Special Issue: RSS2023

The International Journal of Robotics Research
2024, Vol. 0(0) 1–30

DOI: 10.1177/02783649241296160
journals.sagepub.com/home/ijr
Sage

Convex geometric motion planning of multi-body systems on lie groups via variational integrators and sparse moment relaxation

Sangli Teng[1], Ashkan Jasour[2], Ram Vasudevan[1] and Maani Ghaffari[1,3]

Abstract

This paper reports a novel result: with proper robot models based on geometric mechanics, one can formulate the kinodynamic motion planning problems for rigid body systems as exact *polynomial optimization problems. Due to the nonlinear rigid body dynamics, the motion planning problem for rigid body systems is nonconvex. Existing global optimization-based methods do not parameterize 3D rigid body motion efficiently; thus, they do not scale well to long-horizon planning problems. We use Lie groups as the configuration space and apply the variational integrator to formulate the forced rigid body dynamics as quadratic polynomials. Then, we leverage Lasserre's hierarchy of moment relaxation to obtain the globally optimal solution via semidefinite programming. By leveraging the sparsity of the motion planning problem, the proposed algorithm has* linear *complexity with respect to the planning horizon. This paper demonstrates that the proposed method can provide globally optimal solutions or certificates of infeasibility at the second-order relaxation for 3D drone landing using full dynamics and inverse kinematics for serial manipulators. Moreover, we extend the algorithms to multi-body systems via the constrained variational integrators. The testing cases on cart-pole and drone with cable-suspended load suggest that the proposed algorithms can provide rank-one optimal solutions or nontrivial initial guesses. Finally, we propose strategies to speed up the computation, including an alternative formulation using quaternion, which provides empirically tight relaxations for the drone landing problem at the first-order relaxation.*

Keywords

Motion planning, geometric mechanics, rigid body dynamics, moment relaxation, polynomial optimization, matrix lie groups, trajectory optimization, semidefinite programming, inverse kinematics

1. Introduction

The kinodynamic motion planning (Donald et al., 1993), or trajectory optimization (Betts, 1998), which aims to synthesize robot motions subject to kinematics, dynamics, and input constraints, is fundamental in robotics research. A typical formulation of kinodynamic motion planning is a constrained optimization problem, usually nonconvex due to the nonlinear dynamics and obstacle configurations. Despite the nonconvexity of these problems, the optimization methods that exploit the local gradient information have been successfully applied to find local optimal solutions. Unless problem-specific convexification is accessible (Behcet et al., 2013), there is generally no guarantee of global optimality.

Indeed, the complexity of motion planning problems with arbitrary obstacles is high (Canny, 1987; Hopcroft et al., 1984; Reif, 1979) that one should not expect an efficient algorithm to obtain globally optimal solutions for general problems. For motion planning problems in moderate size, global optimization techniques, such as mixed-integer programming (Cohn et al., 2023; Dai et al., 2019; Deits and Tedrake, 2014, 2015; Ding et al., 2018, 2020; Richards and How, 2002; Schouwenaars et al., 2001) and polynomial optimization (Amice et al., 2023; El Khadir et al., 2021; Trutman et al., 2022), have been applied to obtain or approximate the globally optimal solutions. However, these methods do not consider full robot dynamics or apply approximations that sacrifice the modeling fidelity.

Thus, the natural question is how can we obtain the *globally optimal* solution of motion planning problem using *exact* robot dynamics? The main challenges are the scalability of the global optimization algorithm and the absence of proper robot formulations. For the former problem, recent

[1]Robotics Department, University of Michigan, Ann Arbor, MI, USA
[2]Team 347T-Robotic Aerial Mobility, Jet Propulsion Lab., Pasadena, CA, USA
[3]Naval Architecture & Marine Engineering, University of Michigan, Ann Arbor, MI, USA

Corresponding author:
Sangli Teng, Robotics Department, University of Michigan, Ford Motor Company Robotics Building, 2505 Hayward St, Ann Arbor, MI 48109-2106, USA.
Email: sanglit@umich.edu

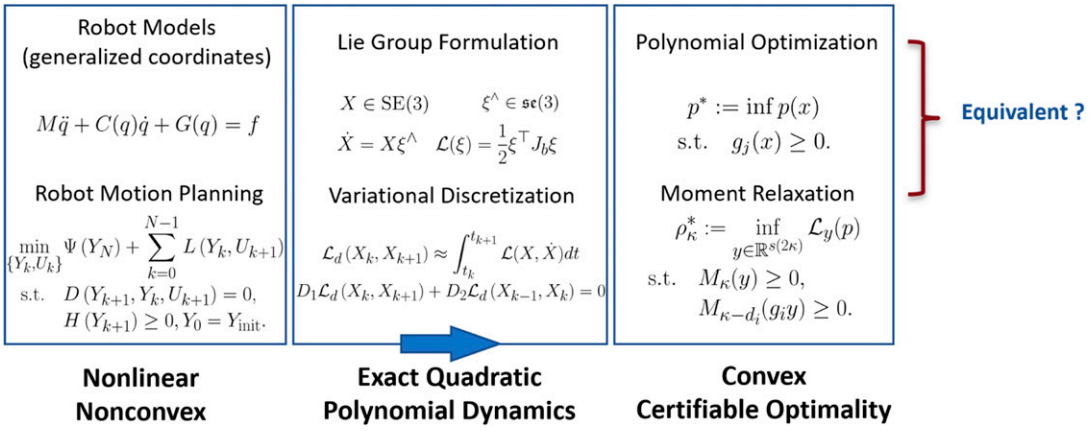

Figure 1. We convert the nonconvex motion planning problem of rigid body systems to convex optimization by formulating the dynamics constraints with the variational integrator on the matrix Lie group. We show that Lasserre's hierarchy of moment relaxation at the second order is capable of providing tight relaxations for our testing cases. A fundamental question and future direction is to explore when the SDP and POP are equivalent, which is critical to solving the motion planning problem of rigid body systems via convex optimization.

progress in the Polynomial Optimization Problem (POP), that is, Lasserre's hierarchy of moment relaxation (Lasserre, 2001, 2015), enables one to compute globally optimal solutions of POPs via a sequence of Semidefinite Programming (SDP). For the latter problem, we apply the variational integrator on Lie groups to formulate rigid body dynamics as quadratic polynomials.

In this paper, we show that bridging the geometric robotics formulation and Lasserre's hierarchy leads to certifiably optimal solutions for motion planning problems using the full dynamics model. We exploit the property of the configuration space of rigid body dynamics and apply variational integrator (Marsden and West, 2001) on Lie groups to generate an exact polynomial dynamics model. Then, we formulate the kinodynamic motion planning problem as a sparse POP with only quadratic polynomials. We further leverage Lasserre's hierarchy to approximate the globally optimal solutions. The proposed geometric motion planning framework on Lie groups is illustrated in Figure 1.

An early version of this work by the current authors was presented as a conference paper (Teng et al., 2023). The major contributions compared to the conference version are:

1. Extension of the proposed algorithm to the multi-body systems via constrained variational integrators.
2. An alternative formulation of the proposed algorithm on quaternion and $\text{SO}(3) \times \mathbb{R}^3$ group to speed up the computation.
3. More insightful discussion about the complexity and convergence of the proposed algorithm.
4. More comparative study with existing algorithms.
5. Closed-loop simulation to verify the dynamic feasibility of the planned trajectory.
6. Open-source implementation of the proposed algorithms to ensure reproducibility of the presented results and enable future studies https://github.com/SangliTeng/LieGroupSDP.git.

The main contributions of this paper are summarized as follows:

1. Derivation of discrete multi-body robotics dynamics model on SE(3) groups as quadratic polynomials using the Lie group variational integrator.
2. Formulation of kinodynamic motion planning problem of rigid body system as a sparse quadratic polynomial optimization. The induced optimization can be solved by Lasserre's hierarchy of moment relaxation at the second order with linear complexity with respect to the planning horizon.
3. Alternative derivation of the proposed algorithm using quaternion to speed up the computation. The first-order relaxation is found empirically tight in the case study.

2. Background and related work

In this section, we review the motion planning algorithms using sampling methods, gradient-based local optimization, and global optimization.

2.1. Sampling-based motion planning

The sampling-based motion planning algorithm has gained success in recent decades (Agha-Mohammadi et al., 2014; Ghaffari Jadidi et al., 2019; Hollinger and Sukhatme, 2014; Horsch et al., 1994; Kavraki et al., 1996; LaValle and Kuffner Jr, 2001; Sertac and Emilio, 2011; Teng et al., 2021). These sampling-based methods are complete (or, resp. optimal), in the sense that the probability of finding the solution (or, resp. finding the optimal solution) converges to one when the sampling is sufficient (LaValle, 2006; Sertac and Emilio, 2011). However, as the completeness of these methods is only in the sense of probability, there is no guarantee of the runtime, and the algorithm may run forever if the solution does not exist. Due to the sampling nature of

these algorithms, the solutions are chattering and need refining by local solvers.

2.2. Gradient-based local search

The gradient-based local optimization has been applied to obtain the optimal trajectory given the system models and control objectives. These methods try to reach the first-order optimality condition to minimize the control objectives. The indirect method first derives the first-order optimality condition in continuous time and then enforces it in discrete time to obtain the optimal trajectory (Kierzenka and Shampine, 2001). On the other hand, the direct method derives the dynamics in discrete time first and then conducts optimizations (Dong et al., 2023; Hereid and Ames, 2017; Li et al., 2023; Manchester et al., 2019; Posa et al., 2014; Schulman et al., 2014; Zucker et al., 2013). The Differential Dynamic Programming iteratively updates the trajectory via simulating the system and then optimizing around the nominal trajectory to decrease the cost function (Jacobson and Mayne, 1970; Saccon et al., 2013; Tassa et al., 2014). Due to the nonlinear nature of the robot dynamics, these optimization problems are nonconvex, and it is hard to justify whether the locally optimal solutions are also globally optimal. Additionally, the quality of the optimized trajectories is also sensitive to poor initialization.

To mitigate the nonconvexity, the convex feasible set algorithm (Liu et al., 2018; Liu and Tomizuka, 2017) iteratively approximates the free regions in the state space by convex sets and conducts local convex optimization. Under the assumption that the dynamics are fully actuated (Liu and Tomizuka, 2017) or linear (Liu et al., 2018), the algorithm is guaranteed to converge to the local optimal solutions. A lossless convexification approach is proposed to slack the nonconvex thrust constraints for rocket landing problems Behcet et al. (2013). The slacked problem is proved to satisfy the same first-order optimality condition as the original problem. However, these convexification methods are problem-specific and are not general to other robot dynamics. Thus, the key to convexifying a wide range of motion planning problems is to find a universal representation of robot dynamics and a systematic way to solve it via convex optimization.

2.3. Global optimization-based motion planning

The mixed-integer programming has been applied to the collision avoidance or path planning problem for aerial robots (Richards and How, 2002), ground vehicles (Deits and Tedrake, 2015; Schouwenaars et al., 2001), and legged robot (Deits and Tedrake, 2014; Ding et al., 2018, 2020). Richards and How (2002); Schouwenaars et al. (2001); Deits and Tedrake (2015) aim at path planning that omits the dynamics of the robots. Ding et al. (2020, 2018); Deits and Tedrake (2014) applied simplified dynamics models for legged robot foot placement planning. These methods are based on simplified models, specifically for legged robots, and do not consider the 3D kinematics or dynamics constraints. Dai et al. (2019) represents the SO(3) surface by the convex hull in partitioned intervals, thus making the Inverse Kinematics (IK) problem a mixed-integer convex optimization. Then, the branch-and-bound process (Lawler and Wood, 1966) is applied to solve the mixed-integer convex optimization. High accuracy approximation of the SO(3) would require more intervals that dramatically increase the runtime. These combinatorial methods are complete and capable of obtaining the globally optimal solutions (LaValle, 2006), while the cost is the exponential time complexity and omission of the geometry of the configuration space. Thus, these combinatorial methods cannot scale well for long-horizon planning and complex rigid body configurations.

The piece-wise linear kinematics model and Lasserre's hierarchy are applied by El Khadir et al. (2021) to plan paths with collisions defined by time-varying polynomial inequalities. With the increase of moment relaxation order, the work of El Khadir et al. (2021) could asymptotically find the collision-free path when a moving obstacle is presented. Jasour et al. (2021a) extends such Lasserre's hierarchy-based method to develop risk-bounded trajectory planners in the presence of uncertain time-varying obstacles using the notion of risk contours (Jasour and Williams, 2019). Moment relaxation methods have been applied to optimal control of hybrid systems (Zhao et al., 2019) in continuous time. Via a sequence of SDPs, the control input and state monotonically converge to the global optimum. Sum-of-squares programming has been applied by Tedrake et al. (2010) to verify the region of attraction for the feedback controller. The work of Zhao et al. (2019) and Tedrake et al. (2010) consider Taylor expansions to approximate the nonlinear robot dynamics. As the Taylor expansion is an infinite series, finite order approximation cannot be exact, and higher order approximation inevitably increases the size of the moment matrix.

Trigonometric functions have been represented as rational functions by Amice et al. (2023); Trutman et al. (2022) to formulate the kinematic constraints in the IK problem. Using this parameterization, Trutman et al. (2022) apply the Lasserre's hierarchy to compute globally optimal solutions of IK and Amice et al. (2023) apply the sum-of-squares (Blekherman et al., 2012; Parrilo, 2003) to certify the collision-free regions. However, both Amice et al. (2023) and Trutman et al. (2022) only consider the rotation about a single axis and need to represent the joint SE(3) pose by a chain of trigonometric polynomials. The drawback is that when the kinematic chain increases, the degree of the polynomial representing the pose also increases. Additionally, Amice et al. (2023); Trutman et al. (2022) propose dense formulations as each equality constraint involves all joint angles. The work of Marić et al. (2020) casts the IK as a problem of finding the nearest point of an algebraic variety and proves all non-singular poses can be tightly solved by sparse-SDP relaxation (Cifuentes et al., 2020). However, this formulation relies on distance geometry that only applies to kinematic

problems. The works of Wang et al. (2020); Jasour et al. (2021b) describe the exact uncertainty propagation of systems parameterized by trigonometric polynomials using the notion of trigonometric moments. Though dynamics models are considered in both cases, the constraints between the Euler angles and their trigonometric functions are not linear. Such constraints make it hard to lift the dynamics constraints to the moment space. To solve the above issues, we propose a sparse geometric-based formulation that only involves quadratic polynomials, which is suitable for long-horizon applications.

2.4. Lasserre's hierarchy and certifiable optimality

Lasserre's hierarchy converts the POP to infinite-dimensional linear programming in the measure space and approximates the globally optimal solution via truncated moment sequences in finite dimension (Lasserre, 2001, 2015). By increasing the order of the moment matrix, one can monotonically approximate the globally optimal solution of POP by solving a sequence of SDPs. When the cost function and constraints satisfy certain technical conditions (Nie, 2014), Lasserre's hierarchy can converge *exactly* to the global optimum in finite relaxation order. For large-scale problems, the sparsity pattern in POP has been used to reduce the size of the SDP by breaking the dense moment matrix into smaller ones (Lasserre, 2006; Waki et al., 2006; Wang et al., 2021b). For control problems that satisfy Markov assumption, the sparsity pattern (Lasserre, 2006; Wang et al., 2022) makes the computation time linear in the problem horizon (Waki et al., 2006).

Lasserre's hierarchy has been applied to perception problems (Yang and Carlone, 2022) as the certifiable algorithm (Bandeira, 2016). The certifiable algorithm (Bandeira, 2016; Yang and Carlone, 2022) requires that (i) the algorithm runs in polynomial time, (ii) returns a globally optimal solution with a certificate of the optimality, (iii) or fails to do so but provides a bound on the objective value. When the relaxation order of Lasserre's hierarchy is determined, the SDP can be solved in polynomial time. The optimum of SDP can also be certified as the globally optimal solution via rank conditions of the moment matrix or serve as a lower bound estimation for the problem (Lasserre, 2015). Such properties are rarely seen in the existing motion planning algorithms based on sampling, combinatorial methods, or nonlinear local solvers.

Continuous time control problems of polynomial systems have been lifted to space of measure to obtain optimal feedback (Henrion et al., 2008; Kamoutsi et al., 2017; Yang et al., 2023; Zhao et al., 2019), region of attraction (Henrion and Korda, 2013) or backward reachable set (Majumdar et al., 2014). Though asymptotically convergent approximation by Lasserre's hierarchy is guaranteed, finite convergence is not observed in these cases. In this work, we leverage Lasserre's hierarchy by converting the discrete-time motion planning problem to exact POP for a single trajectory. We find that our algorithms can provide certified rank-one globally optimal solutions or sufficiently good initialization at the second-order relaxation.

3. Theoretical background and preliminaries

3.1. Polynomial optimization and Lasserre's hierarchy

Let $\mathbb{R}[x]$ be the ring of polynomial with real coefficients and $x := (x_1, x_2, \ldots, x_n)$. Given an integer r, we define the set $\mathbb{N}^n_r := \{\alpha \in \mathbb{N}^n | \sum_i \alpha_i \le r\}$. The monomial with degree up to r can be defined as $x^\alpha := x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}, \alpha \in \mathbb{N}^n_r$, and we have the canonical basis $v_r(x)$ for polynomial degree up to r:

$$v_r(x) := (1, x_1, \ldots, x_n, x_1^2, x_1x_2, \ldots, x_n^2, x_1^r, \ldots, x_n^r). \quad (1)$$

Let $s(r) := (n + r)!/n!r!$ be the dimension of $v_r(x)$. Then any r-degree polynomial $p(x) : \mathbb{R}^n \to \mathbb{R}$ could be expressed as

$$p(x) = \sum_\alpha p_\alpha x^\alpha = \langle p, v_r(x) \rangle, \quad (2)$$

where $p = \{p_\alpha\}$ denotes the coefficients corresponding to the basis defined in (1). We define the POP as follows:

Problem 1. (Polynomial Optimization Problem).

$$\begin{aligned} p^* &:= \inf p(x) \\ \text{s.t.} \quad & g_j(x) \ge 0, \quad \forall j \in \{1, \ldots, m\}. \end{aligned} \quad \text{(POP)}$$

where p, g_i are polynomials. We denote the feasible set $\mathbb{K} := \{x | g_j(x) \ge 0, j = 1, 2, \ldots, m\}$.

We denote the degree, that is, the highest order of monomial, of polynomial g as deg g. Then we have the degree integers as $d_i := \lceil \deg(g_i)/2 \rceil$ and $d_g = \max\{1, d_1, \ldots, d_m\}$, where $\lceil a \rceil$ denotes the smallest integer greater than or equal to a.

Given a probability distribution $\mu(x)$ in $\mathbb{R}^n$ and $\alpha \in \mathbb{N}^n_r$, then the moment of $\mu(x)$ at order α is defined as

$$y_\alpha = y_{\alpha_1, \ldots, \alpha_n} = \mathbb{E}[x^\alpha] = \int x^\alpha \mu(x) dx. \quad (3)$$

We construct the (truncated) moment matrix $M_r(y) \in \mathbb{R}^{s(r) \times s(r)}$ via a $s(2r)$-sequence $y = (y_\alpha)$, with rows and columns labeled the same as in (1). For example, moment matrix $M_2(y)$, with $n = 2$, $r = 2$, is

$$M_2(y) = \begin{bmatrix} y_{0,0} & y_{1,0} & y_{0,1} & y_{2,0} & y_{1,1} & y_{0,2} \\ y_{1,0} & y_{2,0} & y_{1,1} & y_{3,0} & y_{2,1} & y_{1,2} \\ y_{0,1} & y_{1,1} & y_{0,2} & y_{2,1} & y_{1,2} & y_{0,3} \\ y_{2,0} & y_{3,0} & y_{2,1} & y_{4,0} & y_{3,1} & y_{2,2} \\ y_{1,1} & y_{2,1} & y_{1,2} & y_{3,1} & y_{2,2} & y_{1,3} \\ y_{0,2} & y_{1,2} & y_{0,3} & y_{2,2} & y_{1,3} & y_{0,4} \end{bmatrix}. \quad (4)$$

Suppose $y = (y_\alpha) \subset \mathbb{R}$ be a sequence indexed by $\alpha \in \mathbb{N}^n_r$, and let the $\mathcal{L}_y : \mathbb{R}[x] \to \mathbb{R}$ be the linear functional:

$$f(x) = \sum_{\alpha} f_\alpha x^\alpha \mapsto \mathcal{L}_y(f) = \sum_{\alpha} f_\alpha y_\alpha. \tag{5}$$

The functional (5) can be interpreted as substituting the monomials x^α in $f(x)$ by corresponding y_α to obtain the numerical value. Then $M_r(y)$ can be constructed by

$$M_r(y)(\alpha,\beta) = \mathcal{L}_y(x^\alpha x^\beta) = y_{\alpha+\beta}, \alpha,\beta \in \mathbb{N}^n_r, \tag{6}$$

or equivalently, by manipulating the entire $v_r(x)v_r(x)^{\mathrm{T}}$:

$$M_r(y) = \mathcal{L}_y(v_r(x)v_r(x)^{\mathrm{T}}). \tag{7}$$

Finally, we also have the localizing matrix for g_i as

$$M_{r-d_i}(g_i y) = \mathcal{L}_y(g_i v_{r-d_i}(x)v_{r-d_i}(x)^{\mathrm{T}}). \tag{8}$$

One equivalent formulation of (POP) is the infinite-dimensional linear programming over the space of probability with support on the feasible set $\mathbb{K}$ (Lasserre, 2001, 2015), where the objective function is a linear combination of the entries of the moment matrix. The detailed formulation of the infinite-dimensional linear programming formulation is presented in Appendix A.

