

Global Accelerated Nonconvex Geometric Optimization Methods on $SO(3)$

Adeel Akhtar Ricardo G. Sanfelice

Abstract— This paper proposes global accelerated nonconvex geometric (GANG) optimization algorithms for optimizing a class of nonconvex functions on the compact Lie group $SO(3)$. Nonconvex optimization is a challenging problem because the objective function may have multiple critical points, including saddle points. We propose two accelerated geometric algorithms to escape maxima and saddle points using random perturbations. The first algorithm uses the value of the Hessian of the objective function and random perturbations to escape the undesired critical points. In contrast, the second algorithm uses only the gradient information and random perturbations to escape maxima and saddle points. The efficacy of these geometric algorithms is verified in simulations.

I. INTRODUCTION

The cornerstone of many machine learning and data analysis algorithms entails minimizing or maximizing (optimizing) an objective function in an efficient manner. Unlike standard gradient descent, the heavy ball method is an accelerated method, with acceleration achieved by adding velocity to the gradient to speed up the rate of convergence [1], [2]. One of the challenges associated with accelerated optimization methods is when the objective function is defined on a space that is not globally diffeomorphic to a Euclidean space [3]. We call these methods global accelerated nonconvex geometric (GANG) because the underlying space has geometric constraints. It is possible to find a local solution to these geometric optimization problems by selecting a local chart, which locally renders the space as Euclidean space, and then using standard optimization algorithms. However, such solutions lead to local results. In this work, we consider a geometric optimization problem defined on a Lie group without using any local charts or embedding the given manifold in a higher dimensional space [4]. In particular, our space is the set of three-by-three rotation matrices, denoted by $SO(3)$, which has a group structure as well as a smooth manifold structure. We propose a coordinate-free (hence geometric) solution to minimize a class of objective functions defined on the compact manifold $SO(3)$ such that the resulting solution evolves on the manifold and is computationally efficient.

This research has been partially supported by the National Science Foundation under Grant no. ECS-1710621, Grant no. CNS-2039054, and Grant no. CNS-2111688, by the Air Force Office of Scientific Research under Grant nos. FA9550-19-1-0169, FA9550-20-1-0238, and FA9550-23-1-0145, by the Air Force Research Laboratory under Grant nos. FA8651-22-1-0017 and FA8651-23-1-0004, and by the Army Research Office under Grant no. W911NF-20-1-0253.

Adeel Akhtar is with the Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064, and is with the Division of Computer, Electrical and Mathematical Sciences and Engineering, King Abdullah University of Science and Technology, Makkah 23955, Saudi Arabia. Ricardo Sanfelice is with the Department of Electrical and Computer Engineering, University of California, Santa Cruz, CA 95064. Email: {adakhtar, ricardo}@ucsc.edu

Compact Lie groups, such as $SO(3)$, have a number of applications in the area of robotics, computer vision, simultaneous localization and mapping (SLAM), and pose camera estimation [5]. Gradient-based methods and accelerated methods on compact manifolds have been an active area of research [3], [6]. Gradient-based algorithms on Lie groups are generally interpreted as a continuous-time dynamical system, and special care needs to be taken while discretizing the dynamics. In particular, there is a vast amount of literature on methods that preserve the manifold structure even after discretization [7]–[9]. However, due to their associated challenges, considerably less attention has been given to GANG methods.

In this paper, we consider the problem of finding a local minimizer of a non-convex objective function (a Morse function) defined on $SO(3)$, which may have multiple local minimizers, maximizers, and saddle points with global convergence. The term “global convergence” means convergence from everywhere on $SO(3)$ rather than convergence from nearby a global minimum [2]. Using random perturbations, we design two accelerated (heavy ball) methods to minimize a non-convex objective function. In summary, we propose (i) a perturbed accelerated geometric gradient descent algorithm (PAGGD-1) that uses the information of Hessian to escape maximizers and saddle points (Algorithm 3), and (ii) a perturbed accelerated geometric gradient descent algorithm (PAGGD-2) that uses only the gradient information to escape maximizers and saddle points (Algorithm 4). These algorithms are validated numerically in examples.

