' updates over two iterations are equivalent to a contractive mapping applied to their decision variables – which is the first step in solving Problem 8.1.1 – and is used for showing that [59, Chapter 6, Assumption 2.1(b)] holds, is commonly used in the analysis of totally asynchronous multiagent optimization algorithms; see, e.g., [59]. In particular, in [59, Section 6.3.2], it is observed that some form of diagonal dominance is typically required to ensure convergence of totally asynchronous algorithms. Diagonal dominance implies μ -strong convexity¹ of f ; see [80, Lemma 1]. Consequently, when f is μ -diagonally dominant, then f has a unique minimizer x^* .*

We denote the minimizer of f over X as $x^* \in X$. Note that under Assumption 8.2.1 the i -th entry of x^* , namely, x_i^* , belongs to X_i .

Under Assumptions 8.2.1, 8.2.3, and 8.2.5, the following lemma, used in some of the results to follow, shows that $x_i^* \in X_i$ is a fixed point of κ^i in (8.6).

Lemma 8.2.7. *(Fixed point of (8.6)) Let f satisfy Assumption 8.2.3, let $X \subset \mathbb{R}^N$ in (8.1) satisfy Assumption 8.2.1, and let H satisfy Assumption 8.2.5 with $\mu > 0$. Then, for each $i \in \mathcal{V}$, $x_i^* \in X_i$ is a fixed point of κ^i in (8.6).*

Proof. By Assumption 8.2.1, the set $X \subset \mathbb{R}^N$ is nonempty, compact, convex, and can be decomposed as in (8.1). Furthermore, by Assumption 8.2.5, the Hessian of f is diagonally dominant with $\mu > 0$. As discussed in Remark 8.2.6, Assumption 8.2.5 implies μ -strong convexity of f and, consequently, when H is μ -diagonally dominant, then f has a unique² minimizer $x^* \in X$.

¹To illustrate that diagonal dominance is a strictly stronger condition than μ -strong convexity of f , consider $f(d) := d^\top Qd + p^\top d$, where $Q := \begin{bmatrix} 1 & -2 \\ -1 & 3 \end{bmatrix}$. The resulting $H(d)$ has positive eigenvalues, so f is strongly convex, but $H(d)$ fails to be diagonally dominant.

²Whereas convexity of f under Assumption 8.2.3 without the μ -diagonal dominance imposed by Assumption 8.2.5 could allow f to have a continuum of minimizers.

Then, since f is convex by Assumption 8.2.3, and since x^* minimizes f over X , we have, by item (O1), of Proposition C.1.1, that, for each $y_{1,i}^i \in X_i$,

$$\langle y_{1,i}^i - x_i^*, -\nabla_i f(x^*) \rangle \leq 0. \quad (8.15)$$

As a consequence of the property in (8.15), we have, for each $y_{1,i}^i \in X_i$, that

$$\langle y_{1,i}^i - x_i^*, x_i^* - \gamma \nabla_i f(x^*) + \lambda(x_i^* - x_i^*) - x_i^* \rangle \leq 0 \quad (8.16)$$

where $\gamma > 0$ and $\lambda > 0$ come from (1.10). Next, since f is convex by Assumption 8.2.3 and since the set X is nonempty, compact, and convex by Assumption 8.2.1, then by item (P1) of Proposition C.1.2, given $v_{1,i}^i \in X_i$, $u_{1,i}^i \in X_i$ is equal to $\Pi_{X_i}[v_{1,i}^i]$ if and only if $\langle y_{1,i}^i - u_{1,i}^i, v_{1,i}^i - u_{1,i}^i \rangle \leq 0$ for all $y_{1,i}^i \in X_i$. Now, let $u_{1,i}^i = x_i^*$ and $v_{1,i}^i = x_i^* - \gamma \nabla_i f(x^*) + \lambda(x_i^* - x_i^*)$. Then, for each $i \in \mathcal{V}$, $x_i^* = \Pi_{X_i}[x_i^* - \gamma \nabla_i f(x^*) + \lambda(x_i^* - x_i^*)]$ by (8.16). Therefore, for each $i \in \mathcal{V}$, $x_i^* \in X_i$ is a fixed point of κ^i in (8.6). \square

For the results to follow, given X in (8.1), we denote by

$$z := (z_1, z_2) \in X \times X \quad (8.17)$$

the ‘‘true value,’’ in the sense that z in (8.17) contains the current values of the decision variables of all agents. We define z_1 and z_2 as

$$z_1 := (z_{1,1}^1, z_{1,2}^2, \dots, z_{1,N}^N) \quad (8.18a)$$

$$z_2 := (z_{2,1}^1, z_{2,2}^2, \dots, z_{2,N}^N). \quad (8.18b)$$

Namely, z_1 collects the $z_{1,i}^i$ components and z_2 collects the $z_{2,i}^i$ components, for each

$i \in \mathcal{V}$. Then, under Assumptions 8.2.1, 8.2.3, and 8.2.5 the following proposition establishes that, for each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the synchronous algorithm $(D_{\text{sync}}, G_{\text{sync}})$ in (8.12) from $(z_o^1, z_o^2, \dots, z_o^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ is contractive over two steps. To prove it, we use Lemma 8.2.7.

Proposition 8.2.8. *Suppose $X \subset \mathbb{R}^N$ satisfies Assumption 8.2.1, f satisfies Assumption 8.2.3, and H satisfies Assumption 8.2.5 with $\mu > 0$. For each $\gamma \in \left(0, \frac{1}{\max_{i \in \mathcal{V}} \max_{\eta \in X} |H_{ii}(\eta)|}\right)$, $\lambda \in \left(0, \frac{\gamma\mu}{2}\right)$, and each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the synchronous algorithm $(D_{\text{sync}}, G_{\text{sync}})$ in (8.12) from $(z_o^1, z_o^2, \dots, z_o^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ satisfies*

$$\|z(k+1) - z^*\|_\infty \leq a \|z(k-1) - z^*\|_\infty \quad (8.19)$$

for all $k \in \mathbb{N} \setminus \{0\}$, where $a := \max\{a_1, a_2\}$, $a_1 := (1 - \gamma\mu + \lambda)^2 + \lambda + \lambda(1 - \gamma\mu + \lambda) \in [0, 1)$, $a_2 := 1 - \gamma\mu + 2\lambda \in [0, 1)$, and z^* is defined as

$$z^* := (x^*, x^*). \quad (8.20)$$

Proof. To establish the result, we proceed as follows.

- 1) Show that each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to $(D_{\text{sync}}, G_{\text{sync}})$ is complete;
- 2) Show that each true value trajectory $k \mapsto z(k)$, where z is defined via (8.17), associated to each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to $(D_{\text{sync}}, G_{\text{sync}})$ from $(z_o^1, z_o^2, \dots, z_o^N) \in (X \times X)^N$, is such that $\|z_1(k+1) - x^*\|_\infty = \max_{i \in \mathcal{V}} |z_{1,i}^i(k+1) - x_i^*|$ is upper bounded by³

³It will be shown in this step that such an upper bound comes via the non-expansiveness of

$\max_{i \in \mathcal{V}} |\hat{\kappa}^i(z(k)) - \hat{\kappa}^i(z^*)|$ for each $k \in \mathbb{N}$, where $\hat{\kappa}^i$ is defined as

$$\hat{\kappa}^i(z) := z_{1,i}^i - \gamma \nabla_i f(z_1) + \lambda(z_{1,i}^i - z_{2,i}^i); \quad (8.21)$$

- 3) Given $\bar{c}, \underline{c} \in \mathbb{R}^N \times \mathbb{R}^N$, there exists $c \in \mathbb{R}^N \times \mathbb{R}^N$, such that $c_{1,i}^i \in (\bar{c}_{1,i}^i, \underline{c}_{1,i}^i)$ for each $i \in \mathcal{V}$ and $c_{2,i}^i \in (\bar{c}_{2,i}^i, \underline{c}_{2,i}^i)$ for each $i \in \mathcal{V}$, and such that we can use the MVT to upper bound $|\hat{\kappa}^i(\bar{c}) - \hat{\kappa}^i(\underline{c})|$ by

$$(1 - \gamma\mu + \lambda) \|\bar{c}_1 - \underline{c}_1\|_\infty + \lambda \|\bar{c}_2 - \underline{c}_2\|_\infty;$$
- 4) Upper bound both $\|z_1(k+1) - x^*\|_\infty$ and $\|z_2(k+1) - x^*\|_\infty$ using steps 2) and 3) with $\bar{c} = z(k)$, $\underline{c} = z^*$, $\bar{c}_1 = z_1(k)$, $\bar{c}_2 = z_2(k)$, and $\underline{c}_1 = \underline{c}_2 = x^*$. Then, use such bounds to yield (8.19) for all $k \in \mathbb{N} \setminus \{0\}$;
- 5) Show that a_1 and a_2 , defined below (8.19), are both in $[0, 1)$.

Proceeding with step 1), we must show that the map G_{sync}^i in (8.7) is defined on D_{sync} and that $G_{\text{sync}}^i(D_{\text{sync}}) \subset D_{\text{sync}}$. By construction, κ^i in (8.6) is defined for each $z^i \in X \times X$. In addition, by Assumption 8.2.1, the constraint set X is nonempty. Furthermore, by construction, $G_{\text{sync}}^i(z^1, z^2, \dots, z^N) \neq \emptyset$ for each $(z^1, z^2, \dots, z^N) \in (X \times X)^N$. Therefore, by [21, Definition 2.1], $\text{dom } G_{\text{sync}}^i = (X \times X)^N$, which implies that $D_{\text{sync}} \subset \text{dom } G_{\text{sync}}^i$, and each maximal solution to $(D_{\text{sync}}^i, G_{\text{sync}}^i)$ in (8.7)-(8.11) is complete. Since, by construction, $G_{\text{sync}}(z^1, z^2, \dots, z^N) \neq \emptyset$ for each $(z^1, z^2, \dots, z^N) \in (X \times X)^N$, and since $G_{\text{sync}}^i(z^1, z^2, \dots, z^N) \in X \times X$ for each $i \in \mathcal{V}$, due to κ^i in (8.6), then, by construction, $G_{\text{sync}}(z^1, z^2, \dots, z^N) \in (X \times X)^N$. Moreover, by Assumption 8.2.1,

$\Pi_{X_i}[\cdot]$, with respect to the Euclidean norm, which is listed in item (P2) of Proposition C.1.2. Namely, for all $v, y \in \mathbb{R}^n$, $|\Pi_{X_i}[v] - \Pi_{X_i}[y]| \leq |v - y|$.

X is nonempty. Therefore, by [21, Definition 2.1],

$$\text{dom } G_{\text{sync}}(z^1, z^2, \dots, z^N) = (X \times X)^N \quad (8.22)$$

which implies that $D_{\text{sync}} \subset \text{dom } G_{\text{sync}}$, and each maximal solution to $(D_{\text{sync}}, G_{\text{sync}})$ in (8.12) is complete.

Next, for step 2), note that $x_i^* \in X_i$ is a fixed point of κ^i in (8.6) by Lemma 8.2.7, namely, $x_i^* = \kappa^i(z^*)$. Then, picking a solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to $(D_{\text{sync}}, G_{\text{sync}})$ with the associated true value trajectory $k \mapsto z(k)$, the norm $\|z_1(k+1) - x^*\|_\infty$, where z_1 is defined in (8.18a) and x^* comes from (8.20), satisfies

$$\begin{aligned} \|z_1(k+1) - x^*\|_\infty &= \max_{i \in \mathcal{V}} |z_{1,i}^i(k+1) - x_i^*| \quad (8.23) \\ &= \max_{i \in \mathcal{V}} \left| \Pi_{X_i} \left[z_{1,i}^i(k) - \gamma \nabla_i f(z_1(k)) + \lambda (z_{1,i}^i(k) - z_{2,i}^i(k)) \right] \right. \\ &\quad \left. - \Pi_{X_i} [x_i^* - \gamma \nabla_i f(x^*) + \lambda (x_i^* - x_i^*)] \right| \\ &\leq \max_{i \in \mathcal{V}} \left| z_{1,i}^i(k) - \gamma \nabla_i f(z_1(k)) + \lambda (z_{1,i}^i(k) - z_{2,i}^i(k)) \right. \\ &\quad \left. - (x_i^* - \gamma \nabla_i f(x^*) + \lambda (x_i^* - x_i^*)) \right| \end{aligned}$$

for each $k \in \mathbb{N} \setminus \{0\}$ where the last inequality follows from the non-expansiveness of $\Pi_{X_i}[\cdot]$ with respect to the Euclidean norm, precisely, by item (P2) of Proposition C.1.2, since f is convex by Assumption 8.2.3 and X is nonempty, compact, and convex by Assumption 8.2.1. Note also that, in the last inequality, x^* and x_i^* still denote constrained minimizers. The inequality in (8.23) can be rewritten in terms of the function $\hat{\kappa}^i$ in (8.21), as follows:

$$\|z_1(k+1) - x^*\|_\infty \leq \max_{i \in \mathcal{V}} \left| \hat{\kappa}^i(z(k)) - \hat{\kappa}^i(z^*) \right|, \quad (8.24)$$

for each $k \in \mathbb{N}$, where z^* is defined in (8.20).

Then, for step 3), given $\bar{c}, \underline{c} \in \mathbb{R}^N \times \mathbb{R}^N$ and using $\hat{\kappa}^i$ in (8.21) and the MVT in Proposition 2.5.1, there exists $c \in \mathbb{R}^N \times \mathbb{R}^N$ such that $c_{1,i}^i \in (\bar{c}_{1,i}^i, \underline{c}_{1,i}^i)$ for each $i \in \mathcal{V}$ and $c_{2,i}^i \in (\bar{c}_{2,i}^i, \underline{c}_{2,i}^i)$ for each $i \in \mathcal{V}$, and such that

$$\hat{\kappa}^i(\bar{c}) - \hat{\kappa}^i(\underline{c}) = \sum_{j=1}^N \frac{\partial \hat{\kappa}^i(c)}{\partial c_{1,j}^j} (\bar{c}_{1,j}^j - \underline{c}_{1,j}^j) + \sum_{j=1}^N \frac{\partial \hat{\kappa}^i(c)}{\partial c_{2,j}^j} (\bar{c}_{2,j}^j - \underline{c}_{2,j}^j). \quad (8.25)$$

For each $j \neq i$, the partial derivatives are given by

$$\frac{\partial \hat{\kappa}^i(c)}{\partial c_{1,i}^i} = 1 - \gamma \nabla_i^2 f(c_1) + \lambda \quad (8.26a)$$

$$\frac{\partial \hat{\kappa}^i(c)}{\partial c_{1,j}^j} = -\gamma \nabla_j \nabla_i f(c_1) \quad (8.26b)$$

$$\frac{\partial \hat{\kappa}^i(c)}{\partial c_{2,i}^i} = -\lambda \quad (8.26c)$$

$$\frac{\partial \hat{\kappa}^i(c)}{\partial c_{2,j}^j} = 0. \quad (8.26d)$$

This allows (8.25) to be rewritten as

$$\begin{aligned} \hat{\kappa}^i(\bar{c}) - \hat{\kappa}^i(\underline{c}) &= (1 - \gamma \nabla_i^2 f(c_1) + \lambda) (\bar{c}_{1,i}^i - \underline{c}_{1,i}^i) \\ &+ \sum_{j=1, j \neq i}^N (-\gamma \nabla_j \nabla_i f(c_1)) (\bar{c}_{1,j}^j - \underline{c}_{1,j}^j) - \lambda (\bar{c}_{2,i}^i - \underline{c}_{2,i}^i). \end{aligned} \quad (8.27)$$