As searching over an infinitely large moment matrix is impossible, we approximate the solution by sequences y_α with finite order $|\alpha| = 2\kappa$. Thus, we have the relaxed SDP in the space of moment matrix $M_\kappa(y)$ as follows.

Problem 2. (Semidefinite relaxation of POP).

$$\begin{aligned} \rho^*_\kappa := \inf_{y\in\mathbb{R}^{s(2\kappa)}} & \mathcal{L}_y(p) \\ \text{s.t.} \quad & M_\kappa(y) \succeq 0, \\ & M_{\kappa-d_i}(g_i y) \succeq 0, \forall j \in \{1,\ldots,m\}. \end{aligned} \tag{SDP}$$

where p and g_j are polynomials.

As (SDP) gets more constrained as κ increases, we could gradually approximate the globally optimal solution of (POP). This observation leads to the theory of Lasserre's hierarchy as follows:

Theorem 1. (Lasserre's Hierarchy (Lasserre, 2001, 2015)). *Let p^* be the optimum of* (POP) *and the ρ^*_κ (resp. y^*_κ) be the optimum (resp. optimizer) of* (SDP)*, then:*

1. *(Monotone lower bound) ρ^*_κ is monotonically increasing and $\rho^*_\kappa \uparrow p^*$ as $\kappa \to \infty$.*
2. *(Rank condition) If the moment matrix satisfies:*

$$\operatorname{rank}(M_\kappa(y^*_\kappa)) = \operatorname{rank}(M_{\kappa-d_g}(y^*_\kappa)),$$

then $\rho^*_\kappa = p^*$. In this case, y^* is a moment sequence that admits a representing measure on $\mathbb{K}$.

3. *(Number of optimizers) If 2) is satisfied, then the number of optimizers equals to* $\operatorname{rank}(M_\kappa(y^*_\kappa))$.
4. *(Finite convergence) If* (POP) *satisfy some suitable technical condition, under the assumption that Archimedeanness condition*[1] *holds for $\mathbb{K}$, then $\rho^*_\kappa = p^*$ happens at some finite order $\kappa^* < \infty$ (Nie, 2014). We note that the actual order κ^* is unknown in advance.*

Lasserre's hierarchy indicates that one can increase κ from the lowest relaxation order until the rank condition is satisfied. For a wide range of applications that only have a unique optimal solution (Teng et al., 2024; Yang and Carlone, 2020, 2022), the following rank-one optimality condition can be expected.

Remark 1. (rank-one optimality condition). If $\operatorname{rank}(M_\kappa(y^*_\kappa)) = 1$, then $\operatorname{rank}(M_{\kappa-d_g}(y^*_\kappa)) = 1$ as $M_{\kappa-d_g}(y^*_\kappa)$ is a non-zero principle submatrix of $M_\kappa(y^*_\kappa)$. We can further justify whether the unique globally optimal solution is obtained by checking the rank of the solution to (SDP).

Remark 2. (Certificate of optimality). The unique global optimal solution to (POP) can be verified if $\operatorname{rank}(M_\kappa(y^*_\kappa)) = 1$. In the case that $\operatorname{rank}(M_\kappa(y^*_\kappa)) \neq 1$, ρ^*_κ can still serve as a lower bound of the optimal cost p^*. Then, we can extract or approximate the solution of (POP) by matrix decomposition.

Remark 3. (Extract solutions). The solution to (POP) can be extracted by matrix decomposition $M_\kappa(y^*_\kappa) = v_\kappa(x^*)v_\kappa(x^*)^{\mathrm{T}}$, given that $\operatorname{Rank}(M_\kappa(y^*_\kappa)) = 1$. If $\operatorname{Rank}(M_\kappa(y^*_\kappa)) \neq 1$, the Singular Value Decomposition of $M_\kappa(y^*_\kappa)$ provides an approximated $v_r(x^*)$ with the components corresponding to the largest singular value.

3.2. Rigid body dynamics

We now introduce the Lie group-based rigid body dynamics. We consider special Euclidean group SE (3) as the configuration space of rigid body motion:

$$X = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \in \mathrm{SE}(3), \tag{9}$$

where $p \in \mathbb{R}^{3\times 1}$ is the position, and R is the rotation defined on the special orthogonal group

$$\mathrm{SO}(3) := \{R \in \mathbb{R}^{3\times 3} \mid R^T R = I_3, \det(R) = 1\}.$$

On SE (3), the twist is defined as the concatenation of angular velocity ω and linear velocity v in the body frame, that is,

$$\xi := \begin{bmatrix} \omega \\ v \end{bmatrix} \in \mathbb{R}^6, \xi^\wedge = \begin{bmatrix} \omega^\times & v \\ 0 & 0 \end{bmatrix} \in \mathfrak{se}(3), \tag{10}$$

where $(\cdot)^\times$ satisfies $a^\times b = a \times b, a, b \in \mathbb{R}^3$. Note that $\mathfrak{se}(3)$ is the tangent space at the identity $X = I$, and $X\xi^\wedge \in T_X\mathrm{SE}(3)$.

The reconstruction equation gives the Equation of Motion (EOM) in continuous time:

$$\dot{X} = X\xi^{\wedge}. \tag{11}$$

We have the inertia matrix J_b and the kinetic energy:

$$J_b := \begin{bmatrix} I_b & 0 \\ 0 & mI_3 \end{bmatrix}, \quad T(\xi) := \frac{1}{2}\xi^{\mathrm{T}} J_b \xi, \tag{12}$$

where $I_b \in \mathbb{R}^{3\times 3}$ is the moment of inertia in the body frame and m is the body mass. We arrive at the Euler-Poincaré equation (Bloch, 2003; Marsden and Ratiu, 1998) if we take the variation in T_XSE (3):

$$J_b\dot{\xi} + \begin{bmatrix} \omega^{\times} & v^{\times} \\ 0 & \omega^{\times} \end{bmatrix} J_b\,\xi = 0. \tag{13}$$

3.3. Variational integrator

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

$$\mathcal{L}(q, \dot{q}) := T(\dot{q}) - V(q). \tag{14}$$

The key idea of a variational integrator is to discretize the Lagrangian (14) to obtain the discrete-time EOM (Marsden and West, 2001). The discretization scheme ensures that the Lagrangian is conserved in discrete time, thus having superior energy conservation properties over a long duration.

We define the time step $\Delta t \in \mathbb{R}$ and the time sequence $\{t_k = k\Delta t \mid k = 0, \ldots, N\} \subset \mathbb{R}$. Thus, the discrete Lagrangian $\mathcal{L}_d : Q \times Q \to \mathbb{R}$ could be considered as the approximation of the action integral via:

$$\mathcal{L}_d(q_k, q_{k+1}) \approx \int_{t_k}^{t_{k+1}} \mathcal{L}(q, \dot{q}) dt. \tag{15}$$

In this work, we consider the midpoint approximation (Marsden and West, 2001):

$$\mathcal{L}_d(q_k, q_{k+1}) = T\left(\frac{q_{k+1} - q_k}{\Delta t}\right)\Delta t - V\left(\frac{q_{k+1} + q_k}{2}\right)\Delta t. \tag{16}$$

Then, the discrete variant of the action integration becomes:

$$S_d = \sum_{k=0}^{N-1} \mathcal{L}_d(q_k, q_{k+1}). \tag{17}$$

Finally, we take the discrete version of integration by parts (Marsden and West, 2001) to obtain the dynamics. The first step is to take variation in TQ, and then we group the term corresponding to $\delta q_k \in T_{q_k}Q$:

$$\begin{aligned} \delta S_d &= \sum_{k=1}^{N-1} D_1\mathcal{L}_d(q_k, q_{k+1})^{\mathrm{T}}\delta q_k + D_2\mathcal{L}_d(q_k, q_{k+1})^{\mathrm{T}}\delta q_{k+1} \\ &= D_1\mathcal{L}_d(q_0, q_1)^{\mathrm{T}}\delta q_0 + D_2\mathcal{L}_d(q_{N-1}, q_N)^{\mathrm{T}}\delta q_N \\ &+ \sum_{k=1}^{N-1} \left(D_2\mathcal{L}_d(q_{k-1}, q_k) + D_1\mathcal{L}_d(q_k, q_{k+1})\right)^{\mathrm{T}}\delta q_k, \end{aligned} \tag{18}$$

where D_i denotes the derivative with respect to the i-th argument. By the least action principle, the EOM can be characterized by the stationary point:

$$D_1\mathcal{L}_d(q_k, q_{k+1}) + D_2\mathcal{L}_d(q_{k-1}, q_k) = 0. \tag{19}$$

To incorporate the external force $f \in T_q^*Q$, we can compute the action integral again using the midpoint approximation as

$$\int_{t_k}^{t_{k+1}} f(t)^{\mathrm{T}}\delta q\ dt \approx \frac{\Delta t}{2} f(t_k)^{\mathrm{T}}\delta q_k + \frac{\Delta t}{2} f(t_{k+1})^{\mathrm{T}}\delta q_{k+1}. \tag{20}$$

Thus, the integration over $[t_0, t_N]$ can be approximated by the following summation:

$$\begin{aligned} &\int_{t_0}^{t_N} f(t)^{\mathrm{T}}\delta q\ dt = \sum_{k=0}^{N-1}\int_{t_k}^{t_{k+1}} f(t)^{\mathrm{T}}\delta q\ dt \\ &\approx \sum_{k=0}^{N} \frac{\Delta t}{2} f(t_k)^{\mathrm{T}}\delta q_k + \frac{\Delta t}{2} f(t_{k+1})^{\mathrm{T}}\delta q_{k+1} \\ &= \frac{\Delta t}{2} f(t_0)^{\mathrm{T}}\delta q_0 + \frac{\Delta t}{2} f(t_N)^{\mathrm{T}}\delta q_N \\ &\quad + \sum_{k=1}^{N-1} f(t_k)^{\mathrm{T}}\delta q_k \Delta t \end{aligned}. \tag{21}$$

We then have the forced EOM after incorporating it into (19):

$$D_1\mathcal{L}_d(q_k, q_{k+1}) + D_2\mathcal{L}_d(q_{k-1}, q_k) = f_k\Delta t. \tag{22}$$

4. Problem formulation

We consider the robot motion planning problem as the following constrained optimization:

Problem 3. (Discrete Kinodynamic Motion Planning). Consider the robot configuration state Y_k and control input U_k[2] at time step k. Given the terminal cost $\Psi(\cdot)$ and running cost $L(\cdot, \cdot)$, our goal is to plan a trajectory $\{Y_k\}_{k=0}^N$, $\{U_k\}_{k=1}^N$ subject to the initial condition $Y_0 = Y_{\text{init}}$ via the following optimization:

$$\begin{aligned}\min_{\{Y_k\}_{k=0}^N,\{U_k\}_{k=1}^N} \quad & \Psi(Y_N)+\sum_{k=0}^{N-1}L(Y_k,U_{k+1})\\ \text{s.t.} \quad & D(Y_{k+1},Y_k,U_{k+1})=0,\\ & Y_k\in\mathcal{Y},\\ & H(Y_{k+1})\geq 0,\\ & U_{\min}\leq U_k\leq U_{\max},\\ & Y_0=Y_{\text{init}},\\ & k=0,\ldots,N-1.\end{aligned} \tag{23}$$

where $D(\cdot,\cdot,\cdot)$ is the equality constraints induced from robot dynamics, $U_{\min}$ and $U_{\max}$ are the minimal and maximal control input, and $H(\cdot)$ specifies the inequality constraints, such as collision avoidance. $\mathcal{Y}$ specifies the set of states Y, such as the matrix Lie group or quaternion.

For a wide range of robotics systems composed of rigid bodies, the equality $D(\cdot,\cdot,\cdot)$ is nonlinear, thus resulting in nonconvex constraints. The nonconvexity of these equality constraints makes the global optimal solutions to Problem 3 hard to access or certify. In this work, we aim to develop a systematic way to convexify the rigid body dynamics to access the global optimal solution to Problem 3. Specifically, we consider the Y_k as the configuration state of rigid bodies and formulate the Problem 3 as a (POP). A flow chart of the proposed algorithm is presented in Figure 1, which has illustrated the procedure to convert the nonconvex motion planning problem to a convex optimization.

5. Discrete motion of single rigid body

This section introduces our method to formulate Problem 3 as exact POP on matrix Lie groups.

5.1. Polynomial kinematics constraints

We consider the Euler integration on SE (3) as our discrete-time EOM for the kinematic part:

$$\begin{bmatrix}R_{k+1} & p_{k+1}\\ 0 & 1\end{bmatrix}=\begin{bmatrix}R_k & p_k\\ 0 & 1\end{bmatrix}\begin{bmatrix}F_k & hv_k\\ 0 & 1\end{bmatrix}, \tag{24}$$

where $F_k\in\text{SO}(3)$ denotes pose change in discrete time. As $R\in\text{SO}(3)$ contains nine entries while $\dim\text{SO}(3)=3$, we need additional constraints. Consider the column space:

$$R:=[r_1,r_2,r_3]\in\mathbb{R}^{3\times 3},\quad r_1,r_2,r_3\in\mathbb{R}^{3\times 1},$$

$R\in\text{SO}(3)$ is equivalent to the following 15 quadratic equality constraints:

$$\begin{aligned}&\|r_1\|^2-1=\|r_2\|^2-1=\|r_3\|^2-1=0,\\ &r_1^{\mathrm{T}}r_2=r_2^{\mathrm{T}}r_3=r_1^{\mathrm{T}}r_3=0,\\ &r_1\times r_2-r_3=r_2\times r_3-r_1=r_3\times r_1-r_2=0_{3\times 1}.\end{aligned} \tag{25}$$

The first six equations ensure that r_i are orthonormal to each other, and the last nine equations ensure r_i follows the right-hand rule, which is equivalent to the determinant constraints but is quadratic. Due to these quadratic constraints, applying the explicit Euler integration on the vectorized system (11) does not guarantee that the trajectory evolves on the SO(3) manifold.

To indicate rotation about a single axis or planar dynamics on SE (2), we model sine and cosine as two separate variables as $c=\sin(\theta)$ and $s=\cos(\theta)$ and we have:

$$\begin{aligned}R_x&=\begin{bmatrix}1&0&0\\0&c&-s\\0&s&c\end{bmatrix},R_y=\begin{bmatrix}c&0&s\\0&1&0\\-s&0&c\end{bmatrix},\\ R_z&=\begin{bmatrix}c&-s&0\\s&c&0\\0&0&1\end{bmatrix},\text{with } c^2+s^2=1.\end{aligned} \tag{26}$$

We use the notation $R_i(\theta)$ to denote the rotation of θ angle about axis $i\in\{x,y,z\}$.

5.2. Polynomial dynamics constraints

We now derive the dynamics model on SE (3) via LGVI (Lee et al., 2005, 2007; Marsden et al., 1999; Nordkvist and Sanyal, 2010). The derivation for SO(3) has been well-established in (Lee et al., 2005, 2007; Marsden et al., 1999). However, a formulation on SE (3) suitable for POP implementation is absent. We apply the midpoint approximation (16) to represent the twist ξ_k using the configuration state:

$$F_k:=R_k^{-1}R_{k+1}\approx I+\Delta t\omega_k^{\times},\quad \omega_k^{\times}\approx\frac{F_k-I}{\Delta t}, \tag{27}$$

$$\dot{p}_k=R_kv_k\approx\frac{p_{k+1}-p_k}{\Delta t},\quad v_k\approx\frac{R_k^{\mathrm{T}}(p_{k+1}-p_k)}{\Delta t}. \tag{28}$$

We refer to (Lee et al., 2005) for the expression of the kinetic energy of rotation. Then by (16), the discrete kinetic and potential energy takes the form:

$$\begin{aligned}T_d:=&\frac{1}{2\Delta t}\text{tr}((F_k-I)I^b(F_k-I)^{\mathrm{T}})\\ &+\frac{1}{2\Delta t}m\|p_{k+1}-p_k\|^2,\end{aligned} \tag{29}$$

$$V_d:=m\left(\frac{p_{k+1}+p_k}{2}\right)^{\mathrm{T}}g\Delta t, \tag{30}$$

Where $g\in\mathbb{R}^{3\times 1}$ is the gravity and I^b the nonstandard moment of inertia (Marsden et al., 1999) that relate the standard moment of inertia I_b by $I_b=\text{tr}(I^b)I_3-I^b$. The detailed derivation and the explanation of the I^b are well presented in (Lee et al., 2005; Marsden et al., 1999). Then we define the variation $\delta X\in T_X\text{SE}(3)$ as

$$\begin{aligned}&\delta X=X\delta\eta^{\wedge}\in T_X\text{SE}(3),\\ &\delta\eta^{\wedge}=\begin{bmatrix}\delta\omega^{\times}&\delta\rho\\0&0\end{bmatrix}\in\mathfrak{se}(3).\end{aligned} \tag{31}$$

Here, we only derive the position part that is not presented in the existing literature. Consider V_d and

$$T_{d,p}(p_k,p_{k+1})=\frac{1}{2\Delta t}m\|p_{k+1}-p_k\|^2,$$

then we have the variation of position in the world frame as $\delta p = R\delta\rho$. Using (19), we have

$$\begin{aligned} D_1 T_{d,p} &= m\left(\frac{p_k - p_{k+1}}{\Delta t}\right)^{\mathrm{T}} R_k \delta\rho_k \\ &= m\left(R_k^{\mathrm{T}}\frac{p_k - p_{k+1}}{\Delta t}\right)^{\mathrm{T}} \delta\rho_k \\ D_2 T_{d,p} &= m\left(\frac{p_k - p_{k-1}}{\Delta t}\right)^{\mathrm{T}} R_k \delta\rho_k \\ &= m\left(R_k^{\mathrm{T}}\frac{p_k - p_{k-1}}{\Delta t}\right)^{\mathrm{T}} \delta\rho_k \end{aligned}. \tag{32}$$

Similarly, we have the variation for the potential energy as

$$D_1 V_d = D_2 V_d = \frac{mg^{\mathrm{T}}\Delta t}{2} R_k \delta\rho_k = \left(R_k^{\mathrm{T}}\frac{mg\Delta t}{2}\right)^{\mathrm{T}} \delta\rho_k. \tag{33}$$

We wrap up (32) and (33) to obtain

$$mR_k^{\mathrm{T}}\frac{p_{k+1} - p_k}{\Delta t} = mR_k^{\mathrm{T}}\frac{p_k - p_{k-1}}{\Delta t} + R_k^{\mathrm{T}} mg\Delta t. \tag{34}$$

Then, by substituting (27) and (28) into (34), we get

$$mv_{k+1} = mF_k^{\mathrm{T}} v_k + mR_{k+1}^{\mathrm{T}} g\Delta t. \tag{35}$$

Finally, we present the LGVI for the unconstrained rigid body:

$$\begin{aligned} &R_{k+1} = R_k F_k, p_{k+1} = p_k + R_k v_k \Delta t, \\ &F_{k+1} I^b - I^b F_{k+1}^{\mathrm{T}} = I^b F_k - F_k^{\mathrm{T}} I^b, \\ &mv_{k+1} = mF_k^{\mathrm{T}} v_k + mR_{k+1}^{\mathrm{T}} g\Delta t. \end{aligned} \tag{36}$$

We note that the constraints in (36) are exact quadratic polynomials, which enable us to formulate Problem 3 as (POP). To verify the correctness of the derived integrator, we plot the kinetic energy and twist for the system without gravity in Figure 2. We can see that the kinetic energy is conserved for a long time, while the explicit Euler method with dynamics (13) diverges fast. We also compare different choices of Δt in Figure 3 to study its influence on the accuracy of the integrator. We find that smaller Δt enables the trajectory to converge to the continuous time trajectory. Though it is expected that the discretized trajectories with larger Δt deviate from the continuous time one, the integrator still remains stable and conserves the momentum and energy.

6. Extension to multi-body systems

This section introduces the constrained variational integrator and applies it to multi-rigid body systems. We further extract the sparsity patterns and analyze the complexity of the moment relaxation w.r.t the planning horizon and degrees of freedom (DOF) of the rigid body systems.

6.1. Constrained variational integrator

The generalized or minimal coordinates with dimensions equal to the DOF have been widely applied to model the multi-rigid body systems. Though the dimension of the state space model is minimized, the dynamical system is generally evolving on the Riemannian manifold that is highly nonlinear (Bullo and Murray, 1999), which makes it challenging for optimization-based motion planning.