A. Notation and Math Preliminaries

The n -dimensional Euclidean space is represented by \mathbb{R}^n . For a point $x \in \mathbb{R}^n$, the Euclidean norm is denoted by $|x|$, and the distance of a point x from a subset $S \subset \mathbb{R}^n$ is represented by $|x|_S := \inf_{y \in S} |x - y|$. We denote the inner product of two vectors $x, y \in \mathbb{R}^n$ as $\langle x, y \rangle_{\mathbb{R}^n}$. A k -dimensional vector x is represented as $(x_1, x_2, \dots, x_k) := [x_1, x_2, \dots, x_k]^\top$, where $^\top$ denotes transposition. The domain of a map f is represented by $\text{dom } f$. The value of the gradient of the map $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$ with respect to its argument evaluated at x is given by $\text{grad } f(x)$. The trace and determinant of a matrix $A \in \mathbb{R}^{n \times n}$ are represented by $\text{trace}(A)$ and $\det(A)$, respectively. A positive definite, positive semidefinite, negative definite, and negative semidefinite matrix is denoted by $A \succ 0$, $A \succeq 0$, $A \prec 0$, $A \preceq 0$, respectively. The set of 3×3 rotation matrices is defined as $SO(3) = \{R \in \mathbb{R}^{3 \times 3} : R^\top = R^{-1}, \det(R) = +1\}$ and it has a Lie group structure. The associated Lie algebra of $SO(3)$ is the set of 3×3 skew-symmetric matrices $\mathfrak{so}(3) = \{A \in \mathbb{R}^{3 \times 3} : A = -A^\top\}$, which is isomorphic to \mathbb{R}^3 . The

isomorphism is denoted by $\hat{\cdot} : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ and its inverse is denoted $(\cdot)^\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$.

Let M be a smooth manifold and p be a point on M . The tangent space of M at the point p is denoted by T_pM , and the tangent bundle of M is denoted by TM .

Definition 1 (Differential and derivative of a map):

Let $f : M \rightarrow \mathbb{R}$. The differential of f is a smooth section of the cotangent bundle T^*M and is denoted by $Df : M \rightarrow T^*M$. The derivative of f on M at a point $x \in M$ is defined as a map in terms of the differential of f as $Df(x) : T_xM \rightarrow \mathbb{R}$, $\xi \mapsto Df(x) \cdot \xi$.

A Riemannian structure on $SO(3)$ is defined by associating an inner product on its tangent space at each point. It should be noted that the choice of a Riemannian metric is not unique and in this work, we consider the canonical metric on $SO(3)$ and is defined as $\langle \xi_1, \xi_2 \rangle_g := \text{trace}(\xi_1^\top \xi_2)$, for any $g \in SO(3)$ and $\xi_1, \xi_2 \in T_gSO(3)$. The norm of a tangent vector $\xi \in T_gSO(3)$ is $|\xi| = \sqrt{\langle \xi, \xi \rangle} = \sqrt{\text{trace}(\xi^\top \xi)}$. To define the gradient on a smooth manifold, we must first specify a Riemannian metric.

Definition 2 (Gradient on a manifold): Let $SO(3)$ be equipped with the canonical Riemannian metric and $f : SO(3) \rightarrow \mathbb{R}$. The gradient vector field of f on M , denoted by $\text{grad } f : M \rightarrow TM$, with respect to the canonical Riemannian metric on $SO(3)$ is uniquely characterized by the following two properties:

- 1) $\text{grad } f(x) \in T_xSO(3)$, for all $x \in M$, and
- 2) $Df(x) \cdot \xi = \langle \text{grad } f(x), \xi \rangle_x$, for all $\xi \in T_xM$,

Global point stabilization on Lie groups is a nontrivial problem because of topological obstructions [10], [11]. Morse theory is a way to characterize the topological obstructions of a manifold by using the concept of Morse functions [12].

Definition 3 (Set of critical points): Let G be a Lie group. For a smooth (Morse) function $L : G \rightarrow \mathbb{R}$, the gradient and the Hessian (see [13] for details) matrix are $\text{grad } L$ and $\text{Hess } L$, respectively. The set of all critical points of L is defined as $\mathcal{A} := \{g \in G : \text{grad } L(g) = 0\}$. The set of local minimizers of L is defined as $\mathcal{A}_{\min} := \{g \in G : \text{grad } L(g) = 0, \text{Hess } L(g) \succ 0\} \subset \mathcal{A}$.

Definition 4 (Nondegenerate critical points): Suppose $p \in M$ is a critical point. The “second derivative” of L is defined by the symmetric bilinear form $T_p^2L : T_pM \times T_pM \rightarrow \mathbb{R}$. In local coordinates (x^1, \dots, x^n) , this bilinear map can be given by the Hessian matrix [13]. If the bilinear map (Hessian matrix) is nonsingular at p , we call p a nondegenerate point; otherwise, it is a degenerate point.

Definition 5 (Morse function): A map $L : M \rightarrow \mathbb{R}$ is a Morse function if its critical points are nondegenerate; that is, the $\text{Hess } L$ at each critical point is nonsingular.

We present an important consequence of Morse Lemma [12].

Corollary 6: Nondegenerate critical points of Morse functions (smooth, real-valued functions) on smooth manifolds are isolated. Moreover, Morse functions on a compact manifold have finitely many critical points, and they are isolated.