Taking the norm of both sides of (8.27) and applying the triangle inequality gives

$$\begin{aligned} \left| \hat{\kappa}^i(\bar{c}) - \hat{\kappa}^i(\underline{c}) \right| &\leq \left| 1 - \gamma \nabla_i^2 f(c_1) + \lambda \right| \left| \bar{c}_{1,i}^i - \underline{c}_{1,i}^i \right| \\ &+ \gamma \sum_{j=1, j \neq i}^N \left| \nabla_j \nabla_i f(c_1) \right| \left| \bar{c}_{1,j}^j - \underline{c}_{1,j}^j \right| + \lambda \left| \bar{c}_{2,i}^i - \underline{c}_{2,i}^i \right|. \end{aligned} \quad (8.28)$$

Then, denoting $\nabla_i^2 f$ as H_{ii} and $\nabla_j \nabla_i f$ as H_{ij} , where H comes from Assumption

8.2.5, we rewrite (8.28) as

$$\begin{aligned} \left| \hat{\kappa}^i(\bar{c}) - \hat{\kappa}^i(\underline{c}) \right| &\leq |1 - \gamma H_{ii}(c_1) + \lambda| \left| \bar{c}_{1,i}^i - \underline{c}_{1,i}^i \right| \\ &\quad + \gamma \sum_{j=1, j \neq i}^N |H_{ij}(c_1)| \left| \bar{c}_{1,j}^j - \underline{c}_{1,j}^j \right| + \lambda \left| \bar{c}_{2,i}^i - \underline{c}_{2,i}^i \right|. \end{aligned} \quad (8.29)$$

From Assumption 8.2.3, f is convex and thus every diagonal entry of the Hessian matrix is positive; see Remark 8.2.4. Using such a property and noting that, for the unconstrained update law $\hat{\kappa}^i$ in (8.21),

$$\gamma \in \left(0, \frac{1}{\max_{i \in \mathcal{V}} \max_{\eta \in \mathbb{R}^N} |H_{ii}(\eta)|} \right), \quad (8.30)$$

then with $\eta = z_1$, $1 - \gamma H_{ii}(c_1) > 1 - \frac{H_{ii}(c_1)}{\max_{i \in \mathcal{V}} \max_{z_1 \in \mathbb{R}^N} |H_{ii}(z_1)|} \geq 0$. Thus, (8.29) may be rewritten as

$$\begin{aligned} \left| \hat{\kappa}^i(\bar{c}) - \hat{\kappa}^i(\underline{c}) \right| &\leq (1 - \gamma H_{ii}(c_1) + \lambda) \left| \bar{c}_{1,i}^i - \underline{c}_{1,i}^i \right| \\ &\quad + \gamma \sum_{j=1, j \neq i}^N |H_{ij}(c_1)| \left| \bar{c}_{1,j}^j - \underline{c}_{1,j}^j \right| + \lambda \left| \bar{c}_{2,i}^i - \underline{c}_{2,i}^i \right|. \\ &\leq \left(1 - \gamma H_{ii}(c_1) + \lambda + \gamma \sum_{j=1, j \neq i}^N |H_{ij}(c_1)| \right) \|\bar{c}_1 - \underline{c}_1\|_\infty \\ &\quad + \lambda \|\bar{c}_2 - \underline{c}_2\|_\infty, \end{aligned} \quad (8.31)$$

where the last inequality uses the ∞ -norm to group the first and second terms.

Using Assumptions 8.2.3 and 8.2.5, we write the inequality

$$H_{ii}(c_1) = |H_{ii}(c_1)| \geq \sum_{j=1, j \neq i}^N |H_{ij}(c_1)| + \mu. \quad (8.32)$$

This implies that $-\gamma H_{ii}(c_1) + \gamma \sum_{j=1, j \neq i}^N |H_{ij}(c_1)| \leq -\gamma \mu$. Substituting (8.32) into

(8.31) yields

$$\left| \hat{\kappa}^i(\bar{c}) - \hat{\kappa}^i(\underline{c}) \right| \leq (1 - \gamma\mu + \lambda) \|\bar{c}_1 - \underline{c}_1\|_\infty + \lambda \|\bar{c}_2 - \underline{c}_2\|_\infty. \quad (8.33)$$

In step 4), applying (8.33) to (8.24) with $\bar{c} = z(k)$, $\underline{c} = z^*$, $\bar{c}_1 = z_1(k)$, $\bar{c}_2 = z_2(k)$, and $\underline{c}_1 = \underline{c}_2 = x^*$ yields

$$\|z_1(k+1) - x^*\|_\infty \leq (1 - \gamma\mu + \lambda) \|z_1(k) - x^*\|_\infty + \lambda \|z_2(k) - x^*\|_\infty, \quad (8.34)$$

for each $k \in \mathbb{N}$, which is no longer dependent on i . Applying (8.33) to $\|z_1(k) - x^*\|_\infty \leq \max_{i \in \mathcal{V}} |\hat{\kappa}^i(z(k-1)) - \hat{\kappa}^i(z^*)|$, which holds due to (8.24), leads to

$$\begin{aligned} & \|z_1(k+1) - x^*\|_\infty \\ & \leq (1 - \gamma\mu + \lambda) \left((1 - \gamma\mu + \lambda) \|z_1(k-1) - x^*\|_\infty + \lambda \|z_2(k-1) - x^*\|_\infty \right) \\ & \quad + \lambda \|z_2(k) - x^*\|_\infty \end{aligned} \quad (8.35)$$

for each $k \in \mathbb{N} \setminus \{0\}$. Recall that, by the definition of (1.11), $z_2(k) = z_1(k-1)$. Substituting such an expression for $z_2(k)$ into (8.35) and rearranging terms gives

$$\begin{aligned} \|z_1(k+1) - x^*\|_\infty & \leq \left((1 - \gamma\mu + \lambda)^2 + \lambda \right) \|z_1(k-1) - x^*\|_\infty \\ & \quad + \lambda (1 - \gamma\mu + \lambda) \|z_2(k-1) - x^*\|_\infty \end{aligned} \quad (8.36)$$

for each $k \in \mathbb{N} \setminus \{0\}$. Similarly, we use (8.33) to upper bound $\|z_2(k+1) - x^*\|_\infty$. Using (8.34), we obtain

$$\begin{aligned} \|z_2(k+1) - x^*\|_\infty & = \|z_1(k) - x^*\|_\infty \\ & \leq (1 - \gamma\mu + \lambda) \|z_1(k-1) - x^*\|_\infty \end{aligned}$$

$$+ \lambda \|z_2(k-1) - x^*\|_\infty \quad (8.37)$$

for each $k \in \mathbb{N} \setminus \{0\}$. Noting that

$$\|z_1(k-1) - x^*\|_\infty \leq \|z(k-1) - z^*\|_\infty \quad (8.38a)$$

$$\|z_2(k-1) - x^*\|_\infty \leq \|z(k-1) - z^*\|_\infty \quad (8.38b)$$

we further simplify (8.36) and (8.37) to

$$\begin{aligned} & \|z_1(k+1) - x^*\|_\infty \\ & \leq \left((1 - \gamma\mu + \lambda)^2 + \lambda + \lambda(1 - \gamma\mu + \lambda) \right) \|z(k-1) - z^*\|_\infty \end{aligned} \quad (8.39a)$$

$$\|z_2(k+1) - x^*\|_\infty \leq (1 - \gamma\mu + 2\lambda) \|z(k-1) - z^*\|_\infty \quad (8.39b)$$

for each $k \in \mathbb{N} \setminus \{0\}$. Using the bounds in (8.39) yields (8.19), where $a = \max\{a_1, a_2\}$, $a_1 = (1 - \gamma\mu + \lambda)^2 + \lambda + \lambda(1 - \gamma\mu + \lambda)$ and $a_2 = 1 - \gamma\mu + 2\lambda$.

All that remains is step 5), in which we show that $a_1, a_2 \in [0, 1)$. To this end, we first show that $\gamma\mu \in (0, 1)$. Using (8.30) with $\eta = z_1$ and Assumption 8.2.5, which implies that $\max_{i \in \mathcal{V}} \max_{z_1 \in \mathbb{R}^N} |H_{ii}(z_1)| \geq \mu$, we have

$$0 < \gamma\mu < \frac{\mu}{\max_{i \in \mathcal{V}} \max_{z_1 \in \mathbb{R}^N} |H_{ii}(z_1)|} \leq \frac{\mu}{\mu} = 1. \quad (8.40)$$

Then, since by definition $a_1 \geq 0$ and $a_2 \geq 0$, using (8.40) and $\lambda \in \left(0, \frac{\gamma\mu}{2}\right)$, we obtain

$$\begin{aligned} 0 \leq a_1 &= (1 - \gamma\mu + \lambda)^2 + \lambda + \lambda(1 - \gamma\mu + \lambda) \\ &< \left(1 - \gamma\mu + \frac{\gamma\mu}{2}\right)^2 + \frac{\gamma\mu}{2} + \frac{\gamma\mu}{2} \left(1 - \gamma\mu + \frac{\gamma\mu}{2}\right) \end{aligned} \quad (8.41)$$

This upper bound is equal to one since

$$\left(1 - \frac{\gamma\mu}{2}\right)^2 + \frac{\gamma\mu}{2} + \frac{\gamma\mu}{2} \left(1 - \frac{\gamma\mu}{2}\right) = 1 - \gamma\mu + \frac{\gamma^2\mu^2}{4} + \gamma\mu - \frac{\gamma^2\mu^2}{4} = 1. \quad (8.42)$$

Moreover, we have

$$0 \leq a_2 = 1 - \gamma\mu + 2\lambda < 1 - \gamma\mu + \gamma\mu = 1. \quad (8.43)$$

Therefore, for each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the synchronous algorithm $(D_{\text{sync}}, G_{\text{sync}})$ in (8.12) from $(z_{\circ}^1, z_{\circ}^2, \dots, z_{\circ}^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ satisfies (8.19), for each $k \in \mathbb{N} \setminus \{0\}$. \square

Although in Proposition 8.2.8 we establish that, for each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the synchronous algorithm $(D_{\text{sync}}, G_{\text{sync}})$ in (8.12) from $(z_{\circ}^1, z_{\circ}^2, \dots, z_{\circ}^N) \in (X \times X)^N$, the associated true value trajectory is contractive over two steps, contraction at each discrete time $k \in \mathbb{N}$ does not seem possible. We exploit the property in Proposition 8.2.8 in our forthcoming synchronous and asynchronous double-update heavy ball algorithms.

8.3 Synchronous, Double-Update Heavy Ball

In this section, we present Step 2) in our approach to solving Problem 8.1.1. Namely, we propose a synchronous algorithm that updates each agent's value $(z_{1,i}^i, z_{2,i}^i)$ twice for each discrete time $k \in \mathbb{N}$. Then, we establish an exponential convergence rate for the agents' decision variables based on the contractive property established in Section 8.2, and we show that it satisfies the Synchronous Convergence and Box Conditions in [59, Chapter 6, Assumption 2.1].

8.3.1 Modeling

The synchronous, double-update algorithm for the constrained heavy ball algorithm is given in Algorithm 7. In this algorithm, at each discrete time instant $k \in \mathbb{N}$, each agent $i \in \mathcal{V}$ computes the new value of $(z_{1,i}^i, z_{2,i}^i)$; see lines 3-5 of Algorithm 7. In particular, $z_{1,i}^i$ is updated to $\tilde{\kappa}^i(z^i)$ and $z_{2,i}^i$ is updated to $\kappa^i(z^i)$, where

$$\tilde{\kappa}^i(z^i) := \Pi_{X_i} \left[\kappa^i(z^i) - \gamma \nabla_i f(w_1^i(z_1^i)) + \lambda(\kappa^i(z^i) - z_{1,i}^i) \right] \quad (8.44)$$

where $\lambda > 0$ and $\gamma > 0$ come from (1.10), X_i comes from Assumption 8.2.1, z^i is defined in (8.4), z_1^i is defined via (8.5a), κ^i is defined in (8.6), and w_1^i is defined as

$$w_1^i(z_1^i) := (z_{1,1}^i, z_{1,2}^i, \dots, \kappa^i(z^i), \dots, z_{1,N}^i) \in X \quad (8.45)$$

where X is defined in (8.1). The function w_1^i collects the first update to the $z_{1,i}^i$ component of agent i 's decision variable and collects each of the $z_{1,\ell}^i$ components of the local copies of the decision variables of all other agents, for use in the computation of $\nabla_i f$ in (8.44). Then, starting at line 6 of Algorithm 7, each agent $i \in \mathcal{V}$ receives information from its neighbors and updates the entries $(z_{1,\ell}^i, z_{2,\ell}^i)$ of its state z^i , where $\ell \in \mathcal{N}_i$, in the following manner. For each of agent i 's essential neighbors, namely, for each $\ell \in \mathcal{N}_i$, $(z_{1,\ell}^i, z_{2,\ell}^i)$ is updated to $(z_{1,\ell}^\ell, z_{2,\ell}^\ell)$; see lines 7-9 of Algorithm 7. For each ℓ which is not an essential neighbor of i , $(z_{1,\ell}^i, z_{2,\ell}^i)$ is left unchanged; see lines 10-12 of Algorithm 7.

Now we model Algorithm 7 mathematically. From Algorithm 7, for each $i \in \mathcal{V}$, agent i has its state z^i in (8.4) updated via the following difference equation:

$$(z^i)^+ = \tilde{G}_{\text{sync}}^i(z^1, z^2, \dots, z^N) \quad z^i \in \tilde{D}_{\text{sync}}^i := X \times X, \quad (8.46)$$

Algorithm 7 Constrained, Synchronous, Double-Update Heavy Ball

- 1: For each $i \in \mathcal{V}$, set the initial state z_o^i to an arbitrary value in $X \times X$.
 - 2: **for** each $k \in \mathbb{N}$ **do**
 - 3: **for** each $i \in \mathcal{V}$ **do**
 - 4: Update $(z_{1,i}^i, z_{2,i}^i)$ in (8.2) to $(\tilde{\kappa}^i(z^i), \kappa^i(z^i))$, with $\tilde{\kappa}^i$ defined via (8.44) and κ^i defined in (8.6).
 - 5: **end for**
 - 6: **for** each $i \in \mathcal{V}$ **do**
 - 7: **for** each $\ell \in \mathcal{N}_i$ **do**
 - 8: Update $(z_{1,\ell}^i, z_{2,\ell}^i)$ in (8.3) to $(z_{1,\ell}^\ell, z_{2,\ell}^\ell)$.
 - 9: **end for**
 - 10: **for** each $\ell \notin \mathcal{N}_i$ **do**
 - 11: Keep $(z_{1,\ell}^i, z_{2,\ell}^i)$ constant.
 - 12: **end for**
 - 13: **end for**
 - 14: **end for**
-

where $\tilde{G}_{\text{sync}}^i$ is defined as

$$\tilde{G}_{\text{sync}}^i(z^1, z^2, \dots, z^N) := \left(g_1^i(z^1, z^2, \dots, z^N), g_2^i(z^1, z^2, \dots, z^N) \right) \quad (8.47)$$

where, for each $p \in \{1, 2\}$, g_p^i is defined in (8.9) and for each $s \in \mathcal{V}$, $g_{p,s}^i$ is defined as

$$g_{1,s}^i(z^1, z^2, \dots, z^N) := \begin{cases} \tilde{\kappa}^i(z^i) & \text{if } s = i \\ z_{1,s}^s & \text{if } s \in \mathcal{N}_i \\ z_{1,s}^i & \text{if } s \notin \mathcal{N}_i \end{cases} \quad (8.48)$$

if $p = 1$, where $\tilde{\kappa}^i$ is defined in (8.44), and as

$$g_{2,s}^i(z^1, z^2, \dots, z^N) := \begin{cases} \kappa^i(z^i) & \text{if } s = i \\ z_{2,s}^s & \text{if } s \in \mathcal{N}_i \\ z_{2,s}^i & \text{if } s \notin \mathcal{N}_i \end{cases} \quad (8.49)$$

if $p = 2$, where κ^i is defined via (8.6). Then, the full multiagent system corresponding to (8.46) is denoted as

$$\begin{aligned} (z^1, z^2, \dots, z^N)^+ &= \tilde{G}_{\text{sync}}(z^1, z^2, \dots, z^N) \\ (z^1, z^2, \dots, z^N) &\in \tilde{D}_{\text{sync}} := (X \times X)^N \end{aligned} \quad (8.50)$$

where \tilde{G}_{sync} is defined as

$$\begin{aligned} \tilde{G}_{\text{sync}}(z^1, z^2, \dots, z^N) &:= \\ &(\tilde{G}_{\text{sync}}^1(z^1, z^2, \dots, z^N), \tilde{G}_{\text{sync}}^2(z^1, z^2, \dots, z^N), \dots, \tilde{G}_{\text{sync}}^N(z^1, z^2, \dots, z^N)). \end{aligned} \quad (8.51)$$

8.3.2 Results for Algorithm 7

Under Assumptions 8.2.1, 8.2.3, and 8.2.5, the following lemma, used in some of the results to follow, shows that $x_i^* \in X_i$ is a fixed point of $\tilde{\kappa}^i$ in (8.44). To prove it, we use Lemma 8.2.7.

