

Riemannian Direct Trajectory Optimization of Rigid Bodies on Matrix Lie Groups

Sangli Teng^{1†}, Tzu-Yuan Lin¹, William A Clark², Ram Vasudevan¹, and Maani Ghaffari¹
¹University of Michigan, ²Ohio University, [†]Corresponding Author: sanglit@umich.edu

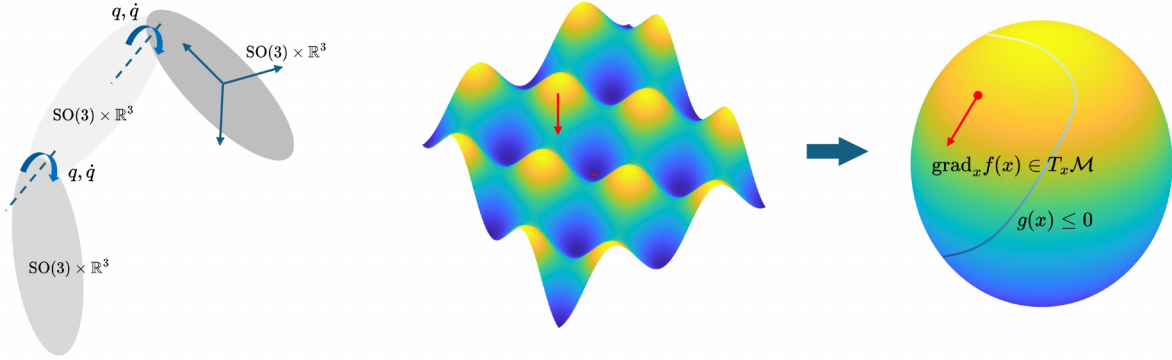


Fig. 1: Constrained Riemannian direct trajectory optimization of rigid bodies formulated on matrix Lie groups. The rigid body dynamics can be formulated in generalized coordinates ($q \in \mathbb{R}^n$) or on maximal coordinates (product space of $\text{SO}(3) \times \mathbb{R}^3$). We compare the landscape of a quadratic cost $\|Rz - z\|^2$ defined on $\mathcal{M} = \text{SO}(3)$ projected on $\mathbb{S}^2 \cong \text{SO}(3)/\text{SO}(2)$ with different coordinates. With generalized coordinates represented by Euler angles, the landscape is highly complicated with saddle points. With the matrix Lie group coordinates, the decision variable is in a symmetric homogeneous space. In this work, we, for the first time, introduce constrained Riemannian optimization to solve motion planning of rigid bodies modeled on matrix Lie groups.

Abstract—Designing dynamically feasible trajectories for rigid bodies is a fundamental problem in robotics. Although direct trajectory optimization is widely applied to solve this problem, inappropriate parameterizations of rigid body dynamics often result in slow convergence and violations of the intrinsic topological structure of the rotation group. This paper introduces a Riemannian optimization framework for direct trajectory optimization of rigid bodies. We first use the Lie Group Variational Integrator to formulate the discrete rigid body dynamics on matrix Lie groups. We then derive the closed-form first- and second-order Riemannian derivatives of the dynamics. Finally, this work applies a line-search Riemannian Interior Point Method (RIPM) to perform trajectory optimization with general nonlinear constraints. As the optimization is performed on matrix Lie groups, it is correct-by-construction to respect the topological structure of the rotation group and be free of singularities. The paper demonstrates that both the derivative evaluations and Newton steps required to solve the RIPM exhibit linear complexity with respect to the planning horizon and system degrees of freedom. Simulation results illustrate that the proposed method is faster than conventional methods by an order of magnitude in challenging robotics tasks.

I. INTRODUCTION

Direct trajectory optimization [1–4] has been extensively applied for motion planning of rigid bodies. Most existing approaches model robot dynamics in generalized coordinates, which evolve on highly complex Riemannian manifolds [5]. This complexity often leads to slow convergence and computationally expensive derivative evaluations during optimization. On the other hand, as the real projective space \mathbb{RP}^3 cannot be embedded differentially into \mathbb{R}^3 [6], there does not exist a globally smooth mapping from \mathbb{R}^3 to $\text{SO}(3)$ (rotation

matrices) or $\text{SU}(2)$ (unit quaternions). Thus, any effort trying to parameterize 3D rotational motions by \mathbb{R}^3 , such as Euler angles, three Degree-Of-Freedom (DOF) parameterization of quaternion [7] or Rodriguez formula [8], breaks the topological structures and introduces singularities.

The Euler-Poincaré equations provide an alternative formulation [9, 10], which describe rigid body dynamics on the matrix Lie group, i.e., a smooth symmetric homogeneous space with superior computational advantages. Despite its potential, a unified framework that fully exploits the Lie group structure for trajectory optimization remains absent. In contrast, *unconstrained* Riemannian optimization [11] has been widely adopted in robot perception, including Simultaneous Localization and Mapping (SLAM) [12], sensor registration [13], and globally optimal pose graph optimization [14, 15]. These perception problems are typically formulated as Maximum A Posteriori estimations that do not involve equality or inequality constraints. However, in motion planning, incorporating constraints such as rigid body dynamics (equality constraints) and collision avoidance (inequality constraints) is necessary to ensure dynamical feasibility.

To mitigate this gap, this work bridges the discrete rigid body dynamics on matrix Lie groups and *constrained* Riemannian optimizations to perform *fast* and *topologically compatible* trajectory optimization. The main contribution is illustrated in Figure 1 and can be summarized as follows:

- 1) Derivation of the discrete multi-rigid body dynamics and a unified formulation for direct trajectory optimization of rigid body systems on matrix Lie groups. The proposed formulation is correct-by-construction to preserve the

- topological structure and conserve the energy.
- 2) Derivation of the exact closed-form Riemannian second-order expansion of the dynamics leveraging Lie group symmetry. We show the complexity to evaluate the first- and second-order derivatives is linear w.r.t. the number of rigid bodies.
 - 3) Development of a line-search Riemannian Interior Point Method to conduct trajectory optimization with general nonlinear constraints.
 - 4) Verifications of the proposed method on challenging robotics tasks, including motion planning of drones and manipulators with full dynamics.
 - 5) Open-source implementation is available at <https://github.com/SangliTeng/RiemannianTrajectoryOptimization>.

The remainder of the paper is organized as follows. The related work is summarized in Section II. The math preliminary is provided in Section III. Then we formulate the direct trajectory optimization in Section IV. The rigid body dynamics and its differentiation are presented in Section VI and V, respectively. A constrained Riemannian optimization is implemented in Section VII and evaluated in Section VIII. Finally, the limitations of the proposed method is discussed in Section IX and the conclusions are summarized in Section X.

II. RELATED WORK

In this section, we review the trajectory optimization of rigid body systems and Riemannian optimizations.

A. Rigid Body Dynamics

The majority of robotics applications model the dynamics in generalized coordinates. By expressing the kinetic energy in terms of joint angles, the robot dynamics evolves on the Riemannian manifold with the metric defined by inertia [16]. Based on this model, the tracking controller [5] and its variants in task space [17–19] can be applied for feedback control of high DOF robots. For optimization-based control, it is more challenging to evaluate the derivatives. The second-order derivative in generalized coordinates has cubic complexity w.r.t. the depth for a kinematic chain, even using the recent state-of-the-art method [20]. Highly complicated tensor computations are also expected in this line of research [21–23].

Other than modeling the dynamics in generalized coordinates, the maximal coordinates formulation represents each rigid body explicitly and enforces the constraints [7, 24]. The graph structure of the kinematic chain [7] preserves the sparsity pattern, thus enabling the computation of gradients to have *linear* complexity w.r.t. the DOF. As each rigid body is explicitly modeled, the manifold structure of the rigid body can be preserved, such as the rigid body simulator [7, 25] that applies the variational integrator [26] in quaternion. However, as [7, 25] considers a three-DOF parameterization of quaternion velocities, the representation has singularities. On the other hand, the Lie Group Variational Integrator (LGVI) [27] in maximal coordinates [28] derives the discrete dynamics of matrix Lie groups that naturally admit a smooth representation that is suitable for Riemannian optimization.

B. Trajectory Optimization

Trajectory optimization aims to synthesize robot motions subject to dynamics, kinematics, input, and environment constraints. The direct trajectory optimization derives the dynamics in discrete time and then conducts optimizations [3, 4, 29–33]. By the state-of-the-art numerical optimization techniques [34, 35], the direct methods can handle large-scale problems with complicated constraints.

For *rigid body* systems, the single rigid body dynamics has also been applied for tracking control or planning of legged robots [36–40]. The simplified models with dynamics of angular rates neglected are applied in trajectory generation of quadrotors [41]. However, none of the above methods can synthesize full dynamic trajectories of the robots. For full rigid body dynamics, [3, 4] incorporated the dynamics in generalized coordinates to synthesize the trajectories. The Differential Dynamic Programming (DDP) is applied in [42] to synthesize an optimal trajectory on the matrix Lie group. The variation integrator-based dynamics has been applied to the Optimal Control Problem (OCP) of satellite [43], while the Lie group version is not fully explored to consider the 3D rotations. The OCP with discrete dynamics on the Lie group is formulated in [44] and can be solved via iterative root finding to meet the first-order optimality condition. The on-manifold Model Predictive Control (MPC) [8] synthesized the trajectory with full dynamics on $SO(3)$ and applied the numerical optimization by lifting the variable to the Lie algebra [27]. As [8, 42, 44] are designed on Lie groups, they do not face the singularity of gimbal locks when using Euler angles. However, the Rodriguez formula [8], a three-DOF parameterization of rotational map, inevitably introduces singularities. Though such singularities can sometimes be avoided by manually rounding the velocity vector [45], an optimization framework that intrinsically handles the topological structures of the rotation group remains absent.

C. Riemannian Optimization

Riemannian optimization has been extensively applied to problems involving variables defined on smooth manifolds [11], such as matrix completion [46] and Semi-Definite Programming (SDP) [47, 48]. The Riemannian derivatives and the retraction map can seamlessly extend traditional first- and second-order numerical optimizations to the manifold setting. Since Riemannian optimization inherently finds search directions on the manifold, it involves far fewer constraints than formulations in the ambient space. For example, the on-manifold Gauss-Newton method has been successfully applied in SLAM [12] on $SE(3)$. Similarly, optimization on the Stiefel manifold has been proposed to solve the relaxed pose graph optimization to obtain globally optimal solutions [14]. Furthermore, Riemannian optimization has demonstrated superior convergence rates in continuous sensor registrations [13].

Beyond perception tasks, OCPs for rigid bodies have been modeled on matrix Lie groups [28, 39, 49–51] and could potentially benefit from Riemannian optimization techniques. However, there is a lack of constrained Riemannian solvers

tailored for motion planning of rigid bodies on matrix Lie groups. To address this gap, the application of *constrained* Riemannian optimization [52–56] holds significant promise to consider general nonlinear constraints on matrix Lie groups.