¹Given $\omega \in \mathbb{R}^3$, we express $\hat{\omega} \in \mathfrak{so}(3)$ or equivalently $(\omega)^\wedge \in \mathfrak{so}(3)$.

The set of critical points \mathcal{A} may consist of minimizers, maximizers, and saddle points.

Definition 7 (strict saddle points): For a smooth function $L : SO(3) \rightarrow \mathbb{R}$, a critical point is called a strict saddle point if the eigenvalues of $\text{Hess } L$ satisfy $\lambda_{\min} < 0$, $\lambda_{\max} > 0$, and $\lambda_i \neq 0$ for each $i = \{1, 2, 3\}$.

II. BACKGROUND AND MOTIVATION

A. Preliminaries

It is well known that a convex objective function defined on \mathbb{R}^n can be globally minimized using a first-order optimization method, such as standard gradient descent, or a first-order accelerated optimization method, such as a heavy ball method [14]. A convex objective function has a unique critical point (or stationary point, i.e., a unique point where the gradient of the function vanishes). Let the convex objective function be $f : \mathbb{R}^n \rightarrow \mathbb{R}$. It is possible to interpret a standard gradient descent algorithm as a first-order dynamical system of the form $\dot{x} = u$, and then design a controller $\kappa : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and assign the value of κ to u ($u = \kappa(x)$) as the negative of the gradient of f , i.e.,

$$\dot{x} = -\text{grad } f(x). \quad (1)$$

A discrete-time realization of (1) gives the standard gradient descent algorithm and is given by

$$x_{k+1} = x_k + dt(-\text{grad } f(x_k)), \quad (2)$$

where $k \in \{1, 2, \dots\}$ denotes the discrete time and $dt > 0$ is the step size. For a convex objective function, this algorithm searches for the global minimum by taking steps along the negative gradient direction.

Similarly, it is possible to associate a second-order dynamical system to minimize the given objective function f with acceleration. One such accelerated method is the heavy ball method, which has tunable “friction” and “gravity” parameters [1]. A discrete-time realization of the heavy ball algorithm for the convex objective function f is given by [15]

$$\begin{aligned} y_k &= \lambda y_{k-1} + dt(-\gamma \text{grad } f(x_k)) \\ x_{k+1} &= x_k + dt(y_k), \end{aligned} \quad (3)$$

where $dt > 0$ is the step size, and λ and γ are the friction and gravity parameters, respectively. When the objective function is convex, both the gradient descent and heavy-ball (accelerated gradient descent) method successfully find the global minimum of the convex objective function [14].

B. Challenges to Optimization on $SO(3)$

Finding a local minimizer of a cost function defined on $SO(3)$ that is not necessarily convex and may have multiple isolated local minimizers with a guarantee of “global” convergence to a minimizer is a challenging problem. By *global*, we mean convergence to the set of minimizers from everywhere on the Lie group. The first challenge in solving this problem is that if the initial condition belongs to the set of critical points excluding the set of local minimizers, under gradient descent (as given in (2)), heavy ball (as given in (3)), or even Nesterov algorithm (as in [16]), convergence to the set of local minimizers is not guaranteed. A second challenge

is that the objective function is defined on a manifold. Standard integration techniques, such as those given in (2) and (3), do not guarantee that the integrated solution stays on the manifold [9].

To address the integration issue, we associate the optimization problem with a dynamical system evolving on $\text{SO}(3)$. A first-order dynamical system evolving on $\text{SO}(3)$ is given by $\dot{R} = R\hat{\Omega}$, where $R \in \text{SO}(3)$ and $\hat{\Omega} \in \mathfrak{so}(3)$, which can be interpreted as the state and the control input, respectively. To assure a decrease of the objective function $L : \text{SO}(3) \rightarrow \mathbb{R}$ in the gradient direction, we set $\Omega = \text{grad } L(R)$. It should be noted that elementary integration techniques used to discretize a continuous-time system defined on \mathbb{R}^n , when applied to a system evolving on a manifold, do not guarantee that the resulting solution stays on the manifold [7]. On the other hand, geometric numerical integration preserves the geometric structure of a dynamical system. Using geometric numerical integration, the first-order system takes the form

$$R_{k+1} = R_k \exp(dt \text{grad } L(R_k))^\wedge, \quad (4)$$

where $dt > 0$ is the step size. The geometric gradient descent (4) can be formulated as Algorithm 1, and we call it *the geometric gradient descent algorithm*.

Algorithm 1 Local geometric gradient descent (LGGD) algorithm on $\text{SO}(3)$

Input: Initialize R in (4) to some R_0 , and pick positive scalar parameters γ, λ , a fixed step size $dt, \epsilon > 0$, and a time horizon T .