Lemma 8.3.1. *(Fixed point of $\tilde{\kappa}^i$ in (8.44)) Let f satisfy Assumption 8.2.3, let $X \subset \mathbb{R}^N$ in (8.1) satisfy Assumption 8.2.1, and let H satisfy Assumption 8.2.5 with $\mu > 0$. Then, for each $i \in \mathcal{V}$, $x_i^* \in X_i$ is a fixed point of $\tilde{\kappa}^i$ in (8.44).*

Proof. By Assumption 8.2.1, the set $X \subset \mathbb{R}^N$ is nonempty, compact, convex, and can be decomposed as in (8.1). Furthermore, by Assumption 8.2.5, the Hessian of f is diagonally dominant with $\mu > 0$. As discussed in Remark 8.2.6, Assumption 8.2.5 implies μ -strong convexity of f and, consequently, when H is μ -diagonally dominant, then f has a unique⁴ minimizer $x^* \in X$. By Lemma 8.2.7, $x_i^* \in X_i$ is a fixed point of κ^i , namely, $x_i^* = \kappa^i(z^*)$, where z^* is defined via (8.20). Then, since f is convex by Assumption 8.2.3, and since x^* minimizes f over X , we have, by

⁴See Footnote 2.

item (O1), of Proposition C.1.1, that, for each $y_{1,i}^i \in X_i$, (8.15) is satisfied. As a consequence of the property in (8.15), we have, for each $y_{1,i}^i \in X_i$, that

$$\begin{aligned} \langle y_{1,i}^i - x_i^*, x_i^* - \gamma \nabla_i f(x^*) + \lambda (x_i^* - x_i^*) - x_i^* \rangle &\leq 0 \\ \langle y_{1,i}^i - x_i^*, \kappa^i(z^*) - \gamma \nabla_i f(x^*) + \lambda (\kappa^i(z^*) - x_i^*) - x_i^* \rangle &\leq 0 \end{aligned} \quad (8.52)$$

where $\gamma > 0$ and $\lambda > 0$ come from (1.10). Next, since f is convex by Assumption 8.2.3 and since the set X is nonempty, compact, and convex by Assumption 8.2.1, then by item (P1) of Proposition C.1.2, given $v_{1,i}^i \in X_i$, $u_{1,i}^i \in X_i$ is equal to $\Pi_{X_i} [v_{1,i}^i]$ if and only if $\langle y_{1,i}^i - u_{1,i}^i, v_{1,i}^i - u_{1,i}^i \rangle \leq 0$ for all $y_{1,i}^i \in X_i$. Now, let $u_{1,i}^i = x_i^*$ and $v_{1,i}^i = \kappa^i(z^*) - \gamma \nabla_i f(x^*) + \lambda (\kappa^i(z^*) - x_i^*)$. Then, for each $i \in \mathcal{V}$, $x_i^* = \Pi_{X_i} [\kappa^i(z^*) - \gamma \nabla_i f(x^*) + \lambda (\kappa^i(z^*) - x_i^*)]$ by (8.52). Therefore, for each $i \in \mathcal{V}$, $x_i^* \in X_i$ is a fixed point of $\tilde{\kappa}^i$ in (8.44). \square

For the result to follow, we denote the portion of the algorithm $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) corresponding to the update of the true value z in (8.17) as

$$z^+ = g(z) \quad z \in X \times X \quad (8.53)$$

where g is defined as

$$g(z) := \left((\tilde{\kappa}^1(z^1), \tilde{\kappa}^2(z^2), \dots, \tilde{\kappa}^N(z^N)), (\kappa^1(z^1), \kappa^2(z^2), \dots, \kappa^N(z^N)) \right) \quad (8.54)$$

where $\tilde{\kappa}^i$ is defined via (8.44) and κ^i is defined in (8.6). Then, under Assumptions 8.2.1, 8.2.3, and 8.2.5, the following lemma establishes that, for each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the synchronous algorithm $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) from $(z_o^1, z_o^2, \dots, z_o^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$, solution to (8.53), has an exponential convergence rate. Additionally,

the following proposition shows that the ∞ -norm contraction property of such a convergence bound allows us to construct sets that satisfy both the Synchronous Convergence Condition in [59, Chapter 6, Assumption 2.1(a)] and the Box Condition in [59, Chapter 6, Assumption 2.1(b)]. To prove it, we use Lemma 8.2.7, Proposition 8.2.8, and Lemma 8.3.1.

Proposition 8.3.2. *Suppose $X \subset \mathbb{R}^N$, defined via (8.1), satisfies Assumption 8.2.1, f satisfies Assumption 8.2.3, H satisfies Assumption 8.2.5 with $\mu > 0$, and $z_\circ \in \mathcal{Z}_\circ := X \times X$ denotes the initial state of z in (8.17). For each $\gamma \in \left(0, \frac{1}{\max_{i \in \mathcal{V}} \max_{\eta \in X} |H_{ii}(\eta)|}\right)$, $\lambda \in \left(0, \frac{\gamma\mu}{2}\right)$, and each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the synchronous algorithm $(\tilde{D}_{sync}, \tilde{G}_{sync})$ in (8.50) from $(z_\circ^1, z_\circ^2, \dots, z_\circ^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ starting from z_\circ satisfies*

$$\|z(k) - z^*\|_\infty \leq a^k \|z_\circ - z^*\|_\infty \quad (8.55)$$

for all $k \in \mathbb{N}$, where $a := \max\{a_1, a_2\}$, $a_1 := (1 - \gamma\mu + \lambda)^2 + \lambda + \lambda(1 - \gamma\mu + \lambda) \in [0, 1)$, and $a_2 := 1 - \gamma\mu + 2\lambda \in [0, 1)$, and z^* is defined via (8.20). Furthermore, for each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the synchronous algorithm $(\tilde{D}_{sync}, \tilde{G}_{sync})$ in (8.50) from $(z_\circ^1, z_\circ^2, \dots, z_\circ^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ starting from z_\circ has sets defined as

$$\mathcal{Z}(k) := \left\{ z(k) \in X \times X : \|z(k) - z^*\|_\infty \leq a^k \|z_\circ - z^*\|_\infty \right\} \quad (8.56)$$

where $k \in \mathbb{N}$, for which the following properties hold:

1) For each $k \in \mathbb{N}$

$$\dots \subset \mathcal{Z}(k+1) \subset \mathcal{Z}(k) \subset \dots \subset \mathcal{Z}_\circ; \quad (8.57)$$

2) Synchronous Convergence Condition:

(a) For each $k \in \mathbb{N}$ and each $b \in \mathcal{Z}(k)$

$$g(b) \in \mathcal{Z}(k+1) \quad (8.58)$$

where g is defined via (8.54);

(b) For each sequence $\{b_k\}_{k \in \mathbb{N}}$ such that $b_k \in \mathcal{Z}(k)$ for each $k \in \mathbb{N}$, $\lim_{k \rightarrow \infty} \{b_k\}_{k \in \mathbb{N}} = z^*$ and z^* is a fixed point of g ;

3) Box Condition: For each $k \in \mathbb{N}$, there exist sets $\mathcal{Z}_i(k) \subset X_i \times X_i$, $i \in \mathcal{V}$, such that

$$\mathcal{Z}(k) = \mathcal{Z}_1(k) \times \mathcal{Z}_2(k) \times \dots \times \mathcal{Z}_N(k). \quad (8.59)$$

Proof. First, we must show that the map $\tilde{G}_{\text{sync}}^i$ in (8.47) is defined on \tilde{D}_{sync} and that $\tilde{G}_{\text{sync}}^i(\tilde{D}_{\text{sync}}) \subset \tilde{D}_{\text{sync}}$. By construction, κ^i in (8.6) is defined for each $z^i \in X \times X$ and $\tilde{\kappa}^i$ in (8.44) is defined for each $z^i \in X \times X$. In addition, by Assumption 8.2.1, the constraint set X is nonempty. Furthermore, by construction, $\tilde{G}_{\text{sync}}^i(z^1, z^2, \dots, z^N) \neq \emptyset$ for each $(z^1, z^2, \dots, z^N) \in (X \times X)^N$. Therefore, by [21, Definition 2.1], $\text{dom } \tilde{G}_{\text{sync}}^i = (X \times X)^N$, which implies that $\tilde{D}_{\text{sync}} \subset \text{dom } \tilde{G}_{\text{sync}}^i$, and each maximal solution to $(\tilde{D}_{\text{sync}}^i, \tilde{G}_{\text{sync}}^i)$ in (8.47)-(8.49) is complete. Since, by construction, $\tilde{G}_{\text{sync}}(z^1, z^2, \dots, z^N) \neq \emptyset$ for each $(z^1, z^2, \dots, z^N) \in (X \times X)^N$, and since $\tilde{G}_{\text{sync}}^i(z^1, z^2, \dots, z^N) \in X \times X$ for each $i \in \mathcal{V}$, due to κ^i in (8.6) and $\tilde{\kappa}^i$ in (8.44), then, by construction, $\tilde{G}_{\text{sync}}(z^1, z^2, \dots, z^N) \in (X \times X)^N$. Moreover, by Assumption 8.2.1, X is nonempty. Therefore, by [21, Definition 2.1],

$$\text{dom } \tilde{G}_{\text{sync}}(z^1, z^2, \dots, z^N) = (X \times X)^N \quad (8.60)$$

which implies that $\tilde{D}_{\text{sync}} \subset \text{dom } \tilde{G}_{\text{sync}}^i$, and each maximal solution to

$(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}}^i)$ in (8.50) is complete.

Next, to show that the associated true value trajectory $k \mapsto z(k)$, starting from $z_\circ \in \mathcal{Z}_\circ$, of each maximal solution to $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) from $(z_\circ^1, z_\circ^2, \dots, z_\circ^N) \in (X \times X)^N$, satisfies (8.55) for each $k \in \mathbb{N}$, we use properties of $(D_{\text{sync}}, G_{\text{sync}})$ in (8.12) and g in (8.54). First, f is \mathcal{C}^2 and convex by Assumption 8.2.1, X is nonempty, compact, and convex by Assumption 8.2.3, and H is μ -diagonally dominant by Assumption 8.2.5. Then, since for the algorithm in (8.50), $(z_{1,i}^i, z_{2,i}^i)$ in (8.2) is updated to $(\tilde{\kappa}^i(z^i), \kappa^i(z^i))$ for each $i \in \mathcal{V}$, during each iteration k – with $\tilde{\kappa}^i$ defined via (8.44) and κ^i defined in (8.6) – then one iteration of the algorithm $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) is the same as two consecutive iterations of the algorithm $(D_{\text{sync}}, G_{\text{sync}})$ in (8.12).

Therefore, since for each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the synchronous algorithm $(D_{\text{sync}}, G_{\text{sync}})$ in (8.12) from $(z_\circ^1, z_\circ^2, \dots, z_\circ^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ satisfies (8.19) for each $k \in \mathbb{N} \setminus \{0\}$ by Proposition 8.2.8, then for each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the algorithm $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) from $(z_\circ^1, z_\circ^2, \dots, z_\circ^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ starting from z_\circ satisfies

$$\|z(k) - z^*\|_\infty \leq a \|z(k-1) - z^*\|_\infty \quad (8.61)$$

for each $k \in \mathbb{N} \setminus \{0\}$, where $a \in [0, 1)$ is defined below (8.55) and z^* is defined via (8.20). Taking the product of (8.61) over k iterations yields (8.55) for each $k \in \mathbb{N}$.

Since, for each maximal solution to the algorithm $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) from $(z_\circ^1, z_\circ^2, \dots, z_\circ^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$, starting from z_\circ satisfies (8.55) for each $k \in \mathbb{N}$, and since $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ is synchronous, then the sets defined in (8.56) for each $k \in \mathbb{N}$ are nonempty.

We first prove that item 1) holds. Since, for each maximal solution to the al-

gorithm $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) from $(z_{\circ}^1, z_{\circ}^2, \dots, z_{\circ}^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$, starting from $z_{\circ} \in \mathcal{Z}_{\circ}$ satisfies (8.55) for each $k \in \mathbb{N}$, where $a \in [0, 1)$ is defined below (8.55) and z^* is defined via (8.20), then the property in (8.57) holds for each $k \in \mathbb{N}$, for the sets in (8.56).

Furthermore, the property (8.58) in item 2)a holds for each $k \in \mathbb{N}$ and each $b \in \mathcal{Z}(k)$. This is true since the associated true value trajectory $k \mapsto z(k)$, starting from $z_{\circ} \in \mathcal{Z}_{\circ}$ satisfies (8.55) for all $k \in \mathbb{N}$. Then, due to $a \in [0, 1)$, $a^{k+1} \leq a^k$ implies

$$\|g(b) - z^*\|_{\infty} \leq a^{k+1} \|z_{\circ} - z^*\|_{\infty} \quad (8.62)$$

for each $k \in \mathbb{N}$ and each $b \in \mathcal{Z}(k)$, where g is defined via (8.54).

Next, we prove that item 2)b holds. Since the sets $\mathcal{Z}(k)$, defined via (8.56) satisfy 8.57, then by [76, Exercise 4.3(b)], each sequence $\{b_k\}_{k \in \mathbb{N}}$ such that $b_k \in \mathcal{Z}(k)$ for each $k \in \mathbb{N}$ has a limit, which is given by

$$\lim_{k \rightarrow \infty} \{b_k\}_{k \in \mathbb{N}} = \lim_{k \rightarrow \infty} \mathcal{Z}(k) = \bigcap_{k \in \mathbb{N}} \overline{\mathcal{Z}(k)} = z^*. \quad (8.63)$$

Such a limit is a fixed point of g in (8.50) since, by Lemma 8.2.7, $\kappa^i(z^*) = x_i^*$ for each $i \in \mathcal{V}$ and, by Lemma 8.3.1, $\tilde{\kappa}^i(z^*) = x_i^*$ for each $i \in \mathcal{V}$. Hence,

$$g(z^*) = z^*. \quad (8.64)$$

Finally, we prove that item 3) holds. Since, for each maximal solution to $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) from $(z_{\circ}^1, z_{\circ}^2, \dots, z_{\circ}^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ starting from z_{\circ} has sets $\mathcal{Z}(k) \subset \mathcal{Z}_{\circ}$ satisfying (8.57) for each $k \in \mathbb{N}$, and since a key property of the ∞ -norm is that its unit sphere has the Box Condition (see [59, Chapter 6.3]), then the sets $\mathcal{Z}(k)$ in (8.56) also satisfy (8.59), for each $k \in \mathbb{N}$. \square

8.4 Asynchronous, Double-Update Heavy Ball

In this section, we design an asynchronous version of the algorithm in Section 8.3 and use Proposition 8.3.2 to establish that, since for each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the synchronous algorithm $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) from $(z_{\circ}^1, z_{\circ}^2, \dots, z_{\circ}^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ starting from $z_{\circ} \in \mathcal{Z}_{\circ}$ has exponential convergence and satisfies the Synchronous Convergence and Box Conditions in [59, Chapter 6, Assumption 2.1], then the forthcoming asynchronous algorithm also has an exponential convergence rate.