In this work, instead of minimal coordinates, we consider the maximal coordinates[3] (Brüdigam and Manchester, 2021b; Leyendecker et al., 2008) that enable the dynamics constraints to remain quadratic. We consider the m-dimensional holonomic constraints as the smooth manifold:

$$g(q) = 0, g : Q \to \mathbb{R}^m. \tag{37}$$

By D'Alembert's principle, the work by the constrained force $\lambda \in \mathbb{R}^m$ is zero; thus, we have the augmented Lagrangian:

$$\overline{\mathcal{L}}(q, \dot{q}) := \mathcal{L}(q, \dot{q}) - \lambda^{\mathrm{T}} g(q). \tag{38}$$

The augmented action integration can be obtained via:

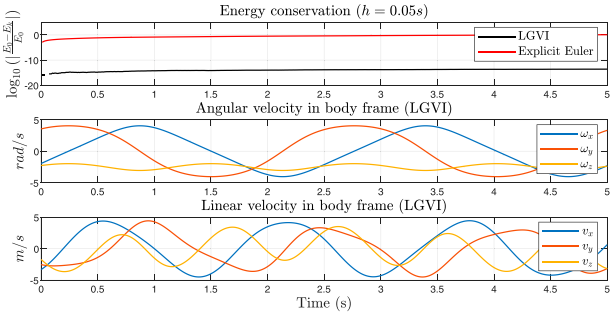

Figure 2. Comparison of the proposed LGVI with explicit Euler integrator for the rigid body system. The presented twists are computed by LGVI. Considering the energy at time step k, the normalized energy loss $|E_k - E_0|/E_0$ is negligible for LGVI while the explicit Euler integrator soon diverges.

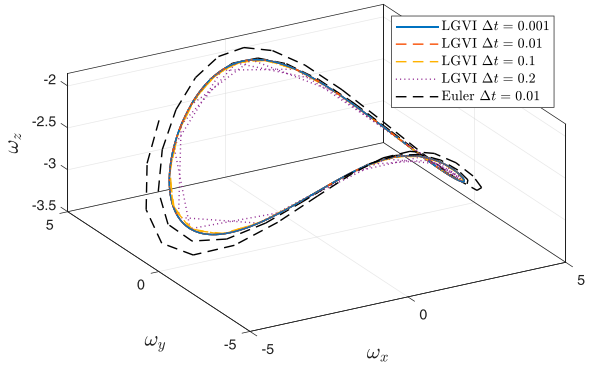

Figure 3. Comparison of the LGVI with different time steps in the phase space. For small time step Δt, the trajectory converges to the continuous time solution. With large Δt, the LGVI still conserves the energy while the explicit Euler method diverges fast.

$$\overline{S}_d = \sum_{k=0}^{N-1} \left(\mathcal{L}_d(q_k, q_{k+1}) - \int_{t_k}^{t_{k+1}} \lambda^{\mathrm{T}} g(q) dt \right). \tag{39}$$

For the latter term about the integration of constrained force, the discrete-time approximation can be derived following (Leyendecker et al., 2008):

$$\int_{t_k}^{t_{k+1}} \lambda^{\mathrm{T}} g(q) dt \approx \frac{\Delta t}{2} \lambda_k^{\mathrm{T}} g(q_k) + \frac{\Delta t}{2} \lambda_{k+1}^{\mathrm{T}} g(q_{k+1}). \tag{40}$$

By taking the variation of δq, we have

$$\begin{aligned} &\delta \int_{t_k}^{t_{k+1}} \lambda^{\mathrm{T}} g(q) dt \\ \approx &\frac{\Delta t}{2} \lambda_k^{\mathrm{T}} \delta g(q_k) + \frac{\Delta t}{2} \lambda_{k+1}^{\mathrm{T}} \delta g(q_{k+1}) \\ \approx &\frac{\Delta t}{2} \lambda_k^{\mathrm{T}} \frac{\partial g(q_k)}{\partial q_k} \delta q_k + \frac{\Delta t}{2} \lambda_{k+1}^{\mathrm{T}} \frac{\partial g(q_{k+1})}{\partial q_{k+1}} \delta q_{k+1} \end{aligned} . \tag{41}$$

Thus, we have the constrained dynamics in the discrete-time by integrating (41) into (18):

$$\begin{aligned} D_1 \mathcal{L}_d(q_k, q_{k+1}) + D_2 \mathcal{L}_d(q_{k-1}, q_k) = G_d^{\mathrm{T}}(q_k) \lambda_k, \\ g(q_{k+1}) = 0, \end{aligned} \tag{42}$$

where $G_d := \partial g / \partial q \Delta t$ is the Jacobians of the constraint manifold in discrete time. The manifold constraints at time step $k+1$ are enforced to ensure the discrete state does not leave the constraint manifold $g(q) = 0$ at the sampled time.

6.2. Main algorithm

We summarize the main procedure of the proposed method in Algorithm 1. We consider a system with N_b rigid bodies, N_g holonomic constraints, N_h state inequality constraints, and m inputs. The state constraints $Y \in \mathcal{Y}$ are polynomial equality constraints with order $\deg \mathcal{Y}$. We first leverage the constrained variational integrator to construct the dynamics constraints $D(\cdot, \cdot, \cdot)$ of the multi-rigid body system. Then we formulate Problem 3 as (POP) and relax it as (SDP) at relaxation order κ. Note that the highest order of the monomials constructed by $v_\kappa(x) v_\kappa(x)^{\mathrm{T}}$ is 2κ. To contain all the monomials involved in the (POP), it is necessary that the relaxation order satisfies $\kappa \geq \kappa_{\min}$, with

$$\kappa_{\min} := \left\lceil \frac{1}{2} \max\{\deg L, \deg \Psi, \deg D, \deg H, \deg \mathcal{Y}\} \right\rceil.$$

We increase κ from $\kappa_{\min}$ until we obtain a rank-one optimal solution or κ reaches the maximal relaxation order $\kappa_{\max}$, which is possibly chosen based on the computational limit. Finally, the solution obtained by (SDP) is rounded by `LocalSearch()` using any local gradient-based method given the solution of (SDP) y_k^* as the initial guess. The `LocalSearch()` can be any Nonlinear Programming (NLP) solver. The objectives of the (SDP) and NLP are also returned to compute the certificate of optimality.

Algorithm 1 Moment relaxation to Problem 3

Require: Configuration state of N_b rigid bodies, $Y := \{Y^{(j)}\}_{j=1}^{N_b}$, feasible set of configuration state $\mathcal{Y}$, holonomic constraints $\{g_j(\cdot)\}_{j=1}^{N_g}$, inequality constraints $\{h_j(\cdot)\}_{j=1}^{N_h}$, running cost $L(\cdot, \cdot)$, terminal cost $\Psi(\cdot)$, input f, input influence map $\{B_d^{(i)}\}_{i=1}^{N_b}$ for each rigid body, planning horizon N and maximal relaxation order $\kappa_{\max}$.

// Initialize the set of dynamics constraints $\mathcal{D}$ and inequality constraints $\mathcal{H}$.
$\mathcal{D} \leftarrow \varnothing, \mathcal{H} \leftarrow \varnothing$
// Obtain the constrained dynamics for each rigid body.
for $i \in \{1, 2, \cdots, N_b\}$ **do**
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{D_1 \mathcal{L}_d(Y_k^{(i)}, Y_{k+1}^{(i)}) + D_2 \mathcal{L}_d(Y_{k-1}^{(i)}, Y_k^{(i)}) = B_d^{(i)} f_{k+1} + \sum_{j=1}^{N_g} G_d^{(i,j)\mathrm{T}} \lambda_{k+1}^{(j)}\}$, with $G_d^{(i,j)} := \frac{\partial g_j}{\partial Y^i} \Delta t$.
end for
for $j \in \{1, 2, \cdots, N_g\}$ **do**
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{g_j(\cdot) = 0\}$
end for
for $j \in \{1, 2, \cdots, N_h\}$ **do**
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{h_j(\cdot) \geq 0\}$
end for
// Formulate Problem 3 as (POP).
(POP) $\leftarrow \{\mathcal{D}, \mathcal{H}, \mathcal{Y}, L, \Psi, U_{\min}, U_{\max}, N\}$.
// Initialize the relaxation order κ.
$\kappa \leftarrow \left\lceil \frac{1}{2} \max\{\deg L, \deg \Psi, \deg D, \deg H, \deg \mathcal{Y}\} \right\rceil$
while true **do**
  $(y_\kappa^*, \rho_\kappa^*) \leftarrow$ Solve (SDP) with relaxation order κ.
  if $\mathrm{Rank}(M_\kappa(y_\kappa^*)) = 1$ or $\kappa = \kappa_{\max}$ **then**
    break
  end if
  $\kappa \leftarrow \kappa + 1$
end while
$\{Y^*, u^*, \lambda^*, \rho_{NLP}^*\} \leftarrow$ `LocalSearch`(y_κ^*)
return $\{Y^*, u^*, \lambda^*, \rho_\kappa^*, \rho_{NLP}^*\}$

6.3. Complexity analysis

By the LGVI, we can now formulate Problem 3 as (POP) that can be relaxed as (SDP). However, the dimension of the moment matrix grows dramatically as the number of variables and relaxation order increases. Here, we provide an analysis of the complexity of the algorithm, considering the size of the moment matrix.

Considering a N_b rigid body system with the dimension of the configuration space of a single rigid body as n. Suppose the dimension of U_k is m. For Problem 3 with planning horizon N and the dense moment relaxation (SDP) with order κ, the dense moment matrix is

$$M_\kappa \in \mathbb{R}^{d(nN_b, m, N, \kappa) \times d(nN_b, m, N, \kappa)}, \tag{43}$$

with the width

$$\begin{aligned} d(nN_b, m, N, \kappa) &:= \frac{((nN_b + m)N + \kappa)!}{((nN_b + m)N)! \kappa!} \\ &= \frac{1}{\kappa!} \prod_{k=1}^{\kappa} (nN_b + m)N + k. \end{aligned} \tag{44}$$

It is straightforward to see that the size of the moment matrix grows polynomially fast w.r.t N_b and N when κ is fixed:

$$\mathcal{O}(((nN_b + m)N)^{\kappa}). \tag{45}$$

Despite that the moment relaxation can provide globally optimal solutions to Problem 3, it is still intractable to implement the dense formulation. However, as Problem 3 satisfies the Markov assumption, the Correlative Sparsity (CS) can be applied to greatly reduce the computational burden (Lasserre, 2006). The detail of CS is presented in Appendix B and we here provide the results.

Consider Problem 3, we partition the state into N sets $y(I_k) = \{(Y_i, Y_{i-1}, U_i) | i \in I_k\}$ indexed by $I_k = \{k, k+1\}, k = 1, 2, \ldots, N$. Then we can verify the running intersection property (Lasserre, 2006):

$$\forall k = 1, \ldots, q-1, \exists s \leq k, I_{k+1} \cap (\cup_{j=1}^{k} I_j) \subseteq I_s,$$

by selecting $s = k$, that is, $I_{k+1} \cap (\cup_{j=1}^{k} I_j) = \{k+1, k+2\} \cap (\cup_{j=1}^{k} I_j) = \{k+1\} \in I_k$. We can see that the cost and dynamic constraints can also be partitioned into such indexed sets as $D(Y_{k+1}, Y_k, U_{k+1}) = 0$, $H(Y_{k+1}) = 0$ and $L(Y_k, U_k)$ only involves variable $y(I_k)$. As shown in Figure 4, CS enables us to optimize the (SDP) with smaller moment matrices instead of a large dense matrix.

By exploiting CS, we have transformed the computational burden of the relaxation (SDP) of Problem 3 to

$$\mathcal{O}(N(nN_b + m)^{\kappa}), \tag{46}$$

which is linear w.r.t the planning horizon N. Similar results in control problems have also been reported by Lasserre (2006, 2015).

Though CS breaks the dense matrix $M_\kappa(y)$ to N smaller ones that only involve variables in $y(I_k)$, the sparse moment relaxation has the same convergence result as the dense version (Lasserre, 2006). The sparsity pattern other than the temporal structure will also be explored for the actual implementation to reduce the computational burden further. Other than CS, Term Sparsity (TS) has also been exploited to reduce the computational burden by eliminating more variables in the moment matrix (Wang et al., 2021b; Yang and Carlone, 2022). Relaxation with TS is looser than only with CS; however, the computation time is greatly reduced and sometimes does not sacrifice the tightness.

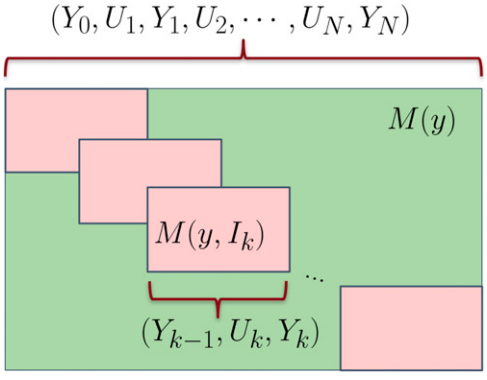

Figure 4. Comparison of dense moment matrix and sparse moment matrix exploiting the correlative sparsity in the temporal structure.

7. Inverse kinematics

In this section, we discuss the simulation setup and the application of the proposed algorithm on IK problems. We also compare the complexity of the proposed algorithm with the existing IK methods based on global optimization.

7.1. Simulation setup

7.1.1. Relaxation order. For our problem with quadratic constraints and quadratic cost function, $\kappa = 1$ is sufficient to ensure $v_\kappa(x)v_\kappa(x)^{\mathrm{T}}$ contains all the monomial involved in Problem 3. For the relaxation with each κ, we check the rank condition in Theorem 1 to justify the global optimality. Specifically, we are interested in the rank-one optimality condition in Remark 1 and Remark 2.

In the studying cases using matrix Lie group formulation, we first try $\kappa = 1$ to implement the moment relaxation, while only a trivial lower bound is accessible. Then, we increase the relaxation order to $\kappa = 2$ and find it provides empirically tight results in the following studying cases.

7.1.2. Sparsity pattern. We use the recent state-of-the-art tool CS-TSSOS (Magron and Wang, 2021; Wang et al., 2021a, 2021b, 2022) to explore the CS and TS pattern. Chordal extension (Wang et al., 2021a) is applied to boost both CS and TS by either extending matrix size (for CS) or reducing more terms (for TS). Due to the Markovian assumption of the control problems, we always exploit the CS pattern. The CS-TSSOS supports the maximal or approximately smallest chordal extension, denoted as MD and block, respectively, referring to the programming API. Assume that the global optimum of (POP) is p^* and the optimum of the (SDP), or local NLP solver is $\rho(\cdot)$, then we have the following inequalities:

$$\begin{aligned} \rho_{TS+MD} \leq \rho_{TS+block} \leq \rho_{CS} \leq \rho_{CS+MD} \\ \leq \rho_{dense} \leq p^* \leq \rho_{NLP}, \end{aligned} \tag{47}$$

where the subscripts denote different sparsity patterns at the same order κ. As the NLP solver is based on local gradient information, only the local optimum is guaranteed. Therefore, ρ_{NLP} serves as an upper bound of p^*.

7.1.3. Evaluation metric. As the optimal value of (POP) cannot be greater than ρ_{NLP}, the objective is upper and lower bounded by the SDP and NLP results. Thus, we can use the relative suboptimality as an index of the suboptimality or relaxation gap:

$$\epsilon := \frac{|\rho_{NLP} - \rho_{SDP}|}{|\rho_{NLP}| + 1e^{-6}}, \qquad \text{(suboptimality)}$$

where a small number is considered in the denominator to avoid $\rho_{NLP} = 0$. As the rank condition is subject to numerical error, we instead check the eigenvalues of each sub-moment matrix as is shown in Remark 4 in Appendix B. We check if each moment matrix is rank-one, assuming the unique minimizer. Assume the eigenvalues of k-th moment matrix $\lambda_{k,i}$ are ranked by their $|\lambda_{k,i}|$ in descending order. Then we compute the ratio between the first and the second one to represent the rank condition:

$$\delta_k = \frac{|\lambda_{k,2}|}{|\lambda_{k,1}|} \leq 1, \delta = \max_k \delta_k. \qquad \text{(rank condition)}$$

As the numerical accuracy is still the bottleneck of the current SDP solvers, δ may not be close to 0 even if the relaxation gap ϵ is small.

7.1.4. Software and hardware setup. The CS-TSSOS is implemented in Julia's open-source package (Wang et al., 2021b, 2022). As MOSEK is based on the primal-dual interior point method, it can generate a certificate for infeasibility (Andersen and Andersen, 2000). We use the certificate returned by MOSEK to indicate the feasibility of the SDP. If (SDP) is infeasible, then its dual will be unbounded (ApS, 2022; Blekherman et al., 2012). Thus, MOSEK will return large dual objective values even if it fails to generate the infeasibility certificate. MOSEK also returns `SLOW_PROGRESS` flag if the problem does not converge successfully, possibly around a minimum. To note that for the problems admitting rank-one solutions, the primal solutions to (SDP) is usually degenerate (Alizadeh et al., 1997; Yang et al., 2022), which may result in slow convergence around the minimum.

As the SDP relaxation of POP does not ensure feasibility or local optimality if the SDP is not tight (Lasserre, 2015), we use the general purpose NLP solver IPOPT (Wächter and Biegler, 2006) to refine the initial guesses provided by the SDPs. All experiments are launched on a desktop equipped with Intel i9-11900 KF CPU, 128 GB memory, and 2 TB harddisk. For the multi-body cases, the virtual memory is configured to consider the increased memory usage.

7.2. Numerical examples on serial manipulator

We consider the IK problem for N DOF serial manipulator with revolute joints. We use $X_k \in \mathrm{SE}(3)$ to represent the pose of k-th joint. Each joint is modeled as rotating about the local z-axis, that is, R_z described in (26) with the angle denoted by c_k and s_k. Note that one can also use DH parameters with similar results. Thus, we have the kinematics chain:

$$X_{k+1} = X_k A_{k+1} T_k, \quad X_0 = I, \qquad (48)$$

with

$$T_k := \begin{bmatrix} R_k^c & 0 \\ 0 & 1 \end{bmatrix}, A_k := \begin{bmatrix} R_k^z & R_k^z p_k^c \\ 0 & 1 \end{bmatrix}. \qquad (49)$$

In this formulation, p_k^c is the constant vector defining the arm, and A_k is the action that describes the rotation of the arm about the z-axis of joint X_k. T_k is a constant that reorientates the pose for joint X_{k+1}. For joint angle constraints $\theta_{\min} \leq \theta \leq \theta_{\max}$, we have the following inequality that is linear w.r.t the cos (θ) and sin (θ):

$$\begin{aligned} &\cos\left(\theta - \frac{\theta_{\max} + \theta_{\min}}{2}\right) \\ &= \cos(\theta)\cos\left(\frac{\theta_{\max} + \theta_{\min}}{2}\right) - \sin(\theta)\sin\left(\frac{\theta_{\max} + \theta_{\min}}{2}\right) \\ &\geq \cos\left(\frac{\theta_{\max} - \theta_{\min}}{2}\right). \end{aligned} \qquad (50)$$

We here formulate an IK problem for an N DOF serial manipulator to reach the target pose $T_g \in \mathrm{SE}(3)$ in the form of Problem 3 as:

Problem 4. (IK for N-DOF serial manipulator).

$$\begin{aligned} \min_{\{c_k, s_k, X_k\}} \quad & \sum_{k=1}^{N} (c_k - c_k^r)^2 + (s_k - s_k^r)^2 \\ \text{s.t.} \quad & X_{k+1} = X_k A_{k+1} T_k \\ & X_k \in \mathrm{SE}(3), \\ & c_{k+1}^2 + s_{k+1}^2 = 1, \\ & c_{k+1}\overline{c}_{k+1} + s_{k+1}\overline{s}_{k+1} \geq c_{\lim}, \\ & k = 0, 1, \ldots, N-1. \\ & X_0 = I, X_N = X_g. \end{aligned} \qquad (51)$$

The $\overline{c}_k, \overline{s}_k$ and $c_{\lim}$ correspond to terms in the expansion of (50), and the cost function is parameterized by a reference joint angle $c_k^r = 1$ and $s_k^r = 0$ in our case to ensure unique solutions.

We tested the proposed algorithm on the 6-DOF manipulator PUMA 560 that has analytical IK solutions (Merat, 1987). As is shown in Figure 5, we uniformly sample 14553 positions in the workspace with random rotations. We validate the algorithm based on the quality of infeasibility detection, and all the seven joint poses X_k (including the base joint) recovered from SDP without refinement. We consider the pose as infeasible if MOSEK returns an infeasibility certificate or unreasonably large objective value. We set the threshold of the dual solution to

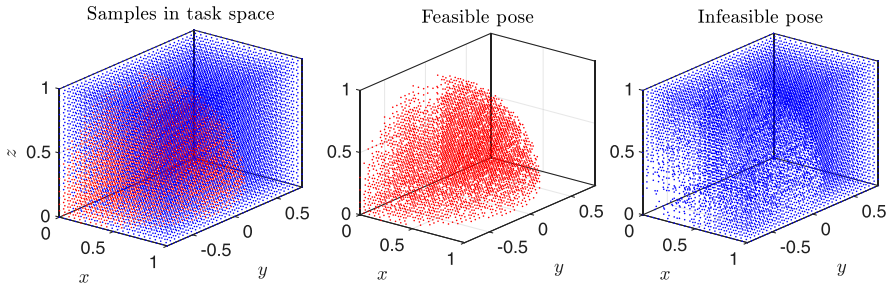

Figure 5. Samples in the workspace for 6-DOF PUMA 560 manipulator. We sampled $21 \times 21 \times 33 = 14553$ points with 5 *cm* intervals in the workspace of size $1m \times 1m \times 1.6\,m$. The 4712 feasible poses are denoted as red, while the 9841 infeasible poses are denoted as blue.