- 1: **for** $k = 0$ to T **do**
 - 2: Compute $\text{grad } L(R_k)$
 - 3: Update R according to (4)
 - 4: **if** $|\text{grad } L(R_k)| < \epsilon$ **then**
 - 5: Break
 - 6: **end if**
 - 7: **end for**
-

To achieve accelerated convergence, we associate the following second-order control system whose configuration space is the Lie group $\text{SO}(3)$ and the state space is the tangent bundle of the Lie group $\text{TSO}(3)$:

$$\dot{R} = R\hat{\Omega}, \quad \dot{\Omega} = u, \quad (5)$$

where $R \in \text{SO}(3)$, $\hat{\Omega} \in \mathfrak{so}(3)$, $\Omega \in \mathbb{R}^3$, and $u \in \mathbb{R}^3$. Designing an appropriate controller for (5) to steer the state (R, Ω) to a desired set is equivalent to designing an appropriate optimization algorithm to minimize the objective function [9], [17], [18]. Using geometric numerical integration, the second-order system takes the following discrete-time form [19]:

$$\begin{aligned} R_{k+1} &= R_k \exp\left(dt \hat{\Omega}_k\right) \\ \Omega_{k+1} &= \Omega_k + dt \left(\kappa \left(\text{grad } L(R_k), (R_k, \hat{\Omega}_k) \right) \right), \end{aligned} \quad (6)$$

where dt is the integration time step, $R_k \in \text{SO}(3)$ and $\Omega_k \in \mathbb{R}^3$ is the rotation matrix and the velocity at the k^{th} iteration,

respectively, and

$$\kappa \left(\text{grad } L(R_k), (R_k, \hat{\Omega}_k) \right) := (-\gamma \text{grad } L(R_k))^\vee - \lambda \Omega_k. \quad (7)$$

We call (6) along with (7) *the accelerated geometric gradient descent algorithm*. These equations can be formulated as Algorithm 2.

Algorithm 2 Local accelerated geometric gradient descent (LAGGD) algorithm on $\text{SO}(3)$

Input: Initialize R and Ω in (6) to some R_0 and Ω_0 , respectively, and pick positive scalar parameters γ, λ , a fixed step size $dt, \epsilon > 0$, and a time horizon T .

- 1: **for** $k = 0$ to T **do**
 - 2: Compute $\text{grad } L(R_k)$
 - 3: Compute κ using (7)
 - 4: Update R and Ω according to (6)
 - 5: **if** $|\text{grad } L(R_k)| < \epsilon$ **then**
 - 6: Break
 - 7: **end if**
 - 8: **end for**
-

Next, we provide an example of an objective function on $\text{SO}(3)$ and apply Algorithms 1 and 2.

Example 1: Let $A \in \mathbb{R}^{3 \times 3}$ be a symmetric and positive definite matrix with distinct eigenvalues. Consider the objective function

$$L : \text{SO}(3) \rightarrow \mathbb{R}, \quad R \mapsto \text{trace}(A(I - R)). \quad (8)$$

The gradient of L at the identity is given by

$$\text{grad } L(R) = (A^\top R - R^\top A)^\vee. \quad (9)$$

It follows from [20, Lemma 2] that L has four isolated critical points, namely, $\mathcal{A} = \{I\} \cup \mathcal{R}(\pi, \mathcal{E}_v^{\mathbb{R}}(A))$, where $\mathcal{E}_v^{\mathbb{R}}(A)$ is the set of real eigenvectors of A , and $\mathcal{R}(\pi, \mathcal{E}_v^{\mathbb{R}}(A))$ represents rotation of 180° about each axis given by real eigenvectors of A . Let $\mathcal{A}_{\min} = \{I\}$. Since $\text{SO}(3)$ is compact, and L is a continuous function, this implies that L is a bounded map. Direct calculation shows that L attains its lower value at $R = I$, and the lower bound is 0. Moreover, L has one maximum point and two saddle points [21, Lemma 2.2].

We apply Algorithms 1 and 2 to the cost function (8) and show results of only Algorithm 2 in Figure 1, due to space limitations. Let A be a symmetric positive definite matrix with distinct eigenvalues, given by $A = [4, 1, 0; 1, 5, 2; 0, 2, 9]$. If $R_k \in (\mathcal{A} \setminus \mathcal{A}_{\min})$, the algorithm fails to converge, as shown in Figure 1 in the red (dashed) plot. However, the algorithm provides convergence from every other point on $\text{SO}(3)$, i.e., $\text{SO}(3) \setminus (\mathcal{A} \setminus \mathcal{A}_{\min})$, as shown by the solid blue line in Figure 1.