8.4.1 Modeling

As has been done in [50], we want to distribute the discrete-time heavy ball among multiple agents while allowing agents to compute and share information as asynchronously as possible. There are two behaviors that could be asynchronous:

- 1) *Computation of Updates to Agents' Variables:* Individual agents may perform updates at different times. Namely, the subset of times at which distinct agents i and ℓ compute updates need not have any relationship;
- 2) *Communication of Updated Agents' Variables:* Communication of the agents' variables is also totally asynchronous.

To allow these behaviors to be totally asynchronous, while still preserving contraction of the update law, we propose an algorithm that uses the update law $\tilde{\kappa}^i$ in (8.44), but with each agent $i \in \mathcal{V}$ executing $\tilde{\kappa}^i$ and communicating to other agents at potentially different times, with arbitrarily long delays between updates.

The asynchronous, double-update algorithm for constrained heavy ball is summarized in Algorithm 8. Since computations are totally asynchronous, then we

denote the set of times at which agent i computes updates as the unbounded set $K^i \subset \mathbb{N}$. Note that each $i \in \mathcal{V}$ has its own K^i . At each discrete time instant $k \in \mathbb{N}$, if $k \in K^i$ – which defines the time of a *computation event* – the corresponding agent i computes the value of $(z_{1,i}^i, z_{2,i}^i)$; see lines 4-5 of Algorithm 8. In particular, if $k \in K^i$, then $z_{1,i}^i$ is updated to $\tilde{\kappa}^i(z^i)$ in (8.44) and $z_{2,i}^i$ is updated to $\kappa^i(z^i)$ in (8.6). Then, agent i sends $(z_{1,i}^i, z_{2,i}^i)$ to all agents $\ell \in \mathcal{N}_i$; see line 6 of Algorithm 8. Due to the possibility of communication delays, such information might not be received for some time, and might be received at different times by different essential neighbors.

Since communications are totally asynchronous, then we denote the set of times at which agent i receives information from agent $\ell \in \mathcal{N}_i$ as the unbounded set $R^{i,\ell} \subset \mathbb{N}$. Note that each $(i, \ell) \in \mathcal{E}$ has its own $R^{i,\ell}$. Then, at each discrete time instant $k \in \mathbb{N}$, if $k \in R^{i,\ell}$ – which defines the time of the reception of information, referred to as a *communication event* – agent i updates $(z_{1,\ell}^i, z_{2,\ell}^i)$ of its state z^i to

$$(z_{1,\ell}^\ell(\tau_\ell^i(k)), z_{2,\ell}^\ell(\tau_\ell^i(k))) \in X_\ell \times X_\ell. \quad (8.65)$$

where

$$\tau_\ell^i(k) \in K^\ell \subset \mathbb{N} \quad (8.66)$$

denotes the time at which agent ℓ originally computed the value of its decision variable onboard agent i at time k . Note that $\tau_i^i(k) = k$ for all $i \in \mathcal{V}$; see lines 8-12 of Algorithm 8.

To explain how we model the delay between sending and receiving information, and the role of τ_ℓ^i in such delay, we use a simple two agent example, as follows. Let $i = 1$ and $\ell = 2$, let⁵ $K^1 = \{1\}$, and let $R^{2,1} = \{3\}$. At $k = 1$, agent 1

⁵Although the sets K^i and $R^{\ell,i}$ are unbounded, we do not consider unboundedness in this example, for simplicity. This two agent example also holds for unbounded sets K^1 and $R^{2,1}$ such that $1 \in K^1$, $0, 2, 3, 4 \notin K^1$, $3 \in R^{2,1}$, and $0, 1, 2, 4 \notin R^{2,1}$.

updates $(z_{1,1}^1, z_{2,1}^1)$ to $(\tilde{\kappa}^1(z^1), \kappa^1(z^1))$, and sends such a value to agent 2. But such a value does not arrive at agent 2 until $k = 3$. When agent 2 receives agent 1's decision variable at $k = 3$, agent 2 updates its value of $(z_{1,1}^2, z_{2,1}^2)$ to $(z_{1,1}^1(\tau_1^2(3)), z_{2,1}^1(\tau_1^2(3)))$, where $\tau_1^2(3) = 1 \in K^1$. In this way, $(z_{1,1}^2(4), z_{2,1}^2(4)) = (z_{1,1}^1(2), z_{2,1}^1(2)) = (\tilde{\kappa}^1(z^1(1)), \kappa^1(z^1(1)))$.

For each ℓ which is not an essential neighbor of i , $(z_{1,\ell}^i, z_{2,\ell}^i)$ is left unchanged; see lines 13-15 of Algorithm 8.

Algorithm 8 Constrained, Asynchronous, Double-Update Heavy Ball

- 1: For each $i \in \mathcal{V}$, set the initial state z_o^i to an arbitrary value in $X \times X$.
 - 2: **for** each $k \in \mathbb{N}$ **do**
 - 3: **for** each $i \in \mathcal{V}$ **do**
 - 4: **if** $k \in K^i$ **then**
 - 5: Update $(z_{1,i}^i, z_{2,i}^i)$ in (8.2) to $(\tilde{\kappa}^i(z^i), \kappa^i(z^i))$, with $\tilde{\kappa}^i$ defined via (8.44) and κ^i defined in (8.6);
 - 6: Agent i sends $(z_{1,i}^i, z_{2,i}^i)$ to all agents $\ell \in \mathcal{N}_i$. Due to communication delays, it may not be received for some time.
 - 7: **end if**
 - 8: **for** each $\ell \in \mathcal{N}_i$ **do**
 - 9: **if** $k \in R^{i,\ell}$ **then**
 - 10: Update $(z_{1,\ell}^i, z_{2,\ell}^i)$ in (8.3) to $(z_{1,\ell}^\ell(\tau_\ell^i(k)), z_{2,\ell}^\ell(\tau_\ell^i(k)))$ in (8.65).
 - 11: **end if**
 - 12: **end for**
 - 13: **for** each $\ell \notin \mathcal{N}_i$ **do**
 - 14: Keep $(z_{1,\ell}^i, z_{2,\ell}^i)$ constant.
 - 15: **end for**
 - 16: **end for**
 - 17: **end for**
-

Now we model Algorithm 8 mathematically. From Algorithm 8, for each $i \in \mathcal{V}$, agent i has its state z^i in (8.4) updated via the following difference equation:

$$(z^i)^+ = G_{\text{async}}^i(z^1, z^2, \dots, z^N) \quad z^i \in D_{\text{async}}^i := X \times X, \quad (8.67)$$

where G_{async}^i is defined as

$$G_{\text{async}}^i(z^1, z^2, \dots, z^N) := (g_1^i(z^1, z^2, \dots, z^N), g_2^i(z^1, z^2, \dots, z^N)) \quad (8.68)$$

where, for each $p \in \{1, 2\}$, g_p^i is defined in (8.9) and, for each $s \in \mathcal{V}$, $g_{p,s}^i$ is defined as

$$g_{1,s}^i(z^1, z^2, \dots, z^N) := \begin{cases} \tilde{\kappa}^i(z^i) & \text{if } s = i, \text{ at each computation event} \\ z_{1,s}^s & \text{if } s \in \mathcal{N}_i, \text{ at each communication event} \\ z_{1,s}^i & \text{otherwise} \end{cases} \quad (8.69)$$

if $p = 1$, where $\tilde{\kappa}^i$ is defined in (8.44), and as

$$g_{2,s}^i(z^1, z^2, \dots, z^N) := \begin{cases} \kappa^i(z^i) & \text{if } s = i, \text{ at each computation event} \\ z_{2,s}^s & \text{if } s \in \mathcal{N}_i, \text{ at each communication event} \\ z_{2,s}^i & \text{otherwise} \end{cases} \quad (8.70)$$

if $p = 2$, where κ^i is defined via (8.6). In (8.69)-(8.70), a computation event by agent i occurs when $k \in K^i$ and a communication event occurs when agent i receives information from agent s at $k \in R^{i,s}$. Due to this, the maps g_1^i and g_2^i depend on the current and past state values. Such a dependency is omitted in (8.69)-(8.70), for simplicity of notation.

Then, the full multiagent system corresponding to (8.67) is denoted as

$$\begin{aligned} (z^1, z^2, \dots, z^N)^+ &= G_{\text{async}}(z^1, z^2, \dots, z^N) \\ (z^1, z^2, \dots, z^N) &\in D_{\text{async}} := (X \times X)^N \end{aligned} \quad (8.71)$$

where G_{async} is defined as

$$G_{\text{async}}(z^1, z^2, \dots, z^N) := \left(G_{\text{async}}^1(z^1, z^2, \dots, z^N), G_{\text{async}}^2(z^1, z^2, \dots, z^N), \dots, G_{\text{async}}^N(z^1, z^2, \dots, z^N) \right) \quad (8.72)$$

8.4.2 Forward Invariance of $(X \times X)^N$ for Algorithm 8

Under Assumptions 8.2.1 and 8.2.3, the set $X \times X$ is forward invariant, as in Definition 2.7.2, for $(D_{\text{async}}^i, G_{\text{async}}^i)$ in (8.67)-(8.70) and, consequently, $(X \times X)^N$ is forward invariant for $(D_{\text{async}}, G_{\text{async}})$ in (8.71), as shown in the following lemma.

Lemma 8.4.1. *(Forward invariance of $(X \times X)^N$ for $(D_{\text{async}}, G_{\text{async}})$ in (8.71))* Let $X \subset \mathbb{R}^N$ satisfy Assumption 8.2.1, let f satisfy Assumption 8.2.3, and let $z_{\circ}^i \in X \times X$ for each $i \in \mathcal{V}$. Then, $(X \times X)^N$ is forward invariant for $(D_{\text{async}}, G_{\text{async}})$ in (8.71).

Proof. First, we show that each maximal solution to $(D_{\text{async}}^i, G_{\text{async}}^i)$ is complete and, by extension, each maximal solution to $(D_{\text{async}}, G_{\text{async}})$ is complete. To that end, we must show that the map G_{async}^i in (8.68) is defined on D_{async} and that $G_{\text{async}}^i(D_{\text{async}}) \subset D_{\text{async}}$. By construction, κ^i in (8.6) is defined for each $z^i \in X \times X$ and $\tilde{\kappa}^i$ in (8.44) is defined for each $z^i \in X \times X$. In addition, by Assumption 8.2.1, the constraint set X is nonempty. Furthermore, by construction, $G_{\text{async}}^i(z^1, z^2, \dots, z^N) \neq \emptyset$ for each $(z^1, z^2, \dots, z^N) \in (X \times X)^N$. Therefore, by [21, Definition 2.1], $\text{dom } G_{\text{async}}^i = (X \times X)^N$, which implies that $D_{\text{async}} \subset \text{dom } G_{\text{async}}^i$, and each maximal solution to $(D_{\text{async}}^i, G_{\text{async}}^i)$ in (8.68)-(8.70) is complete. Since, by construction, $G_{\text{async}}(z^1, z^2, \dots, z^N) \neq \emptyset$ for each $(z^1, z^2, \dots, z^N) \in (X \times X)^N$, and since $G_{\text{async}}^i(z^1, z^2, \dots, z^N) \in X \times X$ for each $i \in \mathcal{V}$, due to κ^i in (8.6) and $\tilde{\kappa}^i$ in (8.44), then, by construction, $G_{\text{async}}(z^1, z^2, \dots, z^N) \in (X \times X)^N$. Moreover, by Assumption 8.2.1, X is

nonempty. Therefore, by [21, Definition 2.1],

$$\text{dom } G_{\text{async}}(z^1, z^2, \dots, z^N) = (X \times X)^N \quad (8.73)$$

which implies that $D_{\text{async}} \subset \text{dom } G_{\text{async}}$, and each maximal solution to $(D_{\text{async}}, G_{\text{async}})$ in (8.71) is complete.

Then, to show forward invariance of $(X \times X)^N$, we start first with a single agent i . First, note that since f is \mathcal{C}^2 by Assumption 8.2.3, then $\nabla_i f$ exists for each $i \in \mathcal{V}$. Then, by definition, κ^i in (8.6) projects into the set X_i and $\tilde{\kappa}^i$ in (8.44) projects into the set X_i . In addition, the communicated values $(z_{1,s}^s, z_{2,s}^s)$ in the second cases of (8.69) and (8.70), for each $s \in \mathcal{N}_i$, are in the set $X_s \times X_s$ since κ^s in (8.6) projects into the set X_s and $\tilde{\kappa}^s$ in (8.44) projects into the set X_s , for each $s \in \mathcal{N}_i$. Moreover, any values $(z_{1,s}^i, z_{2,s}^i)$ which remain unchanged – due to either $s \notin \mathcal{N}_i$, $s \in \mathcal{N}_i$ with no communication event, or $s = i$ with no communication event – are already in the set $X_s \times X_s$ since $z_o^i \in X \times X$ for each $i \in \mathcal{V}$. Therefore, each update $G_{\text{async}}^i(z^1, z^2, \dots, z^N) \in X \times X$, where X in (8.1) is a nonempty set by Assumption 8.2.1, and, since each maximal solution to $(D_{\text{async}}^i, G_{\text{async}}^i)$ is complete, $X \times X$ is forward invariant for $(D_{\text{async}}^i, G_{\text{async}}^i)$, for each $i \in \mathcal{V}$.

Since $X \times X$ is forward invariant for $(D_{\text{async}}^i, G_{\text{async}}^i)$ in (8.67)-(8.70), for each $i \in \mathcal{V}$, then due to the definition of $(D_{\text{async}}, G_{\text{async}})$ in (8.71) and due to X in (8.1) being a nonempty set by Assumption 8.2.1, $G_{\text{async}}(D_{\text{async}}) \subset (X \times X)^N$ and, since each maximal solution to $(D_{\text{async}}, G_{\text{async}})$ is complete, $(X \times X)^N$ is forward invariant for $(D_{\text{async}}, G_{\text{async}})$. \square

8.4.3 Convergence rate of Algorithm 8

For the forthcoming result for the asynchronous algorithm $(D_{\text{async}}, G_{\text{async}})$ in (8.71), we impose the following assumption.

Assumption 8.4.2. (*Computations and Communications*): For each $i \in \mathcal{V}$ and each $\ell \in \mathcal{N}_i$, the sets $K^i \subset \mathbb{N}$ and $R^{i,\ell} \subset \mathbb{N}$ are unbounded. If $\{k_s\}_{s \in \mathbb{N}}$ is an increasing sequence of times in K^i , then for each solution to the algorithm in (8.71), $\lim_{s \rightarrow \infty} \tau_i^\ell(k_s) = \infty$ for each $\ell \in \mathcal{V}$ such that $i \in \mathcal{N}_\ell$, where τ_i^ℓ is defined via (8.66).

Remark 8.4.3. Assumption 8.4.2 ensures that no agent ever stops computing or communicating, though delays can be arbitrarily large.

In the upcoming result, the convergence rate of (8.71) can be computed by leveraging results in [64] and [50] in terms of the number of operations – namely, $\text{ops}(k)$ – the agents have completed (counted in the appropriate sequence). Namely, we count operations as follows. Initially, we set $\text{ops}(0) = 0$. Then, after all agents $i \in \mathcal{V}$ have updated $(z_{1,i}^i, z_{2,i}^i)$ to $(\tilde{\kappa}^i(z^i), \kappa^i(z^i))$ and have sent such updates to and had such updates received by all essential neighbors $\ell \in \mathcal{N}_i$ – say, by time k' – we increment ops to $\text{ops}(k') = 1$. Note that it is possible for any agent i to compute and send – and essential neighbors $\ell \in \mathcal{N}_i$ to receive – multiple updates of $(z_{1,i}^i, z_{2,i}^i)$, between $\text{ops}(0) = 0$ and $\text{ops}(k') = 1$. In other words, different subsequences of $\{k_s\}_s \in \mathbb{N}$ in K^i exist for each $i \in \mathcal{V}$ and different subsequences of $\{k_s\}_s \in \mathbb{N}$ in $R^{i,\ell}$ exist for each $(i, \ell) \in \mathcal{E}$, between $\text{ops}(0) = 0$ and $\text{ops}(k') = 1$. After $\text{ops}(k') = 1$, then we wait until all agents $i \in \mathcal{V}$ have subsequently computed a new update of $(z_{1,i}^i, z_{2,i}^i)$ and such updates have been sent to and received by all other essential neighbors. If this occurs at time k'' , then we set $\text{ops}(k'') = 2$, and this process continues.