Table 1. The Average Performance of SDP Relaxation on PUMA 560 Manipulator. We Launch all the Tests From a Looser Relaxation Scheme and Move to a Tighter Scheme for Failed Cases. The Runtime is Based on Each Individual Relaxation Scheme, While the Other Indices are Accumulated From the Lowest Scheme. The Pose Error is Based on Cases Successfully Solved by MOSEK.

Relaxation order (κ)	2	2	3
Sparsity pattern	CS	CS + MD	CS
Average solution time (s)	7.8	918.7	804.8
Maximum solution time (s)	18.5	2117.9	1886.5
Average joint orientation error (deg)	$6.93e^{-5}$	$6.93e^{-5}$	$6.93e^{-5}$
Maximal joint orientation error (deg)	0.0231	0.0231	0.0231
Average joint position error (cm)	$1.45e^{-5}$	$1.45e^{-5}$	$1.45e^{-5}$
Maximal joint position error (cm)	0.0191	0.0191	0.0191
Percentage of infeasibility detection	97.57%	100.0%	100.0%
Percentage of convergent SDP (feasible poses)	81.35%	98.58%	98.60%
Overall successful rate (all poses)	92.32%	99.54%	99.55%

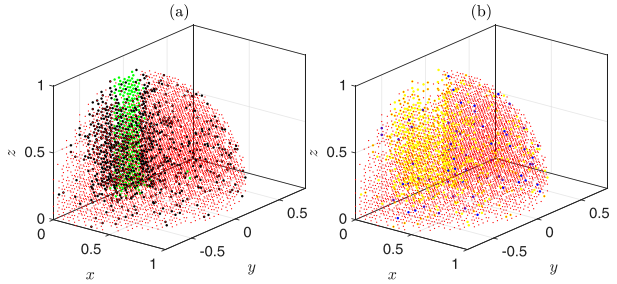

Figure 6. IK solution for 6-DOF PUMA 560 manipulator. (a) Applied the CS at $\kappa = 2$, the green dots denote the infeasible pose that the solver does not generate an infeasibility certificate, and the objective value is smaller than the threshold. The black dots denote the feasible poses that the solver does not converge and terminates as `SLOW_PROGRESS`. (b) Applied CS + MD at $\kappa = 2$ with chordal extension. All the green dots in (a) can be certified as infeasible using this relaxation scheme via the certificate or unreasonably large dual problem values. The yellow dots denote the feasible poses with convergent SDP solution. The blue dots denote the case that is still not convergent at this relaxation scheme. When moving to $\kappa = 3$, only one of the cases converges.

1000, which is sufficiently larger than the maximal possible objective of Problem 4.

The main results are summarized in Table 1 and illustrated in Figure 6. We first apply CS at $\kappa = 2$. As shown in Figure 6(a), we can see that 239 poses are infeasible but detected as feasible. We also have 879 feasible poses that the solver fails to converge and terminates as `SLOW_PROGRESS`. Then, we apply a slightly tighter relaxation, that is, CS + MD at $\kappa = 2$, to improve the result. As is in Figure 6(b), all the infeasible points can be certified, and 812 out of the 879 feasible cases converge. For the other 67 cases, we move to $\kappa = 3$ with CS, but only 1 of them converges. Tighter relaxation, such as CS + MD at $\kappa = 3$, will use up the memory.

For the cases successfully solved by MOSEK, the average rotation and position errors are negligible. For the failed cases that terminated as `SLOW_PROGRESS`, some cases have errors at a comparable level while some have position errors that can reach 1 *m*. Compared to the mix-integer programming-based method (Dai et al., 2019), the proposed algorithm has highly accurate solutions when the SDP converges to the optimum, as the kinematic model we use is exact. As (Dai et al., 2019) cannot avoid the inaccuracy due to the piece-wise approximation of SO(3) manifold, the position error can be centimeters. Both our method and (Trutman et al., 2022) are based on moment relaxation and suffer from similar numerical issues when solving the SDP, while the overall success rates are at the same level.

7.3. Comparison with existing IK methods

Now, we compare the complexity of the proposed methods with other global optimization-based IK algorithms. Problem 4 has the same structure and sparsity pattern as

Problem 3. Thus, we further tested the runtime for Problem 4 with DOF ranging from 6 to 40 with 50 randomly generated parameters R_k^c and p_k^c with CS at $\kappa = 2$ to evaluate the complexity of the algorithm. As is shown in Figure 7, the runtime grows linearly w.r.t the DOF, which is consistent with the complexity analysis presented in (Lasserre, 2006). In comparison, (Dai et al., 2019) has exponential computation time w.r.t the planning horizon due to the combinatorial formulation. As (Trutman et al., 2022) is a dense formulation, the number of variables and degrees grows simultaneously, which results in moment matrices with computationally intractable size for long-horizon tasks. We summarized these global optimization-based IK algorithms in Table 2 to compare their formulation and complexity.

Here, we take the dense formulation (Amice et al., 2023) in the last column as the example. Consider the number of variables to represent each DOF as n and the DOF as N. By the rational function formulation, the order of each pose change X_i is 2. Thus, $\kappa \geq \lceil N/2 \rceil$ is required to contain all the monomials involved in the kinematic chain. As both the number of variables and relaxation order κ are functions of DOF, we note that the width of the moment matrix can be bounded by

$$\begin{aligned}
&\frac{\left(Nn + \left\lceil \frac{N}{2} \right\rceil\right)!}{(Nn)!\left\lceil \frac{N}{2} \right\rceil!} = \frac{Nn+1}{1} \cdot \frac{Nn+2}{2} \cdots \frac{Nn + \left\lceil \frac{N}{2} \right\rceil}{\left\lceil \frac{N}{2} \right\rceil} \\
&\geq \left(\frac{Nn + \left\lceil \frac{N}{2} \right\rceil}{\left\lceil \frac{N}{2} \right\rceil}\right)^{\left\lceil \frac{N}{2} \right\rceil} \geq \left(\frac{\overline{N}n + \left\lceil \frac{N}{2} \right\rceil}{\left\lceil \frac{\overline{N}}{2} \right\rceil}\right)^{\left\lceil \frac{N}{2} \right\rceil} \\
&= \left(\frac{\overline{N}n + \frac{\overline{N}}{2}}{\frac{\overline{N}}{2}}\right)^{\left\lceil \frac{N}{2} \right\rceil} = (2n+1)^{\left\lceil \frac{N}{2} \right\rceil}
\end{aligned} \tag{52}$$

where $\overline{N}$ is the smallest even number that $\overline{N} \geq N$. We can see that this formulation grows faster than the exponential function w.r.t the DOF N. We note that though (Marić et al., 2020) has linear complexity, its tightness relies on a randomly generated reference point to guarantee the optimal solution lies in the tightness region (Cifuentes et al., 2020).

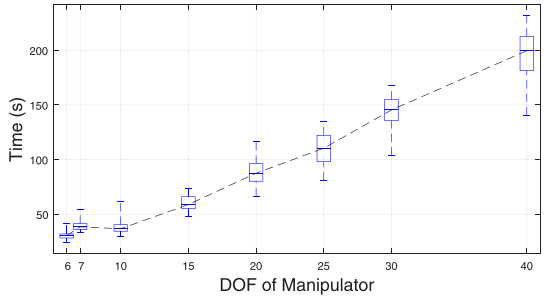

Figure 7. The runtime for the IK problem with DOF ranging from 6 to 40 tested on 50 randomly generated manipulators. We find that the time complexity is linear w.r.t the DOF if CS is applied. The median value is denoted as connected by the black line.

8. Motion planning of rigid body systems

In this section, we apply the proposed algorithms to the motion planning problem of rigid body systems. Then, we verify the dynamic feasibility of the planned motion via physics-based simulation in the Appendix C.

Table 2. Comparison of Global Optimization-Based IK Algorithm. The Complexity is w.r.t the Number of Joints of the Serial Manipulator. Consider the Number of Variables Needed for Each Joint as n and the Number of Joints as N. For the SDP-Based Method, We Evaluate the Space Complexity. For the Mixed-Integer Programming, We Consider the Worst-Case Time Complexity, Which is Exponential w.r.t the Number of Binary Variables.

Method	Parameterization of kinematic chain	Complexity
Proposed	Sparse kinematic chain (48)	$\mathcal{O}(Nn^{\kappa})$
Mixed-Integer Programming (Dai et al., 2019)	SO(3) as piece-wise linear function	$\mathcal{O}(c^{Nn})$
Sum-of-Squares (Marić et al., 2020)	Sparse kinematic chain, distance geometry Given key points on the joints x_i Position constraints: $g_i(x) = \|x_i - x_{i-1}\|^2 - l_i^2 = 0$ Angle constraints: $h_i(y) = 1/l_i^2 \|x_i - x_{i-1}\|^2 \leq 2(1 - \cos\alpha_i)$	$\mathcal{O}(Nn)$ Relies on sampled references
Moment Relaxation (Trutman et al., 2022)	Dense kinematic chain, $X_i(\theta_i) \in \mathrm{SE}(3)$ $\cos(\theta_i) = c_i, \quad \sin(\theta_i) = s_i$ $\prod_{i=1}^{\lceil N/2 \rceil} X_i(\theta_i) - X_g \prod_{i=n}^{N-\lceil N/2 \rceil} X_i^{-1}(\theta_i) = 0$	$\mathcal{O}\left(\left(Nn + \left\lceil \frac{1}{2}\left\lceil \frac{N}{2} \right\rceil \right\rceil\right)!/(Nn)!\left\lceil \frac{1}{2}\left\lceil \frac{N}{2} \right\rceil \right\rceil!\right)$
Sum-of-Squares (Amice et al., 2023)	Dense kinematic chain, $X_i(\theta_i) \in \mathrm{SE}(3)$ Rational function: $\cos(\theta_i) = 1 - s_i^2/1 + s_i^2, \quad \sin(\theta_i) = 2s_i/1 + s_i^2$ $\prod_{i=1}^{\lceil N/2 \rceil} X_i(\theta_i) - X_g \prod_{i=n}^{N-\lceil N/2 \rceil} X_i^{-1}(\theta_i) = 0$	$\mathcal{O}\left(\left(Nn + \left\lceil \frac{N}{2} \right\rceil\right)!/(Nn)!\left\lceil \frac{N}{2} \right\rceil!\right)$

Table 3. Drone Landing Parameters.

Simulation constants	Values	Control parameters	Values
Mass - m	0.5 (kg)	P	[100, 10, 100, 100]
Inertial - I_b	diag (0.3, 0.2, 0.3) (kg/m^2)	Q	[0.1, 10, 0.1, 1]
Gravity - g	$[0, 0, -9.81]^{\mathrm{T}}$ (m/s^2)	U	[0.1, 0.1]
Time step - Δt	$0.1667s$ / $0.125s$	$[\tau_{\min}, \tau_{\max}]$	$[-5, 5]$
Planning horizon - N	30 / 40	$[f_{z,\min}, f_{z,\max}]$	$[-\infty, \infty]$
Obstacle 1	$x^2 + (y - 0.5)^2 \geq 0.25$	p_0, v_0, F_0	$[1, 1, 3]^{\mathrm{T}}, 0_{3\times1}, I_3$
Obstacle 2	$(x - 0.6)^2 + (y - 0.5)^2 \geq 0.25$	$R_0 = R_y(\theta)$	$\theta = 0°, 60°, 90°, 120°, 180°$
Height constraints	$z \geq 0$		

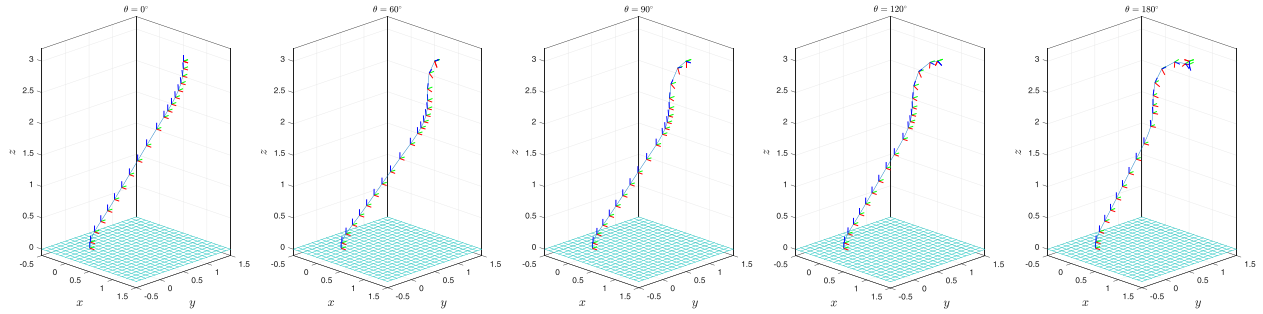

Figure 8. 3D drone landing task in free space. The drone starts at an initial position $x = 1, y = 1, z = 3$ with zero twists and different initial pitch angles. The robot is guided to land at the origin. The blue, red, and green axes indicate the z, x, and y axes in the body frame. For a large initial pitch angle, the drone adjusts the orientation in the first few steps and then moves to the origin. Only part of the waypoints in the tail of trajectories are presented to avoid overlap. The statistics of these cases are presented in Table 4.

8.1. 3D drone landing

In this task, we consider landing a drone while avoiding obstacles from different initial poses. We assume that the drone is controlled by torque $\tau \in \mathbb{R}^3$ and total thrust force $f_z \in \mathbb{R}$ along the z-axis, both in the body frame. Thus, we have the kinematics model (24) and the forced, rigid body dynamics:

$$\begin{aligned} F_{k+1} I^b - I^b F_{k+1}^{\mathrm{T}} &= I^b F_k - F_k^{\mathrm{T}} I^b + \Delta t^2 \tau_{k+1}^{\times}, \\ m v_{k+1} &= m F_k^{\top} v_k + \Delta t (e_z f_{z,k+1} + m R_{k+1}^T g), \end{aligned} \tag{53}$$

with $e_z = [0, 0, 1]^{\mathrm{T}}$. We then define the quadratic cost:

$$\begin{aligned} \Phi(Y_N) = P_1 \|R_N - I\|_{F,I}^2 + P_2 \|F_N - I\|_{F,I}^2 \\ + P_3 \|p_N\|^2 + P_4 \|v_N\|^2, \end{aligned} \tag{54}$$

$$\begin{aligned} L(Y_k, u_{k+1}) = Q_1 \|R_k - I\|_{F,I}^2 + Q_2 \|F_k - I\|_{F,I}^2 \\ + Q_3 \|p_k\|^2 + Q_4 \|v_k\|^2 + U_1 \|\tau_{k+1}\|^2 + U_2 \|f_{z,k+1}\|^2, \end{aligned} \tag{55}$$

where $\|X\|_{F,P} = \sqrt{\mathrm{tr}(X^{\mathrm{T}} P X)}, P \geq 0$, is the weighted Frobenius norm. To show the capabilities of the proposed algorithm in handling obstacle avoidance constraints, we consider one quadratic constraint in each case:

$$\frac{(x - x_c)^2}{r_x^2} + \frac{(y - y_c)^2}{r_y^2} \geq 1. \tag{56}$$

The system parameters, cost functions, and obstacle configurations are shown in Table 3.

We set the drone at an initial position with different pitch angles. We launch the trajectory optimization with one of the obstacles or in free space. Obstacle 2 blocks more waypoints of the trajectory planning in the free space than obstacle 1, which is more challenging. The whole planning horizon is 5 s with 40 steps. For the CS cases without TS, the SDP has too many variables that used up the memory, so we reduced the step number to 30. We start with TS + MD and continue to TS + block or CS if ϵ is large for a former relaxation scheme. Empirically, we find that $\epsilon \leq 1e^{-3}$ results in tight moment relaxation that also guarantees the feasibility of the solution. Thus, we set $\epsilon \leq 1e^{-3}$ as a threshold to decide whether to continue to a tighter relaxation scheme.

The planned trajectories of all cases after refinement by IPOPT are illustrated in Figures 8–10. The statistics of the planned trajectories are presented in Tables 4–6. For all tests, TS + MD does not provide an optimality gap smaller than 0.01. For simple cases in free space or with obstacle 1, TS + block can solve all the cases with $\theta \leq 90°$. For $\theta = 120°$, moving to CS also generates certifiable optimality values. While for $\theta = 180°$, the CS does not provide certifiable optimal values. For hard cases with obstacle 2, more numerical issues are presented. Even the CS sparsity pattern does not provide certifiable solutions for $\theta = 60°$ and $\theta = 180°$.

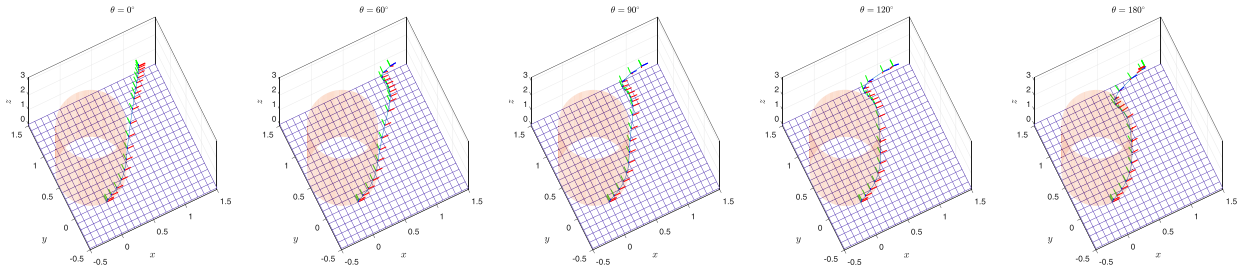

Figure 9. 3D drone landing task with obstacle 1. Obstacle 1 blocks some of the waypoints computed from the free space task. All these plots converge to a locally feasible path after refinement by IPOPT, and the first 4 cases are certified as globally optimal solutions by metric ϵ. The statistics of these cases are presented in Table 5.

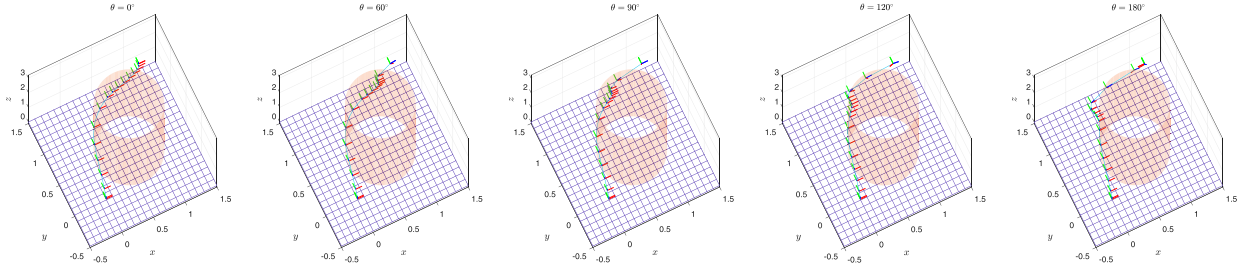

Figure 10. 3D drone landing task with obstacle 2. Obstacle 2 blocks more waypoints in the free space task than obstacle 1, which makes the SDP harder to converge. All these cases converge to locally feasible solutions after refinement by IPOPT, and the first 4 cases have very small optimality gaps. The statistics of these cases are presented in Table 6.

Table 4. Statistics of Drone Landing in Free Space. TS + block Can Solve all Cases With Pitch Angles Smaller Than 120°. For the Harder Cases, CS is Needed to Induce Tighter Relaxation. These Cases are Illustrated in Figure 8.

$R_{y,0}$	0°			60°			90°			120°			180°		
Sparsity	TS + MD	TS + block	CS	TS + MD	TS + block	CS	TS + MD	TS + block	CS	TS + MD	TS + block	CS	TS + MD	TS + block	CS
Suboptimality ϵ	0.0101	$\leq 1e^{-4}$	N/A	0.0235	$\leq 1e^{-4}$	N/A	0.0589	$\leq 1e^{-4}$	N/A	0.3767	0.0046	$\leq 1e^{-4}$	0.4102	0.2054	0.0136
Rank condition δ	0.0968	$\leq 1e^{-4}$	N/A	0.1702	$\leq 1e^{-4}$	N/A	0.4918	$6.302e^{-4}$	N/A	0.8301	0.5824	$\leq 1e^{-4}$	0.9049	0.9653	0.6504
Runtime (s)	397.1	2299.6	N/A	431.4	2185.9	N/A	381.3	2298.5	N/A	373.6	2868.3	6662.0	346.7	3646.5	6820.4

Table 5. Statistics of Drone Landing With Obstacle 1. All Cases Can Be Tightly Solved Except the Last One. These Cases are Illustrated in Figure 9.