Algorithms 1 and 2 fail to minimize the given objective function if initialized at any critical point or a stationary point. Moreover, even if the starting point does not belong to the set of maximizers or saddle points, there is no guarantee that the solution will not get stuck at a saddle point. This

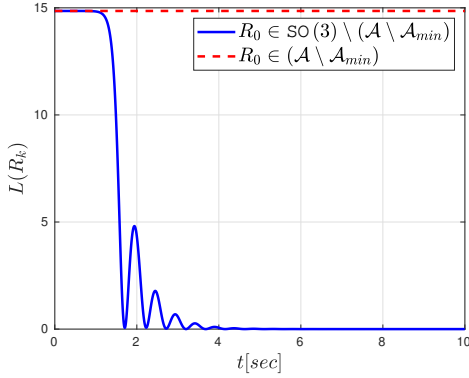


Fig. 1. Optimization of Objective function L using Algorithm 2.

motivates the need for new algorithms that guarantees convergence to the set of minimizers even when initialized from any critical point. In the subsequent sections, we propose two algorithms that promise to address the issues observed in Example 1.

III. PROBLEM FORMULATION AND ASSUMPTIONS

We aim to optimize a non-convex objective function defined on the Special Orthogonal group. By optimizing the objective function, we mean designing an algorithm that seeks (a local) minimum of the objective function starting from every point on the manifold. Formally, we consider the following problem in this work.

Problem 1: Given a continuously differentiable objective function on $\text{SO}(3)$, i.e., $L : \text{SO}(3) \rightarrow \mathbb{R}$, which may have multiple isolated minimizers, maximizers, and saddle points, design an optimization algorithm that guarantees global convergence to a local minimizer (not necessarily a global minimizer) everywhere on $\text{SO}(3)$, including local maximizers and saddle points.

The algorithms aimed at solving this problem require the following assumptions and properties.

Assumption 1: The function L is a Morse function.

Lemma 8: The function L has isolated critical points. Moreover, every saddle point is a strict-saddle point.

Assumption 2: The function L satisfies $L(R) > 0$ for all $R \in \text{SO}(3) \setminus \{I\}$ and $L(R) = 0$ for $R = I$.

Next, we assume a Lipschitz-like condition on the gradient and the Hessian of L . Since L is defined on a Lie group, we exploit the property of left invariance and the exponential map and define the following maps. Let $K : \mathbb{T}_R \text{SO}(3) \rightarrow \mathbb{T}_I \text{SO}(3)$, which is equivalent to $K : \mathbb{T}_R \text{SO}(3) \rightarrow \mathfrak{so}(3)$, because $\mathbb{T}_I \text{SO}(3)$ is isomorphic to $\mathfrak{so}(3)$. With $\exp : \mathfrak{so}(3) \rightarrow \text{SO}(3)$, we define the map $\xi := L \circ \exp \circ K : \mathbb{T}_R \text{SO}(3) \rightarrow \mathbb{R}$. Similar to [22], we make the following assumptions on the gradient and the Hessian of L .

Assumption 3: There exist $b_1 > 0$ and $M > 0$ such that for all $R \in \text{SO}(3)$ and for all $V_R \in \mathbb{T}_R \text{SO}(3)$ with $|V_R| \leq b_1$, we have

$$|\text{grad } \xi(V_R) - \text{grad } \xi(0)| \leq M |V_R|.$$

Assumption 4: There exist $b_2 > 0$ and $N > 0$ such that for all $R \in \text{SO}(3)$ and for all $V_R \in \mathbb{T}_R \text{SO}(3)$ with $|V_R| \leq b_2$,

$$|\text{Hess } \xi(V_R) - \text{Hess } \xi(0)| \leq N |V_R|.$$

Assumptions 3 and 4 imply that the gradient and the Hessian are Lipschitz continuous. In general, verifying Assumptions 3 and 4 is challenging. However, the following lemma, a special case of [22, Lemma 3.2], ensures that the objective function satisfies Assumptions 3 and 4.

Lemma 9: Given a smooth objective function L defined on a compact manifold $\text{SO}(3)$, for arbitrary $b_1, b_2 > 0$, there exist L^* , the minimum value of L , and $M, N > 0$ such that Assumption 3 and 4 are satisfied.

Given the objective function satisfying Assumption 1, 2, 3 and 4, we propose two accelerated geometric algorithms in the next section that minimize the objective function L defined in Problem 1. Finally, for the objective function $L : \text{SO}(3) \rightarrow \mathbb{R}$, it is possible to compute a closed-form expression of its gradient at R , namely, [8]

$$\text{grad } L = \frac{1}{2} \left(DL - R(DL)^\top R \right). \quad (10)$$

IV. PERTURBED GEOMETRIC ACCELERATED ALGORITHMS

Achieving global convergence, i.e., convergence from every point on $\text{SO}(3)$, is a challenging problem because of the possibility of multiple critical points, including saddle points. This requires detecting if the state of the system, or equivalently, the argument of the objective function, is arbitrarily close to the set of critical points except for the set of minimizers. This section proposes two schemes that, based on numerics, escape saddle points and a set of maximizers.

