

Consensus on Lie groups for the Riemannian Center of Mass

Spencer Kraiser, *Student Member, IEEE*, Shahriar Talebi, *Student Member, IEEE*,
and Mehran Mesbahi, *Fellow, IEEE*

Abstract—In this paper, we develop a consensus algorithm for distributed computation of the Riemannian Center of Mass (RCM) on Lie Groups. The algorithm is built upon a distributed optimization reformulation that allows developing an intrinsic, distributed (without relying on a consensus subroutine), and a computationally efficient protocol for the RCM computation. The novel idea for developing this fast distributed algorithm is to utilize a Riemannian version of distributed gradient flow combined with a gradient tracking technique. We first guarantee that, under certain conditions, the limit point of our algorithm is the RCM point of interest. We then provide a proof of global convergence in the Euclidean setting, that can be viewed as a “geometric” dynamic consensus that converges to the average from arbitrary initial points. Finally, we proceed to showcase the superior convergence properties of the proposed approach as compared with other classes of consensus optimization-based algorithms for the RCM computation.

Index Terms—Consensus on Lie Groups, Riemannian Center of Mass, Karcher Mean, Multiagent Systems

I. INTRODUCTION

Consensus algorithms are a ubiquitous class of protocols with pertinent applications to fields such as distributed estimation, optimization, and control of multi-agent systems. At a foundational level, consensus algorithms steer a set of dynamical agents towards a single point. *Average* consensus necessitates this point to be the average of the agents’ initial states. Statistical and computational advantages of average consensus algorithms have found a wide range of applications in distributed resource allocation, formation control, and distributed estimation [1]–[3].

While consensus algorithms have predominantly been studied for Euclidean spaces, there has been a number of efforts in the literature to generalize this protocol to Riemannian manifolds [4]–[10]. One notable application is 3D localization of camera sensors [11]. In this direction, useful notions from Euclidean geometry pertinent to the notion of average consensus generalize to Riemannian manifolds. This includes the Riemannian Center of Mass (RCM), replicating the notion of an average [12].

The Riemannian consensus algorithms discussed so far achieve consensus but not *consensus to the RCM* (or the so-called “RCM consensus”). To the best of our knowledge,

The authors are with the William E. Boeing Department of Aeronautics and Astronautics, University of Washington, Seattle, WA, USA. Emails: {kraiser+shahriar+mesbahi}@uw.edu. The research of the authors has been supported by NSF grant ECCS-2149470 and AFOSR grant FA9550-20-1-0053.

[13, §3.2] is the earliest work on the topic of RCM consensus. The method was later extended to a larger regime of Riemannian manifolds in [14]. In order to guarantee RCM consensus, this approach utilizes a consensus subroutine within each iteration. However, practical implementation of this subroutine may not be always favorable and time-efficient. This calls for a fast distributed RCM consensus algorithm that is, 1) intrinsic, and as such, parameterization of the manifold does not affect its properties; 2) completely distributed without relying a consensus subroutine; and 3) has at least a linear convergence rate.

In this paper, we first reformulate the RCM for a set of points on a Riemannian manifold as the unique optimal solution to a consensus optimization problem. This perspective calls to explore an array of techniques including distributed optimization on Riemannian manifolds and dynamic average consensus. In this direction, we also provide an optimizer for Lie groups equipped with a bi-invariant metric with the aforementioned properties, thus achieving a distributed RCM consensus algorithm. We prove that, under certain conditions, the limit point of our algorithm is the RCM of the original points. We provide global convergence guarantees in the Euclidean case and compare the performance of our proposed algorithm with distributed constrained optimization approaches such as penalty-based and Lagrangian-based methods.

The rest of the paper is organized as follows. §II offers the problem formulation and a technical background on Riemannian geometry, RCM and Lie groups. In §III, we present the RCM optimization reformulation utilized to develop fast and simple distributed RCM consensus. We then proceed with limit point analysis and global convergence guarantees of the proposed algorithm for the Euclidean space in §IV. Finally, several simulation scenarios are presented in §V, followed by concluding remarks in §VI.

II. PROBLEM STATEMENT AND BACKGROUND

Before delving deeper into necessary mathematical background, let us first present the problem statement. Consider a Lie group \mathcal{G} equipped with a bi-invariant Riemannian metric and a communication network of agents $[N] = \{1, 2, \dots, N\}$ represented by an undirected connected graph $G = ([N], E)$. Given N points $z_i \in \mathcal{G}$, the goal is to design distributed dynamics for each agent: $\dot{x}_i(t) = \mathbf{F}_i(t, x_i, x_j : j \sim i)$, steering the agents’ states to the RCM of $\{z_1, \dots, z_N\}$.

A. Riemannian Geometry

For geometric concepts, we follow the standard notation as in [15]. Let $(\mathcal{M}, \langle \cdot, \cdot \rangle)$ be a Riemannian manifold and $d(\cdot, \cdot)$ the corresponding induced geodesic distance. In this paper, we assume that \mathcal{M} has a bounded sectional curvature. We denote its N -fold Cartesian product as \mathcal{M}^N and use boldface letter $\mathbf{x} = (x_1, \dots, x_N) \in \mathcal{M}^N$ to denote a point on the product space.