$R_{y,0}$	0°			60°			90°			120°			180°		
Sparsity	TS + MD	TS + block	CS	TS + MD	TS + block	CS	TS + MD	TS + block	CS	TS + MD	TS + block	CS	TS + MD	TS + block	CS
Suboptimality ϵ	0.0130	$\leq 1e^{-4}$	N/A	0.0252	$\leq 1e^{-4}$	N/A	0.0608	$\leq 1e^{-4}$	N/A	0.3772	0.0070	$\leq 1e^{-4}$	0.3810	0.2083	0.0205
Rank condition δ	0.0965	$\leq 1e^{-4}$	N/A	0.1696	$\leq 1e^{-4}$	N/A	0.4805	$4.631e^{-4}$	N/A	0.8240	0.5687	$\leq 1e^{-4}$	0.9194	0.9661	0.7051
Runtime (s)	424.0	2598.5	N/A	361.6	2549.9	N/A	426.7	3246.1	N/A	433.5	2863.2	7039.4	409.2	3834.5	8641.7

Table 6. Statistics of Drone Landing With Obstacle 2. Due to the configurations of the Obstacle, This Task is More Challenging, but the Overall Optimality Gap Using CS is Small Except for the 180° Case. These Cases are Illustrated in Figure 10.

$R_{y,0}$	0°			60°			90°			120°			180°		
Sparsity	TS + MD	TS + block	CS	TS + MD	TS + block	CS	TS + MD	TS + block	CS	TS + MD	TS + block	CS	TS + MD	TS + block	CS
Suboptimality ϵ	0.0499	0.0275	$7.731e^{-4}$	0.0580	0.0307	0.0035	0.0802	0.0473	$\leq 1e^{-4}$	0.3846	0.0154	$\leq 1e^{-4}$	0.4231	0.2127	0.0167
Rank condition δ	0.1013	0.0738	0.0369	0.1776	0.0661	0.1207	0.5019	0.1849	0.0016	0.8348	0.6049	$\leq 1e^{-4}$	0.9088	0.9632	0.6712
Runtime (s)	391.1	3295.4	8314.3	360.5	3111.7	9310.6	424.2	3471.0	7241.5	416.2	3350.2	7700.8	429.1	3686.7	8180.6

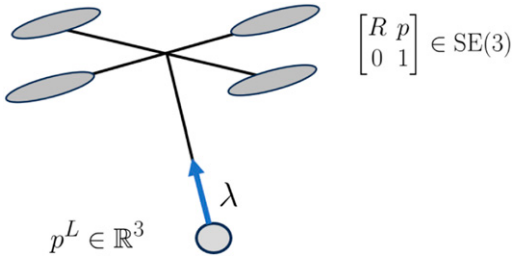

Figure 11. The drone with cable-suspended load. The configuration space of the drone is SE (3), and the load is $p^L \in \mathbb{R}^3$. The cable length is l, and the constrained force is λ.

As the solver returned `SLOW_PROGRESS` for these failed cases, we do not know if the SDP relaxation itself is tight at this order for these initial conditions. However, the solution is still a good initial guess for local search by IPOPT. We can also find that the magnitude of ϵ and δ are closely correlated. When $\epsilon \leq 1e^{-4}$, the rank-one condition will likely satisfy as δ is also small.

8.2. 3D drone landing with cable-suspended load

Now, we extend the algorithms to multi-body dynamical systems. We leverage the constrained LGVI to formulate the dynamics of the drone with a cable-suspended load. The configuration space of the drone is chosen as SE (3), and the load position is described by $p^L \in \mathbb{R}^3$. Thus, we can choose the configuration space as $Q = \mathrm{SE}(3) \times \mathbb{R}^3$. The constrained force $\lambda \in \mathbb{R}$ is also considered to enforce the cable length $l = 0.5\ m$. The drone-load system is illustrated in Figure 11.

To avoid the hybrid dynamics of the cable being taut or not, we enforce the norm constraints and that the multiplier is larger than a small positive number e:

$$g(q) := \frac{1}{2}\left(l^2 - \|p^L - p\|^2\right) = 0, \quad \lambda \geq e > 0. \tag{57}$$

Consider the variation of p on SE (3) as $\delta p := R\delta\rho$ defined in (28), the gradient of g w.r.t the position of the drone and load can be derived as

$$\frac{\partial g}{\partial \rho} = R^{\mathrm{T}}(p_L - p), \quad \frac{\partial g}{\partial p^L} = p - p^L. \tag{58}$$

With e_z denoting the direction of the z-axis in the body frame, we have the constrained dynamics for the position of the drone and the load:

$$\begin{aligned} mv_{k+1} &= mF_k^{\top} v_k + (e_z f_{z,k+1} + R_{k+1}^T mg)\Delta t \\ &\quad + R_{k+1}^T (p_{k+1}^L - p_{k+1})\lambda_{k+1}\Delta t, \\ m^L v_{k+1}^L &= m^L v_k^L + m^L g\Delta t + (p_{k+1} - p_{k+1}^L)\lambda_{k+1}\Delta t, \end{aligned} \tag{59}$$

as well as the quadratic cable length constraint:

$$\|p_{k+1}^L - p_{k+1}\|^2 - l^2 = 0, \lambda_{k+1} \geq e > 0. \tag{60}$$

We consider the tasks to avoid obstacle 2 for both the load and the drone with 40 steps and a time interval of 0.125 s. The drone is initialized with an identity orientation and zero twists at the same initial position, with the load beneath the drone. We assume that $m = m^L$ for our case. We modify the cost function to enable the drone to dock at $z = 0.5\ m$ over the origin while the load is supposed to land at the origin. We set the initial velocity of the load to $v_y^L = -1, 0, 2\ m/s$. The planned trajectories are presented in Figures 12–14. Due to the numerical inaccuracy, the rank-one conditions are not satisfied, while the optimality gaps are 0.2562, 0.1590, and 0.1548, respectively. The computational time using CS only and the computational time are 10540s, 8397.3s, and 9327.6s for the three cases.

8.3. Under-actuated cart-pole

We now launch trajectory optimization for the under-actuated cart-pole to test the algorithms on full rigid body dynamics. The cart-pole system is illustrated in Figure 15. As the planar dynamics, we reduce the DOF of the 3D rigid body via restricting the pose R_k and change F_k to the SO(2) subgroup:

$$R_k = \begin{bmatrix} c_k & -s_k & 0 \\ s_k & c_k & 0 \\ 0 & 0 & 1 \end{bmatrix}, F_k = \begin{bmatrix} c_{v,k} & -s_{v,k} & 0 \\ s_{v,k} & c_{v,k} & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{61}$$

Thus for the rigid body on SE(2), we have the kinematic EOM as

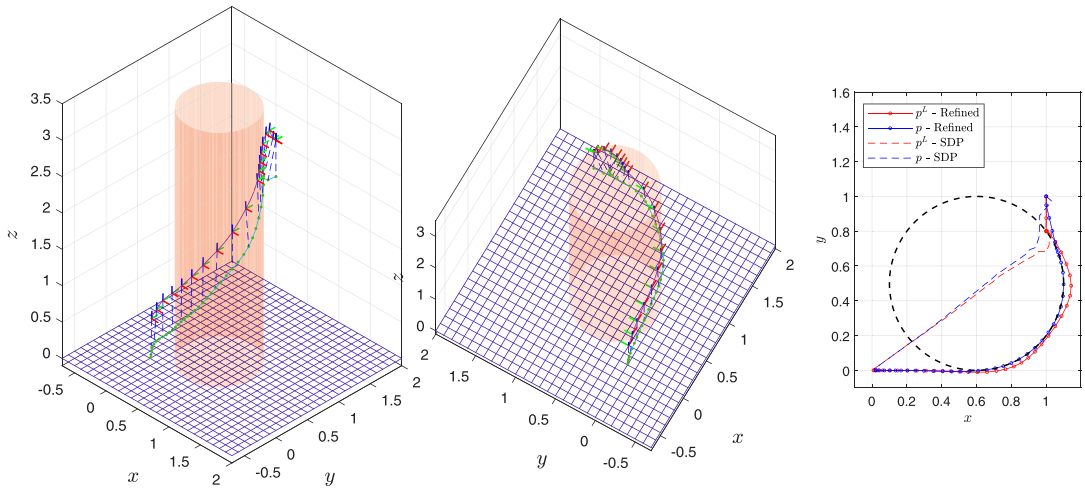

Figure 12. Drone with cable-suspended load. The initial velocity of the load is $-1m \cdot s^{-1}$ in y direction. The relative suboptimality is $\epsilon = 0.2562$. Due to the suboptimality and the numerical issues, the SDP solution does not guarantee the trajectory is collision-free.

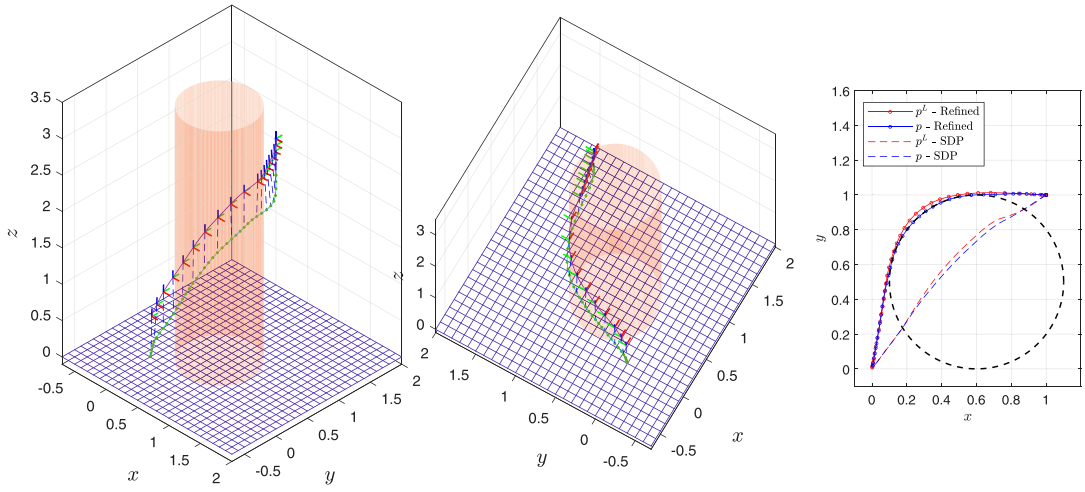

Figure 13. Drone with cable-suspended load. The load has zero velocity at $t = 0$. The relative suboptimality is $\epsilon = 0.1590$.

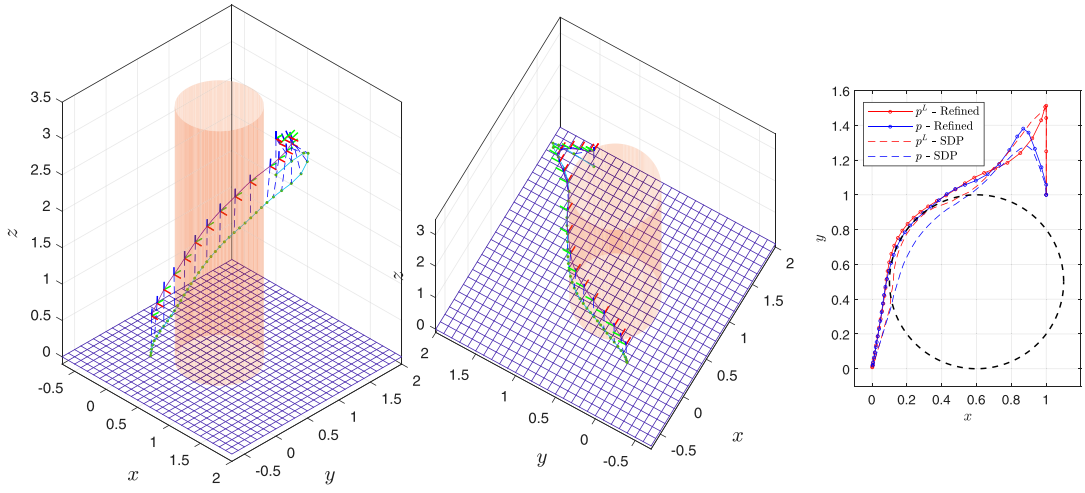

Figure 14. Drone with cable-suspended load. The initial velocity of the load is $2m \cdot s^{-1}$ in y direction. The relative suboptimality is $\epsilon = 0.1548$. In this case, the SDP solution is of good quality and close to the refined solution.

$$\begin{aligned} c_{k+1} &= c_k c_{v,k} - s_k s_{v,k} \\ s_{k+1} &= s_k c_{v,k} + c_k s_{v,k} \\ x_{k+1} &= x_k + \Delta t(c_k v_{x,k} - s_k v_{y,k}) \\ y_{k+1} &= y_k + \Delta t(s_k v_{x,k} + c_k v_{y,k}). \end{aligned} \tag{62}$$

Thus for the rigid body on SE(2), we have the dynamics EOM:

$$\begin{aligned} I^b s_{v,k+1} &= I^b s_{v,k} + \tau_k \Delta t^2 \\ m v_{x,k+1} &= m(c_k v_{x,k} + s_k v_{y,k}) + s_{k+1} mg\Delta t + f_{x,k}\Delta t \\ m v_{y,k+1} &= m(-s_k v_{x,k} + c_k v_{y,k}) + c_{k+1} mg\Delta t + f_{y,k}\Delta t \end{aligned} \tag{63}$$

For the cylindrical joint, we have linear constraints to ensure the two rigid bodies are connected at one point:

$$\begin{bmatrix} R_a & p_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} l_a \\ 1 \end{bmatrix} - \begin{bmatrix} R_b & p_b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} l_b \\ 1 \end{bmatrix} = 0 \tag{64}$$

where l_a and l_b are two vectors attached to the body frame of the link. Thus, we can derive the Jacobian by considering the variation on Lie group. For one link, we have

$$\begin{aligned} & \begin{bmatrix} R_a & p_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta\omega_a^\times & \delta\rho_a \\ 0 & 0 \end{bmatrix} \begin{bmatrix} l_a \\ 1 \end{bmatrix} \\ = & \begin{bmatrix} R_a & p_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \delta\omega_a^\times l_a + \delta\rho_a \\ 0 \end{bmatrix} \\ = & \begin{bmatrix} R_a \delta\omega_a^\times l_a + R_a \delta\rho_a \\ 0 \end{bmatrix} \\ = & \begin{bmatrix} -R_a l_a^\times \delta\omega_a + R_a \delta\rho_a \\ 0 \end{bmatrix} \end{aligned} . \tag{65}$$

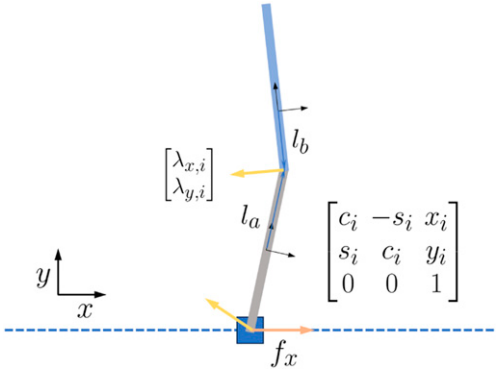

Figure 15. Illustration of the cart-pole systems. The pose of each cart-pole is represented by SE (2). The actuation force is linear at the base link. The multiplier λ is modeled in the world frame to enforce the kinematic constraints. The base link is represented by its x position only as the DOF of y position, and the rotation is subject to the holonomic constraints.

Thus, we have the Jacobians:

$$G_d = [-R_a l_a^\times, R_a, R_b l_b^\times, -R_b]\Delta t. \tag{66}$$

Finally, we present the swing-up trajectory of the under-actuated cart-pole. We optimize an 80-step trajectory with a time interval of $0.125s$ for different initial conditions. We only consider the force f_x along the x axis of the base link as the input. To further reduce the computational burden, we only represent the cart as a single mass due to the holonomic of rotation and y position on the cart. The planned trajectories refined by IPOPT are presented in Figure 16. We tested the initial pole angle being 0.25π, 0.5π, 0.75π, and 0.95π with cart location at $x = 0$. The relaxation gaps for the four cases are 0.0001, 0.054, 0.189, and 0.303, respectively. In all the cases, the sparsity patterns need to be CS + MD at $\kappa = 2$ for a reasonable solution. The computational time is 4378.9, 3223.3s, 2765.2s, and 2779.8s, respectively.

We further tested the algorithms for cart-pole with two arms. Unfortunately, it takes more than 15 h to obtain a trajectory with 20 steps, which is not sufficiently accurate to serve as a good initial guess. The case with the initial angle being 45° is presented in Figure 17. We can find that the solution returned by SDP provides a swing-up trajectory that looks reasonable, while the IPOPT fails to provide a refined solution that achieves similar performance. The reason for the long computational time can be explained by the size of the moment matrix shown in (46). For this 2D planning problem, every single rigid body requires 8 variables, that is, $\{c, s, x, y, c_v, s_v, v_c, v_y\}$, to represent each DOF and 2 variables, that is, $\{\lambda_x, \lambda_y\}$, to represent the constrained forces. Considering the cart with 2 DOF, the total number of variables of the two-arm cart-pole is $n_2 = 2 + 8 + 8$and $n_1 = 2 + 8$ for a one-arm cart-pole. For the number of inputs and constrained forces, we have $m_1 = 1 + 2$ and $m_2 = 1 + 2 + 2$. Given $\kappa = 2$, we have an approximation of the ratio of the size of the two moment matrices by (44), given $\kappa = 2$:

$$\begin{aligned} \frac{Nd(n_2, m_2, 1, 2)}{Nd(n_1, m_1, 1, 2)} &= \frac{\prod_{k=1}^{2}(n_2 + m_2 + k)}{\prod_{k=1}^{2}(n_1 + m_1 + k)} \\ &= \frac{\prod_{k=1}^{2}(18 + 5 + k)}{\prod_{k=1}^{2}(10 + 3 + k)} \\ &\approx 2.86 \end{aligned} \tag{67}$$

Considering the number of entries of the entire moment matrix, the memory required for a two-arm cart-pole is roughly 8 times more than the one-arm case.

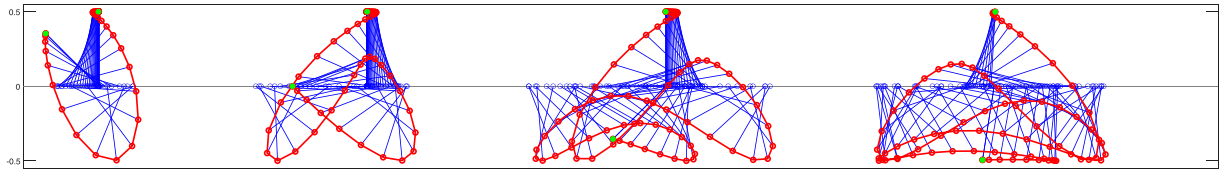

Figure 16. The refined trajectories of the cart-poles with different initial angles: 0.25π, 0.5π, 0.75π, and 0.95π.

8.4. Comparison with local search methods

We further compare the quality of solutions by (SDP) with local gradient-based solvers in the drone landing and cart-pole swing-up problem. We apply IPOPT directly to (POP) using initial guesses with different qualities. As the relaxation gaps of most cases in the drone landing problem studied in Section 8.1 are small as shown from Table 4 to Table 6, we randomly perturb the best-refined solution as initial guesses and compare the quality of the perturbed solutions.

For the orientation R_k and pose change F_k, we perturb it by rotating with a random angle as $\widetilde{R}_k = R_k \exp(\zeta)$, where $\zeta \sim N(0_{3,1}, \Delta^2 I_3)$ is white Gaussian noise. We perform element-wise random perturbation for other vector variables, such as $\widetilde{v}_k = v_k + \zeta$. Then, we assign different Δ to represent the quality of the initial guess. We mainly evaluate the refined solutions by IPOPT via the suboptimality compared with the best-refined solutions (SDP) and the convergent status of IPOPT. For each Δ, we sample 10 times for all the 5 initial conditions and tasks. The result is presented in Table 7. We can see that the local solvers highly rely on the initial guesses, and the random perturbed solutions are no better than the certified solutions. With increasing noise, the solutions by IPOPT become suboptimal or even infeasible.

A similar comparative study is also conducted on cart-pole swing-up cases with different noise levels. We use the perturbed refined solution as our initial guesses and compute the relaxation gap ϵ. As ϵ for the study cases can be larger than 0.3, we further compare the initial guesses by (SDP) with other heuristic-based initializations. We consider the straight line in the joint space to generate the initial trajectory (Schulman et al., 2014). We then perturb the trajectory with the same Gaussian noise and then compare the relaxation gaps. The statistics of relaxation gaps are summarized in Figure 18. We find that the case with initial pole angle $\theta_0 = 0.25\pi$, 0.5π, and 0.75π, the (SDP)-based

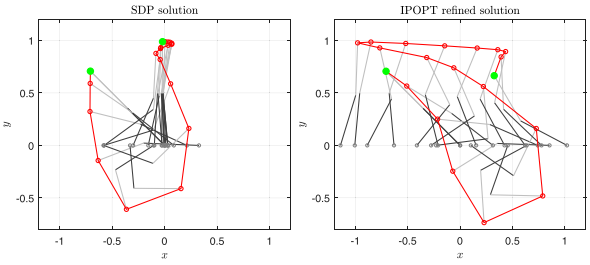

Figure 17. The trajectory of the cart-pole with two arms. The refined trajectory deviates a lot from the SDP solutions as its quality is not sufficient to serve as a good initial guess.