A. Perturbed accelerated geometric gradient descent algorithm (Hessian-based)

First, we propose an accelerated geometric gradient descent algorithm that uses random perturbations to escape critical points other than the local minimizers. We call this the perturbed accelerated geometric gradient descent (PAGGD-1) algorithm. One way to detect if the argument of the objective function is close to a critical point (stationary point) is by using the values of the gradient and Hessian of L . Calculating the Hessian is computationally expensive, especially on manifolds. However, on $\text{SO}(3)$, it is possible to compute the Hessian efficiently in local coordinates. As outlined in [23], $\text{SO}(3)$ can be covered by at least four charts, and by computing a closed-form expression of the Hessian in one of these four charts, the eigenvalues of the Hessian can be analyzed. It should be noted that the Hessian needs to be computed only at the critical points, i.e., the points where the gradient vanishes. Therefore, the proposed scheme is not computationally expensive.

First, we compute the gradient of the objective function $\text{grad } L(R_k)$ at the k^{th} iteration. If at the k^{th} iteration, $\text{grad } L(R_k)$ is arbitrarily close to zero, either we have reached the desired point (the minimum) or another critical point (including a maximum or a saddle point). To determine if the objective function has reached the minimum point, we compute the smallest eigenvalue of the Hessian matrix in the

appropriate chart. If $\lambda_{\min}(\text{Hess } L) > 0$, the objective function has achieved the minimum value. If not, we perturb the system with a random and exponentially decaying rotation matrix. The decaying exponential disturbance is chosen so the system remains on the manifold. For each $i \in \{1, 2, 3\}$, let μ_i be a random number drawn from the open interval $(0, 1)$. The random matrix $R_\mu \in \text{SO}(3)$ is constructed as

$$R_\mu = \exp \left(\begin{bmatrix} \exp(-t\mu_1) & & \\ & \exp(-t\mu_2) & \\ & & \exp(-t\mu_3) \end{bmatrix} \wedge \right). \quad (11)$$

By construction of R_μ , it follows that as $t \rightarrow \infty$, $R_\mu \rightarrow I$. We demonstrate this algorithm in the simulation section and show that, at least numerically, it leads to global convergence to the set of minimizers. A formal analysis of this algorithm is beyond the scope of this paper.

Algorithm 3 Perturbed accelerated geometric gradient descent (PAGGD-1) algorithm on $\text{SO}(3)$

Input: Initialize R and Ω in (6) to some R_0 and Ω_0 , respectively, and pick positive scalar parameters γ, λ , a fixed step size dt , $\epsilon_1 > 0$, and a time horizon T .

```

for  $k = 0$  to  $T$  do
2:   Compute  $\text{grad } L$  using (10)
   if  $|\text{grad } L(R_k)| < \epsilon_1$  then
4:     Compute  $\text{Hess } L$  in local coordinates
     Compute the minimum eigenvalue of  $\text{Hess } L$ , i.e.,
      $\lambda_{\min}(\text{Hess } L)$ 
6:     if  $\lambda_{\min}(\text{Hess } L) < 0$  then
        $R_{k+1} = (R_\mu R_k) \exp(dt \hat{\Omega}_k)$ 
8:     Compute  $\text{grad } L$  using (10)
     else
10:      Break
     end if
12:  end if
   Compute  $\kappa$  using (7)
14:  Update  $R$  and  $\Omega$  according to (6)
end for

```

B. Perturbed accelerated geometric gradient descent algorithm (gradient-based)

Next, we propose an algorithm that escapes the maxima and strict saddle points without using the information of the Hessian. We call this the perturbed accelerated geometric gradient descent (PAGGD-2) algorithm. This algorithm minimizes the objective function until the gradient of L is arbitrarily close to zero. When the gradient of L is sufficiently close to zero, the current value of the objective function is stored in the variable L^* . Next, we perturb the current position using the perturbation matrix R_μ defined in (11). After that, the perturbed system is propagated under geometric gradient descent Algorithm 2 for a small horizon H . Then the value of the objective function is compared with L^* . If $L(R)$ does not achieve a value smaller than L^* , optimization is stopped. Otherwise, the critical point (or stationary point) is escaped, and the procedure is repeated. We formalize this procedure in Algorithm 4. We show, at least numerically, that the algorithm escapes saddle points,

and maxima with high probability, as demonstrated in the simulation section.