We denote the open *geodesic ball* centered at $x \in \mathcal{M}$ with radius $r > 0$ as,

$$\mathcal{B}(x, r) := \{y \in \mathcal{M} : d(x, y) < r\}.$$

A subset $A \subset \mathcal{M}$ is called *geodesically convex* (*g-convex*) if for any $x, y \in A$, there exists a unique *minimizing* geodesic $\gamma : [0, T] \rightarrow \mathcal{M}$ connecting x to y such that γ is contained in A and there exists no other *minimizing* geodesic connecting the pair¹. A function $f : A \rightarrow \mathbb{R}$ is called *g-convex* if for any geodesic γ contained in A , $f \circ \gamma$ is convex in the Euclidean sense [16]. The *convexity radius* of \mathcal{M} is defined as

$$r^* := \frac{1}{2} \min(\text{inj}(\mathcal{M}), \frac{\pi}{\sqrt{\Delta}}),$$

where $\text{inj}(\mathcal{M})$ is the injectivity radius and Δ is the upper bound on the sectional curvature of \mathcal{M} . If $r \leq r^*$, then $\mathcal{B}(x, r)$ is *g-convex* [4], [12].

Next, for $\mathbf{z} = (z_1, \dots, z_N) \in \mathcal{M}^N$, define its *Riemannian Center of Mass (RCM)* as any minimizer of,

$$\min_{y \in \mathcal{M}} \sum_{i=1}^N d(y, z_i)^2; \quad (1)$$

the RCM may not exist nor be unique. We denote the RCM of \mathbf{z} as $\text{RCM}(\mathbf{z})$ or \bar{z} when it exists and is unique. One can induce existence and uniqueness guarantees as follows. Define the *convexity submanifold*:

$$\mathcal{C} := \{\mathbf{z} \in \mathcal{M}^N : \exists y \in \mathcal{M}, \text{ s.t. } z_i \in \mathcal{B}(y, r^*) \forall i\}. \quad (2)$$

If $\mathbf{z} \in \mathcal{C}$, then $\text{RCM}(\mathbf{z})$ exists and is unique [12, Thm. 2.1]. Now, let \mathcal{B} be any geodesic ball satisfying the condition in (2) for \mathbf{z} . If some $z \in \mathcal{B}$ satisfies the *Karcher equation*,

$$\sum_{i=1}^N \log_z(z_i) = 0, \quad (3)$$

then we necessarily have $z = \text{RCM}(\mathbf{z})$.

B. Lie Groups

A *Lie group* \mathcal{G} is a smooth manifold with a group structure where the group and inverse operators are smooth mappings. Every (Lie) group admits the *identity* e such that $xe = ex = x$ and $xx^{-1} = x^{-1}x = e$ for all $x \in \mathcal{G}$. The tangent space of \mathcal{G} at the identity is called the *Lie algebra* $\mathfrak{g} := T_e\mathcal{G}$. The Lie algebra is a vector space equipped with a vector multiplication operator called the commutator. We suggest [17], [18] for introductions to Lie groups with emphasis on Riemannian geometry and control theory.

¹Some authors call this *strong* geodesic convexity.

A Riemannian metric $\langle \cdot, \cdot \rangle$ on \mathcal{G} is called *left-invariant* if

$$\langle \xi, \eta \rangle_e = \langle dL_x\xi, dL_x\eta \rangle_x,$$

for any $x \in \mathcal{G}$, $\xi, \eta \in \mathfrak{g}$. Here, dL_x is the differential of the left-translation map $L_x : y \mapsto xy$. *Right-invariant* metrics are defined similarly with the right-translation map $R_x : y \mapsto yx$. *Bi-invariant* metrics are both left- and right-invariant. If $\mathcal{G} \subset GL(n)$ is a *matrix Lie group*, then $dL_x\xi = x\xi$.

All Lie groups admit left- and right-invariant metrics [15, Lemma 3.10]. A Lie group admits a bi-invariant metric if and only if it is isomorphic to the Cartesian product of a compact Lie group and a vector space [19, Lemma 7.5]. An example of a Lie group with a bi-invariant metric is the unit quaternions equipped with the dot product.

The manifold and Lie group exponential coincide at the identity when the metric is bi-invariant [17, Prop. 3.10.]. Since left-translation is an isometry, the following identities hold for these Lie groups [15, Prop. 5.9]: for any $x, y \in \mathcal{G}$ and $\xi \in \mathfrak{g}$,

$$\begin{aligned} \exp_x(dL_x\xi) &= L_x(\text{Exp}(\xi)), \\ \log_x(y) &= dL_x\text{Log}(x^{-1}y), \end{aligned}$$

where Exp and \exp_x are, respectively, the Lie and manifold exponential at x ; Log and \log_x , on the other hand, are their corresponding inverse mappings (wherever they are well-defined). This also implies that on $T_x\mathcal{G}$, we have

$$\nabla_x \left[\frac{1}{2} d(x, y)^2 \right] = -\log_x(y) = -dL_x\text{Log}(x^{-1}y).$$

III. THE ALGORITHM AND ITS DERIVATION

In this section, we present a reformulation of (1) and propose a solution algorithm that is intrinsic, distributed, and has an (empirically) linear convergence rate.