III. PRELIMINARIES

In this section, we briefly review the basic concepts of Riemannian optimization [11] and matrix Lie groups.

A. Riemannian Geometry

Consider a finite-dimensional smooth manifold \mathcal{M} , we denote the tangent space at $x \in \mathcal{M}$ as $T_x \mathcal{M}$. A vector field is a map $V : \mathcal{M} \rightarrow T\mathcal{M}$ with $V(x) \in T_x \mathcal{M}$, and $T\mathcal{M} := \bigcup_{x \in \mathcal{M}} T_x \mathcal{M}$ is the tangent bundle. We denote all smooth vector field on \mathcal{M} as $\mathfrak{X}(\mathcal{M})$. \mathcal{M} is a Riemannian manifold if an Riemannian metric $\langle \cdot, \cdot \rangle_x : T_x \mathcal{M} \times T_x \mathcal{M} \rightarrow \mathbb{R}$ for each tangent space $x \in \mathcal{M}$, such that the map

$$x \mapsto \langle V(x), U(x) \rangle_x \quad (1)$$

is a smooth function from \mathcal{M} to \mathbb{R} . For a function $f : \mathcal{M}_1 \rightarrow \mathcal{M}_2$, the differential is denoted as $Df(x) : T_x \mathcal{M}_1 \rightarrow T_x \mathcal{M}_2$. Then we define the gradient and the connection:

Definition 1 (Riemannian Gradient). *Let $f : \mathcal{M} \rightarrow \mathbb{R}$ be a smooth function. The Riemannian gradient of f is the vector field $\text{grad } f$ on \mathcal{M} uniquely defined by the following identities:*

$$\forall (x, v) \in T\mathcal{M}, \quad Df(x)[v] = \langle v, \text{grad } f(x) \rangle_x. \quad (2)$$

Definition 2 (Connection). *An affine connection on a manifold \mathcal{M} is a \mathbb{R} -bilinear map $\nabla : \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \rightarrow \mathfrak{X}(\mathcal{M})$ such that*

- 1) $\nabla_f X Y = f \nabla_X Y$, and
- 2) $\nabla_X (fY) = \mathcal{L}_X f \cdot Y + f \nabla_X Y$,

where f is a smooth function, and X and Y are vector fields.

Given a Riemannian metric, there exists a unique affine connection, called the Levi-Civita connection, which preserves the metric and is torsion-free; for more details, see e.g. §6 in [57]. In coordinates, a connection is determined by its Christoffel symbols Γ_{ij}^k . Let $x = (x^1, \dots, x^n)$ be local coordinates, then

$$\nabla_{\partial_i} \partial_j = \sum_k \Gamma_{ij}^k \partial_k, \quad \partial_i := \frac{\partial}{\partial x^i}. \quad (3)$$

Definition 3 (Covariant Derivative). *Let \mathcal{M} be a manifold with connection ∇ , and let $c : [a, b] \rightarrow \mathcal{M}$ be a smooth curve. Let X be a vector field along the curve c . The covariant derivative of X along c is*

$$\frac{DX}{dt} := \nabla_{\dot{c}} \tilde{X}, \quad (4)$$

where \tilde{X} is any extension of X to the whole manifold.

The notion of a covariant derivative allows for the definition of a geodesic. A curve, $c(t)$, is a geodesic if the covariant

derivative of \dot{c} along c vanishes, i.e.

$$\ddot{c}(t) := \frac{D\dot{c}}{dt} = 0. \quad (5)$$

In coordinates, the geodesic equation is a system of second-order differential equations involving the Christoffel symbols:

$$\ddot{x}^k + \sum_{ij} \Gamma_{ij}^k \dot{x}^i \dot{x}^j = 0, \quad k = 1, \dots, n. \quad (6)$$

The Hessian on \mathcal{M} is defined as differentiating the gradients w.r.t. the vector field on $T\mathcal{M}$. Given a tangent vector $u \in T_x \mathcal{M}$, and a vector field V , the derivative of V along u at x is denoted as $\nabla_u V$, given the affine connection ∇ determined by the Riemannian metric $\langle \cdot, \cdot \rangle_x$. Now we proceed to define the Riemannian Hessian:

Definition 4 (Riemannian Hessian). *Let \mathcal{M} be a Riemannian manifold with its Riemannian connection ∇ . The Riemannian Hessian of f at $x \in \mathcal{M}$ is the linear map $\text{Hess } f(x) : T_x \mathcal{M} \rightarrow T_x \mathcal{M}$ defined as follows:*

$$\text{Hess } f(x)[u] = \nabla_u \text{grad } f, \forall u \in T_x \mathcal{M}. \quad (7)$$

Given the Hessian and gradients at $x \in \mathcal{M}$, one could obtain the search direction $v \in T_x \mathcal{M}$ in the tangent space for numerical optimization as in Euclidean space. However, moving along the tangent vector v will generally leave the manifold. Thus, we need the retraction map:

Definition 5 (Retraction). *A retraction on a manifold \mathcal{M} is a smooth map*

$$R : T\mathcal{M} \rightarrow \mathcal{M} : (x, v) \mapsto R_x(v), \quad (8)$$

such that each curve $c(t) = R_x(tv)$ satisfies $c(0) = x$ and $\dot{c}(0) = v$.

Given the retraction map, we have the second-order Taylor expansion on curves:

Definition 6 (Second-order Retraction). *Consider $c(t)$ as the retraction curve:*

$$c(t) = R_x(tv), \quad (9)$$

for $x \in \mathcal{M}$ and $v \in T_x \mathcal{M}$. We have the second-order retraction:

$$\begin{aligned} f(R_x(tv)) &= f(x) + t \langle \text{grad } f(x), v \rangle_x + \frac{t^2}{2} \langle \text{Hess } f(x)[v], v \rangle_x \\ &\quad + \frac{t^2}{2} \langle \text{grad } f(x), \ddot{c}(0) \rangle_x + \mathcal{O}(t^3). \end{aligned} \quad (10)$$

Remark 1. *In the case that the retraction $R_x(\cdot)$ is the Riemannian exponential map, the acceleration of $c(t)$, i.e., $\ddot{c}(0) = 0$.*

B. Matrix Lie Group

Let \mathcal{G} be an n -dimensional matrix Lie group and \mathfrak{g} the associated Lie algebra, i.e., the tangent space of \mathcal{G} at the identity.

For convenience, we define the following isomorphism

$$(\cdot)^\wedge : \mathbb{R}^n \rightarrow \mathfrak{g}, \quad (\cdot)^\vee : \mathfrak{g} \rightarrow \mathbb{R}^n. \quad (11)$$

that maps between the vector space \mathbb{R}^n and \mathfrak{g} . Then, $\forall \phi \in \mathbb{R}^n$, we can define the Lie exponential map as

$$\exp(\cdot) : \mathbb{R}^n \rightarrow \mathcal{G}, \quad \exp(\phi) = \exp_m(\phi^\wedge), \quad (12)$$

where $\exp_m(\cdot)$ is the exponential of square matrices. We also define the Lie logarithmic map as the inverse of the Lie exponential map:

$$\log(\cdot) : \mathcal{G} \rightarrow \mathbb{R}^n. \quad (13)$$

For every $X \in \mathcal{G}$, the adjoint action, $\text{Ad}_X : \mathfrak{g} \rightarrow \mathfrak{g}$, is a Lie algebra isomorphism that enables change of frames:

$$\text{Ad}_X(\phi^\wedge) = X\phi^\wedge X^{-1}. \quad (14)$$

Its derivative at the identity gives rise to the adjoint map in Lie Algebra as

$$\text{ad}_\phi(\eta) = [\phi^\wedge, \eta^\wedge], \quad (15)$$

where $\phi^\wedge, \eta^\wedge \in \mathfrak{g}$ and $[\cdot, \cdot]$ is the Lie bracket.

We say a Riemannian metric is left (resp. right) invariant if they are invariant under left (resp. right) group translation:

$$\langle X\phi^\wedge, X\eta^\wedge \rangle_{T_X \mathcal{G}} = \langle \phi^\wedge, \eta^\wedge \rangle_{\mathfrak{g}}, \quad (\text{left-invariant metric})$$

$$\langle \phi^\wedge X, \eta^\wedge X \rangle_{T_X \mathcal{G}} = \langle \phi^\wedge, \eta^\wedge \rangle_{\mathfrak{g}}. \quad (\text{right-invariant metric})$$

A metric is *bi-invariant* if it is both left and right invariant. In general, the Lie exponential and Riemannian exponential are not identical, as bi-invariant metrics may not exist for \mathcal{G} . In particular, for a bi-invariant metric to exist, Ad_g must be an isometry on \mathfrak{g} [58]; this is always possible when the group is either compact or Abelian. In our work, we assume \mathcal{G} is equipped with a bi-invariant metric, which is not restrictive for robotics applications as rigid body motion in \mathbb{R}^n space can be modeled on $\text{SO}(n) \times \mathbb{R}^n$, a product space of a compact and an Abelian group:

Assumption 1. *We assume the group we study, namely \mathcal{G} , admits a bi-invariant metric, and the Riemannian exponential is identical to the Lie exponential in the following derivation.*

IV. PROBLEM FORMULATIONS

Now we formally define the direct trajectory optimization of rigid bodies in 3D spaces. For a rigid body system composed of N_b single rigid bodies modeled on $\text{SO}(3) \times \mathbb{R}^3$, we define the configuration space as the product space:

$$\mathcal{M}_{\text{RB}} := \underbrace{(\text{SO}(3) \times \mathbb{R}^3) \times \cdots \times (\text{SO}(3) \times \mathbb{R}^3)}_{N_b}. \quad (16)$$

We define the time step $\Delta t \in \mathbb{R}$ and the time sequence $\{t_k = k\Delta t \mid k = 0, \dots, N\} \subset \mathbb{R}$. We assume that the set of all feasible control inputs is $u \in \mathcal{U} \subseteq \mathbb{R}^m$, and we have the implicit rigid body dynamics:

$$f_d(x_{k+1}, x_k, u_k) = 0, x \in \mathcal{M}_{\text{RB}}, u \in \mathcal{U}, \quad (17)$$

that relates the discrete configuration states at time t_k and t_{k+1} . Then we have the optimization:

Problem 1 (Direct Trajectory Optimization of Rigid Bodies). *Consider the configuration space \mathcal{M}_{RB} of a N_b rigid body system and the set of feasible control input \mathcal{U} , we synthesize the optimal trajectory of $x \in \mathcal{M}_{\text{RB}}$ via the optimization:*

$$\begin{aligned} \min_{\{x_k\}_{k=0}^N, \{u_k\}_{k=0}^{N-1}} \quad & P(x_N) + \sum_{k=0}^{N-1} L(x_k, u_k) \\ \text{s.t.} \quad & f_d(x_{k+1}, x_k, u_k) = 0, \\ & g(x_{k+1}) \leq 0, \\ & u_k \in \mathcal{U}, x_{k+1} \in \mathcal{M}_{\text{RB}} \\ & k = 0, 1, 2, \dots, N-1, \\ & x_0 = x_{\text{init}}. \end{aligned} \quad (18)$$

with $L(\cdot, \cdot)$ the stage cost, $P(\cdot, \cdot)$ the terminal cost, $g(\cdot)$ the inequality constraints, and x_{init} the initial condition.