Algorithm 4 Perturbed accelerated geometric gradient descent (PAGGD-2) algorithm on $\text{SO}(3)$

Input: Initialize R and Ω in (6) to some R_0 and Ω_0 , respectively, and pick positive scalar parameters γ, λ , a fixed step size dt , $\epsilon_1 > 0$ and $\epsilon_2 > 0$, a time horizon T , and a small horizon window $H > 0$.

```

for  $k = 0$  to  $T$  do
  Compute  $\text{grad } L$  using (10)
3:  if  $|\text{grad } L(R_k)| < \epsilon_1$  then
     $L^* \leftarrow L(R)$ .
    Generate perturbation  $R_\mu \in \text{SO}(3)$  as in (11)
6:     $R_k \leftarrow R_\mu R_k$ 
    for  $j = 0$  to  $H$  do
      Compute  $\kappa$  using (7)
9:      Update  $R$  and  $\Omega$  according to (6)
      Compute  $\text{grad } L$  using (10)
      Compute  $L(R_k)$ 
12:    end for
    if  $(L^* - L(R_k)) < \epsilon_2$  then
      Break
15:    end if
    end if
    Compute  $\kappa$  using (7)
18:  Update  $R$  and  $\Omega$  according to (6)
end for

```

V. SIMULATIONS

We simulate the objective function introduced in Example 1 with $A = [4, 1, 0; 1, 5, 2; 0, 2, 9]$. In all simulations, gravity γ and friction λ parameters are selected as 3 and 2, respectively. It follows from [21, Lemma 2.2] that L has a maximum point, a minimum point, and two saddle points. As mentioned in Example 1, the minimum point, denoted by C_m , is the identity matrix. The maximum point, and the two saddle points, denoted by C_M, C_{s_1} , and C_{s_2} , respectively, are computed by applying a rotation of 180° about each eigenvector of A , i.e., $\mathcal{R}(\pi, \mathcal{E}_v^{\mathbb{R}}(A))$. In summary, the set of critical points consists of $\{C_M, C_{s_1}, C_{s_2}, C_m\}$. Next, we simulate Algorithms 3 and 4; the code is available online².

A. Simulation results of Algorithm 3 (PAGGD-1)

The goal is to optimize the objective function (8) given in Example 1 using Algorithm 3. We ran the algorithm three times, starting from different initial conditions, as shown by three plots in Figure 2. The initial conditions are chosen to be the most challenging possible, i.e., starting from the maximum point and each of the two saddle points. It should be noted that non-perturbed algorithms, such as Algorithm 1 and Algorithm 2 fail and these failed results are not shown in Figure 2. In the first case, the initial condition is $R_k = C_M$, i.e., a stationary point, which is a maximum point. As seen in Figure 2, the value of the objective function is $L(R_0) = 14.8$,

²<https://github.com/HybridSystemsLab/AcceleratedGeometricGD>

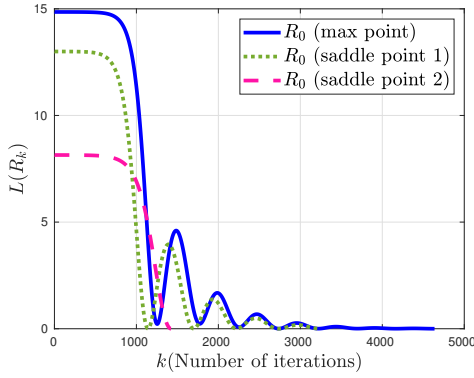


Fig. 2. Optimizing the objective function L using Algorithm 3.

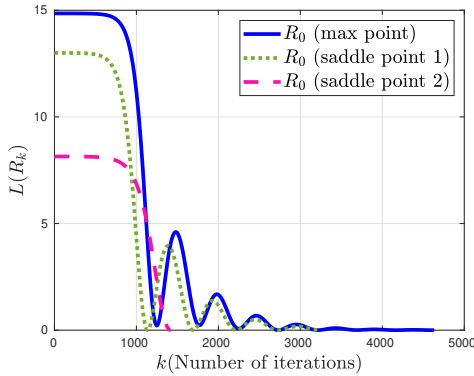


Fig. 3. Optimizing the objective function L using Algorithm 4.

and under Algorithm 3, the objective function achieves its minimum value in less than 5000 iterations, as seen by the solid blue curve in Figure 2. In the second and third cases, the initial conditions are chosen to be the saddle points C_{s_1} and C_{s_2} . The value of the objective function $L(R_0)$ at each saddle point is 13 and 8.1, respectively. As seen by the dotted green and dashed magenta plots in Figure 2, in each case, Algorithm 3 practically minimizes the objective function successfully in a finite number of steps. We want to highlight that, the Hessian information is used only near the critical points, and hence the algorithm is not computationally very expensive.