A. RCM Consensus Reformulation

Using a consensus reformulation, we can reformulate (1) as a *g-convex consensus optimization problem*,

$$\min_{\mathbf{x} \in \mathcal{M}^N} \mathbf{f}(\mathbf{x}) := \frac{1}{2} \sum_{i=1}^N d(x_i, z_i)^2 \quad (5a)$$

$$\text{s.t. } \mathbf{x} \in \mathcal{A}, \quad (5b)$$

where $\mathcal{A} = \{(x, \dots, x) : x \in \mathcal{M}\}$ is the *agreement submanifold*. Also, note that on a connected graph, (5b) is equivalent to zero *consensus error* $\varphi(\mathbf{x}) = 0$, defined as

$$\varphi(\mathbf{x}) := \frac{1}{2} \sum_{\{i, j\} \in E} d(x_i, x_j)^2 = \frac{1}{4} \sum_{i=1}^N \sum_{j \sim i} d(x_i, x_j)^2. \quad (6)$$

Thus, the unique solution to (5) on \mathcal{C} (where its existence and uniqueness is guaranteed) corresponds to $\mathbf{x}^* := (\bar{z}, \dots, \bar{z})$.

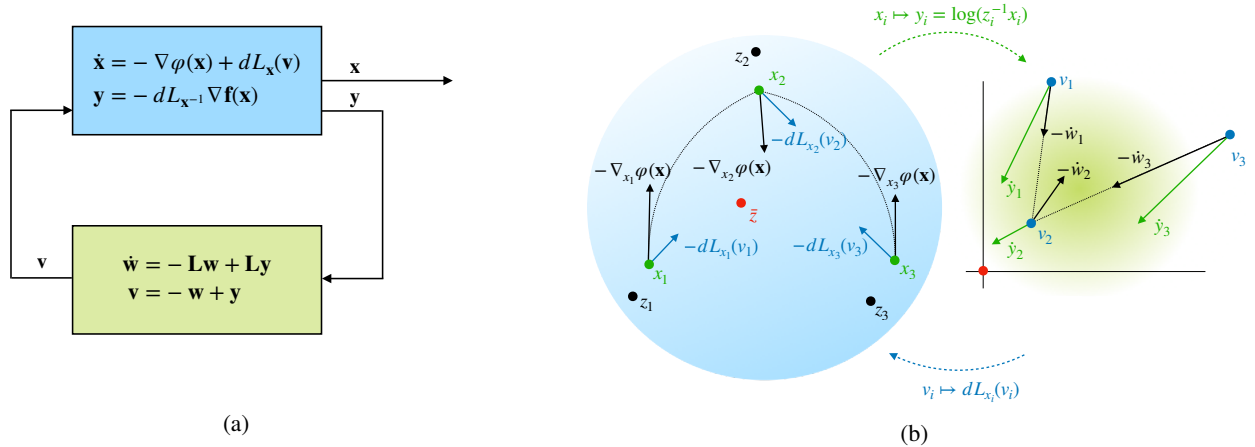


Fig. 1: (a) Block diagram of Algorithm 1. (b) Illustration of the correspondence in Algorithm 1 between the state consensus on the Lie group (left) and the gradient dynamic consensus on its Lie algebra (right), interconnected by two mappings.

This optimization reformulation enables us to design a distributed algorithm for RCM consensus using *local first order information*. That is, for agent i we use²

$$\nabla_{x_i}\mathbf{f}(\mathbf{x}) = -dL_{x_i}\text{Log}(x_i^{-1}z_i), \quad (7a)$$

$$\nabla_{x_i}\varphi(\mathbf{x}) = -dL_{x_i}\sum_{j\sim i}\text{Log}(x_i^{-1}x_j), \quad (7b)$$

which depend only on the variables $\{x_i, x_j : j \sim i\}$ that are locally available. Therefore, a fast first order optimizer will be a solution to our problem.

B. Our Algorithm

Algorithm 1. Given $\mathbf{z} = (z_1, \dots, z_N) \in \mathcal{C} \subset \mathcal{G}^N$, let $\mathbf{L} : \mathfrak{g}^N \rightarrow \mathfrak{g}^N$ be the graph Laplacian of G , represented as a linear operator on the product Lie algebra \mathfrak{g}^N . For each agent i , set $x_i(0) := z_i \in \mathcal{G}$ and $w_i(0) := 0 \in \mathfrak{g}$. The proposed algorithm assumes the following dynamics:

$$\dot{\mathbf{x}} = -\nabla\varphi(\mathbf{x}) - dL_{\mathbf{x}}\mathbf{v} \quad (8a)$$

$$\dot{\mathbf{w}} = \mathbf{L}\mathbf{v}, \quad (8b)$$

where $\mathbf{v} = -\mathbf{w} + dL_{\mathbf{x}^{-1}}\nabla\mathbf{f}(\mathbf{x})$.

In order to show that Algorithm 1 is fully distributed, notice that the dynamics for each agent i reduces to

$$\dot{x}_i = dL_{x_i}\left(\sum_{j\sim i}\text{Log}(x_i^{-1}x_j) - v_i\right),$$

$$\dot{w}_i = \sum_{j\sim i}(v_i - v_j),$$

with $v_i = -w_i + \text{Log}(z_i^{-1}x_i)$. Later, we show that each $v_i \in \mathfrak{g}$ is tracking a global information about the distributed cost \mathbf{f} . Also, w_i is a latent state for a dynamic consensus algorithm on the Lie algebra (10).