Table 7. Statistics of drone landing task solved by IPOPT. We perturb the best optimality guaranteed SDP solution with noise of different levels as the initial guess. The average suboptimality is evaluated on each task with 5 initial pitch angles. For each test case, we sample the initial conditions 10 times. Thus, the average suboptimality is evaluated by finding feasible solutions among the 50 random samples. We show that with a small noise level, the IPOPT converges to the best-refined solutions. However, as the noise level increases, the optimality gap grows significantly, and even local feasibility is not guaranteed. We set the computational budget as 1e4 iterations.

Initialization noise Δ	0.001			0.01			0.1			0.5			1		
Task	Free	Obs-1	Obs-2	Free	Obs-1	Obs-2	Free	Obs-1	Obs-2	Free	Obs-1	Obs-2	Free	Obs-1	Obs-2
Average suboptimality ϵ	$\leq 1e^{-4}$	$\leq 1e^{-4}$	$\leq 1e^{-4}$	$\leq 1e^{-4}$	$\leq 1e^{-4}$	$\leq 1e^{-4}$	0.0115	0.0134	$\leq 1e^{-4}$	0.3644	0.4757	0.2406	0.7727	0.7543	0.5652
Timeout (out of 50)	0	0	0	0	0	0	0	0	0	12	8	2	25	15	6
Infeasibility (out of 50)	0	0	0	0	0	0	0	0	0	13	11	13	16	23	13

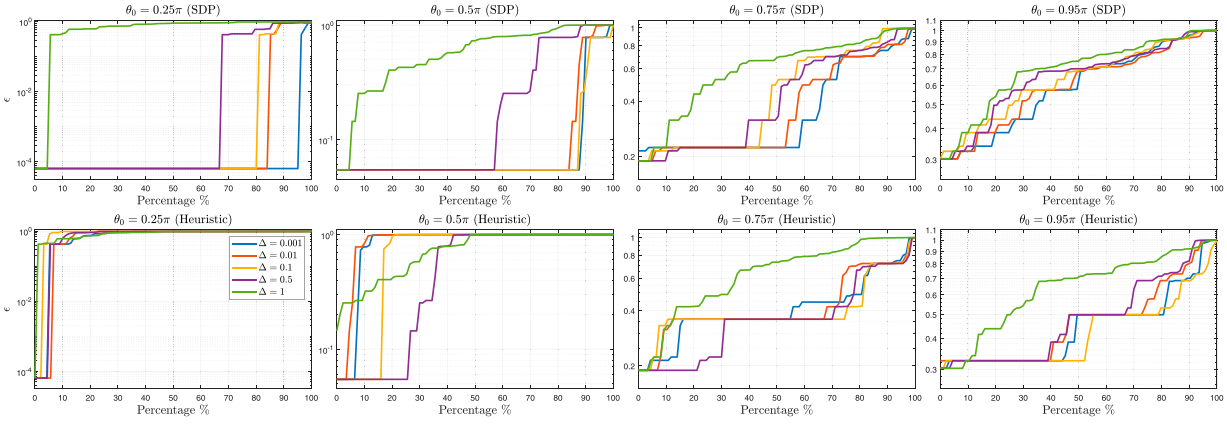

Figure 18. The suboptimality of solutions by IPOPT with perturbed initializations in different qualities. We plot the percentage of the cases with the suboptimality smaller than ϵ. For (SDP), the ϵ are statistically smaller than heuristic-based initializations when $\theta_0 = 0.25\pi$, 0.5π and 0.75π, for all the corresponding level of noise perturbations. As the relaxation gap is large when $\theta_0 = 0.95\pi$, the heuristic-based solution statistically outperforms the (SDP). However, the suboptimality metric ϵ is only accessible when solving (SDP).

initial guesses statistically provide better solutions. Though for $\theta_0 = 0.95\pi$, the perturbed straight-line initialization performs statistically better, we note that the evaluation metric ϵ is only accessible by solving (SDP). For the case with tight relaxation, the (SDP) based initializations are guaranteed to provide a good warm start to the solver and nontrivial lower bounds.

9. Alternative formulations

In this section, we extend the proposed method to alternative variational integrators on $\mathrm{SO}(3)\times\mathbb{R}^3$ and quaternion to speed up the computation. We compare the integrators by the tightness of the relaxation and the computation time in the drone landing problems in Section 8.1 and Section 8.2.

9.1. LGVI on $\mathrm{SO}(3)\times\mathbb{R}^3$

The proposed algorithm has been applied to LGVI on SE(3), where the linear velocity is derived in the body frame and thus coupled with the rotational motion. Here, we derive the alternative LGVI on $\mathrm{SO}(3)\times\mathbb{R}^3$ to decouple the linear and rotational motion.

Consider the configuration state of a single rigid body as the orientation $R\in\mathrm{SO}(3)$ and position $p\in\mathbb{R}^3$. The twists are the angular velocity $\omega^\times\in\mathfrak{so}(3)$ and the velocity in the world frame $u\in\mathbb{R}^3$. Thus, the unforced EOM of the single rigid body is

$$\begin{aligned}\dot{R}=R\omega^\times,\dot{p}=u,\\ I_b\dot{\omega}=-\omega^\times I_b\omega,\dot{u}=0.\end{aligned}\tag{68}$$

Then we follow the procedure in Section 3.3 to derive the LGVI on $\mathrm{SO}(3)\times\mathbb{R}^3$. As the rotational motion is identical to the SE(3) case, we only derive the linear part here. Consider the kinetic energy of the linear motion:

$$T_p(u)=\frac{1}{2}mu^{\mathrm{T}}u,\tag{69}$$

and the potential energy:

$$V(p)=mpg.\tag{70}$$

Given the midpoint approximation of the linear velocity as

$$u_k=\frac{p_{k+1}-p_k}{\Delta t},\tag{71}$$

we can approximate the discrete kinetic and potential energy as

$$T_{p,d}=\frac{1}{2\Delta t}m\|p_{k+1}-p_k\|^2,V_d=m\left(\frac{p_{k+1}+p_k}{2}\right)^{\mathrm{T}}g\Delta t.\tag{72}$$

The variation on p, for example, δp, is taken in Euclidean space, which is not coupled with the rotational motion. Thus, we have the derivative of the kinetic energy:

$$\begin{aligned}D_1T_{d,p}=m\frac{(p_k-p_{k+1})^{\mathrm{T}}}{\Delta t}\delta p_k,\\ D_2T_{d,p}=m\frac{(p_k-p_{k-1})^{\mathrm{T}}}{\Delta t}\delta p_k,\end{aligned}\tag{73}$$

and the potential energy as

$$D_1V_d=D_2V_d=\frac{mg^{\mathrm{T}}\Delta t}{2}\delta p_k.\tag{74}$$

Finally, we have the linear motion using $\mathrm{SO}(3)\times\mathbb{R}^3$ as

$$\begin{aligned}p_{k+1}=p_k+u_k\Delta t,\\ mu_{k+1}=mu_k+mg\Delta t.\end{aligned}\tag{75}$$

We notice that the linear velocity u in the world frame is no longer coupled with the pose change F.

For the drone system, the rotational dynamics are identical to the SE(3) case, while the translational dynamics become

$$mu_{k+1}=mu_k+mg\Delta t+R_{k+1}e_zf_{z,k+1}\Delta t.\tag{76}$$

For the case with cable-suspended load, the dynamics considering the constrained force becomes

$$\begin{aligned}mu_{k+1}=mu_k+mg\Delta t+R_{k+1}e_zf_{z,k+1}\Delta t\\ +(p_{k+1}-p^L_{k+1})\lambda_{k+1}\Delta t\end{aligned}.\tag{77}$$

9.2. Quaternion-based variational integrator

The quaternion-based variational integrator has been proposed by Manchester and Peck (2016) and applied to the simulation of multi-rigid body systems (Brüdigam and Manchester, 2021b). Compared with the rotational matrix, the quaternion has fewer variables and thus can potentially lower the computational burden.

We briefly review the integrator by Manchester and Peck (2016) and convert it to the polynomial constraints. Consider the quaternion as a stack of the variables:

$$q=\begin{bmatrix}q_s\\ q_{v_1}\\ q_{v_2}\\ q_{v_3}\end{bmatrix}=:\begin{bmatrix}q_s\\ q_v\end{bmatrix},\tag{78}$$

where $q_s\in\mathbb{R}$ is the scalar and $q_v\in\mathbb{R}^3$ is the vector part. The rotational motion can be obtained via the matrix-vector product:

$$q_1\cdot q_2=L(q_1)q_2,\tag{79}$$

with

$$L(q)=\begin{bmatrix}q_s & -q_v^T\\ q_v & q_sI_3+q_v^\times\end{bmatrix}.\tag{80}$$

Thus, the pose change can be parameterized by

$$\omega_k = \begin{bmatrix} \omega_{s,k} \\ \omega_{v,k} \end{bmatrix} = L(q_k)^{\mathrm{T}} q_{k+1}, \tag{81}$$

where ω_k is the discrete quaternion angular velocity. We have the rotational motion as follows:

$$\begin{aligned} &q_{k+1} = L(q_k)\omega_k, \\ &I_b\omega_{v,k+1}\omega_{s,k+1} + \omega_{v,k+1}^{\times} I_b\omega_{v,k+1} = \\ &\quad I_b\omega_{v,k}\omega_{s,k} + \omega_{v,k}^{\times} I_b\omega_{v,k}. \end{aligned} \tag{82}$$

To have unit quaternion, we also enforce the norm constraints that

$$\|q\|_2^2 - 1 = \|\omega\|_2^2 - 1 = 0. \tag{83}$$

As the double cover property of quaternion results in non-unique representations of the same rotation, we consider enforcing the scalar part to be non-negative to ensure the uniqueness:

$$\omega_s \geq 0, q_s \geq 0. \tag{84}$$

To rotate a vector in the world frame, we have the rotation matrix:

$$Q(q) = VR(q)^{\mathrm{T}} L(q) V^{\mathrm{T}}, \tag{85}$$

with

$$R(q) = \begin{bmatrix} q_s & -q_v^{\mathrm{T}} \\ q_v & q_s I_3 - q_v^{\times} \end{bmatrix}, V = [\, 0_{3\times 1} \quad I_{3\times 3}\,]. \tag{86}$$

By (21), the forced rotational motion of the done can be derived as

$$\begin{aligned} &I_b\omega_{v,k+1}\omega_{s,k+1} + \omega_{v,k+1}^{\times} I_b\omega_{v,k+1} = \\ &I_b\omega_{v,k}\omega_{s,k} + \omega_{v,k}^{\times} I_b\omega_{v,k} + \frac{\Delta t^2}{2}\tau_{k+1}. \end{aligned} \tag{87}$$

The unforced linear motion is identical to the $\mathrm{SO}(3)\times\mathbb{R}^3$ case, which is not coupled with the rotation. Considering the quaternion rotation, we have the linear motion for the drone as follows:

$$mu_{k+1} = mu_k + mg\Delta t + Q(q_{k+1}) f_{z,k+1}\Delta t, \tag{88}$$

and for the drone with cable-suspended load as

$$\begin{aligned} mu_{k+1} = mu_k + mg\Delta t + Q(q_{k+1}) f_{z,k+1}\Delta t \\ +(p_{k+1} - p_{k+1}^L)\lambda_{k+1}\Delta t \end{aligned}. \tag{89}$$

To regularize the orientation and angular velocity to the desired value $\overline{q}$ and $\overline{\omega}$, we consider the quadratic costs:

$$\|q_k - \overline{q}_k\|_{Q_q}^2, \ \|\omega_k - \overline{\omega}_k\|_{Q_\omega}^2, \tag{90}$$

where Q_q and Q_ω are the weighting matrices.

9.3. Comparison of integrators

We now compare different integrators for rigid body systems and list the candidate Lie integrators in Table 8. For the explicit Euler and Runge-Kutta method, the exponential map is required to integrate the continuous motion to discrete time, which does not have an exact polynomial solution in finite degree (Iserles et al., 2000). For variational integrators, derivation on SE(3) has been studied by

Table 8. Comparison of Lie Integrators.

Method	Equations
Explicit Euler (Celledoni et al., 2014)	$y \in \mathcal{G}, \dot{y} = yf(y), y_{k+1} = y_k \exp(f(y_k)\Delta t)$
Runge-Kutta (Celledoni et al., 2014)	$y \in \mathcal{G}, \dot{y} = yf(y), f_{n,1} = f(y_n)\Delta t, f_{n,2} = f(\exp(\frac{1}{2}f_{n,1})\cdot y_n)\Delta t,$ $f_{n,3} = f(\exp(\frac{1}{2}f_{n,2} - \frac{1}{8}[f_{n,1}, f_{n,2}])\cdot y_n)\Delta t, f_{n,4} = f(\exp(f_{n,3})\cdot y_n)\Delta t,$ $y_{n+1} = \exp(\frac{1}{6}(f_{n,1} + 2f_{n,2} + 2f_{n,3} + f_{n,4} - \frac{1}{2}[f_{n,1}, f_{n,4}]))\cdot y_n.$
LGVI (Nordkvist and Sanyal, 2010)	$(I_b\omega_{v,k})^{\times}\Delta t = F_k I^b - I^b F_k^{\mathrm{T}},$ $R_{k+1} = R_k F_k, \quad p_{k+1} = R_k v_k \Delta t + p_k,$ $mv_{k+1} = F_k^{\mathrm{T}} mv_k + mR_{k+1} g\Delta t, I_b\omega_{v,k+1} = F_k^{\mathrm{T}} I_b\omega_{v,k}.$
LGVI (Lee et al., 2007)	$R_{k+1} = R_k F_k, \quad F_{k+1} I^b - I^b F_{k+1}^{\mathrm{T}} = I^b F_k - F_k^{\mathrm{T}} I^b,$ $p_{k+2} - 2p_{k+1} + p_k = \Delta t^2 mg.$
Variational integrator, quaternion (Manchester and Peck, 2016)	$q_{k+1} = L(q_k)\omega_k, \quad p_{k+1} = p_k + u_k\Delta t, \quad mu_{k+1} = mu_k + mg\Delta t$ $I_b\omega_{v,k+1}\omega_{s,k+1} + \omega_{v,k+1}^{\times} I_b\omega_{v,k+1} = I_b\omega_{v,k}\omega_{s,k} + \omega_{v,k}^{\times} I_b\omega_{v,k}, \omega_{s,k} \geq 0.$
Proposed, SE(3)	$R_{k+1} = R_k F_k, \quad p_{k+1} = R_k v_k \Delta t + p_k$ $F_{k+1} I^b - I^b F_{k+1}^{\mathrm{T}} = I^b F_k - F_k^{\mathrm{T}} I^b, \quad mv_{k+1} = mF_k^{\mathrm{T}} v_k + mR_{k+1}^{\mathrm{T}} g\Delta t.$
Proposed, $\mathrm{SO}(3)\times\mathbb{R}^3$	$R_{k+1} = R_k F_k, \quad p_{k+1} = u_k\Delta t + p_k,$ $F_{k+1} I^b - I^b F_{k+1}^{\mathrm{T}} = I^b F_k - F_k^{\mathrm{T}} I^b, \quad mu_{k+1} = mu_k + mg\Delta t.$

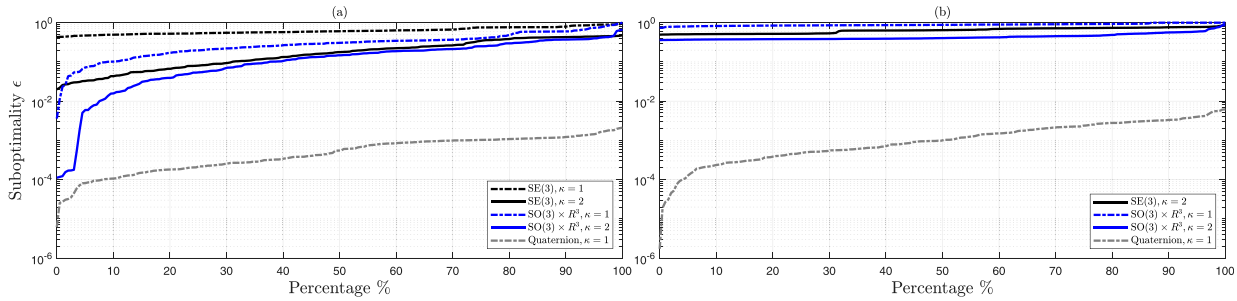

Figure 19. Comparison of integrators on (a) drone landing problem and (b) with cable-suspended load. Within the computational budget, the formulation on $\mathrm{SO}(3)\times\mathbb{R}^3$ provides better initial guesses than SE(3). The relaxation gap by quaternion-based formulation is also much smaller than that of the other methods. The matrix Lie group version considers the TS + block as the sparsity pattern at $\kappa = 2$. When $\kappa = 1$, both quaternion and matrix Lie group versions consider CS + MD.

Nordkvist and Sanyal (2010); Lee et al. (2007). However, the presented derivation directly applies variations on $T\mathrm{SE}(3)$ in discrete time and does not rely on the angular velocity in continuous time as done by Nordkvist and Sanyal (2010). The elimination of the angular velocity in continuous time is more suitable for controller design as it is redundant as its discrete variant F_k is presented. The method by Lee et al. (2007) does not consider the linear velocity defined in the tangent space, which makes the states at three consecutive time steps appear in the same moment matrix. This formulation is denser and thus can dramatically increase the computational burden for moment relaxation.

We then conduct numerical simulations to compare the quaternion-based integrator (Brüdigam and Manchester, 2021b) and the derived LGVI on SE(3) and $\mathrm{SO}(3)\times\mathbb{R}^3$ in the drone landing problem in Section 8.1. We uniformly sample 200 initial orientations and fix the other initial parameters to compare the relaxation gaps. As the rotation matrix $Q(q)$ involves quadratic polynomial of q, the input force $VR(q)^{\mathrm{T}}L(q)V^{\mathrm{T}}e_z f_z$ contains third-order monomials. Thus, $\kappa = 2$ is required to implement the moment relaxation. However, we find that relaxation with $\kappa = 2$ does not provide a sufficiently good initial guess for local solvers.

To fix this problem, we reparameterize the input to lower the relaxation order. We consider the force in the world frame $f_w \in \mathbb{R}^3$ as the decision variables and then enforce it to align with the z axis of the body frame via an additional scalar variable s:

$$\begin{aligned} mu_{k+1} &= mu_k + mg\Delta t + f_{w,k+1}\Delta t, \\ Q(q_{k+1})e_z - f_{w,k+1}s_{k+1} &= 0. \end{aligned} \tag{91}$$

We note that s always exists as long as the total thrust force $f_z \neq 0$. We did not meet this singularity, while a reformulation without singularity is proffered in the future.

Via this reformulation, we now have transformed the third-order dynamics quadratic. We find that the reformulated dynamics have tight relaxation with $\kappa = 1$, which greatly reduces the computational burden. The relaxation gap in the logarithmic scale is illustrated in Figure 19. It is interesting to observe that the quaternion-based version provides relaxation gaps that are dramatically smaller than the version with the rotational matrix.

We further compare the variational integrators on the drone landing problem with cable-suspended load in Section 8.2. For the quaternion-based integrator, the modified linear motion of the drone is the following quadratic polynomial:

$$\begin{aligned} mu_{k+1} &= mu_k + mg\Delta t + f_{w,k+1}\Delta t \\ &\quad + (p_{k+1} - p^L_{k+1})\lambda_{k+1}\Delta t, \\ Q(q_{k+1})e_z - f_{w,k+1}s_{k+1} &= 0. \end{aligned} \tag{92}$$

For the SE(3) case, the dynamics (59) involves third-order monomials in $R^{\mathrm{T}}(p^L - p)\lambda$, which requires $\kappa \geq 2$ for moment relaxation. Again, we sample 200 initial orientations and load velocities to compare different formulations. The relaxation gap of this case is presented in Figure 19. Within the computational budget of local solvers, the optimization gaps by the quaternion-based formulation are all smaller than 0.01, while the relaxation gap by the matrix Lie group formulation is dramatically larger.

In both cases, the $\mathrm{SO}(3)\times\mathbb{R}^3$ formulation provides tighter relaxation compared to the SE(3) formulation. The alternative quaternion-based formulation also exhibits great advantages in computational time. For the case without load, the mean and maximal time consumed by (SDP) is $0.84s$ and $2.07s$. For the case with load, the computational time is $1.37s$ and $2.42s$, respectively. For the relaxation with rotational matrix at the second order that provides relatively tight results, the time consumed is larger than $1000s$.

9.4. Collision avoidance in cluttered environments

As the quaternion-based formulation can efficiently provide tight relaxation at $\kappa = 1$ for the drone landing problem, we extend this algorithm to plan dynamically feasible trajectories with nontrivial obstacle configurations. We consider a world with nonconvex obstacles (Tedrake, 2023), where the collision-free space is decomposed to a union of convex polytopes.