As the state $x \in \mathcal{M}_{\text{RB}}$ admits manifold structure on matrix Lie groups, in the following sections, we apply Riemannian optimization to solve Problem 1. The key to this goal is to formulate the discrete dynamics on matrix Lie groups and derive their Riemannian derivative.

V. DISCRETE RIGID BODY DYNAMICS

In this section, we derive the rigid body dynamics based on the variational integrators on \mathcal{M}_{RB} .

A. Variation-based Discretization

Consider a mechanical system with the configuration space \mathcal{M} . We denote the configuration state as $x \in \mathcal{M}$ and the generalized velocity as $\dot{x} \in T_x \mathcal{M}$. Then we have the Lagrangian given the kinetic and potential energy $T(\dot{x}), V(x)$:

$$L(x, \dot{x}) := T(\dot{x}) - V(x). \quad (19)$$

The key idea of a variational integrator is to discretize the Lagrangian (19) to obtain the discrete-time EoM [26]. The discretization scheme ensures that the Lagrangian is conserved in discrete time, thus having superior energy conservation properties over long durations. The discrete Lagrangian $L_d : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}$ could be considered as the approximation of the action integral via:

$$L_d(x_k, x_{k+1}) \approx \int_{t_k}^{t_{k+1}} L(x, \dot{x}) dt. \quad (20)$$

Then the discrete variant of the action integral becomes:

$$S_d = \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}). \quad (21)$$

Finally, we take variation in $T\mathcal{M}$ and group the term corresponding to $\delta x_k \in T_{x_k} \mathcal{M}$ as the discrete version of

integration by parts [26]:

$$\begin{aligned} \delta S_d = & \langle D_1 L_d(x_0, x_1), \delta x_0 \rangle + \langle D_2 L_d(x_{N-1}, x_N), \delta x_N \rangle \\ & + \sum_{k=1}^{N-1} \langle D_2 L_d(x_{k-1}, x_k) + D_1 L_d(x_k, x_{k+1}), \delta x_k \rangle. \end{aligned} \quad (22)$$

where D_i denotes the derivative with respect to the i -th argument and $\langle \cdot, \cdot \rangle$ the canonical inner product. By the least action principle, the stationary point can be determined by letting the derivative of δx_k be zero:

$$D_1 L_d(x_k, x_{k+1}) + D_2 L_d(x_{k-1}, x_k) = 0. \quad (23)$$

For the holonomic constraints specified by the manifold:

$$h(x) = 0 \in \mathbb{R}^m, \quad (24)$$

another action integral can be considered:

$$\begin{aligned} \int_{t_k}^{t_{k+1}} \langle \lambda(t), h(x) \rangle dt \approx & \frac{\Delta t}{2} \langle \lambda(t_k), J(x_k) \delta x_k \rangle \\ & + \frac{\Delta t}{2} \langle \lambda(t_{k+1}), J(x_{k+1}) \delta x_{k+1} \rangle. \end{aligned} \quad (25)$$

with

$$J(x_k) := \frac{\partial h(x_k)}{\partial x_k} \quad (26)$$

the Jacobian of the constraints, $\delta h(x_k) = J(x_k) \delta x_k$, and λ the constrained force. By adding these terms to the unconstrained Lagrangian and taking variation w.r.t x_k , we have the constrained dynamics:

$$D_1 L_d(x_k, x_{k+1}) + D_2 L_d(x_{k-1}, x_k) = J(x_k)^\top \lambda_k \Delta t, \quad (27)$$

$$h(x_{k+1}) = 0. \quad (28)$$

The constraints $h(x_{k+1}) = 0$ are needed to ensure the states in discrete time do not leave the manifold, which is essential in the multi-body case. To consider the external force u we consider the action integral of control input $u(t)$:

$$\int_{t_k}^{t_{k+1}} \langle u(t), \delta x \rangle dt \approx \frac{\Delta t}{2} \langle u(t_k), \delta x_k \rangle + \frac{\Delta t}{2} \langle u(t_{k+1}), \delta x_{k+1} \rangle. \quad (29)$$

B. Dynamics on $\text{SO}(3) \times \mathbb{R}^3$

Now, we derive the EoM on $\text{SO}(3) \times \mathbb{R}^3$. Consider the discrete equation of motion:

$$\begin{aligned} R_{k+1} &= R_k F_k \in \text{SO}(3), \\ p_{k+1} &= p_k + v_k \Delta t, \end{aligned} \quad (30)$$

with R_k the orientation, F_k the discrete pose change, p_k the position, and v_k the linear velocity. The mid-point approximation can be applied:

$$F_k := R_k^{-1} R_{k+1} \approx I + \Delta t \omega_k^\times, \quad \omega_k^\times \approx \frac{F_k - I}{\Delta t}, \quad (31)$$

$$\dot{p}_k = v_k \approx \frac{p_{k+1} - p_k}{\Delta t}. \quad (32)$$

The kinetic and potential energy can be approximated by:

$$\begin{aligned} T_d &:= \frac{1}{2\Delta t} \text{tr}((F_k - I)I^b(F_k - I)^\top) + \frac{1}{2\Delta t} m \|p_{k+1} - p_k\|^2, \\ V_d &:= m \left(\frac{p_{k+1} + p_k}{2} \right)^\top g \Delta t, \end{aligned} \quad (33)$$

where I^b the nonstandard moment of inertia [59] that relate the standard moment of inertia I_b by $I_b = \text{tr}(I^b)I_3 - I^b$. Via the taking variation on $\text{SO}(3) \times \mathbb{R}^3$, we have the dynamics:

$$F_{k+1}I^b - I^b F_{k+1}^\top = I^b F_k - F_k^\top I^b, \quad (34)$$

$$mv_{k+1} = mv_k + mg\Delta t. \quad (35)$$

With the constraints formulated on $\text{SO}(3) \times \mathbb{R}^3$, we can obtain the dynamics for multi-body systems. Compared to an explicit integration scheme, the LGVI naturally obeys the manifold constraints and conserves the energy [26, 27]. As the LGVI is completely in matrix form, there is no need to move back and forth between the Lie group and its Lie algebra for integration. The reader can refer to [28] for a detailed comparison of integrators on Lie groups for motion planning,

VI. DIFFERENTIATE THE RIGID BODY DYNAMICS

In this section, we derive the first- and second-order Riemannian derivatives of kinematics and dynamics of rigid body motion. Under Assumption 1 and remembering Remark 1, the BCH formula is sufficient to obtain the Riemannian gradients and Hessians via the retraction in Definition 6.

A. Differentiate the Kinematic Chain

Without loss of generality, we consider the kinematic chain constraints on \mathcal{G} with finite length n :

$$X_1 X_2 \cdots X_{n-1} X_n = I \in \mathcal{G}. \quad (36)$$

Now we leverage the BCH formula to derive the second-order retraction at the operating point $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_N$. We denote $\bar{Y} := \bar{X}_1 \bar{X}_2 \cdots \bar{X}_N$. To avoid operating points other than $I \in \mathcal{G}$, we reformulated constraints (36) as:

$$\bar{Y}^{-1} X_1 X_2 \cdots X_{n-1} X_n = \bar{Y}^{-1}, \quad (37)$$

and vectorize it via the logarithmic map:

$$\log(\bar{Y}^{-1} X_1 X_2 \cdots X_{n-1} X_n) = \log(\bar{Y}^{-1}). \quad (38)$$

Consider the tangent vector $\bar{X}_i \xi_i^\wedge \in T_{\bar{X}_i} \mathcal{G}$, the retraction by the Riemannian exponential under Assumption 1 is:

$$c_i(t) = \bar{X}_i \exp(t\xi_i). \quad (39)$$

With $X_{i,j} = X_i X_{i+1} \cdots X_j$, $i \leq j$ and $X_{i+1,i} = I$, the second-order retraction of (38) can be derived as:

$$\begin{aligned} & \log(\bar{Y}^{-1} X_1 \exp(t\xi_1) \cdots X_{n-1} \exp(t\xi_{n-1}) X_n \exp(t\xi_n)) = \\ & t \sum_{i=1}^n \text{Ad}_{X_{i+1,n}^{-1}} \xi_i + \frac{t^2}{2} \sum_{i < j} [\text{Ad}_{X_{i+1,n}^{-1}} \xi_i, \text{Ad}_{X_{j+1,n}^{-1}} \xi_j] + \mathcal{O}(t^3) \end{aligned} \quad (40)$$

TABLE I: Summary of equations of motions for discrete rigid body dynamics and conventional constraints.

Constraints		Second-order Expansions
Rotational	$\log Y := \log(R_{k+1}^{-1} R_k F_k) = 0 \in \mathfrak{so}(3)$	$-Y^{-1} \xi_{k+1}^R + F_k^{-1} \xi_k^R + \xi_k^F +$
Kinematics		$0.5[-Y^{-1} \xi_{k+1}^R, F_k^{-1} \xi_k^R] + 0.5[-Y^{-1} \xi_{k+1}^R, \xi_k^F]$ $+ 0.5[F_k^{-1} \xi_k^R, \xi_k^F]$
Translational	$p_{k+1} - p_k - v_k \Delta t = 0 \in \mathbb{R}^3$	$\xi_{k+1}^p - \xi_k^p - \xi_k^p \Delta t$
Kinematics		
Rotational	$(F_{k+1} I^b - I^b F_{k+1}^\top - (I^b F_k - F_k^\top I^b))^\vee = 0 \in \mathfrak{so}(3)$	$I_b F_{k+1} \xi_{k+1}^F - I_b F_k \xi_k^F$
Dynamics		$+ 0.5(F_{k+1} \xi_{k+1}^{F \wedge 2} I^b - I^b \xi_{k+1}^{F \wedge 2} F_{k+1}^\top)^\vee$ $+ 0.5(I^b F_k \xi_k^{F \wedge 2} - \xi_k^{F \wedge 2} F_k^\top I^b)^\vee$
Translational	$mv_{k+1} - mv_k - mg \Delta t = 0 \in \mathbb{R}^3$	$m \xi_{k+1}^v - m \xi_k^v$
Dynamics		
Pivot	$R_1 r_1 + p_1 - (R_2 r_2 + p_2) = 0 \in \mathbb{R}^3$	$R_1(\xi_1^{R \wedge} + 0.5 \xi_1^{R \wedge 2}) r_1 + \xi_1^p -$
Constraints		$R_2(\xi_2^{R \wedge} + 0.5 \xi_2^{R \wedge 2}) r_2 + \xi_2^p$
Axis	$(R_1 e_i)^\top (R_2 e_z) = 0, i = x, y$	$-(R_1 e_i)^\top (R_2 e_z) \xi_2^{R \wedge} - (R_2 e_z)^\top (R_1 e_i) \xi_1^{R \wedge}$
Constraints		$+ 0.5(R_1 e_i)^\top (R_2 \xi_2^{R \wedge} e_z)$ $+ 0.5(R_2 e_z)^\top (R_1 \xi_1^{R \wedge} e_i)$

The Riemannian gradient and Hessian can be obtained by evaluating the first- and second-order terms of t . For clarity, the detailed derivation is deferred to Appendix A.