C. Derivation of our Algorithm

Similar to the gradient flow as the simplest (continuous) optimizer for an optimization problem, the Distributed Gradient Flow (DGF) can be viewed as the simplest optimizer

²These closed-form expressions only hold for Lie groups with bi-invariant metrics. Nonetheless, (5), is still distributed for arbitrary Riemannian manifolds but (7) will be different.

for a *distributed* optimization problem. The state dynamics for DGF of the distributed cost \mathbf{f} assumes the form,

$$\dot{\mathbf{x}} = -\nabla\varphi(\mathbf{x}) - \nabla\mathbf{f}(\mathbf{x}). \quad (9)$$

Locally, each agent's state follows the dynamics

$$\dot{x}_i = dL_{x_i}\sum_{j\sim i}\text{Log}(x_i^{-1}x_j) + dL_{x_i}\text{Log}(x_i^{-1}z_i).$$

The first forcing term above drives the consensus error φ to zero, whereas the second term attempts to minimize the *local cost* $x \mapsto \frac{1}{2}d(x, z_i)^2$. This is a first attempt at minimizing the cost $x \mapsto \mathbf{f}(x, \dots, x) = \sum_{i=1}^N \frac{1}{2}d(x, z_i)^2$ in a distributed manner, whose global minimizer is $x = \bar{z}$. However, this method easily fails due to the excess of points that satisfy $\nabla\varphi(\mathbf{x}) = -\nabla\mathbf{f}(\mathbf{x})$ with $\mathbf{x} \notin \mathcal{A}$.

One can think of DGF as an “open loop” distributed optimizer for (5), i.e., while (9) includes global state feedback with the consensus term, there is no global *cost gradient feedback*. To resolve this issue, each agent needs access to the global information about the average gradients of local costs of all agents. The simplest way to introduce this cost gradient feedback is through Gradient Tracking (GT) [20]. This involves implementing a dynamic consensus algorithm on the Lie algebra for tracking this average gradient:

$$\dot{\mathbf{w}} = -\mathbf{L}\mathbf{w} + \mathbf{L}\mathbf{u}, \quad (10a)$$

$$\mathbf{v} = -\mathbf{w} + \mathbf{u}. \quad (10b)$$

Here, $w_i(0) = 0$ and we set $u_i = dL_{x_i^{-1}}\nabla_{x_i}\mathbf{f}(\mathbf{x})$. Under these dynamics, each v_i will track the same global information $\frac{1}{N}\sum_{j=1}^N dL_{x_j^{-1}}\nabla_{x_j}\mathbf{f}(\mathbf{x})$.

Finally, combining DGF with GT, which is achieved by replacing the term $-\nabla\mathbf{f}(\mathbf{x})$ in (9) with $-dL_{\mathbf{x}}\mathbf{v}$, gives us (8). In other words, we are “closing the loop” on the DGF by introducing a cost gradient feedback for each agent using GT. A schematic diagram of this design is also depicted in Figure 1a with its geometric interpretation in Figure 1b.

Remark 1. We do not require z_i 's to be constant. Suppose each z_i is time-varying and we wish for each x_i to track the time-varying RCM in a distributed manner. Indeed, we

have observed that our algorithm works empirically in this dynamic consensus setup.

Remark 2. Note that we have initialized $\mathbf{x}(0) = z_i$. This can be relaxed to only requiring $\mathbf{x}(0) \in \mathcal{C}$. The “closer” the points are initialized to \bar{z} , the faster the convergence. However, since \bar{z} is unknown a priori, we simply chose $\mathbf{x}_i(0) = z_i$; another candidate would be $\mathbf{x}_i(0) = e$.

IV. MAIN RESULTS

In this section, we first provide the limit point analysis of Algorithm 1 in the general case and then guarantee convergence for the Euclidean case.

Proposition 1 (Limit point). *Let $\mathcal{B} \subset \mathcal{G}$ be a geodesic ball with radius $r < r^*$ containing $\{z_1, \dots, z_N\}$, and consider the trajectory $(\mathbf{x}(\cdot), \mathbf{w}(\cdot))$ generated by Algorithm 1. Suppose $(\mathbf{x}(\cdot), \mathbf{w}(\cdot))$ converges to some $(\mathbf{x}^*, \mathbf{w}^*)$ such that $\mathbf{x}^* \in \mathcal{B}^N$. Then $x_i^* = \bar{z}$ and $w_i^* = \text{Log}(z_i^{-1}\bar{z})$ for each i .*

Proof. We can compactly express Algorithm 1 as

$$\begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} -\nabla\varphi(\mathbf{x}) - dL_{\mathbf{x}}[-\mathbf{w} + \text{Log}(\mathbf{z}^{-1}\mathbf{x})] \\ \mathbf{L}[-\mathbf{w} + \text{Log}(\mathbf{z}^{-1}\mathbf{x})] \end{bmatrix}.$$

Since we assume $(\mathbf{x}^*, \mathbf{w}^*)$ is a fixed point,

$$\mathbf{L}[-\mathbf{w}^* + \text{Log}(\mathbf{z}^{-1}\mathbf{x}^*)] = 0,$$

and hence,

$$-\mathbf{w}^* + \text{Log}(\mathbf{z}^{-1}\mathbf{x}^*) = (\xi, \dots, \xi) =: \xi \mathbf{1} \in \mathfrak{g}^N,$$

for some $\xi \in \mathfrak{g}$. Hence, $dL_{(\mathbf{x}^*)^{-1}}\nabla\varphi(\mathbf{x}^*) = \xi \mathbf{1}$. Then

$$N\|\xi\|^2 = \langle \xi \mathbf{1}, \xi \mathbf{1} \rangle = \langle dL_{(\mathbf{x}^*)^{-1}}\nabla\varphi(\mathbf{x}^*), \xi \mathbf{1} \rangle.$$