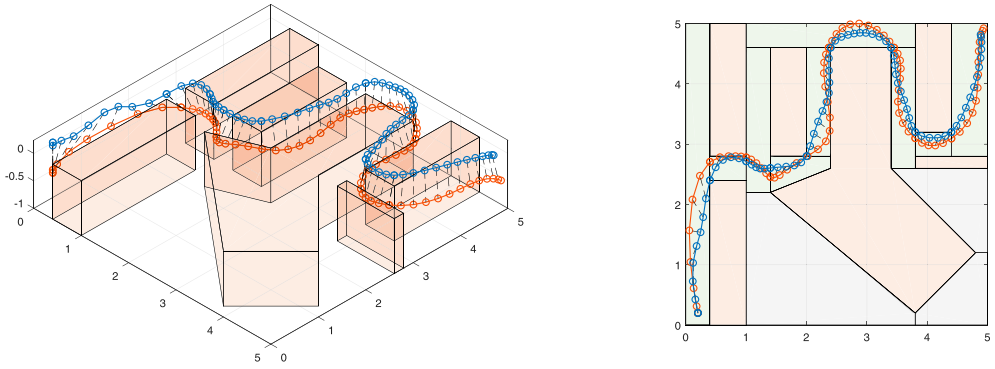

Figure 20. Trajectory optimization of drone with cable-suspended load in a cluttered environment using quaternion-based model. We assign each collocation point to a safe corridor in the form of convex polytopes. The relaxation gap is less than 0.001, and the time used by the SDP solver is 5.35 s. The blue points indicate the drone position, while the orange points denote the load position. The obstacles are depicted in red. The traversed obstacle-free regions are in green, while the collision-free regions not traversed are in gray.

We allocate each collocation point to the prespecified segments and then conduct the trajectory optimization. For each collocation point, we consider the following linear constraints to ensure the load and drone position are within the collision-free polytopes:

$$\mathcal{A}_i p_k \leq b_i, \mathcal{A}_i p_k^L \leq b_i, k \in \mathcal{I}_i, \tag{93}$$

where $\mathcal{I}_i$ is the index set that specifies the polytope constraints defined by $\mathcal{A}_i$ and b_i.

The planned trajectory and obstacle configurations are presented in Figure 20. We can see that the load and drone successfully avoids the obstacle. The total number of collocation points of the trajectory is 80, while the time consumed by the MOSEK is 5.35 s. The relaxation gap $\epsilon = 0.00042$, which is sufficient to justify that the refined trajectory is optimal up to the allocation of the convex collision-free polytopes.

We note that the optimal allocation of the polytope can be solved by mixed-integer programming, which is outside of the scope of this work. Some candidate frameworks, such as the Graph of Convex Sets (GCS) (Marcucci et al., 2024), can provide convex relaxation to this problem, which, however, is not a (POP). Combining the GCS algorithm with the proposed methods could be an interesting future direction.

10. Discussions and future work

In this work, we show that bridging the geometric mechanics and polynomial optimization enables globally optimal motion planning using full rigid body dynamics. We show that the LGVI can formulate the rigid body dynamics as exact quadratic polynomials, which is not possible using conventional integrators. The resulting formulation enables us to obtain certified globally optimal solutions to motion planning problems, such as IK and drone landing problems. The proposed method is also extended to multi-body systems considering the constrained variational integrators. By the polynomial formulation, Lasserre's hierarchy at the second order can provide rank-one globally optimal solutions or certificate of infeasibility.

Though rank-one optimal solutions are available for study cases with easier initial conditions, the solutions are not tight for challenging cases, such as drone landing and cart-pole swing-up with large initial angles. For these cases without an exact rank-one solution, increasing the relaxation order can better approximate the solution, owing to the asymptotic convergence of Lasserre's hierarchy. Despite the absence of rank-one solutions, the rounded solution can also serve as a nontrivial initial guess for local search. We also note that the rank-one solution of the moment relaxation is usually degenerate, which results in slow convergence around the optimum (Alizadeh et al., 1997; Yang et al., 2022). The slow convergence caused by degeneracy can be one reason that the solution is not exactly rank-one but still serves as a good approximation.

Due to the limit of the memory, a tighter relaxation scheme is not implementable for these suboptimal cases. Therefore, it is unclear if such gaps are caused by the numerical inaccuracy of the SDP solver or if the relaxation order is not sufficient. Thus, a more intriguing future direction is to find conditions when the geometric mechanics-based motion planning problems can be tightly solved via Lasserre's hierarchy despite the numerical challenges. A matrix completion problem has been tightly solved by the second-order Lasserre's hierarchy (Cosse and Demanet, 2021), which analytically constructs the dual polynomial certificate. Though such construction techniques are hard to generalize to the complicated dynamics constraints, they inspire future research in this direction. As the sparsity pattern is the key to the implementation of the proposed method, a more fundamental question is how to determine the most efficient sparsity pattern while guaranteeing tightness.

With the alternative quaternion-based formulations, the first-order relaxation can provide tight relaxation for the drone landing cases, which is more efficient than the matrix

Lie group version. The reformulation of the input is key to obtaining tight relaxation at the first-order relaxation. However, the presented reformulation introduces one singular point when the total thrust force becomes zero. Though, in our case, the solution does not reach the singular state, a systematic way of input formulation that guarantees tightness should be derived in the future.

Though the LGVI can model arbitrary multi-body dynamics, the computational cost for two full rigid bodies is still too high. Thus, we are only able to limit the numerical studies to the presented cases. Future work can be extended to more complicated systems using the quaternion-based formulation. Likewise, reformulating the constrained force, as in the drone landing case, needs extra attention to ensure the dynamics remain quadratic. From the point of view of global optimization, the complexity of the presented algorithm is linear w.r.t the planning horizon and polynomial order w.r.t the system dimension (consider the size of the moment matrix) when the relaxation order is determined. Such property is an improvement compared to other methods, such as combinatorial optimization with exponential complexity in the worst case. To improve the scalability for real-time deployment, combining the global convergence property of SDP (Kang et al., 2024; Yang et al., 2022) and fast local search on Lie groups (Brockett, 1991; Bloch et al., 1992; Clark et al., 2021; Teng et al., 2022b, 2022a; Ghaffari et al., 2022; Jang et al., 2023), should be considered in the future. The proposed method can also be extended to nonholonomic systems using LGVI, while the discretization of nonholonomic constraints needs more attention (Fernandez et al., 2012; Kobilarov et al., 2010).

11. Conclusions

In this paper, we present the novel result: by leveraging geometric mechanics-based robot dynamics, one can formulate the motion planning problem as polynomial optimization problems that can be solved via Lasserre's hierarchy. We show that the proposed formulation converts rigid body dynamics as exact quadratic polynomials. We further formulate the motion planning problem as a sparse moment relaxation problem. Attributed to the low-order and sparse formulation, the resulting SDP has linear complexity with respect to the planning horizon and is computationally tractable for the current solvers. The case study on the inverse kinematics for serial manipulators and 3D drone landing problems suggests that the proposed formulation can successfully provide certified globally optimal solutions or certificates of infeasibility for most cases. The proposed method is also extended to multi-body dynamical systems using constrained Lie group variational integrators. The numerical study on drones with cable-suspended load and cart-pole systems suggests that the proposed method can provide tight solutions or nontrivial initial guesses. Finally, an alternative quaternion-based formulation is presented to greatly speed up the computation.

Acknowledgment

The authors thank the editors and anonymous reviewers for the constructive feedback and in-depth review of this work, in particular, the suggestion on the extension to the presented quaternion-based formulation.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by AFOSR MURI FA9550-23-1-0400. Funding for M. Ghaffari was in part provided by NSF Award No. 2118818.

ORCID iDs

Sangli Teng https://orcid.org/0000-0001-8892-5144
Maani Ghaffari https://orcid.org/0000-0002-4734-4295

Notes

1. The Archimedeanness condition is satisfied if a two-norm ball can be constructed by the constraints $\{g_j\}$, that is, $\exists\ \{\sigma_j\}$ a set of sum-of-squares polynomials, such that $M - \|x\|_2^2 = \sigma_0(x) + \sum_j \sigma_j(x) g_j(x)$, with $M \geq 0$.
2. For constrained multi-rigid body dynamics, the U_k also contains the constrained force determined by the external control input.
3. The maximal coordinates formulation explicitly models the full 6 DOF of each rigid body and adds constrained forces.

References

Agha-Mohammadi AA, Chakravorty S and Amato NM (2014) FIRM: sampling-based feedback motion-planning under motion uncertainty and imperfect measurements. *The International Journal of Robotics Research* 33(2): 268–304.

Alizadeh F, Haeberly JPA and Overton ML (1997) Complementarity and nondegeneracy in semidefinite programming. *Mathematical programming* 77(1): 111–128.

Amice A, Dai H, Werner P, et al. (2023) Finding and optimizing certified, collision-free regions in configuration space for robot manipulators. In: *International Workshop on the Algorithmic Foundations of Robotics*. Berlin, Germany: Springer, 328–348.

Andersen ED and Andersen KD (2000) The MOSEK interior point optimizer for linear programming: an implementation of the homogeneous algorithm. In: *High performance optimization*. Boston, MA: Springer, 197–232.

ApS M (2022) MOSEK modeling cookbook. https://docs.mosek.com/MOSEKModelingCookbook-letter.pdf.

Bandeira AS (2016) A note on probably certifiably correct algorithms. *Comptes Rendus Mathematique* 354(3): 329–333.

Behcet A, Carson JM and Blackmore L (2013) Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem. *IEEE*

Transactions on Control Systems Technology 21(6): 2104–2113.

Betts JT (1998) Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics* 21(2): 193–207.

Blekherman G, Parrilo PA and Thomas RR (2012) *Semidefinite Optimization and Convex Algebraic Geometry*. Philadelphia, PA: SIAM.

Bloch AM (2003) Nonholonomic mechanics. In: *Nonholonomic Mechanics and Control*. Berlin, Germany: Springer, 207–276.

Bloch AM, Brockett RW and Ratiu TS (1992) Completely integrable gradient flows. *Communications in Mathematical Physics* 147(1): 57–74.

Brockett RW (1991) Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems. *Linear Algebra and its Applications* 146: 79–91.

Brüdigam J and Manchester Z (2021a) Linear-quadratic optimal control in maximal coordinates. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Piscataway, NJ: IEEE, 9775–9781.

Brüdigam J and Manchester Z (2021b) Linear-time variational integrators in maximal coordinates. In: *International Workshop on the Algorithmic Foundations of Robotics*. Berlin, Germany: Springer, 194–209.

Bullo F and Murray RM (1999) Tracking for fully actuated mechanical systems: a geometric framework. *Automatica* 35(1): 17–34.

Canny J (1987) A new algebraic method for robot motion planning and real geometry. In: *Annual Symposium on Foundations of Computer Science*. Piscataway, NJ: IEEE, 39–48.

Celledoni E, Marthinsen H and Owren B (2014) An introduction to Lie group integrators–basics, new developments and applications. *Journal of Computational Physics* 257: 1040–1061.

Cifuentes D, Agarwal S, Parrilo PA, et al. (2020) On the local stability of semidefinite relaxations. *Mathematical Programming* 193: 1–35.

Clark W, Ghaffari M and Bloch A (2021) Nonparametric continuous sensor registration. *Journal of Machine Learning Research* 22(271): 1–50.

Cohn TB, Petersen M, Simchowitz M, et al. (2023) Non-euclidean motion planning with graphs of geodesically-convex sets. In: Proceedings of the Robotics: Science and Systems Conference, Daegu, Republic of Korea, 2023. DOI: 10.15607/RSS.2023.XIX.057.

Cosse A and Demanet L (2021) Stable rank-one matrix completion is solved by the level 2 Lasserre relaxation. *Foundations of Computational Mathematics* 21: 891–940.

Dai H, Izatt G and Tedrake R (2019) Global inverse kinematics via mixed-integer convex optimization. *The International Journal of Robotics Research* 38(12-13): 1420–1441.

Deits R and Tedrake R (2014) Footstep planning on uneven terrain with mixed-integer convex optimization. In: *IEEE-RAS International Conference on Humanoid Robots*. Piscataway, NJ: IEEE, 279–286.

Deits R and Tedrake R (2015) Efficient mixed-integer planning for UAVs in cluttered environments. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Piscataway, NJ: IEEE, 42–49.

Ding Y, Li C and Park HW (2018) Single leg dynamic motion planning with mixed-integer convex optimization. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Piscataway, NJ: IEEE, 1–6.

Ding Y, Li C and Park HW (2020) Kinodynamic motion planning for multi-legged robot jumping via mixed-integer convex program. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Piscataway, NJ: IEEE, 3998–4005.

Donald B, Xavier P, Canny J, et al. (1993) Kinodynamic motion planning. *Journal of the ACM* 40(5): 1048–1066.

Dong DE, Berger HP and Abraham I (2023) Time optimal ergodic search. In: Proceedings of the Robotics: Science and Systems Conference, Daegu, Republic of Korea, 2023. DOI: 10.15607/RSS.2023.XIX.082.

El Khadir B, Lasserre JB and Sindhwani V (2021) Piecewise-linear motion planning amidst static, moving, or morphing obstacles. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Piscataway, NJ: IEEE, 7802–7808.

Fernandez OE, Bloch AM and Olver PJ (2012) Variational integrators from Hamiltonizable nonholonomic systems. *Journal of Geometric Mechanics* 4(2): 137–163.

Ghaffari M, Zhang R, Zhu M, et al. (2022) Progress in symmetry preserving robot perception and control through geometry and learning. *Frontiers in Robotics and AI* 9: 969380.

Ghaffari Jadidi M, Valls Miro J and Dissanayake G (2019) Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring. *The International Journal of Robotics Research* 38(6): 658–685.

Henrion D and Korda M (2013) Convex computation of the region of attraction of polynomial control systems. *IEEE Transactions on Automatic Control* 59(2): 297–312.

Henrion D, Lasserre JB and Savorgnan C (2008) Nonlinear optimal control synthesis via occupation measures. In: *Proceedings of the IEEE Conference on Decision and Control*. Piscataway, NJ: IEEE, 4749–4754.

Hereid A and Ames AD (2017) FROST: fast robot optimization and simulation toolkit. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Piscataway, NJ: IEEE, 719–726.

Hollinger GA and Sukhatme GS (2014) Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research* 33(9): 1271–1287.

Hopcroft JE, Schwartz JT and Sharir M (1984) On the complexity of motion planning for multiple independent objects; PSPACE-hardness of the "warehouseman's problem". *The International Journal of Robotics Research* 3(4): 76–88.

Horsch T, Schwarz F and Tolle H (1994) Motion planning with many degrees of freedom-random reflections at c-space obstacles. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Piscataway, NJ: IEEE, 3318–3323.

Howell TA, Le Cleac'h S, Kolter JZ, et al. (2022) *Dojo: a differentiable simulator for robotics*. Ithaca, NY: arXiv.

Iserles A, Munthe-Kaas HZ, Nørsett SP, et al. (2000) Lie-group methods. *Acta Numerica* 9: 215–365.

Jacobson DH and Mayne DQ (1970) *Differential Dynamic Programming*. Amsterdam: Elsevier Publishing Company, Vol. 24.

Jang J, Teng S and Ghaffari M (2023) Convex geometric trajectory tracking using Lie algebraic MPC for autonomous marine vehicles. *IEEE Robotics and Automation Letters* 8(12): 8374–8381.

Jasour A and Williams BC (2019) Risk contours map for risk bounded motion planning under perception uncertainties. In: Proceedings of the Robotics: Science and Systems Conference, Breisgau, Germany, June 22–June 26, 2019.

Jasour A, Han W and Williams BC (2021a) Convex risk bounded continuous-time trajectory planning in uncertain nonconvex environments. In: Proceedings of the Robotics: Science and Systems Conference, Daegu, Republic of Korea, 2023. Virtual. DOI: 10.15607/RSS.2021.XVII.069.

Jasour A, Wang A and Williams BC (2021b) *Moment-based Exact Uncertainty Propagation through Nonlinear Stochastic Autonomous Systems*. Ithaca, NY: arXiv.

Kamoutsi A, Sutter T, Mohajerin Esfahani P, et al. (2017) On infinite linear programming and the moment approach to deterministic infinite horizon discounted optimal control problems. *IEEE Control Systems Letters* 1(1): 134–139.

Kang S, Xu X, Sarva J, et al. (2024) *Fast and certifiable trajectory optimization*. Ithaca, NY: arXiv.

Kavraki LE, Svestka P, Latombe JC, et al. (1996) Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation* 12(4): 566–580.

Kierzenka J and Shampine LF (2001) A BVP solver based on residual control and the Maltab PSE. *ACM Transactions on Mathematical Software* 27(3): 299–316.

Kobilarov M, Marsden JE and Sukhatme GS (2010) Geometric discretization of nonholonomic systems with symmetries. *Discrete and Continuous Dynamical Systems - Series S* 3(1): 61–84.

Lasserre JB (2001) Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization* 11(3): 796–817.

Lasserre JB (2006) Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization* 17(3): 822–843.

Lasserre JB (2015) *An Introduction to Polynomial and Semi-algebraic Optimization*. Cambridge, UK: Cambridge University Press, Vol. 52.

LaValle SM (2006) *Planning Algorithms*. Cambridge, UK: Cambridge University Press.

LaValle SM and Kuffner JJ Jr (2001) Randomized kinodynamic planning. *The International Journal of Robotics Research* 20(5): 378–400.

Lawler EL and Wood DE (1966) Branch-and-bound methods: a survey. *Operations Research* 14(4): 699–719.

Lee T, McClamroch NH and Leok M (2005) 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*. Piscataway, NJ: IEEE, 962–967.

Lee T, Leok M and McClamroch NH (2007) Lie group variational integrators for the full body problem. *Computer Methods in Applied Mechanics and Engineering* 196(29-30): 2907–2924.

Leyendecker S, Marsden JE and Ortiz M (2008) 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.

Li Z, Zeng J, Chen S, et al. (2023) Autonomous navigation of underactuated bipedal robots in height-constrained environments. *The International Journal of Robotics Research* 42(8): 565–585.

Liu C and Tomizuka M (2017) Real time trajectory optimization for nonlinear robotic systems: relaxation and convexification. *Systems & Control Letters* 108: 56–63.

Liu C, Lin CY and Tomizuka M (2018) The convex feasible set algorithm for real time optimization in motion planning. *SIAM Journal on Control and Optimization* 56(4): 2712–2733.

Magron V and Wang J (2021) *TSSOS: A Julia Library to Exploit Sparsity for Large-Scale Polynomial Optimization*. Ithaca, NY: arXiv.

Majumdar A, Vasudevan R, Tobenkin MM, et al. (2014) Convex optimization of nonlinear feedback controllers via occupation measures 33(9): 1209–1230.

Manchester ZR and Peck MA (2016) Quaternion variational integrators for spacecraft dynamics. *Journal of Guidance, Control, and Dynamics* 39(1): 69–76.

Manchester Z, Doshi N, Wood RJ, et al. (2019) Contact-implicit trajectory optimization using variational integrators. *The International Journal of Robotics Research* 38(12-13): 1463–1476.

Marcucci T, Umenberger J, Parrilo P, et al. (2024) Shortest paths in graphs of convex sets. *SIAM Journal on Optimization* 34(1): 507–532.

Marić F, Giamou M, Khoubyarian S, et al. (2020) Inverse kinematics for serial kinematic chains via sum of squares optimization. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Piscataway, NJ: IEEE, 7101–7107.

Marsden JE and Ratiu TS (1998) *Introduction to Mechanics and Symmetry*. Berlin, Germany: Springer.

Marsden JE and West M (2001) Discrete mechanics and variational integrators. *Acta Numerica* 10: 357–514.

Marsden JE, Pekarsky S and Shkoller S (1999) Discrete euler-poincaré and lie-Poisson equations. *Nonlinearity* 12(6): 1647–1662.

Merat F (1987) Introduction to robotics: mechanics and control. *IEEE Journal of Robotics and Automation* 3(2): 166.

Nie J (2014) Optimality conditions and finite convergence of Lasserre's hierarchy. *Mathematical programming* 146(1): 97–121.

Nordkvist N and Sanyal AK (2010) A Lie group variational integrator for rigid body motion in SE(3) with applications to underwater vehicle dynamics. In: *Proceedings of the IEEE Conference on Decision and Control*. Piscataway, NJ: IEEE, 5414–5419.

Parrilo PA (2003) Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming* 96(2): 293–320.

Posa M, Cantu C and Tedrake R (2014) A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research* 33(1): 69–81.

Reif JH (1979) Complexity of the mover's problem and generalizations. In: *Annual Symposium on Foundations of Computer Science*. Washington, DC: IEEE Computer Society, 421–427.

Richards A and How JP (2002) Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In: *Proceedings of the American Control Conference*. Piscataway, NJ: IEEE, Vol. 3, 1936–1941.

Saccon A, Hauser J and Aguiar AP (2013) Optimal control on Lie groups: the projection operator approach. *IEEE Transactions on Automatic Control* 58(9): 2230–2245.

Schouwenaars T, De Moor B, Feron E, et al. (2001) *Mixed Integer Programming for Multi-Vehicle Path Planning*. Piscataway, NJ: IEEE, 2603–2608.

Schulman J, Duan Y, Ho J, et al. (2014) Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research* 33(9): 1251–1270.