B. Differentiate the Dynamics and Constraints

Consider the rotational dynamics in (34), we again use the the retraction curve along the direction $\xi_i^F \in \mathfrak{so}(3)$:

$$c(t) = \bar{F}_i \exp(t \xi_i^F) \in SO(3). \quad (41)$$

As the (34) is already a skew matrix, the vectorized perturbed dynamics can be obtained by substituting the retraction curve and applying the $(\cdot)^\vee$ map:

$$(F_{k+1} \exp(t \xi_{k+1}^F) I^b - I^b \exp(-t \xi_{k+1}^F) F_{k+1}^\top)^\vee = (I^b F_k \exp(t \xi_k^F) - \exp(-t \xi_k^F) F_k^\top I^b)^\vee. \quad (42)$$

Via the Taylor expansion of the exponential map:

$$\exp(t \xi) \approx I + \frac{t \xi^\wedge}{1!} + \frac{t^2 \xi^{\wedge 2}}{2!} + \mathcal{O}(t^3), \quad (43)$$

the second-order expansion can be obtained by substituting the series and keeping the first and second-order terms of t .

Then we apply the second-order retraction again to differentiate the holonomic constraints. Here, we consider two common constraints: pivot constraints and axis constraints. Given two rigid bodies connected by a pivot joint, we have:

$$R_1 r_1 + p_1 = R_2 r_2 + p_2, \quad (\text{Pivot Constraints})$$

where R_i is the orientation of the rigid body in the world frame, p_i is the position of the rigid body, and r_i is the position vector pointing from the center of mass (CoM) of each rigid body to the joint, represented in each rigid body frame. The associated second-order retraction can be computed from:

$$R_1 \exp(t \xi_1^R) r_1 + (p_1 + t \xi_2^p) = R_2 \exp(t \xi_2^R) r_2 + (p_2 + t \xi_2^p). \quad (44)$$

To constrain the rotation to one axis, e.g., e_z , we have:

$$(R_1 e_i)^\top (R_2 e_z) = 0, i = x, y, \quad (\text{Axis Constraints})$$

where e_i is the unit vector of each axis. Then, the second-order retraction can be computed from:

$$(R_1 \exp(t \xi_1^R) e_i)^\top (R_2 \exp(t \xi_2^R) e_z) = 0, i = x, y, \quad (45)$$

Finally, we summarize all the constraints and the corresponding second-order retraction in Table I.

C. Complexity of Differentiation

As the maximal coordinate formulation admits a graph structure [7], we conclude that the complexities of evaluating the first- and second-order derivatives are linear w.r.t. the number of rigid bodies and kinematic pairs. The derivative for each rigid body can be evaluated by the first four rows of Table I. As the kinematic pair only involves the configuration state of two rigid bodies, the Hessian only involves the variables involved in the two rigid bodies. An example of the

Jacobian is illustrated in Figure 2.

VII. CONSTRAINED RIEMANNIAN OPTIMIZATION

In this section, we apply the Riemannian Interior Point (RIPM) to conduct trajectory optimization. Though an RIPM is implemented in [56], we provide a customized implementation without dependencies on Manopt [60]. We mainly refer to IPOPT [34] for the line-search implementation.

A. Riemannian KKT Condition

We introduce the optimality condition of constrained optimization defined on smooth manifolds. By the language of Riemannian geometry, the condition coincides with its counterparts in Euclidean space [61]. Consider an optimization problem with general nonlinear constraints:

Problem 2 (Constrained Riemannian Optimization).

$$\begin{aligned} \min_{x \in \mathcal{M}} \quad & f(x) \\ & h_i(x) = 0, i = 1, 2, \dots, l, \\ & g_j(x) \leq 0, j = 1, 2, \dots, m. \end{aligned} \quad (46)$$

with the variable x defined on smooth manifold \mathcal{M} , $\{h_i(x)\}_{i=1}^l$ the equality constraints and $\{g_i(x)\}_{i=1}^m$ the inequality constraints.

With the multiplier $y \in \mathbb{R}^l$ and $z \in \mathbb{R}^m$, we further have the Lagrangian defined on the product manifold $\mathcal{M} \times \mathbb{R}^l \times \mathbb{R}^m$:

$$\mathcal{L}(x, y, z) := f(x) + \sum_{i=1}^l y_i h_i(x) + \sum_{i=1}^m z_i g_i(x). \quad (47)$$

The Riemannian gradients can then be obtained:

$$\begin{aligned} \text{grad}_x \mathcal{L}(x, y, z) = & \text{grad}_x f(x) + \sum_{i=1}^l y_i \text{grad}_x h_i(x) \\ & + \sum_{i=1}^m z_i \text{grad}_x g_i(x) \end{aligned} \quad (48)$$

Then we have the following optimality conditions:

Definition 7 (First-Order Optimality Conditions). $\hat{x} \in \mathcal{M}$ is said to satisfy the KKT conditions if exist the multipliers \hat{y} and \hat{z} , such that:

- 1) Stationary condition: $\text{grad}_x \mathcal{L}(\hat{x}, \hat{y}, \hat{z}) = 0$,
- 2) Primal feasibility: $h_i(\hat{x}) = 0, g_j(\hat{x}) \geq 0, \forall i, j$,
- 3) Dual feasibility: $\hat{z}_j \geq 0, \forall j$,
- 4) Complementarity condition: $\hat{z}_j g_j(\hat{x}) = 0, \forall j$.

We then proceed to introduce the RIPM to search for the critical point that satisfies the KKT condition.

B. Riemannian Interior Point Method

In our application, we apply a line-search RIPM to find the KKT pair $(\hat{x}, \hat{y}, \hat{z}) \in \mathcal{M} \times \mathbb{R}^l \times \mathbb{R}^m$ that satisfies the optimality condition in Definition 7. We consider the log-barrier problem with homotopy parameter μ and slack variables s :

Problem 3 (Log-barrier Problem).

$$\begin{aligned} \min_{x \in \mathcal{M}} \quad & \varphi_\mu := f(x) - \mu \sum_{i=1}^m \log s_i \\ & h_i(x) = 0, i = 1, 2, \dots, l, \\ & g_j(x) + s_j = 0, j = 1, 2, \dots, m, \\ & s \geq 0. \end{aligned} \quad (49)$$

The convergence to Problem 2 (φ_0) can be guaranteed if $\mu \rightarrow 0$ [62]. The KKT condition of the barrier function can be written as:

$$\begin{aligned} \text{grad}_x f(x) + A_E(x)y + A_I(x)z &= 0 \\ h(x) &= 0 \\ g(x) + s &= 0 \\ Sz - \mu e &= 0 \end{aligned} \quad (50)$$

with the Jacobian evaluated at $x \in \mathcal{M}$ defined as:

$$A_E(x) : \mathbb{R}^l \rightarrow T_x \mathcal{M} : A_E(x)y := \sum_{i=1}^l y_i \text{grad}_x h_i(x), \quad (51)$$

$$A_I(x) : \mathbb{R}^m \rightarrow T_x \mathcal{M} : A_I(x)z := \sum_{i=1}^m z_i \text{grad}_x g_i(x). \quad (52)$$

Then we differentiate the KKT vector field:

$$\begin{bmatrix} \text{Hess}_x \mathcal{L}(x, y, z) & A_E^*(x) & A_I^*(x) & 0 \\ A_E(x) & 0 & 0 & 0 \\ A_I(x) & 0 & 0 & I \\ 0 & 0 & S & Z \end{bmatrix} \begin{bmatrix} d_x \\ d_y \\ d_z \\ d_s \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + A_E^* y + A_I^* z \\ h(x) \\ g(x) + s \\ Sz - \mu e \end{bmatrix} \quad (53)$$

with $(d_x, d_y, d_z, d_s) \in T_x \mathcal{M} \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}^m$ the search direction, $Z = \text{diag}(z)$, $S = \text{diag}(s)$ and A^* the adjoint of A . The Hessian of the Lagrangian can be obtained by:

$$\begin{aligned} \text{Hess}_x \mathcal{L}(x, y, z) = & \text{Hess}_x f(x) + \sum_{i=1}^l y_i \text{Hess}_x h_i(x) \\ & + \sum_{i=1}^m z_i \text{Hess}_x g_i(x). \end{aligned} \quad (54)$$

After obtaining the search direction via Riemannian Newton's method according to (53), we implement a backtracking line-search method to decide the step size [34]. The procedure is summarized in Algorithm 1. The logic of Algorithm 1 is to accept the Newton step only if the improvement in feasibility or the reduction in cost is sufficient. The condition for adjusting the homotopy parameter for the log-barrier problem is determined by the violation of the optimality condition E_μ :

$$E_\mu = \max \{ \epsilon_{\text{KKT}}, \epsilon_E, \epsilon_I, \}, \quad (55)$$

with the violation of the KKT condition, equality constraints

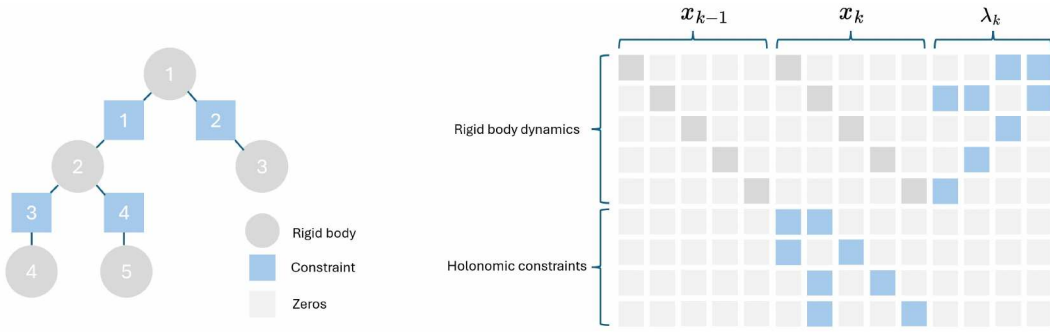


Fig. 2: Example of a rigid body system in maximal coordinates and its Jacobian. The blue block indicates the Jacobians with nonzero elements. The lighter gray box indicates zero block, while the darker gray box indicates the Jacobians of the autonomous single rigid body dynamics.