Since the metric is bi-invariant, we have

$$dL_{(x_i^*)^{-1}}\text{Log}_{x_i^*}(x_j^*) = \text{Log}((x_i^*)^{-1}x_j^*) = -dL_{(x_j^*)^{-1}}\text{Log}_{x_j^*}(x_i^*)$$

implying $\sum_{i=1}^N dL_{(x_i^*)^{-1}} \sum_{j \sim i} \text{Log}_{x_i^*}(x_j^*) = 0$. Therefore

$$\langle dL_{(\mathbf{x}^*)^{-1}}\nabla\varphi(\mathbf{x}^*), \xi \mathbf{1} \rangle = \left\langle \sum_{i=1}^N \sum_{j \sim i} \text{Log}((x_i^*)^{-1}x_j^*), \xi \right\rangle = 0.$$

Thus we must have $\xi = 0$, and therefore $\nabla\varphi(\mathbf{x}^*) = 0$.

Next, it is shown in [4, Theorem 5] that the critical points of φ restricted to \mathcal{C} coincide with \mathcal{A} . So, $\mathbf{x}^* \in \mathcal{C}$ and $\nabla\varphi(\mathbf{x}^*) = 0$ together imply $\mathbf{x}^* \in \mathcal{A}$. Next, note that

$$\frac{d}{dt} \sum_{i=1}^N \dot{w}_i(t) = 0.$$

As such, since $\mathbf{w}(0) = 0$, we must have $\sum_{i=1}^N w_i^* = 0$. Therefore, by the fact that $-\mathbf{w}^* + \text{Log}(\mathbf{z}^{-1}\mathbf{x}^*) = 0$, we arrive at

$$\sum_{i=1}^N \text{Log}(z_i^{-1}\mathbf{x}^*) = 0.$$

Since $x^*, z_1, \dots, z_N \in \mathcal{B}$ and x^* satisfies (3), $x^* = \bar{z}$, and thus $x_i^* = \bar{z}$ and $w_i^* = \text{Log}(z_i^{-1}\bar{z})$. \square

Next, we provide the convergence guarantees of our algorithm for $\mathcal{G} = \mathbb{R}^n$; the corresponding analysis for arbitrary Lie groups is the subject of our future work. Herein, we write $\mathbf{x} = \text{col}(x_1, \dots, x_N) \in \mathbb{R}^{Nn}$ to denote vertical concatenation

and use similar notation for \mathbf{z} , \mathbf{x} , and \mathbf{u} . Then the dynamics of Algorithm 1 reduces to the following matrix form:

$$\dot{\mathbf{x}} = -(\mathbf{L} \otimes I_n)\mathbf{x} + \mathbf{w} - (\mathbf{x} - \mathbf{z}) \quad (11a)$$

$$\dot{\mathbf{w}} = -(\mathbf{L} \otimes I_n)\mathbf{w} + (\mathbf{L} \otimes I_n)(\mathbf{x} - \mathbf{z}), \quad (11b)$$

where \otimes denote the Kronecker product. The RCM of initial states \mathbf{z} reduces to the Euclidean average

$$\bar{z} = \text{RCM}(\mathbf{z}) = \frac{1}{N} \sum_{i=1}^N z_i \in \mathbb{R}^n,$$

and so $\mathbf{x}^* = \text{col}(\bar{z}, \dots, \bar{z}) = \mathbf{1}_N \otimes \bar{z}$. The next result establishes guaranteed average consensus starting from arbitrary initial points.

Theorem 1 (Convergence in \mathbb{R}^n). *Suppose $\mathbf{x}(t), \mathbf{w}(t)$ is the trajectory generated by (11) over a connected graph G with $\mathbf{w}(0) = 0$. Then, starting from any arbitrary starting point $\mathbf{x}(0)$, we have an exponentially convergent trajectory with $\lim_{t \rightarrow \infty} x_i(t) = \bar{z}$ and $\lim_{t \rightarrow \infty} w_i(t) = \bar{z} - z_i$ for all i .*

Proof. First, by noting that $\mathbf{v} = -\mathbf{w} + \mathbf{x} - \mathbf{z}$ we reformulate the system in (11) as

$$\begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{x}} \end{bmatrix} = (A \otimes I_n) \begin{bmatrix} \mathbf{v} \\ \mathbf{x} \end{bmatrix}, \quad (12)$$

where

$$A := \begin{bmatrix} -\mathbf{L} - I_N & -\mathbf{L} \\ -I_N & -\mathbf{L} \end{bmatrix}. \quad (13)$$

Before proceeding, we need the following results on characterizing the spectrum of A with its proof deferred to the end of this section.

Lemma 1. *Suppose G is connected. Then zero is a simple eigenvalue of the matrix A in (13). Furthermore, all non-zero eigenvalues of this matrix have negative real parts.*

The spectral properties of A established above then implies that the dynamics of (13) is marginally stable. Let p, q be, respectively, the right and left eigenvectors of A associated with $\lambda = 0$ such that $p^\top q = 1$. So, $p \in \mathcal{N}(A)$. Also, since 0 is a simple eigenvalue of A^\top , we get

$$\mathcal{N}(A^\top) = \text{span} \left(\begin{bmatrix} \mathbf{1}_N \\ -\mathbf{1}_N \end{bmatrix} \right).$$

Thus, we set $p = \begin{bmatrix} 0_N \\ \mathbf{1}_N \end{bmatrix}$, $q = \frac{1}{N} \begin{bmatrix} -\mathbf{1}_N \\ \mathbf{1}_N \end{bmatrix}$. By properties of Kronecker product, we obtain that $A \otimes I_n$ is also marginally stable with a zero eigenvalue of algebraic and geometric multiplicity n . This eigenvalue has corresponding right eigenvectors $p \otimes e_i$ and left eigenvectors $q \otimes e_i$. Here, $e_i \in \mathbb{R}^n$ is the i -th standard basis for $i = 1, 2, \dots, n$.