B. Simulation results of Algorithm 4 (PAGGD-2)

Next, we optimize the same objective function (8), used in the previous section and as given in Example 1, but using Algorithm 4. This algorithm aims to escape saddle points and maxima without using the Hessian information. Similar to the previous subsection, we ran Algorithm 4 three times, where at each time, the initial conditions are the maximum point and the two saddle points. As seen in Figure 3, starting from all three points, the objective function approximately achieves the minimum value in a finite number of steps.

VI. CONCLUSIONS

This work proposes two accelerated geometric algorithms to minimize a nonconvex objective function defined on a Lie

group. The algorithms are based on the idea of detecting a critical point other than the minimizer and then adding a random perturbation to escape maxima and saddle points. Numerical simulations demonstrate that the algorithms successfully and efficiently escape saddle points and maxima. Future work includes formalizing mathematically the properties of the algorithms seen numerically.

REFERENCES

- [1] B. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.
- [2] D. Hustig-Schultz and R. G. Sanfelice, "A hybrid algorithm for practical nonconvex optimization," in *Proceedings of the 2021 International Symposium on Mathematical Theory of Networks and Systems*, August 2021.
- [3] M. Betancourt, M. I. Jordan, and A. C. Wilson, "On symplectic optimization," *arXiv: Computation*, 2018.
- [4] A. Akhtar, S. Saleem, and S. L. Waslander, "Feedback linearizing controllers on $SO(3)$ using a global parametrization," in *2020 American Control Conference (ACC)*, 2020, pp. 1441–1446.
- [5] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ: Princeton University Press, 2008.
- [6] T. Strizic, J. I. Poveda, and A. R. Teel, "Hybrid gradient descent for robust global optimization on the circle," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 2985–2990.
- [7] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration*, 2nd ed., ser. Springer Series in Computational Mathematics. Springer-Verlag, Berlin, 2006, vol. 31, structure-preserving algorithms for ordinary differential equations.
- [8] H. Sharma, T. Lee, M. Patil, and C. Woolsey, "Symplectic accelerated optimization on $SO(3)$ with lie group variational integrators," in *2020 American Control Conference (ACC)*, 2020, pp. 2826–2831.
- [9] T. Lee, M. Tao, and M. Leok, "Variational symplectic accelerated optimization on lie groups," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 233–240.
- [10] A. Akhtar and S. L. Waslander, "Controller class for rigid body tracking on $SO(3)$," *IEEE Transactions on Automatic Control*, vol. 66, no. 5, pp. 2234–2241, 2021.
- [11] A. Akhtar and R. G. Sanfelice, "A class of hybrid geometric controllers for robust global asymptotic stabilization on S^1 ," in *2022 American Control Conference (ACC)*, 2022, pp. 874–879.
- [12] J. Milnor, *Morse Theory. (AM-51)*. Princeton University Press, 2016. [Online]. Available: <https://doi.org/10.1515/9781400881802>
- [13] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems*, ser. Texts in Applied Mathematics. New York-Heidelberg-Berlin: Springer-Verlag, 2004, vol. 49.
- [14] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [15] D. M. Hustig-Schultz and R. G. Sanfelice, "A robust hybrid heavy ball algorithm for optimization with high performance," in *2019 American Control Conference (ACC)*, 2019, pp. 151–156.
- [16] Y. Nesterov, *Lectures on Convex Optimization*, 2nd ed. Springer Publishing Company, Incorporated, 2018.
- [17] U. Helmke and M. J. B., *Optimization and dynamical systems / Uwe Helmke and John B. Moore*. Springer-Verlag London ; New York, 1994.
- [18] W. Su, S. Boyd, and E. J. Candès, "A differential equation for modeling nesterov's accelerated gradient method: Theory and insights," *Journal of Machine Learning Research*, vol. 17, no. 153, pp. 1–43, 2016.
- [19] A. M. Bloch, I. I. Hussein, M. Leok, and A. K. Sanyal, "Geometric structure-preserving optimal control of a rigid body," *Journal of Dynamical and Control Systems*, vol. 15, pp. 307–330, 2009.
- [20] C. G. Mayhew and A. R. Teel, "Synergistic potential functions for hybrid control of rigid-body attitude," in *Proceedings of the 2011 American Control Conference*, 2011, pp. 875–880.
- [21] M. Izadi and A. K. Sanyal, "Rigid body attitude estimation based on the lagrange-d'alembert principle," *Automatica*, vol. 50, no. 10, pp. 2570–2577, 2014.
- [22] C. Criscitiello and N. Boumal, "Efficiently escaping saddle points on manifolds," in *Neural Information Processing Systems*, 2019.
- [23] E. W. Grafarend and W. Kühnel, "A minimal atlas for the rotation group $SO(3)$," *GEM - International Journal on Geomathematics*, vol. 2, pp. 113–122, 2011.