Algorithm 1 Line-Search Riemannian Interior Point Method

Require: Initial guess $(x_1, y_1, z_1, s_1) \in \mathcal{M} \times \mathbb{R}^l \times \mathbb{R}^m \times \mathbb{R}^m$, homotopy parameter μ , parameters in Table IV

for $k = 1, \dots, N_{\max}$ **do**

 // Decide the search direction

$(d_k^x, d_k^y, d_k^z, d_k^s) \leftarrow$ solving (53) evaluated at (x_k, y_k, z_k, s_k)

 // Check the termination condition

if $E_0(d_k^x, d_k^y, d_k^z, d_k^s) \leq \epsilon_{\text{tol}}$ **then**

break

end if

 // Update the homotopy parameter

if $E_\mu(d_k^x, d_k^y, d_k^z, d_k^s) \leq 10\mu$ **then**

$\mu \leftarrow \max\{\frac{\epsilon_{\text{tol}}}{10}, \min\{\kappa_\mu \mu, \mu^{\theta_\mu}\}\}$

end if

 // Step size by fraction to boundary rule

$\tau \leftarrow \max\{\tau_{\min}, 1 - \mu\}$

$\alpha_k^z \leftarrow \max\{\alpha \in (0, 1) \mid z + \alpha d_k^z \geq (1 - \tau_j)z\}$

$\alpha_k^s \leftarrow \max\{\alpha \in (0, 1) \mid s + \alpha d_k^s \geq (1 - \tau_j)s\}$

$d_k^z \leftarrow \alpha_k^z d_k^z, d_k^s \leftarrow \alpha_k^s d_k^s$

 // Backtracking line-search

$\varphi_{\mu,k}^* \leftarrow \varphi_\mu(x_k)$

$\theta_k^* \leftarrow \|h(x_k)\|_1 + \sum_{i, g_i(x_k) \geq 0} g_i(x_k)$

for $j = 1, \dots, J_{\max}$ **do**

$\tilde{x} \leftarrow R_{x_k}(d_k^x)$ // Apply the retraction map

 // Log barrier loss and constraints violation

$\tilde{\varphi}_\mu \leftarrow \varphi_\mu(\tilde{x}), \tilde{\theta} \leftarrow \|h(\tilde{x})\|, c \leftarrow \langle \text{grad}_x f(\tilde{x}), d_k^x \rangle$

if $\theta_k^* \leq \theta_{\min}$ **and** $c < 0$ **and** $\alpha_{k,j} c^{s_\varphi} > \delta \theta^{*s_\theta}$ **then**

if $\varphi_\mu(\tilde{x}) \leq \varphi_\mu(x_k) + \eta_\varphi c$ **then**

break // Armijo condition satisfied

end if

else if $\tilde{\varphi}_\mu \leq \varphi_{\mu,k}^* - \gamma_\theta \theta_k^*$ **or** $\tilde{\theta} \leq (1 - \gamma_\theta) \theta_k^*$ **then**

break // Feasibility or cost is improved

end if

$d_k^x \leftarrow \beta d_k^x$ // Update step size

end for

 // Update the variables

$x_{k+1} \leftarrow \tilde{x}, y_{k+1} \leftarrow y_k + d_k^y, z_{k+1} \leftarrow z_k + d_k^z, s_{k+1} \leftarrow s_k + d_k^s$

end for

return (x_k, y_k, z_k, s_k)

and inequality constraints being

$$\epsilon_{\text{KKT}} = \frac{\|\text{grad}_x f(x) + A_E y + A_I z\|_\infty}{s_d}, \quad (56)$$

$$\epsilon_E = \|h(x)\|_\infty, \quad \epsilon_I = \frac{\|S z - \mu e\|_\infty}{s_c},$$

respectively. The s_d and s_c are normalizing parameters, and the termination condition for Problem 2 is E_0 . The parameters are shown in Table IV in the Appendix. B.

VIII. NUMERICAL EXPERIMENTS

In this section, we provide a comprehensive evaluation of the proposed method.

A. System setup

1) *Cost function design:* We consider the quadratic cost function on $\text{SO}(3) \times \mathbb{R}^3$ to indicate the difference between the desired and the current configurations. For orientation and the discrete angular velocity, we consider the square of the weighted chordal distance [63]:

$$L_{\text{SO}(3)}(R, R_d) = \frac{1}{2} \text{tr}((RR_d^\top - I)W_R(RR_d^\top - I)^\top), \quad (57)$$

where W_R is a positive-definite weighting matrix. The derivative of the cost function can be obtained by considering the retraction curve:

$$L_{\text{SO}(3)}(R \exp(t\xi), R_d), \xi \in \mathfrak{so}(3). \quad (58)$$

For distance, we consider the quadratic cost function in \mathbb{R}^3 :

$$L_{\mathbb{R}^3}(p, p_d) = \frac{1}{2} (p - p_d)^\top W_p (p - p_d). \quad (59)$$

where W_p is a positive-definite weighting matrix.

2) *Software setup:* The main loop of Algorithm 1 is implemented in MATLAB with the internal linear system solvers for the Newton steps. We use CasADi[64] to evaluate the second-order retraction curve in Table I to obtain the first- and second-order derivatives. To improve the performance, we use code generation to obtain the C++ file and compile it in MATLAB as MEX files for execution. We consider CasADi as our interface for implementing direct trajectory optimization for the baselines using IPOPT and YALMIP [65] when using SNOPT [35]. Due to the difference between the evaluation of the derivatives, we compare the time spent in the solver and the number of iterations as the metric.

3) *Hardware setup:* We use a laptop with an Intel i7-11850H CPU to run all the experiments.

B. Single Rigid Body

In the first experiment, we demonstrate the proposed method on a single rigid body. We consider a quadrotor modeled as a full rigid body with input limits. The input is assumed to be a 4-DOF vector composed of the total linear force along the

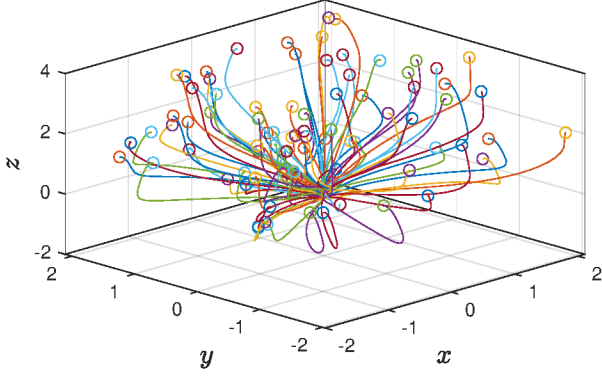


Fig. 3: Testing cases for drone docking from randomly generated initial poses in $\text{SO}(3) \times \mathbb{R}^3$.

z -axis in the body frame and a 3-DOF torque. As a result, the dynamics part can be expressed as:

$$\begin{aligned} F_{k+1}I^b - I^b F_{k+1}^\top &= I^b F_k - F_k^\top I^b + \tau^\wedge \Delta t^2, \\ mv_{k+1} &= mv_k + mg\Delta t + R_{k+1}e_z u_z \Delta t. \end{aligned} \quad (60)$$

We assume that the torque admit a symmetric box limits $\tau \in [-\tau_{\text{lim}}, \tau_{\text{lim}}]^3$ and the linear force is within a non-negative interval $u_z \in [0, u_{z,\text{lim}}]$.

As the minimal dimension of a singularity-free representation of $\text{SO}(3)$ is 4, we do not consider the 3-DOF parameterization, such as Euler angle, Rodriguez formula, or the modified quaternion representation [7]. We compare the proposed formulation with a singularity-free quaternion-based variational integrator. As the quaternion parameterizes the rotation with a four-dimensional vector $q := [q_v, q_x, q_y, q_z] \in \mathbb{R}^4$, it does not have singularities. However, the constraint $\|q\|^2 = 1$ is required to ensure $q \in \mathbb{S}^3$, and $q_v \geq 0$ is necessary to ensure the state lies in one branch of the manifold. Consider the integrator [7]¹ with the singularity-free representation of quaternions, we directly apply the CasADi and its default IPOPT solver to conduct trajectory optimization with the gradient derived in the ambient space.

We consider the task of docking the drone from randomly sampled initial poses to the origin with an identity pose and zero twists as shown in Figure 3. The location of the drone is uniformly sampled from a box. The orientation is also randomly sampled such that $\log(R_0)$ is uniform in the interval $[0, \pi]$ and all directions. We apply an initialization strategy that uniformly interpolates the geodesic curves between the initial and the goal states. The convergence result of the RIPM is presented in Figure 4. As we can see, 82 out of 100 cases converge while 93 out of 100 of the unconstrained cases converge to the tolerance $\epsilon_{\text{tol}} = 1e^{-14}$. The ability to converge super-linearly to such high precision demonstrates the power of the proposed second-order method. For the constrained cases, more than 50% cases converge in 50 iterations if the

¹The original implementation of [7] considers $q_v = \sqrt{1 - q_x^2 - q_y^2 - q_z^2}$ to avoid manifold constraints and reduce dimensions, which, however, introduces a singularity that results in NaN error in numerical optimizations.

tolerance is set to the default value of IPOPT $1e^{-6}$.

We also present the result of the quaternion-based method in Table II. We consider the baseline solver as the IPOPT based on the Interior Point Method and the SNOPT [35] based on Sequential Quadratic Programming (SQP). Due to the polynomial nature of the dynamics and the manifold constraints, most cases do not converge in 1000 iterations with the baselines. As the gradient-based nonlinear programming is sensitive to initializations, we also provide high-quality initialization from the globally optimal motion planning methods [28] based on moment relaxation (dual of sum-of-square optimization)[66, 67]. We apply the open-source code in [28] to generate a certified optimal trajectory and perturb the solutions with different levels of Gaussian noise as initialization. We perturbed the position and velocity of the SDP-based solution by Gaussian noise with covariance $0.1I$. We perturb the orientation and the discrete angular velocity on a quaternion with Gaussian noise with covariances $\Sigma_1 = 0.01I$ and $\Sigma_2 = 0.05I$, project them to \mathbb{S}^3 by rounding the norm to 1. As shown in Table II, the convergence rate of IPOPT and SNOPT soon degrades when the perturbation increases.

Finally, we show that the proposed method is capable of handling general nonlinear constraints by designing trajectories for the drone in cluttered environments as shown in Figure 5. We consider nonconvex norm constraints $(x - x_c)^2 + (y - y_c)^2 \geq r^2$ for the drones to avoid the cylindrical regions. As the proposed RIPM considers general nonlinear constraints, future work can also incorporate more sophisticated representations of collision-free regions, such as [68, 69].

C. Multi-Rigid Body

Complexity analysis: We verify the complexity of the proposed differentiation via a kinematic chain on $\text{SO}(3) \times \mathbb{R}^3$ with a single branch and different depths. We compare the time to evaluate the zeroth- to the second-order differentiation of the dynamics. As is shown in Figure 6, the time scales linearly w.r.t. the depth of the kinematic chain. The sparsity pattern of a kinematic chain and the Hessian of the Lagrangian of the following example is shown in Figure 7. The block diagonal structure indicates that the KKT system scales linearly w.r.t. the planning horizon.