Next, by computing $e^{(A \otimes I_n)t}$ using the Jordan decomposition and taking the limit as $t \rightarrow \infty$ we obtain that (see [21, Proposition 3.11] for a similar computation)

$$\lim_{t \rightarrow \infty} e^{(A \otimes I_n)t} = \sum_{i=1}^n (p \otimes e_i)(q \otimes e_i)^\top = (pq^\top) \otimes I_n.$$

Therefore, (13) converges as follows

$$\begin{bmatrix} \mathbf{v}^* \\ \mathbf{x}^* \end{bmatrix} := \lim_{t \rightarrow \infty} \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{x}(t) \end{bmatrix} = ((pq^\top) \otimes I_n) \begin{bmatrix} \mathbf{v}(0) \\ \mathbf{x}(0) \end{bmatrix},$$

where $\mathbf{w}(0) = \mathbf{0}_{Nn}$ and $\mathbf{x}(0)$ is the arbitrary starting point. Thus, $\mathbf{v}(0) = -\mathbf{w}(0) + \mathbf{x}(0) - \mathbf{z} = \mathbf{x}(0) - \mathbf{z}$, and

$$\begin{aligned} \begin{bmatrix} \mathbf{v}^* \\ \mathbf{x}^* \end{bmatrix} &= \frac{1}{N} \left(\begin{bmatrix} \mathbf{0}_{N \times N} & \mathbf{0}_{N \times N} \\ -\mathbf{1}_N \mathbf{1}_N^\top & \mathbf{1}_N \mathbf{1}_N^\top \end{bmatrix} \otimes I_n \right) \begin{bmatrix} \mathbf{x}(0) - \mathbf{z} \\ \mathbf{x}(0) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{0}_{Nn} \\ \frac{1}{N} (\mathbf{1}_N \mathbf{1}_N^\top \otimes I_n) \mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{Nn} \\ \mathbf{1}_N \otimes \bar{\mathbf{z}} \end{bmatrix}. \end{aligned}$$

Therefore $\mathbf{x}^* = \mathbf{1}_N \otimes \bar{\mathbf{z}}$ which completes the proof. \square

Proof of Lemma 1. Let $\begin{bmatrix} \mathbf{v} \\ \mathbf{x} \end{bmatrix} \in \mathcal{N}(A)$ be non-zero. Then

$$-\mathbf{L}\mathbf{v} - \mathbf{v} - \mathbf{L}\mathbf{x} = 0, \quad (15a)$$

$$-\mathbf{v} - \mathbf{L}\mathbf{x} = 0. \quad (15b)$$

By combining the two, we get $\mathbf{L}\mathbf{v} = 0$ implying that $\mathbf{v} \in \text{span}(\mathbf{1}_N)$. It follows from (15b) that $\mathbf{L}\mathbf{x} = -\mathbf{v} = \alpha \mathbf{1}_N$. Then $\mathbf{1}^\top \mathbf{L}\mathbf{x} = 0 = \alpha \mathbf{1}_N^\top \mathbf{1}_N$, and thus $\alpha = 0$. But this implies that $\mathbf{v} = 0$. Thus $\mathbf{L}\mathbf{x} = 0$, which implies that $\mathbf{x} \in \text{span}(\mathbf{1}_N)$. Therefore, we can conclude that

$$\mathcal{N}(A) = \left\{ \begin{bmatrix} \mathbf{0}_N \\ \mathbf{x} \end{bmatrix} : \mathbf{x} \in \text{span}(\mathbf{1}_N) \right\}.$$

Since the nullspace has dimension 1, it follows that 0 is a simple eigenvalue of A . Next, let $\begin{bmatrix} \mathbf{v} \\ \mathbf{x} \end{bmatrix}$ be an eigenvector of A associated with a non-zero eigenvalue λ . For the sake of contradiction, suppose $\Re[\lambda] \geq 0$. Then

$$\begin{cases} \lambda \mathbf{v} &= -\mathbf{L}\mathbf{v} - \mathbf{v} - \mathbf{L}\mathbf{x} \\ \lambda \mathbf{x} &= -\mathbf{v} - \mathbf{L}\mathbf{x} \end{cases}$$

Note, if $\mathbf{x} = 0$, then $\mathbf{v} = 0$. So, in order for $\begin{bmatrix} \mathbf{v} \\ \mathbf{x} \end{bmatrix}$ to be an eigenvector, we need $\mathbf{x} \neq 0$. By solving for \mathbf{x} , we obtain

$$-\lambda \mathbf{x} = (\lambda \mathbf{I} + \mathbf{L})^2 \mathbf{x}. \quad (16)$$