For the result to follow, we denote the portion of the algorithm $(D_{\text{async}}, G_{\text{async}})$ in (8.71) corresponding to the update of the true value z in (8.17) as

$$z^+ = g(z) \quad z \in X \times X \quad (8.74)$$

where g is defined as

$$g(z) := \left(\left(\tilde{g}_{1,1}^1(z^1), \tilde{g}_{1,2}^2(z^2), \dots, \tilde{g}_{1,N}^N(z^N) \right), \left(\tilde{g}_{2,1}^1(z^1), \tilde{g}_{2,2}^2(z^2), \dots, \tilde{g}_{2,N}^N(z^N) \right) \right) \quad (8.75)$$

where, for each $i \in \mathcal{V}$, $\tilde{g}_{1,i}^i$ and $\tilde{g}_{2,i}^i$ are defined as

$$\tilde{g}_{1,i}^i(z^i) := \begin{cases} \tilde{\kappa}^i(z^i) & \text{at each computation event by agent } i \\ z_{1,i}^i & \text{otherwise} \end{cases} \quad (8.76a)$$

$$\tilde{g}_{2,i}^i(z^i) := \begin{cases} \kappa^i(z^i) & \text{at each computation event by agent } i \\ z_{2,i}^i & \text{otherwise} \end{cases} \quad (8.76b)$$

where $\tilde{\kappa}^i$ is defined via (8.44), κ^i is defined in (8.6), and where a computation event by agent i occurs when $k \in K^i$. Since (8.74)-(8.76) represents only the updates of the true values of the decision variables, and not the local copies onboard each agent i , then communication events are not represented in (8.76). The model in (8.74)-(8.76) is the asynchronous version of (8.53)-(8.54), which we use in the forthcoming proof of Theorem 8.4.4 to show that, since the true value trajectory $k \mapsto z(k)$ of each maximal solution to $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ satisfies items 1)-?? of Proposition 8.3.2, then z^* is a fixed point of g in (8.75) for the true value trajectory $k \mapsto z(k)$ of each maximal solution to $(D_{\text{async}}, G_{\text{async}})$.

The following theorem establishes that each maximal solution

$k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the algorithm $(D_{\text{async}}, G_{\text{async}})$, defined via (8.71) from $(z_o^1, z_o^2, \dots, z_o^N) \in (X \times X)^N$, converges exponentially to (z^*, z^*, \dots, z^*) . To prove it, we use Propositions 8.3.2 and B.1.2.

Theorem 8.4.4. *(Exponential convergence rate for $(D_{\text{async}}, G_{\text{async}})$ in (8.71))*
 Suppose $X \subset \mathbb{R}^N$ satisfies Assumption 8.2.1, f satisfies Assumption 8.2.3, H satisfies Assumption 8.2.5 with $\mu > 0$, and $z_o \in \mathcal{Z}_o := X \times X$ denotes the initial state of z in (8.17). For each $\gamma \in \left(0, \frac{1}{\max_{i \in \mathcal{V}} \max_{\eta \in X} |H_{ii}(\eta)|}\right)$ and $\lambda \in \left(0, \frac{\gamma\mu}{2}\right)$, each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the asynchronous, double-update heavy ball algorithm $(D_{\text{async}}, G_{\text{async}})$ from $(z_o^1, z_o^2, \dots, z_o^N) \in \mathcal{Z}_o^N$, for which Assumption 8.4.2 holds, satisfies

$$\max_{i \in \mathcal{V}} \|z^i(k) - z^*\|_\infty \leq a^{\text{ops}(k)} \max_{i \in \mathcal{V}} \|z_o^i - z^*\|_\infty \quad (8.77)$$

for all $k \in \mathbb{N}$, where $a := \max\{a_1, a_2\}$, $a_1 := (1 - \gamma\mu + \lambda)^2 + \lambda + \lambda(1 - \gamma\mu + \lambda) \in [0, 1)$, $a_2 := 1 - \gamma\mu + 2\lambda \in [0, 1)$, and z^* is defined via (8.20).

Proof. To establish the result, we proceed as follows.

- 1) We use Propositions 8.3.2 and B.1.2 to show that, for each maximal solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to the algorithm $(D_{\text{async}}, G_{\text{async}})$ in (8.71) from $(z_o^1, z_o^2, \dots, z_o^N) \in \mathcal{Z}_o^N$, the associated true value trajectory $k \mapsto z(k)$ starting from $z_o \in \mathcal{Z}_o$ has the property that, for each sequence $\{b_{\text{ops}(k)}\}_{\text{ops}(k) \in \mathbb{N}}$ such that $b_{\text{ops}(k)} \in \mathcal{Z}(\text{ops}(k))$ for each $\text{ops}(k) \in \mathbb{N}$, $\lim_{\text{ops}(k) \rightarrow \infty} \{b_{\text{ops}(k)}\}_{\text{ops}(k) \in \mathbb{N}} = z^*$ and z^* is a fixed point of g in (8.75). Note that the function ops , which was described below Remark 8.4.3, is solution dependent.
- 2) We use Proposition 8.3.2, Lemma 8.4.1, and the function ops to compute the desired convergence rate for the asynchronous, double-update algorithm

$(D_{\text{async}}, G_{\text{async}})$ in (8.71) by picking an arbitrary solution, analyzing its behavior between $\text{ops}(0) = 0$ and $\text{ops}(k'') = 1$:

- (a) First, we analyze computation events between $\text{ops}(0) = 0$ and $\text{ops}(k') = 1$;
- (b) Then, we analyze communication events between $\text{ops}(0) = 0$ and $\text{ops}(k'') = 1$;
- (c) Then, we complete the proof by iterating the process in 2)a-2)b above over subsequent cycles of ops.

Proceeding with step 1), we pick a solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to $(D_{\text{async}}, G_{\text{async}})$ from $(z_o^1, z_o^2, \dots, z_o^N) \in \mathcal{Z}_o^N$, which has the solution dependent function ops. By Proposition B.1.2, for such a solution to $(D_{\text{async}}, G_{\text{async}})$, the associated true value trajectory $k \mapsto z(k)$ starting from $z_o \in \mathcal{Z}_o$ is such that each sequence $\{b_{\text{ops}(k)}\}_{\text{ops}(k) \in \mathbb{N}}$ such that $b_{\text{ops}(k)} \in \mathcal{Z}(\text{ops}(k))$ for each $\text{ops}(k) \in \mathbb{N}$ has a limit, which is given by $\lim_{\text{ops}(k) \rightarrow \infty} \{b_{\text{ops}(k)}\}_{\text{ops}(k) \in \mathbb{N}} = z^*$, and such a limit is a fixed point of $g(z)$ in (8.74), due to the following. By Proposition 8.3.2, for each maximal solution to the algorithm $(\tilde{D}_{\text{sync}}, \tilde{G}_{\text{sync}})$ in (8.50) from $(z_o^1, z_o^2, \dots, z_o^N) \in (X \times X)^N$, the associated true value trajectory $k \mapsto z(k)$ starting from z_o satisfies the bound in (8.55) for all $k \in \mathbb{N}$, where a is defined below (8.77) and z^* is defined via (8.20). Furthermore, by Proposition 8.3.2, the associated true value trajectory $k \mapsto z(k)$ of such solutions, starting from z_o , has nonempty sets defined in (8.56) for which (8.57) is satisfied, the Synchronous Convergence Condition in item 2) of Proposition 8.3.2 holds, and the Box Condition in item 3) of Proposition 8.3.2 holds.

Next, for step 2), the solution $k \mapsto (z^1(k), z^2(k), \dots, z^N(k))$ to $(D_{\text{async}}, G_{\text{async}})$ from $(z_o^1, z_o^2, \dots, z_o^N) \in \mathcal{Z}_o^N$ that we picked in step 1) has the

solution-dependent sets of computation times K^i for each $i \in \mathcal{V}$, the solution-dependent sets of communication times $R^{i,\ell}$ for each $(i, \ell) \in \mathcal{E}$, and the solution-dependent function τ_ℓ^i defined via (8.66). Recall that for a single incrementation of the solution-dependent function ops by 1, it is possible for any agent i to update $(z_{1,i}^i, z_{2,i}^i)$ to $(\tilde{\kappa}^i(z^i), \kappa^i(z^i))$ multiple times, where $\tilde{\kappa}^i$ is defined via (8.44) and κ^i is defined in (8.6). However, each agent $i \in \mathcal{V}$ computes such an update, and such an update is sent to and received by each essential neighbor $\ell \in \mathcal{N}_i$, at least once.

For step 2)a, the initial state of the solution is set to $z_\circ^i \in \mathcal{Z}_\circ$ for each $i \in \mathcal{V}$. Suppose that, for such a solution, at time $k_i \in K^i$, agent i updates $(z_{1,i}^i, z_{2,i}^i)$ to $(\tilde{\kappa}^i(z^i), \kappa^i(z^i))$. Then, $(z_{1,i}^i(k_i + 1), z_{2,i}^i(k_i + 1)) \in \mathcal{Z}_i(1) \subset X_i \times X_i$, where X_i comes from Assumption 8.2.1. Then, by Proposition 8.3.2 and step 1), at the discrete instant $k' = \max_{i \in \mathcal{V}} k_i + 1$, we find that $(z_{1,i}^i(k'), z_{2,i}^i(k')) \in \mathcal{Z}_i(1)$ for each $i \in \mathcal{V}$. Note that, for each discrete time instant $k \in \{0, 1, \dots, k_i\}$, $(z_{1,i}^i(k), z_{2,i}^i(k)) \in \mathcal{Z}_{i_\circ} = X_i \times X_i$ for each $i \in \mathcal{V}$, whether or not an update of $(z_{1,i}^i, z_{2,i}^i)$ to $(\tilde{\kappa}^i(z^i), \kappa^i(z^i))$ has occurred for any agent $i \in \mathcal{V}$. Such a property is due to $z_\circ^i \in X \times X$, the forward invariance of $X \times X$ by Lemma 8.4.1, and – in the case of a computation event – the contractivity of $(\tilde{\kappa}^i(z^i), \kappa^i(z^i))$ by Proposition 8.3.2. Moreover, since $\mathcal{Z}_i(1) \subset \mathcal{Z}_{i_\circ}$, then $(z_{1,i}^i(k_i + 1), z_{2,i}^i(k_i + 1)) \in \mathcal{Z}_{i_\circ}$ is also satisfied for each $i \in \mathcal{V}$.

Then, for step 2)b, suppose that, after each agent $i \in \mathcal{V}$ has computed at least one update $(\tilde{\kappa}^i(z^i), \kappa^i(z^i))$ to its decision variable, each agent i sends such information, which is received by each agent $\ell \in \mathcal{N}_i$ after some time. Consequently, each agent $i \in \mathcal{V}$ receives $(z_{1,\ell}^\ell(\tau_\ell^i(k_{i,\ell})), z_{2,\ell}^\ell(\tau_\ell^i(k_{i,\ell})))$ from each agent $\ell \in \mathcal{N}_i$ at $k_{i,\ell} \in R^{i,\ell}$. Then, by Proposition 8.3.2 and step 1), at the discrete instant $k'' = \max_{i \in \mathcal{V}} \max_{\ell \in \mathcal{N}_i} k_{i,\ell}$, each agent $i \in \mathcal{V}$ has $(z_{1,\ell}^i(k''), z_{2,\ell}^i(k'')) \in \mathcal{Z}_\ell(1) \subset X_\ell \times X_\ell$ for each $\ell \in \mathcal{N}_i$. Note that, for each discrete time instant $k \in \{0, 1, \dots, k_{i,\ell}\}$, whether

or not any agent i receives information from any agent $\ell \in \mathcal{N}_i$, $(z_{1,\ell}^i(k), z_{2,\ell}^i(k)) \in \mathcal{Z}_{\ell_o} = X_\ell \times X_\ell$ for each $i \in \mathcal{V}$ and each $\ell \in \mathcal{N}_i$, where X_ℓ comes from Assumption 8.2.1. Such a property is due to $z_o^i \in X \times X$, the forward invariance of $X \times X$ by Lemma 8.4.1, and the contractivity of $(\tilde{\kappa}^\ell(z^\ell), \kappa^\ell(z^\ell))$ by Proposition 8.3.2.

Finally, for step 2)c, $z(k'') \in \mathcal{Z}(1)$ due to steps 2)a-2)b, and this is satisfied precisely when a single cycle has completed, namely, when $\text{ops}(k'') = 1$. Iterating such a process for subsequent cycles of ops completes the proof.

□

8.5 Numerical Example

Example 8.5.1. *To demonstrate the effectiveness of the asynchronous, double-update heavy ball algorithm $(D_{\text{async}}, G_{\text{async}})$ in (8.71), we compare it in simulation with a multiagent constrained gradient descent algorithm. In particular, we compare $(D_{\text{async}}, G_{\text{async}})$ with a version of the asynchronous primal-dual algorithm for constrained gradient descent, in [50], with the dual variables $\eta(t)$ fixed to zero. First, we compare the convergence rates of $(D_{\text{async}}, G_{\text{async}})$ and the algorithm [50] analytically. For the asynchronous primal-dual algorithm in [50], when the dual variables $\eta(t)$ are fixed at zero, the constrained update law for block i simplifies to*

$$\bar{\kappa}^i(z_1^i) := \Pi_{X_i} [z_{1,i}^i - \gamma \nabla_i f(z_1^i)] \quad (8.78)$$

where

$$\gamma < \frac{1}{\max_{i \in \mathcal{V}} \max_{\eta \in X} \sum_{\ell=1}^N |H_{ij}(\eta)|}. \quad (8.79)$$

In [50], the asynchronous primal-dual algorithm is designed and analyzed for convex functions f which satisfies Assumption 8.2.3, H satisfies Assumption 8.2.5,

the constraint set X satisfies Assumption 8.2.1, and the algorithm itself satisfies Assumption 8.4.2.

It is shown in [50, Theorem 2] that, for the dual variables fixed, each maximal solution $k \mapsto z_1(k)$ to the asynchronous primal-dual algorithm satisfies

$$\max_{i \in N} \|z_1^i(k) - x^*\|_\infty \leq q_p^{ops(k)} \max_{\ell \in N} \|z_{1,0}^\ell - x^*\|_\infty \quad (8.80)$$

where $q_p := (1 - \gamma\mu) \in [0, 1)$ and $x^* \in X$ denotes the fixed point of the constrained gradient descent update law, with the dual variables fixed. Comparing the constant q_p in (8.80) with the constant a , defined below (8.77), the primal convergence rate of asynchronous primal-dual algorithm in [50] is faster than the convergence rate of (D_{async}, G_{async}) .