Motion planning of manipulator: In the second example, we apply the proposed method to motion planning of a 7-DOF KUKA iiwa manipulator. We model each link of the manipulator and enforce the revolute joint constraints via considering one pivot constraint and a 2-DOF axis constraint as shown in Table I. We consider the baseline as the direct methods with full dynamics in the generalized coordinates:

$$\begin{aligned} q_{k+1} &= q_k + \dot{q}_k \Delta t, \\ M(q_k) \frac{(\dot{q}_{k+1} - \dot{q}_k)}{\Delta t} + C(q_k, \dot{q}_k) \dot{q}_k + G(q_k) &= \tau_k. \end{aligned} \quad (61)$$

We consider moving the manipulator from the initial pose to the target pose while avoiding obstacles, a cylinder with fixed location and radius. For simplification, we only penalize the distance between the center of mass of each link and the

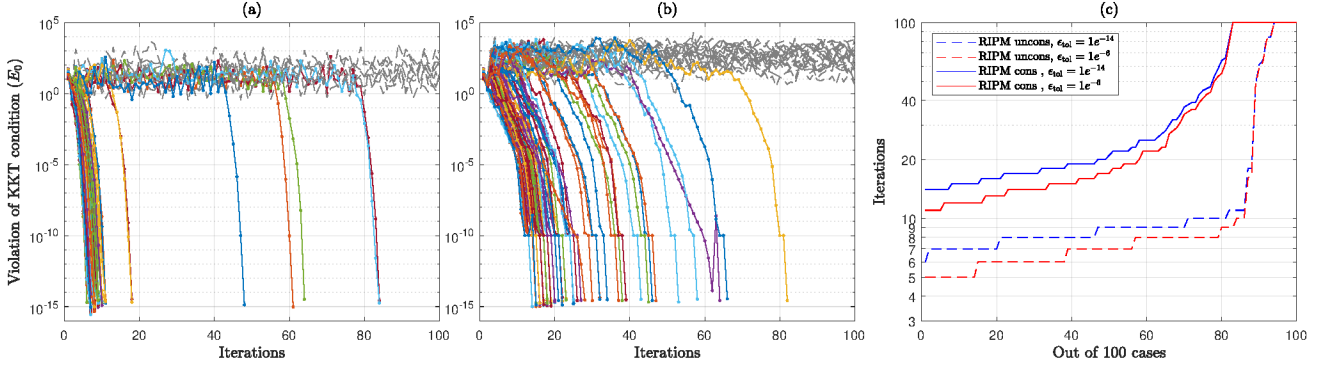


Fig. 4: Convergence of RIMP on drone on $SO(3) \times \mathbb{R}^3$ with full dynamics with 100 different initial conditions. The tolerance for convergence is set to $E_0 = 1e^{-14}$. The convergence rate of unconstrained cases in (a) is much faster than that of constrained cases in (b). The dashed lines indicate the cases that do not converge, while the others are converged cases. Accelerated convergence or super-linear convergence is observed to indicate the success of the second-order method. In (c), we compared the iterations to converge. We can see that it takes fewer iterations for both cases to stop with the default tolerance of IPOPT $\epsilon_{tol} = 1e^{-6}$. The convergence rate and speed outperform the quaternion-based method IPOPT computed in the ambient space.

TABLE II: Comparison of convergence rate of single rigid body case. We summarize the convergent cases for the proposed method and baselines. Otherwise noted, all the cases are without input inequality constraints. Without certified optimal solutions (SDP), the baseline can not converge for either IPOPT with exact Hessian or SNOPT with approximated Hessian. With certified initialization, more than half of the cases using IPOPT can converge, while the performance degrades soon after noises are present. Similar performance is observed for SNOPT, which is even more sensitive to noise. The time in each iteration for the IPM-based method is also listed. The proposed method also takes only about 30% of the time as in IPOPT for the unconstrained cases in each iteration. The planning horizon is 40 steps.

Methods	Initialization Strategy	Convergence in 1000 Iters. (in 100 Iters. for proposed)	Iters. to Convergence		Avg Time per Iteration (sec) w/o function evaluation
			Mdn	Avg	
Proposed	Straight line	82/100 (constrained)	19	24	0.0078
		93/100	9	12	0.0046
IPOPT	Straight line	0	—	—	0.0136
	SDP	68/100	5	73	0.0133
	SDP(Σ_1)	45/100	106	211	0.0113
	SDP(Σ_2)	31/100	317	382	0.0101
SNOPT	Straight line	0	—	—	—
	SDP	56/100	170	177	—
	SDP(Σ_1)	1/100	975	975	—
	SDP(Σ_2)	0	—	—	—

TABLE III: Comparison of proposed formulation using RIMP on \mathcal{M}_{RB} with IPOPT in generalized coordinates. Without infeasibility restoration, the proposed method has a 25% convergence rate. With the infeasibility restoration, IPOPT greatly outperforms the proposed method at the cost of more iterations to solve feasibility sub-problems. In terms of iterations to converge, the proposed method greatly outperforms IPOPT, while the time consumed in each iteration is three times higher, due to the increasing number of variables.

Methods	Convergence in 500 Iterations (w/o infeasibility restoration)	Iters. to Convergence			Time per Iteration (sec) w/o function evaluation
		Mdn	Avg	Max	
RIMP (\mathcal{M}_{RB})	21/100	21	20	42	0.1159
IPOPT ($q \in \mathbb{R}^n$)	64/100	147	201	498	0.0319
	(16/100)	57	76	197	0.0329

obstacle. Thus, the proposed method will have constraints for the i -th link as:

$$(x_i - x_{obs})^2 + (y_i - y_{obs})^2 \geq r_{obs}^2, i = 1, 2, \dots, 7. \quad (62)$$

For generalized coordinates formulation, forward kinematics is required to map the joint angle $q \in \mathbb{R}^n$ to the task space.

We initialize the trajectory via linear interpolation of joint space in both cases. As the manipulator is fully actuated, we can warm start the trajectory with feasible primal solutions by computing the constrained force and the input torque through

inverse dynamics. In all cases, we consider a fixed target pose with a random initial pose and a fixed obstacle and target pose. The convergence rates are shown in Table III. As IPOPT admits advanced features like infeasibility restoration that is not applied in this method, we list the results of IPOPT with and w/o this feature for fair comparison. We find that our method outperforms IPOPT w/o infeasibility restoration in terms of convergence rate and number of iterations to converge. With infeasibility restoration, the performance of IPOPT improves at the cost of more iterations to converge.

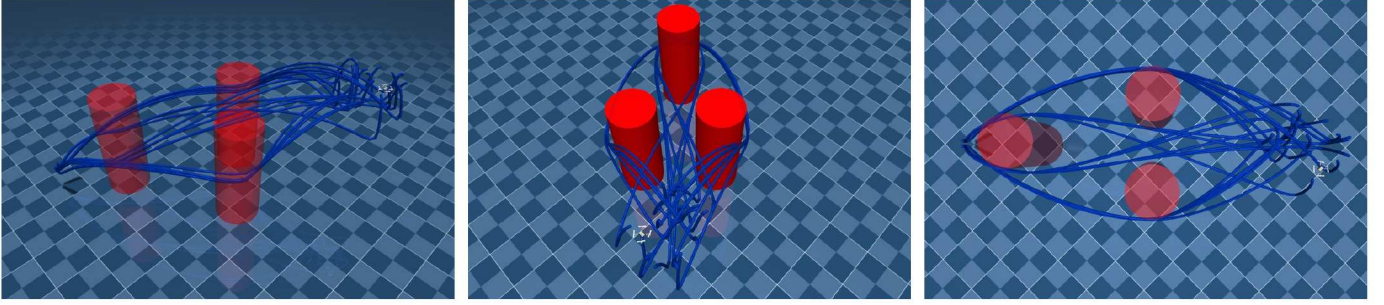


Fig. 5: Trajectories optimization for drone traversing complex environments considering general nonlinear constraints.

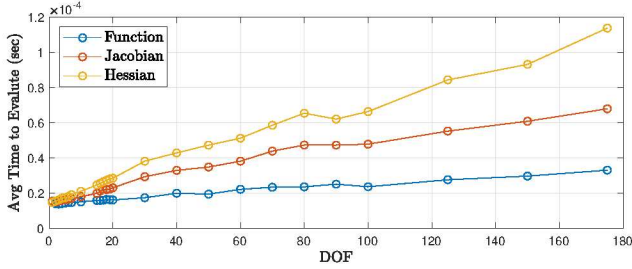


Fig. 6: Average time to evaluate the zeroth-, first-, and second-order derivative of rigid body dynamics w.r.t. the DOF of a kinematic chain with one branch. The proposed formulation scales linearly w.r.t. the DOF.

For both cases, we consider $\epsilon_{\text{tol}} = 1e^{-6}$ as the termination condition.

IX. LIMITATIONS AND DISCUSSIONS

Here, we analyze the limitations of the proposed algorithm.

Increasing number of variables: The proposed method considers all the degrees of freedom of the rigid body, thus making the KKT system multiple times larger than the version in the generalized coordinates. Thus, the trade-off between the number of iterations and computational time in each iteration should be carefully investigated. First-order Riemannian optimization [55] can be considered in future work to bypass the need to solve the expensive Newton steps. The discrete null space mechanics [70–72] can also be applied to eliminate the constrained force, while the null space of the Jacobians of constraints needs to be carefully computed.

Guarantees on global optimality: The proposed method is based on local gradients. As a result, global optimality is not guaranteed. Conducting optimization landscape analysis is also challenging due to the nonconvex nature of motion planning problems. However, we note that $\text{SO}(3) \times \mathbb{R}^3$ is constrained by polynomial quality, thus can be seamlessly integrated with the globally optimal motion planning framework using moment relaxations [28, 73, 74].

Holonomic constraints: The proposed method only considers the holonomic constraints, while it does not incorporate more complicated structures, such as complementarity constraints [3], nonholonomic constraints [10], or the hybrid zero dynamics [75]. As the RIPM is a general framework for constrained optimization, we plan to include these constraints by taking these special structures into consideration [4, 76, 77].

Code optimization: The proposed RIPM is implemented in MATLAB and does not fully replicate all the features of IPOPT [34]. As we do not include the advanced features, such as inertial correction, second-order correction, and infeasibility restoration, the proposed method is not as robust as IPOPT. It is crucial to include these features to improve the performance of the proposed solver further.

X. CONCLUSIONS

In this work, we, for the first time, applied the constrained Riemannian optimization to the paradigm of direct trajectory optimization, which is fast and preserves the topological structure of rotation groups. We first model the multi-body dynamics on matrix Lie groups using the Lie group variational integrator. Then we derive the closed-form Riemannian derivative and apply the line-search Riemannian Interior Point Method to conduct trajectory optimization. We demonstrate that the complexity in the derivative evaluation and Newton steps is linear w.r.t. both the system DOF and the planning horizon. We also show that the proposed method outperforms the conventional methods by an order of magnitude in challenging robotics tasks.