So, $(-\lambda, \mathbf{x})$ is an eigenpair of the matrix $(\lambda \mathbf{I} + \mathbf{L})^2$. The eigenvalues of $(\lambda \mathbf{I} + \mathbf{L})^2$ are $\{(\lambda + \rho)^2 : \rho \in \sigma(\mathbf{L})\}$. So, there exists some $\rho \in \sigma(\mathbf{L})$ such that $-\lambda = (\lambda + \rho)^2$. Let $\lambda = a + bi$ and note the hypothesis of contradiction is $a \geq 0$ and $a^2 + b^2 \neq 0$. By separating the real and imaginary parts, we obtain

$$\begin{aligned} -a &= (a + \rho)^2 - b^2, \\ -b &= 2(a + \rho)b. \end{aligned}$$

Suppose $b = 0$, then the first equation and the hypothesis $a \geq 0$ implies that $a = 0$. But, this implies $\lambda = 0$ which is a contradiction. Next, suppose $b \neq 0$, then the second equation implies $-1 = 2a + 2\rho$. Thus $a = \frac{-1-2\rho}{2} < 0$, which is a contradiction with $a \geq 0$ as $\rho \geq 0$. Therefore, if $\lambda \neq 0$ then we must have $a < 0$, and so all non-zero eigenvalues of A have negative real part. This completes the proof. \square

V. SIMULATIONS

In this section and for illustration purposes, we consider the special orthogonal Lie group $\mathcal{G} = SO(3)$, with its parameterization of 3×3 rotation matrices. The associated Lie algebra, denoted $\mathfrak{so}(3)$, is the space of 3×3 skew-symmetric matrices. We equip $SO(3)$ with the following bi-invariant Riemannian metric:

$$\langle u, v \rangle_x := \frac{1}{2} \text{tr}(u^\top v) = \frac{1}{2} \text{tr}(\xi^\top \eta),$$

where $x \in SO(3)$, $u, v \in T_x SO(3)$, and $\xi, \eta \in \mathfrak{so}(3)$ with $\xi = x^{-1}u, \eta = x^{-1}v$. Then, the corresponding geodesic distance reduces to

$$d(x, y)^2 = \|\text{Log}(x^\top y)\|^2 = -\frac{1}{2} \text{tr}[\text{Log}(x^\top y)^2].$$

In the following, two scenarios are in order where we randomly initialize agents on $SO(3)$.

A. Scenario 1 (Comparison of Consensus Algorithms)

In this section, we run simulations comparing our solver to three other consensus algorithms. We randomly initialized $N = 10$ agents $\mathbf{x} \in \mathcal{C}$ and run each algorithm with the same initial conditions in order to compare them. We choose the following two metrics for comparison: the *consensus error* (6), and the *RCM Error* $E_{\text{RCM}}(\mathbf{x}) = \sum_{i=1}^N d(x_i, \bar{\mathbf{z}})^2$ which aggregates the error of each point x_1, \dots, x_N from the RCM $\bar{\mathbf{z}}$. Using these two metrics, we compare our solver against three other algorithms described in the following and illustrate the results in Figure 2. For the technical details of these algorithm and their implementation see the extended version of this work [22] and the code [23].

1) *Algorithm 1:* To implement the continuous dynamics of our algorithm, we consider their forward-Euler discretization on each tangent space with step size $\epsilon = 0.1$ and use the Lie group exponential mapping as our choice of retraction.

2) *Riemannian Consensus Algorithm:* As stated before, [4] was one of the first works to introduce a consensus algorithm for arbitrary Riemannian manifolds, referred to as ‘‘R. Tron et. al’’ in Figure 2.

3) *Penalty method:* The penalty approach is a commonly used solver for constrained optimization problems. We implemented [24, Algorithm 14.3.1], while introducing a distributed implementation for it. In particular, we chose penalty parameter $\mu(s) = \frac{1}{\sqrt{s}}$, number of gradient descent iterations $S = 50$, and step size $\epsilon = 0.1$.

4) *Lagrangian method:* A Lagrangian method is an approach to solving constrained optimization problems that involves finding saddle points of the Lagrangian function. We implemented the solver described in [25, 4.4.1].

It can be seen in Figure 2 that both error metrics for our algorithm are vanishing with linear rate. Also, note in Figure 2 that Riemannian Consensus Algorithm has a seemingly linear rate of decrease for the consensus error, similar to our algorithm but with faster rate. However, we emphasize that this algorithm does not converge to the RCM, and it only synchronizes the agents to a point. This is evident in Figure 2, where the RCM Error for this algorithm stops

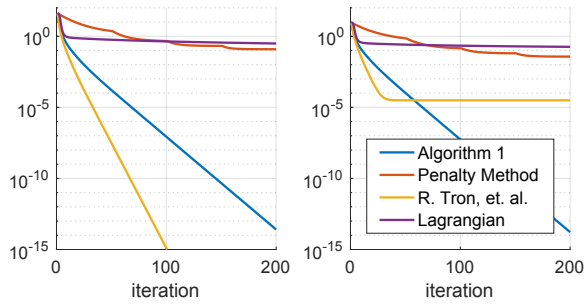


Fig. 2: The consensus error (left) and the RCM Error (right) at each iteration, comparing our algorithm (in blue) with three other algorithms.