Sertac K and Emilio F (2011) Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research* 30(7): 846–894.

Tassa Y, Mansard N and Todorov E (2014) Control-limited differential dynamic programming. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Piscataway, NJ: IEEE, 1168–1175.

Tedrake R (2023) Robotic manipulation. https://manipulation.mit.edu/.

Tedrake R, Manchester IR, Tobenkin M, et al. (2010) LQR-trees: feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research* 29(8): 1038–1052.

Teng S, Gong Y, Grizzle JW, et al. (2021) *Toward Safety-Aware Informative Motion Planning for Legged Robots*. Ithaca, NY: arXiv.

Teng S, Chen D, Clark W, et al. (2022a) 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*. Piscataway, NJ: IEEE, 8850–8857.

Teng S, Clark W, Bloch A, et al. (2022b) Lie algebraic cost function design for control on Lie groups. In: *Proceedings of the IEEE Conference on Decision and Control*. Piscataway, NJ: IEEE, 1867–1874.

Teng S, Jasour A, Vasudevan R, et al. (2023) Convex geometric motion planning on lie groups via moment relaxation. In: Proceedings of the Robotics: Science and Systems Conference. Daegu, Republic of Korea, 2023. DOI:10.15607/RSS.2023.XIX.058.

Teng S, Zhang H, Jin D, et al. (2024) *GMKF: Generalized Moment Kalman Filter for Polynomial Systems with Arbitrary Noise*. Ithaca, NY: arXiv.

Trutman P, Din MSE, Henrion D, et al. (2022) Globally optimal solution to inverse kinematics of 7DOF serial manipulator. *IEEE Robotics and Automation Letters* 7(3): 6012–6019.

Wächter A and Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming* 106: 25–57.

Waki H, Kim S, Kojima M, et al. (2006) Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization* 17(1): 218–242.

Wang A, Jasour A and Williams B (2020) *Moment State Dynamical Systems for Nonlinear Chance-Constrained Motion Planning*. Ithaca, NY: arXiv.

Wang J, Magron V and Lasserre JB (2021a) Chordal-TSSOS: a moment-sos hierarchy that exploits term sparsity with chordal extension. *SIAM Journal on Optimization* 31(1): 114–141.

Wang J, Magron V and Lasserre JB (2021b) TSSOS: a moment-SOS hierarchy that exploits term sparsity. *SIAM Journal on Optimization* 31(1): 30–58.

Wang J, Magron V, Lasserre JB, et al. (2022) CS-TSSOS: correlative and term sparsity for large-scale polynomial optimization. *ACM Transactions on Mathematical Software* 48(4): 1–26.

Yang H and Carlone L (2020) One ring to rule them all: certifiably robust geometric perception with outliers. *Proceedings of the Advances in Neural Information Processing Systems Conference* 33: 18846–18859.

Yang H and Carlone L (2022) *Certifiably Optimal Outlier-Robust Geometric Perception: Semidefinite Relaxations and Scalable Global Optimization*. Piscataway, NJ: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Yang H, Liang L, Carlone L, et al. (2022) An inexact projected gradient method with rounding and lifting by nonlinear programming for solving rank-one semidefinite relaxation of polynomial optimization. *Mathematical Programming* 201: 409–472.

Yang L, Dai H, Amice A, et al. (2023) *Approximate Optimal Controller Synthesis for Cart-Poles and Quadrotors via Sums-of-Squares*. Piscataway, NJ: IEEE Robotics and Automation Letters.

Zhao P, Mohan S and Vasudevan R (2019) Optimal control of polynomial hybrid systems via convex relaxations. *IEEE Transactions on Automatic Control* 65(5): 2062–2077.

Zucker M, Ratliff N, Dragan AD, et al. (2013) Chomp: covariant Hamiltonian optimization for motion planning. *The International Journal of Robotics Research* 32(9-10): 1164–1193.

Appendices

A. POP as infinite-dimensional linear programming

(POP) can be converted to the following infinite-dimensional linear programming problem over the space of measure (Lasserre, 2001, 2015):

Problem 5. (Infinite dimension linear programming).

$$p^* := \inf_{\mu \in \mathcal{M}^+(\mathbb{K})} \int_{\mathbb{K}} p(x) d\mu, \tag{LP}$$

with $\mathbb{K}$ the feasible set defined in (POP), $\mathcal{M}(\mathbb{K})$ the set of vector space of finite signed Borel measure and $\mathcal{M}^+(\mathbb{K})$ the convex cone of non-negative finite Borel measure on $\mathbb{K}$.

Then, the optimization problem (POP) is equivalent to finding the Dirac measure δ_{x^*} that is associated with the minimizer x^* of (POP). Recall that the Dirac measure has the property $\int_{\mathbb{K}} p(x) d\delta_{\overline{x}} = p(\overline{x})$, which enables one to select the value of $p(x)$ at a given point $\overline{x}$.

B. Sparse moment relaxation of POP

Though Lasserre's Hierarchy enables one to approximate (POP) by (SDP), the size of (SDP) increases dramatically as κ and n increase. Thus, it is critical to explore the structure of the problem fully to reduce the computational burden. For many applications in control and planning satisfying the Markov assumption, only states at consecutive time steps appear in the system dynamics. The cost function is usually the sum of stage costs that only contain states within one step. Motivated by this observation, we introduce the correlative sparsity.

We define the index set $I_0 = \{1, \ldots, n\} = \cup_{k=1}^{q} I_k$, as the union of q subsets $I_k \subset I_0$ that partition the variable x. For arbitrary $I_k \subseteq I_0$, let $\mathbb{R}[x(I_k)]$ denote the ring of polynomials in the variable $x(I_k) \in \{x_i | i \in I_k\}$. We also define the index set $J = \{1, \ldots, m\}$ that is partitioned in to q different disjoint sets J_k, $k = 1, \ldots, q$ to group the constraints g_j, $j = 1, \ldots, m$.

Assumption 2. (Sparse structure of (POP), (Lasserre, 2006)).

1. For feasible set $\mathbb{K}$, there is a large number M, such that $\|x\|_\infty \le M$ for $\forall x \in \mathbb{K}$.
2. For every $j \in J_k$, $g_j \in \mathbb{R}[x(I_k)]$, such that each constraint $g_j(x) \ge 0$ only involves variables in the set $x(I_k) = \{x_i | i \in I_k\}$.
3. The objective function $p(x) \in \mathbb{R}[x]$ can be written as $p(x) = \sum_{k=1}^{q} f_k$, with $f_k \in \mathbb{R}[x(I_k)], k = 1, \ldots, q$.
4. The index set I_k satisfy the running intersection property:

$$\forall k = 1, \ldots, q-1, \exists s \le k, I_{k+1} \cap \left(\cup_{j=1}^{k} I_j \right) \subseteq I_s.$$

If (POP) satisfies the assumptions, the following sparse moment relaxations can dramatically reduce the problem size and have the same convergence property as the dense moment relaxation:

Problem 6. (Sparse moment relaxation (Lasserre, 2006, 2015)).

$$\begin{aligned} \rho_\kappa^* := & \inf_{y \in \mathbb{R}^{s(2\kappa)}} \mathcal{L}_y(p) \\ s.t. \quad & M_\kappa(y, I_k) \ge 0, \\ & M_{\kappa - d_j}(g_j y, I_k) \ge 0, \\ & j \in J_k, k = 1, \ldots, q. \end{aligned} \tag{sparse-SDP}$$

Where $M_\kappa(y, I_k)$ denotes the moment matrix formed by the variables that appear in the set I_k. We also have a slightly different rank condition and the special rank-one case for the sparse moment relaxation.

Theorem 3. (Rank condition for (sparse-SDP) (Lasserre, 2006, 2015)). (sparse-SDP) *is tight, if:*

1. *Assumption* 2 *is satisfied for* (POP)*, and,*
2. $\text{rank}(M_\kappa(y_\kappa^*, I_k)) = \text{rank}(M_{\kappa - d_g}(y_\kappa^*, I_k))$, $k = 1, \ldots, q$, *and,*
3. $\text{rank}(M_\kappa(y_\kappa^*, I_{jk})) = 1$ *for all pairs* (i, k) *with* $I_{jk} := I_j \cap I_k \neq \emptyset$.

Remark 4. (Rank-one optimality condition for (sparse-SDP)). For the special case of rank-one condition, we will only need to check the first two conditions in Theorem 3 as any $M_\kappa(y, I_{jk})$ can become a principle submatrix of the $M_\kappa(y, I_k)$ or $M_\kappa(y, I_j)$ after proper invertible row and column permutations. Thus, $M_\kappa(y, I_{jk})$ is rank-one if $M_\kappa(y, I_k)$ or $M_\kappa(y, I_j)$ are rank-one matrix.

C. Closed-loop simulation

In this appendix, we apply tracking controllers to track the planned trajectories to demonstrate they are dynamically feasible. We consider a customized simulator using the variation-based dynamics in maximal coordinates (Brüdigam and Manchester, 2021b; Howell et al., 2022).

C.1. Drone landing. We develop the following nonlinear tracking controller to track the desired trajectory planned by the proposed method. We apply the controller in a receding horizon manner to minimize the quadratic cost w.r.t the reference trajectory:

Problem 7. (Nonlinear tracking controller). Consider the configuration state Y_k, input U_k, and dynamics $D(\cdot, \cdot, \cdot)$ as in Problem 3. We track the desired trajectory $\{\overline{Y}_k\}_{k=0}^{N}$ and $\{\overline{U}_k\}_{k=1}^{N}$ via minimizing the quadratic tracking cost:

$$
\begin{aligned}
\min_{\{Y_k\}_{k=0}^N,\{U_k\}_{k=1}^N} \quad & \sum_{k=0}^{N-1} \|Y_k - \overline{Y}_k\|_Q^2 + \|U_{k+1} - \overline{U}_{k+1}\|_R^2 \\
& + \|Y_N - \overline{Y}_N\|_P^2 \\
\text{s.t.} \quad & D(Y_{k+1}, Y_k, U_{k+1}) = 0, \\
& Y_k \in \mathcal{Y}, \\
& U_{\min} \le U_k \le U_{\max}, \\
& Y_0 = Y_{\text{init}}, \\
& k = 0, \ldots, N-1.
\end{aligned} \tag{94}
$$

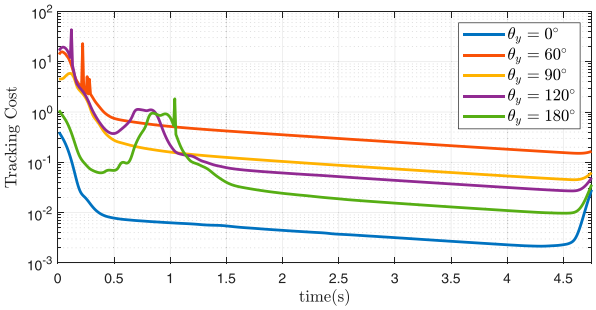

Figure 21. Tracking cost along the planned path of the drone landing problem. The tracking cost converges exponentially fast after the transient state at the first 1.5 s. The tracking cost can be considered a Lyapunov function for the closed-loop system to certify stability.

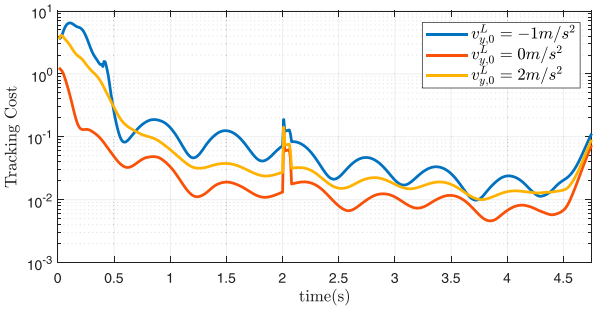

Figure 22. Tracking cost along the planned path of the drone landing problem with cable-suspended load. The tracking cost converges with some oscillation due to the motion of the load. The controller successfully stabilizes the system after a perturbation was acted on the load at $t = 2s$.

where the P, Q, and R are the cost matrices w.r.t. the terminal state, running state, and control input. The desired state $\{\overline{Y}_k\}_{k=0}^N$ and control input $\{\overline{U}_k\}_{k=1}^N$ are interpolated at the planned trajectory given the sampling time of the tracking controller.

As the tracking controller considers the nonlinear dynamics, the nonconvexity still makes the computation challenging. To solve the problem, we initialize the optimization using the nominal control input and state obtained by the trajectory optimization $\{\overline{Y}_k\}_{k=0}^N$ and $\{\overline{U}_k\}_{k=1}^N$. In our simulation, the tracking horizon is set to $N = 20$ with the sample time $\Delta t = 0.01s$. The tracking cost for cases in Figure 10 are shown in Figure 21, where we can see that the tracking cost decays exponentially fast after the transient state at the first few seconds. The actual control input and the references are illustrated in Figure 23.

C.2. Drone landing with cable-suspended load. We apply the same tracking controller to the case with cable-suspended loads. The cost function incorporates additional terms to regularize the tracking error of the load position and velocity. We note that the cable dynamics are hybrid by the complementarity condition (Posa et al., 2014):

$$
\begin{aligned}
&\|p_{k+1}^L - p_{k+1}\|^2 - l^2 \ge 0, \quad \lambda_{k+1} \ge 0, \\
&(l^2 - \|p_{k+1}^L - p_{k+1}\|^2)\lambda_{k+1} = 0.
\end{aligned} \tag{95}
$$

As (95) is challenging for optimization, we introduce the slack variable $\delta^{(\cdot)}$ in the optimization to eliminate the infeasibility:

$$
\begin{aligned}
&l^2 - \|p_{k+1}^L - p_{k+1}\|^2 + \delta_{k+1}^{(1)} = 0, \\
&(l^2 - \|p_{k+1}^L - p_{k+1}\|^2)\lambda_{k+1} + \delta_{k+1}^{(2)} = 0.
\end{aligned} \tag{96}
$$

Then, we penalize δ in the cost function with large weights. The tracking cost and control inputs are presented in Figures 22 and 24, respectively. We can see that in case $v_y^L = -1m/s^2$, though the cable tension λ becomes 0 for some time, the controller can still stabilize the system.

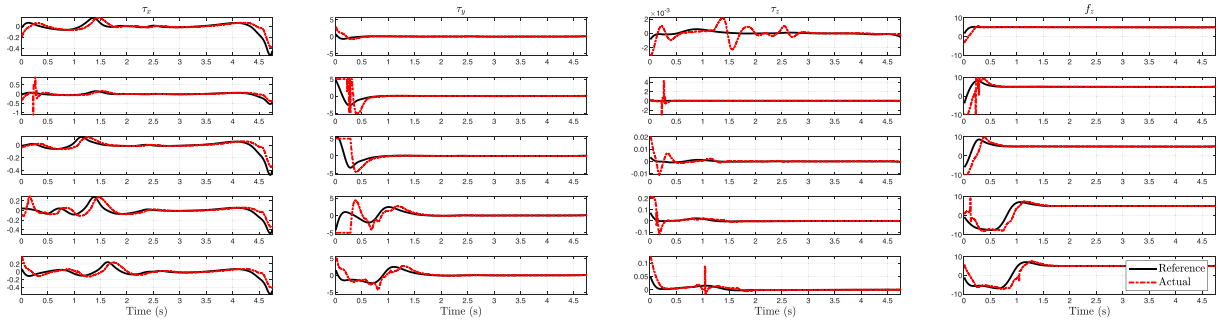

Figure 23. Control input along the planned trajectory of the drone landing problems. The initial pitch angles from top to bottom are 0°, 30°, 90°, 120°, and 180°.

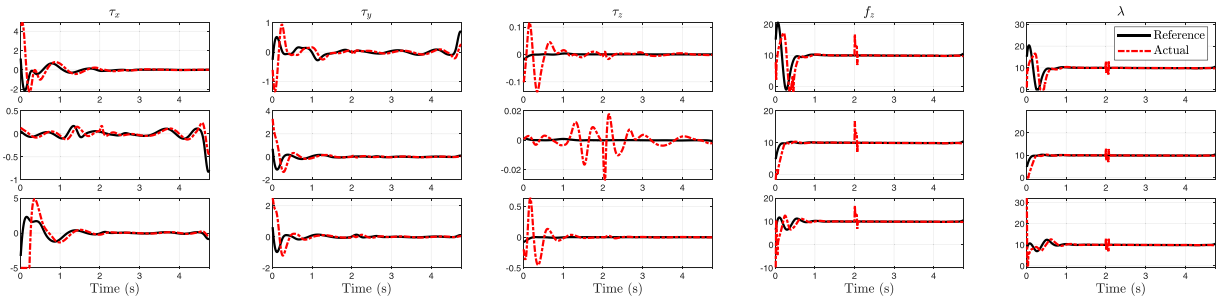

Figure 24. Control input along the planned trajectory. The initial velocity v_y^L of each case from top to bottom are -1 m/s^2, 0 m/s^2 and 2 m/s^2.

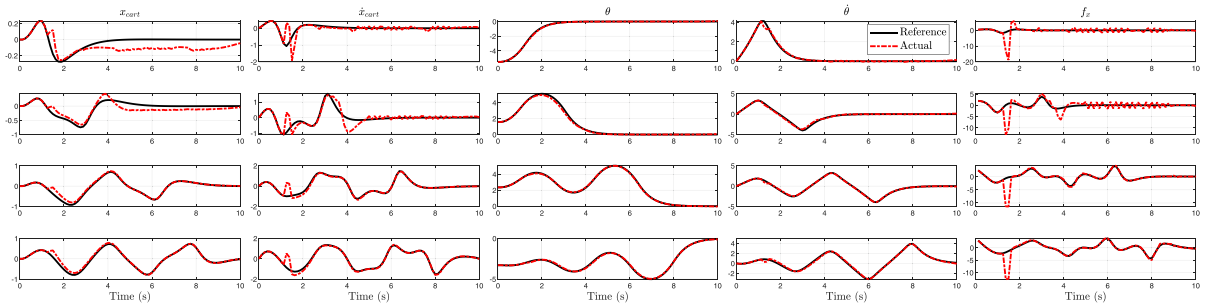

Figure 25. Trajectory of the cart-pole state in minimal coordinates. The states corresponding to the cases with initial cart angle are $\theta_0 = 0.25\pi$, 0.5π, 0.75π, and 0.95π. The pole angles are wrapped to ensure the terminal states are around $\theta = 0$. Note that the actual controllers are designed in the maximal coordinates and, thus, do not have the rounding issue of joint angles in minimal coordinates. The controller successfully stabilizes the system after a perturbation on the cat at $t = 0.5s$.

C.3. Cart-pole swing-up. We implement the linear quadratic regulator (Brüdigam and Manchester, 2021a) to track the trajectory in the maximal coordinates.

Consider the dynamics (62), (63), and the holonomic constraints (64). We define the residual state w.r.t the nominal trajectory as $\widetilde{y} := Y - \overline{Y}$, $\widetilde{u} := f_x - \overline{f}_x$ and $\widetilde{\lambda} := \lambda - \overline{\lambda}$. We can then derive the linearized dynamics around the reference trajectory as

$$\widetilde{y}_{k+1} = A_k \widetilde{y}_k + B_k \widetilde{u}_k + C_k \widetilde{\lambda}_k. \tag{97}$$

where A_k, B_k, and C_k are the Jacobians obtained by differentiating the dynamics $D(\cdot, \cdot, \cdot)$ using implicit function theorem. The linearized holonomic constraints can similarly obtained by differentiating the holonomic constraints by

$$G_{k+1}\widetilde{y}_{k+1} = G_{k+1}(A_k \widetilde{y}_k + B_k \widetilde{u}_k + C_k \widetilde{\lambda}_k) = 0, \tag{98}$$

with $G_k := \partial g(Y_k)/\partial Y_k$. Finally, we have the linear quadratic regulator in the maximal coordinates as

$$\begin{aligned} \min_{\{\widetilde{u}_k, \widetilde{\lambda}_k\}_{k=1}^{N-1}, \{\widetilde{y}\}_{k=0}^{N}} \quad & \frac{1}{2}\widetilde{y}_N^{\mathrm{T}} P \widetilde{y}_N + \frac{1}{2}\sum_{k=0}^{N-1}\left(\widetilde{y}_k^{\mathrm{T}} Q \widetilde{y}_k + \widetilde{u}_k^{\mathrm{T}} R \widetilde{u}_k\right) \\ \text{s.t.} \quad & \widetilde{y}_{k+1} = A_k \widetilde{y}_k + B_k \widetilde{u}_k + C_k \widetilde{\lambda}_k, \\ & G_{k+1}\left(A_k \widetilde{y}_k + B_k \widetilde{u}_k + C_k \widetilde{\lambda}_k\right) = 0, \\ & f_{x,\min} \le \overline{f}_{x,k} + \widetilde{u}_k \le f_{x,\max}, \\ & \overline{Y}_0 + \widetilde{y}_0 = Y_{init}, \\ & k = 0, 1, \ldots, N-1. \end{aligned} \tag{99}$$

We apply the batch optimization in a receding horizon manner to track the desired trajectory obtained from the global optimization. The tracking performance of the cart position, the pole angle, and the control input are illustrated in Figure 25. The controller can stabilize the trajectories after a disturbance is acted on the cart at $t = 1.25s$.