Next, we compare the algorithms in simulation⁶. For this simulation, we utilize an example with $N = 10$ agents with the objective function

$$f(z_1) := \frac{3}{10} \sum_{i=1}^N (z_{1,i}^i)^2 + \frac{1}{200} \sum_{i=1}^N \sum_{\substack{\ell=1 \\ \ell \neq i}}^N (z_{1,i}^i - z_{1,\ell}^i)^2 \quad (8.81)$$

for which $\mu = \frac{1}{2}$. We also require that $z_{1,i}^i \in X_i = [1, 10]$ and $z_{2,i}^i \in X_i = [1, 10]$ for each $i \in \mathcal{V}$. We use the parameter value $\gamma = 0.3$ for the step size of both algorithms, which satisfies $\gamma \in \left(0, \frac{1}{\max_{i \in \mathcal{V}} \max_{z_1 \in X} |H_{ii}(z_1)|}\right)$ for (D_{async}, G_{async}) in (8.71) and the definition of γ in (8.79) for the asynchronous primal-dual algorithm in [50]. Additionally, we use the value $\lambda = 0.075$ for (D_{async}, G_{async}) , which satisfies $\lambda \in (0, \frac{\gamma\mu}{2})$. In this example, both algorithms have a communication rate of 0.5 (i.e., each agent has a 50% probability of communicating the latest update to another agent at each iteration) and a computation rate of 1 (i.e., each decision variable has a 100% probability of updating at each iteration). The initial conditions for (D_{async}, G_{async})

⁶Code at github.com/HybridSystemsLab/MultiagentHBF.

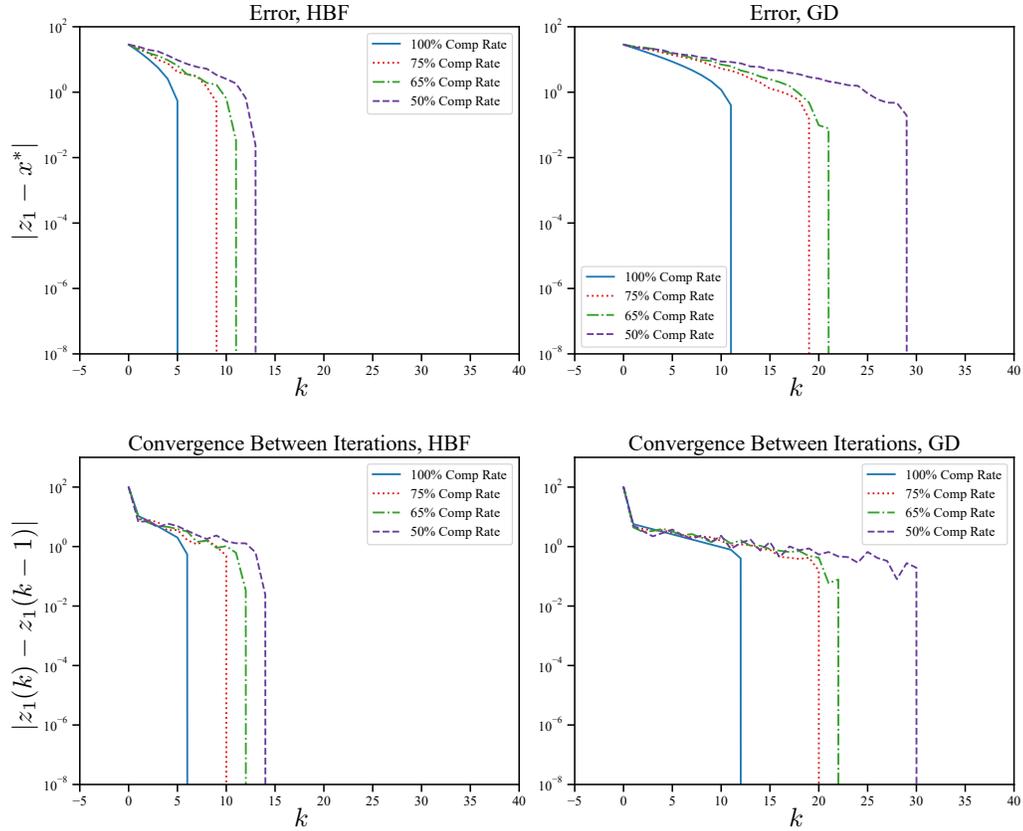


Figure 8.1: Comparing the effect of different computation rates on solutions, for the objective function in (8.81) with the constraint set $X_i = [1, 10]$. Top: A comparison of the evolution over time of $|z_1 - x^*|$, with $(D_{\text{async}}, G_{\text{async}})$ on the left and the asynchronous primal-dual algorithm in (8.80) on the right. Bottom: A comparison of the evolution over time of $|z_1(k) - z_1(k-1)|$, with $(D_{\text{async}}, G_{\text{async}})$ on the left and the asynchronous primal-dual algorithm in (8.80) on the right.

are $z_{\circ}^i = ((10, 10, 10, 10, 10, 10, 10, 10, 10, 10), (10, 10, 10, 10, 10, 10, 10, 10, 10, 10))$, for all $i \in \mathcal{V}$, and for the asynchronous primal-dual algorithm in [50] initial conditions are $z_{1,\circ}^i = (10, 10, 10, 10, 10, 10, 10, 10, 10, 10)$, for all $i \in \mathcal{V}$.

Figure 1.6 demonstrates marked performance improvement of $(D_{\text{async}}, G_{\text{async}})$ over the asynchronous primal-dual algorithm, with $(D_{\text{async}}, G_{\text{async}})$ finishing in 6 iterations and the asynchronous primal-dual algorithm finishing in 12 iterations. In other words, $(D_{\text{async}}, G_{\text{async}})$ converges twice as fast as the asynchronous primal-dual algorithm. From this example, we see that although the theoretical convergence bound of $(D_{\text{async}}, G_{\text{async}})$ in (8.77) is slower than the convergence bound of the asynchronous primal-dual algorithm in (8.80), such a bound on $(D_{\text{async}}, G_{\text{async}})$ is quite conservative compared to its numerical performance.

For both algorithms, we also compare the effect of different computation rates, while the communication rate is set at 1. Figure 8.1 compares both the evolution over time of $|z_1 - x^*|$ and the distance between specific iterations, namely, $|z_1(k) - z_1(k-1)|$, for $(D_{\text{async}}, G_{\text{async}})$ and the asynchronous primal-dual algorithm in (8.80), with computation rates of 1, .75, .65, and .5. The objective function, constraint set, parameter values, and initial conditions are the same as those listed above, for Figure 1.6.

Chapter 9

Conclusion

In this dissertation, we addressed several problems involving accelerated gradient methods. The solutions to a subset of such problems were solved via the use of hybrid system tools, while the solutions to the remaining problem was solved via discrete-time methods. In this chapter, we present a summary of the major content and describe several potential future research directions.

9.1 Summary

In Chapter 3, we analyzed key properties of the ODEs in (1.1), (1.2), and (1.5). First, for (1.2), we established UGAS of the minimizer and an exponential convergence rate. Next, for (1.5), we established UGAS of the minimizer and a convergence rate of $\frac{1}{(t+2)^2}$. Then, for (1.1), we established UGAS of the minimizer for nonstrongly convex objective functions L , and we established exponential convergence rates for both strongly and nonstrongly convex L . Finally, for (1.1), we established almost global asymptotic stability of a local minimizer for nonconvex Morse functions L .

In Chapter 4, we developed a hybrid algorithm uniting two heavy ball al-

gorithms with properly designed $\lambda_q > 0$ and $\gamma_q > 0$, using hybrid system tools. Designed for \mathcal{C}^1 , nonstrongly convex objective functions L with a single minimizer, the algorithm renders the minimizer uniformly globally asymptotically stable, with a hybrid exponential convergence rate, and with robustness. Two different sets of switching rules were derived: one which uses measurements of L and ∇L , and one in which only measurements of ∇L are used. Whereas the first set of switching rules requires knowledge of L^* , the second set of switching rules requires no knowledge of the minimizer.

In Chapter 5, we presented two algorithms, designed using hybrid system tools, that properly unites Nesterov's algorithm globally and the heavy ball algorithm with large $\lambda > 0$ locally to ensure fast convergence and uniform global asymptotic stability of the minimizer, with robustness. The first such algorithm, designed for \mathcal{C}^2 , strongly convex objective functions L , has a hybrid convergence rate that is exponential. The second such algorithm, designed for \mathcal{C}^1 , nonstrongly convex objective functions L with a single minimizer, has a hybrid convergence rate of $\frac{1}{(t+2)^2}$ globally and exponential locally. Both algorithms use measurements of ∇L , and neither algorithm requires knowledge of the minimizer.

In Chapter 6, we presented a general framework, designed using hybrid system tools, for uniting local and global optimization algorithms, which allows either the local and global controllers to be any accelerated gradient algorithm. We then determined sufficient conditions for well-posedness, existence of solutions, and uniform global asymptotic stability of the minimizer for the hybrid closed-loop system. The framework allows for the optimization of objective functions that are either \mathcal{C}^2 and strongly convex, or \mathcal{C}^1 and nonstrongly convex with a single minimizer. We then outlined how some of the algorithms in Chapters 4 and 5 satisfy this framework.

In Chapter 7, we developed a hybrid optimization algorithm, based on the heavy ball ODE in (1.1), to ensure practical global attractivity to a neighborhood of a local minimizer of a nonconvex Morse objective function L , even when the state $z \in \mathbb{R}^2$ starts at a local maximizer. Designed using hybrid system tools, this algorithm utilizes a switching strategy that uses measurements of the gradient of L to detect whether the state z is near a critical point. If z is near a critical point, the supervisor selects linear feedback to push the state away from such a critical point. If z is far from a critical point, the supervisor selects the heavy ball method to converge to a small neighborhood of a local minimizer. In simulation we demonstrated the robustness of the algorithm to small noise in measurements of ∇L .

In Chapter 8, we developed a totally asynchronous, block-based optimization algorithm, utilizing the constrained heavy ball method, which guarantees fast convergence to the unique minimizer of f . Specifically, we show that our algorithm has an exponential convergence rate under the assumption that f is \mathcal{C}^2 , convex, and the Hessian of f is diagonally dominant. Although such an exponential convergence rate is no better than the primal convergence rate, with a fixed dual variable, in [50, Theorem 2] for the asynchronous primal-dual algorithm in [50], nevertheless we demonstrate in simulation that our algorithm is twice as fast.

9.2 Future Directions

The results for the uniting algorithms and uniting framework in Chapters 4, 5, and 6 could be extended to allow for nonstrongly convex objective functions with a compact, connected continuum of minimizers via the modifications described in Sections 3.1.3, 3.2.3, 4.5, and 5.2.10, namely, via the use of the Lyapunov functions in (3.69) and (3.97) and the use of Clarke's generalized derivative. The results in

Chapters 4, 5, and 6 could also be further extended to allow for nonsmooth, non-strongly convex objective functions L using such Lyapunov functions and Clarke’s generalized derivative.

The uniting algorithms in Chapters 4 and 5 could be applied to machine learning problems, via backpropagation in a neural network. Another potential extension would be to make the uniting algorithms adaptive, i.e., learning parameters such as λ , γ , or the Lipschitz parameter M online.

The results for the uniting framework in Chapter 6 could be extended to allow for more types of optimization algorithms to be used as either the local or global controller, including gradient-free accelerated optimization schemes.

The results for the hybrid algorithm for nonconvex Morse functions L , in Chapter 7, could be extended to allow for Morse functions $L : \mathbb{R}^n \rightarrow \mathbb{R}$. One challenge in doing so would be to design a mechanism to detect when the state z_1 is stuck in a saddle point, and then facilitate escape from such a saddle point. Some possible mechanisms which could be utilized include the “gradient restart” scheme in [81] or the “speed restart” scheme in [15], which restart when the momentum is taking the state z in a bad direction (as determined by the momentum term and the negative gradient making an obtuse angle).

Another possible method for extending the hybrid algorithm for nonconvex Morse functions L , in Chapter 7, is as follows. In [82], it was shown that first order methods¹, such as the heavy ball method, almost always avoid strict saddle points – namely, saddle points where the Hessian of the objective function admits at least one direction of negative curvature; see [83]. Moreover, in [84], it was shown that such an advantage could also be leveraged for non-strict saddle points by regularizing the objective function when the state is in the vicinity of such a

¹In [82], the term “first order method” refers to gradient-based methods, since such methods involve a gradient term.

non-strict saddle point. To do so, a linear regularizer is chosen in [84], since it was proved in [85] that such a regularizer also renders the objective function Morse.

Other extensions to the algorithm in Chapter 7 would be to investigate ways of shrinking the parameters $(\varepsilon_1, \varepsilon_2, \rho_1, \rho_2)$ in real time, in order to actually converge to the set \mathcal{A} , and to analyze the impact of different settings for such parameters on the performance of the algorithm. Moreover, the addition of stopping criteria for the algorithm could be explored.

The results for the totally asynchronous, multiagent algorithm in Chapter 8 can be extended to allow for each decision variable to be such that $(z_{1,i}^i, z_{2,i}^i) \in X_i \times X_i \subset \mathbb{R}^m \times \mathbb{R}^m$.

Appendix A

General Results on Hybrid Systems

A.1 Existence of Solutions, Stability, and Invariance

The following proposition, from [21], is used to prove the existence of solutions to many of the the hybrid closed-loop algorithms proposed in this dissertation.

Proposition A.1.1. (*Basic existence of solutions*) Let $\mathcal{H} = (C, F, D, G)$ satisfy Definition 2.1.1. Take an arbitrary $x_o \in C \cup D$. If $x_o \in D$ or

(VC) there exists a neighborhood U of x_o such that for every $x \in U \cap C$,

$$F(x) \cap T_C(x) \neq \emptyset,$$

then there exists a nontrivial solution x to \mathcal{H} with $x(0, 0) = x_o$. If (VC) holds for every $x_o \in C \setminus D$, then there exists a nontrivial solution to \mathcal{H} from every initial

point in $C \cup D$, and every $x \in \mathcal{S}_{\mathcal{H}}$ satisfies exactly one of the following conditions:

- (a) x is complete;
- (b) $\text{dom } x$ is bounded and the interval I^J , where $J = \sup_j \text{dom } x$, has nonempty interior and $t \mapsto x(t, J)$ is a maximal solution to $\dot{x} \in F(x)$, in fact $\lim_{t \rightarrow T} |x(t, J)| = \infty$, where $T = \sup_t \text{dom } x$;
- (c) $x(T, J) \notin C \cup D$, where $(T, J) = \sup \text{dom } x$.

Furthermore, if $G(D) \subset C \cup D$, then (c) above does not occur.

The following definition, from [22, Definition 3.17], describes the basic properties that a function must satisfy to serve as a Lyapunov function for the hybrid closed-loop algorithm \mathcal{H} .

Definition A.1.2 (Lyapunov function candidate). *The sets \mathcal{U} , $\mathcal{A} \subset \mathbb{R}^n$, and the function $V : \text{dom } V \rightarrow \mathbb{R}$ define a Lyapunov function candidate on \mathcal{U} with respect to \mathcal{A} for the hybrid closed-loop system $\mathcal{H} = (C, F, D, G)$ if the following conditions hold:*

1. $(\overline{C} \cup D \cup G(D)) \cup \mathcal{U} \subset \text{dom } V$;
2. \mathcal{U} contains an open neighborhood of $\mathcal{A} \cap (C \cup D \cup G(D))$;
3. V is continuous on \mathcal{U} and locally Lipschitz on an open set containing $\overline{C} \cap \mathcal{U}$;
4. V is positive definite on $\overline{C} \cup D \cup G(D)$ with respect to \mathcal{A} .

The following theorem is used to prove the uniform global asymptotic stability of the hybrid closed-loop system, via Lyapunov stability and an invariance principle.