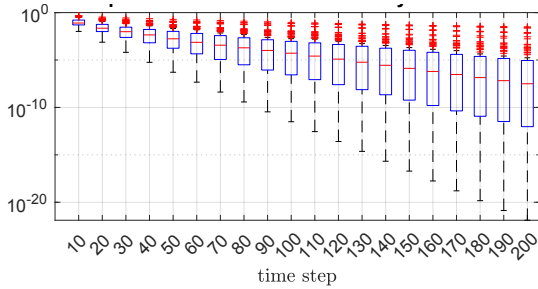


Fig. 3: The RCM error of our proposed algorithm over 100 randomly sampled problem instances.

decreasing. Finally, as illustrated in Figure 2, the Penalty and Lagrangian method have a sub-linear rate of decrease for both the consensus error and the RCM error.

B. Scenario 2 (Linear rate of convergence)

In this scenario, we generate 100 different problem instances with $N = 10$ agents randomly initialized on $\mathcal{C} \subset SO(3)^{10}$. We run the algorithm on each instance for 200 iterations for visualization of the converge rate. We illustrate the result in Figure 3 showing the statistics of the RCM Error at each time step, confirming a linear rate. We note that although convergences analysis of the algorithm has only been provided for the Euclidean case, these numerical experiments—using different problem parameters—demonstrate the effectiveness of the proposed approach for distributed RCM consensus on Lie groups.

VI. CONCLUSIONS

In this work, we reformulate the Riemannian Center of Mass (RCM) as a distributed optimization problem and propose a novel and fast distributed solver on bi-invariant Lie groups. This in turn provides the first (completely) distributed solver for the RCM consensus problem with no subroutines. The key idea behind the proposed algorithm is a correspondence between a consensus protocol on the Lie group and a dynamic consensus protocol on its Lie algebra. We have shown the properties of limit points of our algorithm in the general setting, and guaranteed its convergence in the Euclidean setting; the convergence analysis for the general case is the subject of our ongoing work.

REFERENCES

- [1] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [2] S. H. Strogatz, *Sync: How Order Emerges from Chaos in the Universe, Nature, and Daily Life*. Hachette UK, 2012.
- [3] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 40–72, 2019.
- [4] R. Tron, B. Afsari, and R. Vidal, "Riemannian consensus for manifolds with bounded curvature," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 921–934, 2012.
- [5] R. Olfati-Saber, "Swarms on sphere: A programmable swarm with synchronous behaviors like oscillator networks," in *Proceedings of the 45th IEEE Conference on Decision and Control*, pp. 5060–5066, IEEE, 2006.
- [6] A. Sarlette, S. Bonnabel, and R. Sepulchre, "Coordinated motion design on Lie groups," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1047–1058, 2010.
- [7] R. Sepulchre, "Consensus on nonlinear spaces," *IFAC Proceedings Volumes*, vol. 43, no. 14, pp. 1029–1039, 2010.
- [8] S. Chen, L. Zhao, W. Zhang, and P. Shi, "Consensus on compact Riemannian manifolds," *Inf. Sci.*, vol. 268, pp. 220–230, June 2014.
- [9] S. Chen, P. Shi, W. Zhang, and L. Zhao, "Finite-time consensus on strongly convex balls of Riemannian manifolds with switching directed communication topologies," *J. Math. Anal. Appl.*, vol. 409, pp. 663–675, Jan. 2014.
- [10] S. Kraissler, S. Talebi, and M. Mesbahi, "Distributed consensus on manifolds using the Riemannian center of mass," *IEEE Conference on Control Technology and Applications*, 2023.
- [11] R. Tron and R. Vidal, "Distributed 3-d localization of camera sensor networks from 2-d image measurements," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3325–3340, 2014.
- [12] B. Afsari, "Riemannian L^p center of mass: Existence, uniqueness, and convexity," *Proceedings of the American Mathematical Society*, vol. 139, no. 2, pp. 655–673, 2011.
- [13] R. Tron, R. Vidal, and A. Terzis, "Distributed pose averaging in camera networks via consensus on $SE(3)$," in *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1–10, IEEE, 2008.
- [14] R. Tron, B. Afsari, and R. Vidal, "Average consensus on riemannian manifolds with bounded curvature," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 7855–7862, IEEE, 2011.
- [15] J. M. Lee, *Introduction to Riemannian Manifolds*, vol. 176. Springer, 2nd ed., 2018.
- [16] N. Boumal, *An Introduction to Optimization on Smooth Manifolds*. Cambridge University Press, 2023.
- [17] A. Arvanitogeorgos, *An Introduction to Lie Groups and the Geometry of Homogeneous Spaces*, vol. 22. American Mathematical Soc., 2003.
- [18] Y. L. Sachkov, "Control theory on Lie groups," *Journal of Mathematical Sciences*, vol. 156, no. 3, pp. 381–439, 2009.
- [19] J. Milnor, "Curvatures of left invariant metrics on Lie groups," *Advances in mathematics*, vol. 21, no. 3, pp. 293–329, 1976.
- [20] A. Nedic, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM Journal on Optimization*, vol. 27, no. 4, pp. 2597–2633, 2017.
- [21] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multi-agent Networks*. Princeton University Press, 2010.
- [22] S. Kraissler, S. Talebi, and M. Mesbahi, "Consensus on lie groups for the riemannian center of mass (extended)," *arXiv preprint arXiv:2308.08054*, 2023.
- [23] S. Kraissler, S. Talebi, and M. Mesbahi, "RCM-Consensus-Lie-Group," *GitHub repository*, 2023. Available online at <https://github.com/spencerkraissler/RCM-Consensus-Lie-Group>.
- [24] A. R. Conn, N. I. Gould, and P. L. Toint, *Trust Region Methods*. SIAM, 2000.
- [25] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic press, 2014.