ACKNOWLEDGMENTS

W. Clark, R. Vasudevan, and M. Ghaffari were supported by AFOSR MURI FA9550-23-1-0400. The authors thank the area chairs and anonymous reviewers for the constructive feedback and in-depth review.

APPENDIX A DERIVATION OF SECOND-ORDER RETRACTION OF KINEMATIC CHAIN

Consider the vectorized kinematic chain (38):

$$\log(\bar{Y}^{-1} X_1 X_2 \cdots X_{n-1} X_n) = \log(\bar{Y}^{-1}).$$

Denote $X_{i,j} = X_i X_{i+1} \cdots X_j$ when $i \leq j$ and $X_{i+1,i} = I$, we have the identity using the property of adjoint action:

$$\begin{aligned} \exp(t\xi) \bar{X}_{i,j} &= \bar{X}_{i,j} (\bar{X}_{i,j}^{-1} \exp(t\xi) \bar{X}_{i,j}) \\ &= \bar{X}_{i,j} \exp(t \text{Ad}_{\bar{X}_{i,j}^{-1}} \xi). \end{aligned} \quad (63)$$

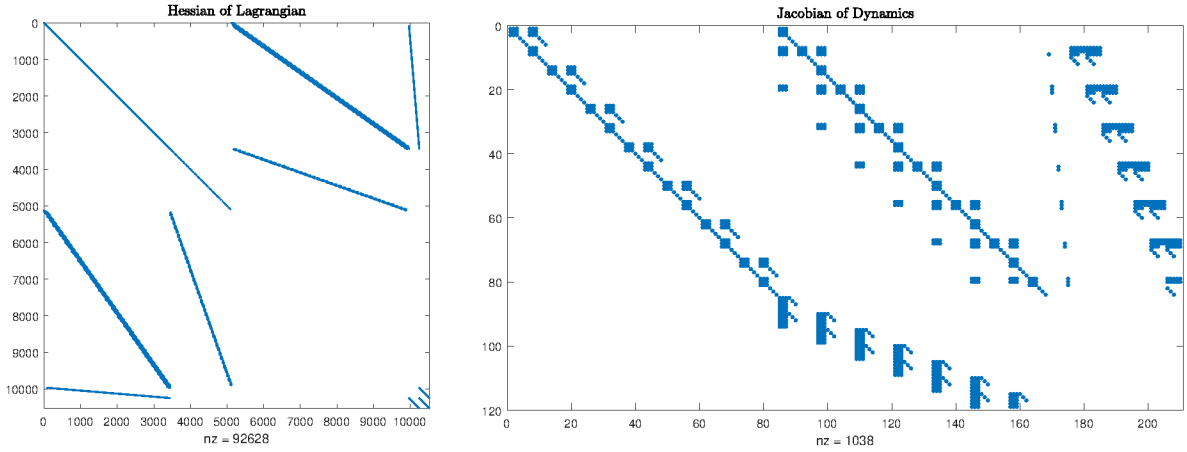


Fig. 7: Illustration of the Hessian of the Lagrangian and Jacobians of the rigid body dynamics of the manipulator example. The block diagonal structure shows that the KKT system scales linearly w.r.t. the planning horizon. The planning horizon is 40 steps for the manipulator task using full dynamics.

The perturbed kinematic chain becomes:

$$\begin{aligned}
& Y^{-1} \bar{X}_1 \exp(t\xi_1) \cdots \bar{X}_{n-1} \exp(t\xi_{n-1}) \bar{X}_n \exp(t\xi_n) \\
& = Y^{-1} \bar{X}_1 \exp(t\xi_1) \cdots \\
& \quad \bar{X}_{n-1} \bar{X}_n \exp(t \operatorname{Ad}_{\bar{X}_{n-1}^{-1}} \xi_{n-1}) \exp(t\xi_n) \\
& = Y^{-1} \bar{X}_1 \exp(t\xi_1) \cdots \bar{X}_{n-2} \\
& \quad \bar{X}_{n-1} \bar{X}_n (\bar{X}_{n-1}^{-1} \bar{X}_{n-2}^{-1} \exp(t\xi_{n-2}) \bar{X}_{n-1} \bar{X}_n) \\
& \quad \exp(t \operatorname{Ad}_{\bar{X}_{n-1}^{-1}} \xi_{n-1}) \exp(t\xi_n) \\
& = Y^{-1} \bar{X}_1 \exp(t\xi_1) \cdots \bar{X}_{n-2} \bar{X}_{n-1} \bar{X}_n \exp(t \operatorname{Ad}_{\bar{X}_{n-1,n}^{-1}}) \\
& \quad \exp(t \operatorname{Ad}_{\bar{X}_{n-1,n}^{-1}} \xi_{n-1}) \exp(t\xi_n) \\
& = Y^{-1} \bar{X}_{1,n} \exp(t \operatorname{Ad}_{\bar{X}_{2,n}^{-1}} \xi_1) \exp(t \operatorname{Ad}_{\bar{X}_{3,n}^{-1}} \xi_2) \cdots \exp(t\xi_n) \\
& = \exp(t \operatorname{Ad}_{\bar{X}_{2,n}^{-1}} \xi_1) \exp(t \operatorname{Ad}_{\bar{X}_{3,n}^{-1}} \xi_2) \cdots \exp(t\xi_n).
\end{aligned} \tag{64}$$

Then we proceed to apply the BCH formula to obtain the second-order retraction:

$$\begin{aligned}
& \log \left(\prod_{i=1}^n \exp \left(t \operatorname{Ad}_{\bar{X}_{i+1,n}^{-1}} \xi_i \right) \right) \\
& = t \operatorname{Ad}_{\bar{X}_{2,n}^{-1}} \xi_1 + \log \left(\prod_{i=2}^n \exp \left(t \operatorname{Ad}_{\bar{X}_{i+1,n}^{-1}} \xi_i \right) \right) \\
& + \frac{1}{2} \left[t \operatorname{Ad}_{\bar{X}_{2,n}^{-1}} \xi_1, \log \left(\prod_{i=2}^n \exp \left(t \operatorname{Ad}_{\bar{X}_{i+1,n}^{-1}} \xi_i \right) \right) \right] + \mathcal{O}(t^3)
\end{aligned} \tag{65}$$

By repeated application of BCH formula to $\log \left(\prod_{i=k}^n \exp \left(t \operatorname{Ad}_{\bar{X}_{i+1,n}^{-1}} \xi_i \right) \right)$, $k \geq 2$, we have the second-order retraction as:

$$\begin{aligned}
& \log \left(\prod_{i=1}^n \exp \left(t \operatorname{Ad}_{\bar{X}_{i+1,n}^{-1}} \xi_i \right) \right) = \\
& t \sum_{i=1}^n \operatorname{Ad}_{\bar{X}_{i+1,n}^{-1}} \xi_i + \sum_{i < j} \frac{t^2}{2} [\operatorname{Ad}_{\bar{X}_{i+1,n}^{-1}} \xi_i, \operatorname{Ad}_{\bar{X}_{j+1,n}^{-1}} \xi_j] + \mathcal{O}(t^3)
\end{aligned} \tag{66}$$

TABLE IV: Part of hyper-parameters in Algorithm 1.

Parameter	Notation	Value
Maximal iteration	N_{\max}	200
Maximal iteration in line search	J_{\max}	30
Tolerance to KKT of Problem 2	ϵ_{tol}	$1e^{-11}$
Linear decaying rate of μ	κ_{μ}	0.99
Superlinear decaying rate of μ	θ_{μ}	1.99
Minimal fraction to boundary	τ_{\min}	0.995
Barrier problem cost progress	γ_{θ}	$1e^{-6}$
Minimal infeasibility	θ_{\min}	$1e^{-4}$
Progress of barrier cost	η_{φ}	$1e^{-4}$
Progress of feasibility	γ_{θ}	$1e^{-4}$
Decaying rate for line search	β	0.5

We note that the derivation does not involve the Jacobian of the matrix logarithmic map due to the shift of the operating point by \bar{Y} .

APPENDIX B

IMPLEMENTATION DETAILS OF RIPM

We consider the following parameters shown in Table IV to implement the RIPM in Algorithm 1.

REFERENCES

- [1] Charles R Hargraves and Stephen W Paris. Direct trajectory optimization using nonlinear programming and collocation. *Journal of guidance, control, and dynamics*, 10(4):338–342, 1987.
- [2] John T Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, 1998.
- [3] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *International Journal of Robotics Research*, 33(1):69–81, 2014.
- [4] Ayonga Hereid and Aaron D Ames. FROST: Fast robot optimization and simulation toolkit. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 719–726. IEEE, 2017.