Theorem A.1.3. (*Hybrid Lyapunov theorem*) Given sets $\mathcal{U}, \mathcal{A} \subset \mathbb{R}^n$ and a function $V : \text{dom } V \rightarrow \mathbb{R}$ defining a Lyapunov candidate on \mathcal{U} with respect to \mathcal{A} for the closed-loop hybrid system $\mathcal{H} = (C, F, D, G)$, suppose

- \mathcal{H} satisfies the hybrid basic conditions;
- \mathcal{A} is compact and \mathcal{U} contains a nonzero open neighborhood of \mathcal{A} ;
- \dot{V} and ΔV satisfy

$$\dot{V}(x) = \max_{\xi \in F(x)} \langle \nabla V(x), \xi \rangle \leq 0 \quad \forall x \in C \cap \mathcal{U} \quad (\text{A.1})$$

$$\Delta V(x) := \max_{\xi \in G(x)} V(\xi) - V(x) \leq 0 \quad \forall x \in D \cap \mathcal{U} \quad (\text{A.2})$$

Then \mathcal{A} is stable. Furthermore, \mathcal{A} is attractive and, hence, pre-asymptotically stable if any of the following conditions hold:

1. *Strict decrease during flows and jumps:*

$$\dot{V}(x) < 0 \quad \forall x \in (C \cap \mathcal{U}) \setminus \mathcal{A} \quad (\text{A.3})$$

$$\Delta V(x) < 0 \quad \forall x \in (D \cap \mathcal{U}) \setminus \mathcal{A} \quad (\text{A.4})$$

2. *Strict decrease during flows and no instantaneous Zeno:*

(a) $\dot{V}(x) < 0$ for each $x \in (C \cap \mathcal{U}) \setminus \mathcal{A}$,

(b) any instantaneous Zeno solution x to \mathcal{H} where $\text{rge } x \subset \mathcal{U}$ converges to \mathcal{A} ;

3. *Strict decrease during jumps and no complete continuous solution:*

(a) $\Delta V(x) < 0$ for each $x \in (D \cap \mathcal{U}) \setminus \mathcal{A}$,

(b) any complete continuous solution x to \mathcal{H} where $\text{rge } x \subset \mathcal{U}$ converges to \mathcal{A} ;

4. Weak decrease during flows and jumps: for each $\chi \in \mathcal{U}$ with $r := V(\chi) > 0$ there is no complete solution x to \mathcal{H} , $x(0,0) = \chi$ such that

$$\text{rge } x \subset \{x : V(x) = r\} \cap \mathcal{U} \quad (\text{A.5})$$

and the set \mathcal{U} is the subset of the basin of pre-attraction.

Observe that, if the set \mathcal{A} is pre-asymptotically stable via Theorem A.1.3 and the Lyapunov function V also has compact sublevel sets, namely, for each $c_V > 0$, $\{x : V(x) \leq c_V\}$ is compact, then the origin is *globally pre-asymptotically stable*.

The following result is used to show that, when a hybrid closed-loop algorithm \mathcal{H} has a set \mathcal{A} globally asymptotically stable, then when \mathcal{H} satisfies the hybrid basic conditions, the set \mathcal{A} is also uniformly globally asymptotically stable¹ for \mathcal{H} .

Theorem A.1.4. *(Pre-asymptotic stability implies \mathcal{KL} pre-asymptotic stability)*
 Suppose that the hybrid closed-loop system \mathcal{H} satisfies the hybrid basic conditions and that a compact set \mathcal{A} is pre-asymptotically stable with basin of pre-attraction $\mathcal{B}_{\mathcal{A}}^p$. Then, $\mathcal{B}_{\mathcal{A}}^p$ is open and \mathcal{A} is \mathcal{KL} pre-asymptotically stable on $\mathcal{B}_{\mathcal{A}}^p$ for \mathcal{H} ; namely, there exists a function $\beta \in \mathcal{KL}$ such that

$$|x(0,0)|_{\mathcal{A}} \leq \beta(|x(0,0)|_{\mathcal{A}}, t+j) \quad \forall (t,j) \in \text{dom } x \quad (\text{A.6})$$

for each $x \in \mathcal{S}_{\mathcal{H}}(\mathcal{B}_{\mathcal{A}}^p)$.

¹Uniform global asymptotic stability allows an equivalent characterization involving a class- \mathcal{KL} function [21].

For Proposition 3.2.8 and Theorem A.1.6 we use the following definition of weak invariance, from [21].

Definition A.1.5 (Weak invariance). *Given a hybrid system \mathcal{H} , a set $S \subset \mathbb{R}^n$ is said to be*

- *weakly forward invariant if for every $x_o \in S$ there exists at least one complete $x \in \mathcal{S}_{\mathcal{H}}(x_o)$ with $\text{rge } x \subset S$;*
- *weakly backward invariant if for every $x_o \in S$ and every $T > 0$, there exists at least one $x \in \mathcal{S}_{\mathcal{H}}(S)$ such that for some $(t^*, j^*) \in \text{dom } x$, $t^* + j^* \geq T$, it is the case that $x(t^*, j^*) = x_o$ and $x(t, j) \in S$ for all $(t, j) \in \text{dom } x$ with $t + j \leq t^* + j^*$;*
- *weakly invariant if it is both weakly forward invariant and weakly backward invariant.*

The following *hybrid invariance principle*, from [22, Theorem 3.23], is used to establish attractivity when only a “weak” Lyapunov function is available – meaning that the function does not strictly decrease along both flows and jumps of the hybrid system. It is also useful to check where particular solutions of interest converge to.

Theorem A.1.6. (*Hybrid Invariance Principle*) *Given a hybrid closed-loop system $\mathcal{H} = (C, F, D, G)$ with state $x \in \mathbb{R}^n$ satisfying the hybrid basic conditions, nonempty $\mathcal{U} \subset \mathbb{R}^n$, and a function $V : \text{dom } V \rightarrow \mathbb{R}$, suppose that A.1.2 is satisfied, and that (A.1) and (A.2) hold. With $X := C \cup D \cup G(D)$, we employ the following definitions:*

$$V^{-1}(r) := \{x \in X : V(x) = r\} \tag{A.7}$$

$$\dot{V}^{-1}(0) := \{x \in C : \dot{V}(x) = 0\} \quad (\text{A.8})$$

$$\Delta V^{-1}(0) := \{x \in D : \Delta V(x) = 0\} \quad (\text{A.9})$$

Let x be a precompact solution to \mathcal{H} with $\overline{\text{rge } x} \subset \mathcal{U}$. Then, for some constant $r \in V(\mathcal{U} \cap X)$, the following hold:

1. The solution x converges to the largest weakly invariant set in

$$V^{-1}(r) \cap \mathcal{U} \cap \left[\dot{V}^{-1}(0) \cup \left(\Delta V^{-1}(0) \cap G \left(\Delta V^{-1}(0) \right) \right) \right]; \quad (\text{A.10})$$

2. The solution x converges to the largest weakly invariant set in

$$V^{-1}(r) \cap \mathcal{U} \cap \Delta V^{-1}(0) \cap G \left(\Delta V^{-1}(0) \right) \quad (\text{A.11})$$

if in addition the solution x is Zeno;

3. The solution x converges to the largest weakly invariant set in

$$V^{-1}(r) \cap \mathcal{U} \cap \dot{V}^{-1}(0) \quad (\text{A.12})$$

if, in addition, the solution x is such that, for some $a > 0$ and some $J \in \mathbb{N}$, $t_{j+1} - t_j > a$ for all $j \geq J$; i.e., the given solution x is such that the elapsed time between consecutive jumps is eventually bounded below by a positive constant a .

Appendix B

General Results on Totally Asynchronous Multiagent Algorithms

B.1 Totally Asynchronous Convergence

The general fixed point problem that pertains to Proposition B.1.2 below, and which comes from [59], is described as follows. For $i \in \{1, 2, \dots, N\}$ agents, let X_1, \dots, X_N be subsets of the Euclidean spaces $\mathbb{R}^{m_1}, \dots, \mathbb{R}^{m_N}$, respectively. Let $m = m_1 + \dots + m_N$, and let $X \subset \mathbb{R}^m$ be the Cartesian product

$$X = X_1 \times \dots \times X_N. \tag{B.1}$$

Accordingly, elements in X are written as N -tuples of their components, i.e., for $x \in X$, we write

$$x := (x_1^1, \dots, x_N^N) \tag{B.2}$$

where x_i^i are the corresponding elements of X_i for agents $i \in \{1, 2, \dots, N\}$. We assume there is a notion of sequence convergence defined on X . For $i \in \{1, 2, \dots, N\}$, let the function $g^i : X \rightarrow X_i$ be a given convergence algorithm, and let $g : X \rightarrow X$ be the synchronous function defined by

$$g(x) = (g^1(x), \dots, g^N(x)) \quad (\text{B.3})$$

for all $x \in X$. The problem is to find the fixed point of g , namely, an element $x^* \in X$ such that $x^* := g(x^*)$.

Now, to describe a distributed asynchronous version of $x_i^i := g^i(x^i)$, $i \in \{1, 2, \dots, N\}$. Let $x_i^i(t)$ be the value of the i -th component at time $t \in \mathbb{N}$. We assume there is a set of times $t \in \mathbb{N}$ at which one or more components of x is updated by some agent of a distributed system. We also assume there is a set of times at which x_i^i is updated. We assume that agent i might not have access to the most recent value of the components of x . Therefore, we assume that

$$x_i^i(t+1) := g^i(x_1^i(\tau_1^i(t)), \dots, x_N^i(\tau_N^i(t))), \quad (\text{B.4})$$

for all t in the set of update times for i , where $\tau_\ell^i(t)$ are times satisfying

$$0 \leq \tau_\ell^i(t) \leq t$$

for all $t \in \mathbb{N}$, and all $\ell \in N$. At all times t not in the set of update times for i , x_i^i is left unchanged, namely

$$x_i^i(t+1) = x_i^i(t). \quad (\text{B.5})$$

The set of times $t \in \mathbb{N}$ should be viewed as the indices of the sequence of physical times at which updates take place. The sequence of physical times at which agent

i updates does not need to be known to any one agent, since their knowledge is not needed to execute (B.4)-(B.5), namely, there is no need for a global clock. The difference $t - \tau_\ell^i(k)$ between the current time t and the time $\tau_\ell^i(t)$ corresponding to the ℓ -th component available at the agent updating x^i can be viewed as a form of communication delay.

Assumption 8.4.2 is imposed on the general asynchronous multiagent algorithm described in (B.4)-(B.5). The following assumptions are also imposed on the asynchronous algorithm in (B.4)-(B.5):

Assumption B.1.1 (Synchronous and box conditions). *There is a sequence of nonempty sets $\mathcal{X}(k) \subset X$, $k \in \mathbb{N}$ such that*

$$\dots \subset \mathcal{X}(k+1) \subset \mathcal{X}(k) \subset \dots \subset \mathcal{X}_\circ \tag{B.6}$$

satisfying the following conditions:

(M1) (Synchronous Convergence Condition)

- (i) *We have $g(x(k)) \in \mathcal{X}(k+1)$ for all $k \in \mathbb{N}$ and all $x \in \mathcal{X}(k)$.*
- (ii) *Furthermore, if $\{x_k\}_{k \in \mathbb{N}}$ is a sequence such that $x_k \in \mathcal{X}(k)$ for all $k \in \mathbb{N}$, then every limit point of $\{x_k\}_{k \in \mathbb{N}}$ is a fixed point of g .*

(M2) (Box Condition) *For every $k \in \mathbb{N}$, there exist sets $\mathcal{X}_i(k) \subset \mathcal{X}_{i_\circ}$ such that*

$$\mathcal{X}(k) = \mathcal{X}_1(k) \times \mathcal{X}_2(k) \times \dots \times \mathcal{X}_N(k). \tag{B.7}$$

Note that the Synchronous Convergence Condition implies that the limit points of sequences generated by the (synchronous) iteration $x := g(x)$ are fixed points of $\mathcal{X}(k)$, assuming the initial condition $x_\circ \in \mathcal{X}_\circ$.

The following proposition, which comes from [59], establishes that the totally asynchronous algorithm in (B.4)-(B.5) converges to a fixed point of g .

Proposition B.1.2. (*Asynchronous Convergence*) [59, Proposition 2.1, Chapter 6]: *Let Assumption 8.4.2 hold for the synchronous algorithm in (B.3) and let $x_\circ \in \mathcal{X}_\circ := X$. Then, every limit point of the sequence $\{x(t)\}_{t \in \mathbb{N}}$ generated by (B.4)-(B.5) is a fixed point of g .*

Proof. We show by induction that for each $k \in \mathbb{N}$, there is a time $t_k \in \mathbb{N}$ such that:

- (a) $x \in \mathcal{X}(k)$ for all $t \geq t_k$;
- (b) For all t in the set of update times for agent i , with $t \geq t_k$ and all $i \in N$, we have $x^i(t) \in \mathcal{X}(k)$, where

$$x^i(t) = \left(x_1^1(\tau_1^i(t)), \dots, x_N^N(\tau_N^i(t)) \right). \quad (\text{B.8})$$

In words, after some time, all solution estimates will be in $\mathcal{X}(k)$ and all estimates used in iteration B.4 will come from $\mathcal{X}(k)$.

The induction hypothesis in (a)-(b) is true for $k = 0$, since the initial estimate is assumed to be in \mathcal{X}_\circ . Assuming it is true for a given k , we will show that there exists a time t_{k+1} with the required properties. For each $i \in N$, let t^i be the first element in the set of times at which agent i is updated, such that $t^i \geq t_k$. Then, by the Synchronous Convergence Condition in item (M1) of Assumption B.1.1, we have $g(x^i(t^i)) \in \mathcal{X}(k+1)$. This implies, in view of the Box Condition in item (M2) of Assumption B.1.1, that

$$x_i^i(t^i + 1) = g_i^i(x^i(t^i)) \in \mathcal{X}_i(k+1). \quad (\text{B.9})$$

Similarly, for every t in the set of update times for agent i such that $t \geq t^i$, we have $x_i^i(t+1) \in \mathcal{X}_i(k+1)$. Between elements t in the set of update times for agent i , $x_i^i(t)$ does not change. Thus,

$$x_i^i(t) \in \mathcal{X}_i(k+1) \tag{B.10}$$

for all $t \geq t^i + 1$.

Let $t'_k := \max_i \{t^i\} + 1$. Then, using the Box Condition in item (M2) of Assumption B.1.1, we have

$$x^i(t) \in \mathcal{X}(k+1) \tag{B.11}$$

for all $t \geq t'_k$.

Finally, since by Assumption 8.4.2, we have $\tau_\ell^i \rightarrow \infty$ as $t \rightarrow \infty$, where t is in the set of update times for agent i , then we can choose a time $t_{k+1} \geq t'_k$ that is sufficiently large so that $\tau_\ell^i \geq t'_k$ for all $i, \ell \in N$ and all t in the set of update times for agent i with $t \geq t_{k+1}$. We then have $x_\ell^\ell(\tau_\ell^i(t)) \in \mathcal{X}_\ell(k+1)$ for all t in the set of update times for agent i with $t \geq t_{k+1}$ and all $\ell \in N$, which by the Box Condition in item (M2) of Assumption B.1.1 implies that, at time t ,

$$x^i(t) = \left(x_1^i(\tau_1^i(t)), \dots, x_N^i(\tau_N^i(t)) \right) \in \mathcal{X}(k+1) \tag{B.12}$$

and the induction is complete. □

Appendix C

General Results on Optimality and Projection

C.1 Optimality Conditions and Projection Theorem

In [59, Chapter 3, Proposition 3.1], necessary and sufficient conditions for a vector $x \in X$ to be optimal are listed. They are as follows

Proposition C.1.1. *(Optimality conditions): Let f satisfy Assumption 8.2.3 and let X be nonempty, compact, and convex. Then, the following hold:*

(O1) *If a vector $\xi \in X$ minimizes f over X , then $\langle y - \xi, \nabla f(\xi) \rangle \geq 0$ for all $y \in X$;*

(O2) *If f is also convex on the set X , then the condition of part (O1) is also sufficient for ξ to minimize f over X .*

A useful property of projection is also stated in [59, Chapter 3, Proposition 3.2(b)-(c)], which is as follows.