- [5] Francesco Bullo and Richard M Murray. Tracking for fully actuated mechanical systems: a geometric framework. *Automatica*, 35(1):17–34, 1999.
- [6] John M. Lee. *Smooth Manifolds*, pages 1–29. Springer New York, New York, NY, 2003. ISBN 978-0-387-21752-9. doi: 10.1007/978-0-387-21752-9_1.
- [7] Jan Brüdigan and Zachary Manchester. Linear-time variational integrators in maximal coordinates. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 194–209. Springer, 2021.
- [8] Uroš V Kalabić, Rohit Gupta, Stefano Di Cairano, Anthony M Bloch, and Ilya V Kolmanovsky. MPC on manifolds with an application to the control of spacecraft attitude on $SO(3)$. *Automatica*, 76:293–300, 2017.
- [9] Jerrold E Marsden and Tudor S Ratiu. Introduction to mechanics and symmetry. 1998.
- [10] Anthony M Bloch. Nonholonomic mechanics. In *Nonholonomic mechanics and control*, pages 207–276. Springer, 2003.
- [11] Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.
- [12] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2016.
- [13] William Clark, Maani Ghaffari, and Anthony Bloch. Non-parametric continuous sensor registration. *Journal of Machine Learning Research*, 22(271):1–50, 2021.
- [14] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special euclidean group. *International Journal of Robotics Research*, 38(2-3):95–125, 2019.
- [15] Haoyu Han and Heng Yang. Building rome with convex optimization. *arXiv preprint arXiv:2502.04640*, 2025.
- [16] Francesco Bullo and Andrew D Lewis. *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*, volume 49. Springer, 2019.
- [17] Nathan D Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian motion policies. *arXiv preprint arXiv:1801.02854*, 2018.
- [18] Luis Sentis and Oussama Khatib. Synthesis of whole-body behaviors through hierarchical control of behavioral primitives. *International Journal of Humanoid Robotics*, 2(04):505–518, 2005.
- [19] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.
- [20] Shubham Singh, Ryan P. Russell, and Patrick M. Wensing. On second-order derivatives of rigid-body dynamics: Theory and implementation. *IEEE Transactions on Robotics*, 40:2233–2253, 2024. doi: 10.1109/TRO.2024.3370002.
- [21] Roy Featherstone. *Rigid body dynamics algorithms*. Springer, 2014.
- [22] Roy Featherstone. The calculation of robot dynamics using articulated-body inertias. *International Journal of Robotics Research*, 2(1):13–30, 1983.
- [23] Frank C Park, James E Bobrow, and Scott R Ploen. A Lie group formulation of robot dynamics. *International Journal of Robotics Research*, 14(6):609–618, 1995.
- [24] Sigrid Leyendecker, Jerrold E Marsden, and Michael Ortiz. Variational integrators for constrained dynamical systems. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics*, 88(9):677–708, 2008.
- [25] A. Howell Taylor, Simon Le Cleac’h, Zico Kolter, Mac Schwager, and Zachary Manchester. Dojo: A differentiable simulator for robotics. *arXiv preprint arXiv:2203.00806*, 2022.
- [26] Jerrold E Marsden and Matthew West. Discrete mechanics and variational integrators. *Acta Numerica*, 10:357–514, 2001.
- [27] Taeyoung Lee, N Harris McClamroch, and Melvin Leok. A Lie group variational integrator for the attitude dynamics of a rigid body with applications to the 3D pendulum. In *Proceedings of IEEE Conference on Control Applications*, pages 962–967. IEEE, 2005.
- [28] Sangli Teng, Ashkan Jasour, Ram Vasudevan, and Maani Ghaffari. Convex geometric motion planning of multi-body systems on Lie groups via variational integrators and sparse moment relaxation. *International Journal of Robotics Research*, page 02783649241296160, 2024.
- [29] Matt Zucker, Nathan Ratliff, Anca D Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M Dellin, J Andrew Bagnell, and Siddhartha S Srinivasa. Chomp: Covariant Hamiltonian optimization for motion planning. *International Journal of Robotics Research*, 32(9-10):1164–1193, 2013.
- [30] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *International Journal of Robotics Research*, 33(9):1251–1270, 2014.
- [31] Zachary Manchester, Neel Doshi, Robert J Wood, and Scott Kuindersma. Contact-implicit trajectory optimization using variational integrators. *International Journal of Robotics Research*, 38(12-13):1463–1476, 2019.
- [32] Dayi E Dong, Henry P Berger, and Ian Abraham. Time Optimal Ergodic Search. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.082.
- [33] Zhongyu Li, Jun Zeng, Shuxiao Chen, and Koushil Sreenath. Autonomous navigation of underactuated bipedal robots in height-constrained environments. *International Journal of Robotics Research*, 42(8):565–585, 2023.
- [34] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.
- [35] Philip E Gill, Walter Murray, and Michael A Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
- [36] Donghyun Kim, Jared Di Carlo, Benjamin Katz, Gerardo Bledd, and Sangbae Kim. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv preprint arXiv:1909.06586*, 2019.
- [37] Yanran Ding, Abhishek Pandala, Chuanzheng Li, Young-Ha Shin, and Hae-Won Park. Representation-free model predictive control for dynamic motions in quadrupeds. *IEEE Transactions on Robotics*, 37(4):1154–1171, 2021.
- [38] Ayush Agrawal, Shuxiao Chen, Akshara Rai, and Koushil Sreenath. Vision-aided dynamic quadrupedal locomotion on discrete terrain using motion libraries. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4708–4714. IEEE, 2022.
- [39] Sangli Teng, Dianhao Chen, William Clark, and Maani Ghaffari. An error-state model predictive control on connected matrix Lie groups for legged robot control. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 8850–8857. IEEE, 2022.
- [40] Ruben Grandia, Fabian Jenelten, Shaohui Yang, Farbod Farshidian, and Marco Hutter. Perceptive locomotion through nonlinear model-predictive control. *IEEE Transactions on Robotics*, 39(5):3402–3421, 2023.
- [41] Mark W Mueller, Markus Hehn, and Raffaello D’Andrea. A computationally efficient algorithm for state-to-state quadcopter trajectory generation and feasibility verification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3480–3486. IEEE, 2013.

- [42] George I Boutselis and Evangelos Theodorou. Discrete-time differential dynamic programming on Lie groups: Derivation, convergence analysis, and numerical results. *IEEE Transactions on Automatic Control*, 66(10):4636–4651, 2020.
- [43] Oliver Junge, Jerrold E Marsden, and Sina Ober-Blobaum. Discrete mechanics and optimal control. *IFAC Proceedings Volumes*, 38(1):538–543, 2005.
- [44] Marin B Kobilarov and Jerrold E Marsden. Discrete geometric optimal control on Lie groups. *IEEE Transactions on Robotics*, 27(4):641–655, 2011.
- [45] Mingyang Wang, Qianhao Wang, Ze Wang, Yuman Gao, Jingping Wang, Can Cui, Yuan Li, Ziming Ding, Kaiwei Wang, Chao Xu, et al. Unlocking aerobatic potential of quadcopters: Autonomous freestyle flight generation and execution. *Science Robotics*, 10(101):eadp9905, 2025.
- [46] Bart Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013.
- [47] Jie Wang and Liangbing Hu. Solving low-rank semidefinite programs via manifold optimization. *arXiv preprint arXiv:2303.01722*, 2023.
- [48] Samuel Burer and Renato DC Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical programming*, 95(2):329–357, 2003.
- [49] Sangli Teng, Ashkan Jasour, Ram Vasudevan, and Maani Ghaffari Jadidi. Convex Geometric Motion Planning on Lie Groups via Moment Relaxation. In *Proceedings of the Robotics: Science and Systems Conference*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.058.
- [50] Sangli Teng, William Clark, Anthony Bloch, Ram Vasudevan, and Maani Ghaffari. Lie algebraic cost function design for control on Lie groups. In *Proceedings of the IEEE Conference on Decision and Control*, pages 1867–1874. IEEE, 2022.
- [51] Junwoo Jang, Sangli Teng, and Maani Ghaffari. Convex geometric trajectory tracking using lie algebraic mpc for autonomous marine vehicles. *IEEE Robotics and Automation Letters*, 8(12):8374–8381, 2023. doi: 10.1109/LRA.2023.3328450.
- [52] Anton Schiela and Julian Ortiz. An SQP method for equality constrained optimization on manifolds. *arXiv preprint arXiv:2005.06844*, 2020.
- [53] Mitsuaki Obara, Takayuki Okuno, and Akiko Takeda. Sequential quadratic optimization for nonlinear optimization problems on Riemannian manifolds. *SIAM Journal on Optimization*, 32(2):822–853, 2022.
- [54] Yuya Yamakawa and Hiroyuki Sato. Sequential optimality conditions for nonlinear optimization on Riemannian manifolds and a globally convergent augmented Lagrangian method. *Computational Optimization and Applications*, 81(2):397–421, 2022.
- [55] Changshuo Liu and Nicolas Boumal. Simple algorithms for optimization on Riemannian manifolds with constraints. *Applied Mathematics & Optimization*, 82(3):949–981, 2020.
- [56] Zhijian Lai and Akiko Yoshise. Riemannian interior point methods for constrained optimization on manifolds. *Journal of Optimization Theory and Applications*, 201(1):433–469, 2024.
- [57] Loring W Tu. *Differential Geometry*. Springer Cham, 2017.
- [58] John Milnor. Curvatures of left invariant metrics on Lie groups. *Advances in Mathematics*, 21(3):293–329, 1976.
- [59] Jerrold E Marsden, Sergey Pekarsky, and Steve Shkoller. Discrete Euler-Poincaré and Lie-Poisson equations. *Nonlinearity*, 12(6):1647, 1999.
- [60] Nicolas Boumal, Bamdev Mishra, P-A Absil, and Rodolphe Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. *The Journal of Machine Learning Research*, 15(1):1455–1459, 2014.
- [61] Wei Hong Yang, Lei-Hong Zhang, and Ruyi Song. Optimality conditions for the nonlinear programming problems on Riemannian manifolds. *Pacific Journal of Optimization*, 10(2):415–434, 2014.
- [62] Stephen J Wright. Numerical optimization, 2006.
- [63] Taeyoung Lee, Melvin Leok, and N Harris McClamroch. Geometric tracking control of a quadrotor uav on SE(3). In *Proceedings of the IEEE Conference on Decision and Control*, pages 5420–5425. IEEE, 2010.
- [64] Joel AE Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.
- [65] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [66] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.
- [67] Jean Bernard Lasserre. *An introduction to polynomial and semi-algebraic optimization*, volume 52. Cambridge University Press, 2015.
- [68] Alexandre Amice, Hongkai Dai, Peter Werner, Annan Zhang, and Russ Tedrake. Finding and optimizing certified, collision-free regions in configuration space for robot manipulators. In *International Workshop on the Algorithmic Foundations of Robotics*, pages 328–348. Springer, 2023.
- [69] Yuwei Wu, Igor Spasojevic, Pratik Chaudhari, and Vijay Kumar. Towards optimizing a convex cover of collision-free space for trajectory generation. *IEEE Robotics and Automation Letters*, 2025.
- [70] Peter Betsch. The discrete null space method for the energy consistent integration of constrained mechanical systems: Part i: Holonomic constraints. *Computer Methods in Applied Mechanics and Engineering*, 194(50-52):5159–5190, 2005.
- [71] Peter Betsch and Sigrid Leyendecker. The discrete null space method for the energy consistent integration of constrained mechanical systems. part ii: Multibody dynamics. *International journal for numerical methods in engineering*, 67(4):499–552, 2006.
- [72] Sigrid Leyendecker, Sina Ober-Blobaum, Jerrold E Marsden, and Michael Ortiz. Discrete mechanics and optimal control for constrained systems. *Optimal Control Applications and Methods*, 31(6):505–528, 2010.
- [73] Shucheng Kang, Xiaoyang Xu, Jay Sarva, Ling Liang, and Heng Yang. Fast and certifiable trajectory optimization. *arXiv preprint arXiv:2406.05846*, 2024.
- [74] Heng Yang, Ling Liang, Luca Carlone, and Kim-Chuan Toh. An inexact projected gradient method with rounding and lifting by nonlinear programming for solving rank-one semidefinite relaxation of polynomial optimization. *Mathematical Programming*, pages 1–64, 2022.
- [75] Eric R Westervelt, Jessy W Grizzle, and Daniel E Koditschek. Hybrid zero dynamics of planar biped walkers. *IEEE transactions on automatic control*, 48(1):42–56, 2003.
- [76] Marin Kobilarov, Jerrold E Marsden, and Gaurav S Sukhatme. Geometric discretization of nonholonomic systems with symmetries. *Discrete and Continuous Dynamical Systems Series S*, 3(1):61–84, 2010.
- [77] Taylor A Howell, Kevin Tracy, Simon Le Cleac’h, and Zachary Manchester. CALIPSO: A differentiable solver for trajectory optimization with conic and complementarity constraints. In *The International Symposium of Robotics Research*, pages 504–521. Springer, 2022.