Proposition C.1.2. (*Projection theorem, parts (b) and (c)*): Let f satisfy Assumption 8.2.3 and let X be nonempty, compact, and convex. Then,

(P1) Given some $v \in \mathbb{R}^n$, a vector $u \in X$ is equal to $\Pi_X[v]$ if and only if $\langle y - u, v - u \rangle \leq 0$ for all $y \in X$;

(P2) $\Pi_X[v]$ is continuous and nonexpansive, that is, for all $v, y \in \mathbb{R}^n$, $|\Pi_X[v] - \Pi_X[y]| \leq |v - y|$.

Appendix D

Code for Numerical Examples

Most of the numerical examples in this dissertation were simulated in MATLAB, using the the Hybrid Equation (HyEQ) Toolbox version 2.0.4. The Hybrid Equation (HyEQ) Toolbox can be downloaded for free from <https://www.mathworks.com/matlabcentral/fileexchange/41372-hybrid-equations-toolbox-v2-04>.

Code for each of the numerical examples simulated in MATLAB can be found at the following GitHub repositories.

- Figure 1.1: github.com/HybridSystemsLab/UnitingMotivationHBF
- Figure 1.2: github.com/HybridSystemsLab/UnitingMotivationSC
- Figure 1.3: github.com/HybridSystemsLab/UnitingMotivation
- Figure 1.4 and Example 5.2.7:
github.com/HybridSystemsLab/UnitingTradeoff
- Figure 1.5: github.com/HybridSystemsLab/RobustnessHeavyBall
- Example 4.3.4: github.com/HybridSystemsLab/UnitingLevelSetsHBF

- Example 4.4.5: github.com/HybridSystemsLab/UnitingRobustnessHBF
- Example 4.4.6: github.com/HybridSystemsLab/UnitingGradientsHBF
- Example 5.1.5: github.com/HybridSystemsLab/UnitingRobustnessSC
- Example 5.1.6: github.com/HybridSystemsLab/UnitingSC
- Example 5.2.5: github.com/HybridSystemsLab/UnitingRobustness
- Example 5.2.6: github.com/HybridSystemsLab/UnitingNSC
- Example 7.5.1: github.com/HybridSystemsLab/PGASHeavyBall

The numerical examples for the totally asynchronous heavy ball algorithm were simulated using Python 3, with the following installed package and libraries.

- NumPy: <https://numpy.org>
- Matplotlib: <https://matplotlib.org/>
- Seaborn: <https://seaborn.pydata.org/>

Code for the numerical examples simulated in Python 3 can be found at the following GitHub repository.

- Figure 1.6 and 8.1: github.com/HybridSystemsLab/MultiagentHBF

Bibliography

- [1] Simon Michalowsky and Christian Ebenbauer. The multidimensional n-th order heavy ball method and its application to extremum seeking. In *53rd IEEE Conference on Decision and Control*, pages 2660–2666, 2014.
- [2] Euhanna Ghadimi. *Accelerating convergence of large-scale optimization algorithms*. PhD thesis, KTH Royal Institute of Technology, 2015.
- [3] Jia Liu, Atilla Eryilmaz, Ness B Shroff, and Elizabeth S Bentley. Heavy-ball: A new approach to tame delay and convergence in wireless network optimization. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.
- [4] Gianluca Bianchin, Jorge I Poveda, and Emiliano Dall’Anese. Online optimization of switched lti systems using continuous-time and hybrid accelerated gradient flows. *arXiv preprint arXiv:2008.03903*, 2020.
- [5] Marcello Colombino, Emiliano Dall’Anese, and Andrey Bernstein. Online optimization as a feedback controller: Stability and tracking. *IEEE Transactions on Control of Network Systems*, 7(1):422–432, 2019.
- [6] Sandeep Menta, Adrian Hauswirth, Saverio Bolognani, Gabriela Hug, and Florian Dörfler. Stability of dynamic feedback optimization with applications to power systems. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 136–143. IEEE, 2018.
- [7] Gianluca Bianchin and Fabio Pasqualetti. Gramian-based optimization for the analysis and control of traffic networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(7):3013–3024, 2019.
- [8] Ronny Kutadinata, Will Moase, Chris Manzie, Lele Zhang, and Tim Garoni. Enhancing the performance of existing urban traffic light control through extremum-seeking. *Transportation Research Part C: Emerging Technologies*, 62:1–20, 2016.

- [9] Euhanna Ghadimi, Hamid Reza Feyzmahdavian, and Mikael Johansson. Global convergence of the heavy-ball method for convex optimization. In *14th IEEE European Control Conference*, pages 310–315, 2015.
- [10] Stefano Sarao Mannelli and Pierfrancesco Urbani. Analytical study of momentum-based acceleration methods in paradigmatic high-dimensional non-convex problems. *Advances in Neural Information Processing Systems*, 34, 2021.
- [11] Guilherme Franca, Daniel P Robinson, and Rene Vidal. A dynamical systems perspective on nonsmooth constrained optimization. *arXiv preprint arXiv:1808.04048*, 2018.
- [12] Michael Muehlebach and Michael Jordan. A dynamical systems perspective on nesterov acceleration. In *International Conference on Machine Learning*, pages 4656–4662. PMLR, 2019.
- [13] Boris Polyak and Pavel Shcherbakov. Lyapunov functions: an optimization theory perspective. *IFAC-PapersOnLine*, 50(1):7456–7461, 2017.
- [14] Hedy Attouch, Xavier Goudou, and Patrick 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*, 2(01):1–34, 2000.
- [15] Weijie Su, Stephen Boyd, and Emmanuel J Candès. A differential equation for modeling nesterov’s accelerated gradient method: Theory and insights. *The Journal of Machine Learning Research*, 17(1):5312–5354, 2016.
- [16] Walid Krichene, Alexandre Bayen, and Peter Bartlett. Accelerated mirror descent in continuous and discrete time. *Advances in neural information processing systems*, 28:2845–2853, 2015.
- [17] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.
- [18] Ashia C Wilson, Ben Recht, and Michael I Jordan. A lyapunov analysis of accelerated methods in optimization. *Journal of Machine Learning Research*, 22(113):1–34, 2021.
- [19] Arman Sharifi Kolarijani, Peyman Mohajerin Esfahani, and Tamás Keviczky. Continuous-time accelerated methods via a hybrid control lens. *IEEE Transactions on Automatic Control*, pages 3425–3440, October 2019.

- [20] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. Hybrid dynamical systems. *IEEE Control Systems*, 29(2):28–93, 2009.
- [21] R. Goebel, R. G. Sanfelice, and A. R. Teel. *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton University Press, New Jersey, 2012.
- [22] Ricardo G. Sanfelice. *Hybrid Feedback Control*. Princeton University Press, New Jersey, 2021.
- [23] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.
- [24] Justin H Le and Andrew R Teel. Hybrid heavy-ball systems: reset methods for optimization with uncertainty. In *2021 American Control Conference (ACC)*, pages 2236–2241. IEEE, 2021.
- [25] Othmane Sebbouh, Ch Dossal, and Aude Rondepierre. Convergence rates of damped inertial dynamics under geometric conditions and perturbations. *SIAM Journal on Optimization*, 30(3):1850–1877, 2020.
- [26] Simon Michalowsky and Christian Ebenbauer. Extremum control of linear systems based on output feedback. In *55th IEEE Conference on Decision and Control*, pages 2963–2968, 2016.
- [27] Laurent Lessard, Benjamin Recht, and Andrew Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.
- [28] Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.
- [29] Nicolas Flammarion and Francis Bach. From averaging to acceleration, there is only a step-size. In *Conference on Learning Theory*, pages 658–695, 2015.
- [30] Yurii 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*, volume 269, pages 543–547, 1983.
- [31] Arman Sharifi Kolarijani, Peyman Mohajerin Esfahani, and Tamas Keviczky. Fast gradient-based methods with exponential rate: A hybrid control framework. In *International Conference on Machine Learning*, pages 2728–2736, 2018.

- [32] Jorge I Poveda and Na Li. Inducing uniform asymptotic stability in non-autonomous accelerated optimization dynamics via hybrid regularization. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 3000–3005. IEEE, 2019.
- [33] Andrew R Teel, Jorge I Poveda, and Justin Le. First-order optimization algorithms with resets and hamiltonian flows. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 5838–5843. IEEE, 2019.
- [34] 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)*, pages 151–156, 2019.
- [35] Jorge I Poveda and Andrew R Teel. The heavy-ball ODE with time-varying damping: Persistence of excitation and uniform asymptotic stability. In *2020 American Control Conference (ACC)*, pages 773–778. IEEE, 2020.
- [36] Jelena Diakonikolas and Michael I Jordan. Generalized momentum-based methods: A hamiltonian perspective. *SIAM Journal on Optimization*, 31(1):915–944, 2021.
- [37] Jorge I Poveda and Na Li. Robust hybrid zero-order optimization algorithms with acceleration via averaging in time. *Automatica*, 123:109361, 2021.
- [38] 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 Systems Letters*, 2(1):49–54, 2017.
- [39] Boya Sun, Jemin George, and Solmaz Kia. High-resolution modeling of the fastest first-order optimization method for strongly convex functions. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 4237–4242. IEEE, 2020.
- [40] SK Zavriev and FV Kostyuk. Heavy-ball method in nonconvex optimization problems. *Computational Mathematics and Modeling*, 4(4):336–341, 1993.
- [41] Sébastien Gadat, Fabien Panloup, Sofiane Saadane, et al. Stochastic heavy ball. *Electronic Journal of Statistics*, 12(1):461–529, 2018.
- [42] Peter Ochs, Yunjin Chen, Thomas Brox, and Thomas Pock. ipiano: Inertial proximal algorithm for nonconvex optimization. *SIAM Journal on Imaging Sciences*, 7(2):1388–1419, 2014.
- [43] Thomas Pock and Shoham Sabach. Inertial proximal alternating linearized minimization (ipalm) for nonconvex and nonsmooth problems. *SIAM Journal on Imaging Sciences*, 9(4):1756–1787, 2016.

- [44] Radu Ioan Boț, Ernő Robert Csetnek, and Szilárd Csaba László. An inertial forward–backward algorithm for the minimization of the sum of two nonconvex functions. *EURO Journal on Computational Optimization*, 4(1):3–25, 2016.
- [45] Tom Strizic, Jorge I Poveda, and Andrew R Teel. Hybrid gradient descent for robust global optimization on the circle. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2985–2990. IEEE, 2017.
- [46] Matina Baradaran, Jorge I Poveda, and Andrew R Teel. Stochastic hybrid inclusions applied to global almost sure optimization on manifolds. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6538–6543. IEEE, 2018.
- [47] Tzuu-Shuh Chiang, Chii-Ruey Hwang, and Shuenn Jyi Sheu. Diffusion for global optimization in \mathbb{R}^n . *SIAM Journal on Control and Optimization*, 25(3):737–753, 1987.
- [48] Hongbin Wang and Paul C Miller. Scaled heavy-ball acceleration of the richardson-lucy algorithm for 3d microscopy image restoration. *IEEE Transactions on Image Processing*, 23(2):848–854, 2013.
- [49] Peter Ochs, Thomas Brox, and Thomas Pock. ipiasco: inertial proximal algorithm for strongly convex optimization. *Journal of Mathematical Imaging and Vision*, 53(2):171–181, 2015.
- [50] Katherine Hendrickson and Matthew Hale. Towards totally asynchronous primal-dual optimization in blocks. *arXiv preprint arXiv:2004.05142*, 2020.
- [51] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.
- [52] Yurii Nesterov. Introductory lectures on convex optimization, vol. 87, 2004.
- [53] Jingzhao Zhang, César A Uribe, Aryan Mokhtari, and Ali Jadbabaie. Achieving acceleration in distributed optimization via direct discretization of the heavy-ball ode. In *2019 American Control Conference (ACC)*, pages 3408–3413. IEEE, 2019.
- [54] Huaqing Li, Huqiang Cheng, Zheng Wang, and Guo-Cheng Wu. Distributed nesterov gradient and heavy-ball double accelerated asynchronous optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 32(12):5723–5737, 2020.

- [55] Annie I Chen and Asuman Ozdaglar. A fast distributed proximal-gradient method. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 601–608. IEEE, 2012.
- [56] Daniel E Ochoa, Jorge I Poveda, César A Uribe, and Nicanor Quijano. Robust optimization over networks using distributed restarting of accelerated dynamics. *IEEE Control Systems Letters*, 5(1):301–306, 2020.
- [57] Alex Olshevsky. Linear time average consensus and distributed optimization on fixed graphs. *SIAM Journal on Control and Optimization*, 55(6):3990–4014, 2017.
- [58] Dušan Jakovetić, Joao Xavier, and José MF Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.
- [59] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.
- [60] Dimitri P Bertsekas. Distributed asynchronous computation of fixed points. *Mathematical Programming*, 27(1):107–120, 1983.
- [61] Stefan Hochhaus and Matthew T Hale. Asynchronous distributed optimization with heterogeneous regularizations and normalizations. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4232–4237. IEEE, 2018.
- [62] Matthew Ubl and Matthew T Hale. Totally asynchronous distributed quadratic programming with independent stepsizes and regularizations. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7423–7428. IEEE, 2019.
- [63] MT Hale and Magnus Egerstedt. Cloud-based optimization: A quasi-decentralized approach to multi-agent coordination. In *53rd IEEE Conference on Decision and Control*, pages 6635–6640. IEEE, 2014.
- [64] Matthew T Hale, Angelia Nedić, and Magnus Egerstedt. Asynchronous multiagent primal-dual optimization. *IEEE Transactions on Automatic Control*, 62(9):4421–4435, 2017.
- [65] Jayash Koshal, Angelia Nedić, and Uday V Shanbhag. Multiuser optimization: Distributed algorithms and error analysis. *SIAM Journal on Optimization*, 21(3):1046–1081, 2011.
- [66] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004.

- [67] Dmitriy Drusvyatskiy and Adrian S Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 2018.
- [68] Hamed Karimi, Julie Nutini, and Mark 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*, pages 795–811. Springer, 2016.
- [69] Ion Necoara, Yu Nesterov, and Francois Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175(1-2):69–107, 2019.
- [70] Ji Liu and Stephen J Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.
- [71] A. S. Kolarijani, P. M. Esfahani, and T. Keviczky. Continuous-time accelerated methods via a hybrid control lens. *IEEE Transactions on Automatic Control*, 65(8):3425–3440, 2020.
- [72] Michèle Audin and Mihai Damian. *Morse Theory and Floer Homology*. Springer, 2014.
- [73] Ron Larson, Robert Hostetler, and Bruce H Edwards. *Calculus: Early transcendental functions*. Houghton Mifflin, 4th edition, 2007.
- [74] Frank H Clarke. *Optimization and nonsmooth analysis*. SIAM, 1990.
- [75] Ricardo G Sanfelice, Rafal Goebel, and Andrew R Teel. Invariance principles for hybrid systems with connections to detectability and asymptotic stability. *IEEE Transactions on Automatic Control*, 52(12):2282–2297, 2007.
- [76] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2009.
- [77] H. K. Khalil. *Nonlinear Systems*. Prentice Hall, Upper Saddle River, New Jersey, 3 edition, 2002.
- [78] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1724–1732. JMLR. org, 2017.
- [79] Othmane Sebbouh, Charles Dossal, and Aude Rondepierre. Nesterov’s acceleration and polyak’s heavy ball method in continuous time: convergence rate analysis under geometric conditions and perturbations. July 2019.

- [80] Gabriel Behrendt and Matthew Hale. Technical report: A totally asynchronous algorithm for tracking solutions to time-varying convex optimization problems. *arXiv preprint arXiv:2110.06705*, 2021.
- [81] Brendan O’donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2015.
- [82] Jason D Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I Jordan, and Benjamin Recht. First-order methods almost always avoid strict saddle points. *Mathematical programming*, 176(1):311–337, 2019.
- [83] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015.
- [84] Matthew Ubl, Kasra Yazdani, and Matthew T Hale. Linear regularizers enforce the strict saddle property. *arXiv preprint arXiv:2205.09160*, 2022.
- [85] John Milnor. *Lectures on the h-cobordism theorem*, volume 2258. Princeton university press, 2015.