Discrete Geometric Optimal Control on Lie Groups

Marin Kobilarov and Jerrold E. Marsden

***Abstract*—We consider the optimal control of mechanical systems on Lie groups and develop numerical methods which exploit the structure of the state space and preserve the system motion invariants. Our approach is based on a coordinate-free variational discretization of the dynamics leading to structure-preserving discrete equations of motion. We construct necessary conditions for optimal trajectories corresponding to discrete geodesics of a higher order system and develop numerical methods for their computation. The resulting algorithms are simple to implement and converge to a solution in very few iterations. A general software implementation is provided and applied to two example systems: an underactuated boat and a satellite with thrusters.**

I. Introduction

We consider the optimal control of mechanical systems evolving on a finite dimensional Lie group. Our primary motivation is the control of autonomous vehicles modeled as rigid bodies. The goal is to actuate the system to move from its current state to a desired state in an optimal way, e.g. with minimum control effort or time.

The standard way to solve such problems is to first derive the continuous nonlinear equations of motion, for example using a variational principle such as Lagrange-d'Alembert. Two general methods, termed direct and indirect, are then available to compute a minimum-cost trajectory [1]. In the direct method, the differential equations are discretized (or represented using a finite set of parameters) and enforced as algebraic constraints in a nonlinear optimization program. Such a formulation is then computationally solved using a package such as sequential quadratic programming. The indirect method is to derive necessary conditions for optimality, i.e. using Pontryagin-maximum principle by formulating another variational problem based on the original continuous equations and cost function. The necessary conditions are expressed through the evolution of additional adjoint variables satisfying a set of ordinary differential and transversality equations. These equations are then discretized and solved iteratively, e.g. using Newton's method, in order to compute an approximate numerical solution.

The framework proposed in this work uses a different computational strategy. It employs the theory of *discrete mechanics* based on a discrete variational formulation. In particular, we employ a discrete Lagrange-d'Alembert (DLA) variational principle yielding a set of discrete trajectories that approximately satisfy the dynamics and that respect the state space structure. Among these trajectories we find the extremal one without any further discretization or approximation. This is achieved by formulating a higher order variational problem to be solved through a repeated application of the DLA principle to obtain the optimal control trajectory. Such a construction enables the preservation of important properties of the mechanical system–group structure, momentum, and symplectic structure–and results in algorithms with provable accuracy and stability.

Our approach is designed for accurate and numerically stable computation and is especially suited for nonlinear systems that require iterative optimal control solvers. We construct a general optimization framework for systems on Lie groups and demonstrate its application to rigid body motion groups as well as to any real matrix subgroup. Finally, in addition to developing a computational control theory for Lie groups, we spell out the details necessary for a practical implementation.

The Lagrangian Mechanical System

We consider mechanical systems evolving on an n-dimensional Lie group G. The fundamental property of Lie groups is that each tangent vector on the manifold can be generated by translating a unique tangent vector at the identity using the group operation. More formally, each vector $\dot{g} \in T_g G$ at configuration $g \in G$ corresponds to a unique vector $\xi \in \mathfrak{g}$ through $\dot{g} = g\xi$, where $\mathfrak{g} := T_e G$ denotes the Lie algebra and $e \in G$ is the group identity.

In view of this structure the dynamics can be derived through the *reduced Lagrangian* $\ell : G \times \mathfrak{g} \to \mathbb{R}$ defined by $\ell(g,\xi) = L(g, g\xi)$ where $L : TG \to \mathbb{R}$ is the standard Lagrangian. As we will show working with the reduced state (g,ξ) has important implications for constructing numerical optimization schemes.

In this work we employ general Lagrangians of the form

$$\ell(g,\xi) = K(\xi) - V(g), \tag{1}$$

where $K : \mathfrak{g} \to \mathbb{R}$ and $V : G \to \mathbb{R}$ are given kinetic and potential energy functions. The kinetic energy Hessian $\partial_\xi^2 K$ is assumed non-singular over the control problem domain.

The system is actuated using a control force $f(t) \in \mathfrak{g}^*$ defined in the body reference frame[1]. We will begin our development with fully actuated systems, i.e. such that f can act in any direction of the linear space $\mathfrak{g}^*$. We will then consider underactuated systems with control parameters, i.e. systems such that $f = \sum_{i=1}^c f^i(\phi) u^i$, where $f^i \in \mathfrak{g}^*$ define the allowed control directions (covectors) which depend on the controllable parameters $\phi \in \mathbb{M} \subset \mathbb{R}^m$. Here, $u \in \mathbb{U} \subset \mathbb{R}^c$ denotes the control inputs. The control input set $\mathbb{U}$ and the control shape space $\mathbb{M}$ are bounded vector spaces. The reader

Marin Kobilarov and Jerrold Marsden are with the Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA, 91125 USA (e-mail: marin@cds.caltech.edu, marsden@cds.caltech.edu).

Manuscript received June 11, 2010; revised March 29, 2011.

[1] In the Lagrangian setting a force is an element of the Lie algebra dual $\mathfrak{g}^*$, i.e. a one-form $\langle f, \cdot \rangle$ that pairs with velocity vectors to produce the total work $\int_0^T \langle f, \xi \rangle dt$ done by the force along a path between $g(0)$ and $g(T)$.

can also consult e.g. [2], [3], [4] regarding standard notation as well as more formal introduction to Lie groups and robot dynamics.

The Optimization problem

The system is required to move from a fixed initial state $(g(0), \xi(0))$ to a fixed final state $(g(T), \xi(T))$ during a time interval $[0, T]$. The problem is to find the optimal control $u^* = \text{argmin}_u\, J(u, T)$ where the cost function J is defined by

$$J(u,T) = \frac{1}{2}\int_0^T \|u(t)\|^2 \mathrm{dt}, \tag{2}$$

subject to the dynamics and boundary state conditions.

Related work

Our approach is based on recently developed *structure-preserving* numerical integration and optimal control methods. While standard optimization methods are based on shooting, multiple shooting, or collocation techniques, recent work on Discrete Mechanics and Optimal Control (DMOC) [5] employs *variational integrators* [6], [7] that are derived from the discretization of variational principles such as Hamilton's principle for conservative systems or the Lagrange-D'Alembert principle for dissipative systems. Momentum preservation and symplecticity are automatically enforced, avoiding numerical issues (like numerical dissipation) that generic algorithms often possess.

Structure preservation plays an especially important role for systems on manifolds such as Lie groups which has led to a number of *geometric integration* methods for ordinary differential equations [8]. *Symplectic-momentum integrators* on Lie groups [9], [10] are a particular class of such methods that were combined with ideas developed in the context of *Lie group methods* [11] to construct more general and higher order integrators on Lie groups [12], [13], [14].

The optimal control problem considered in this work has a rich history both in the analytical exploration of its interesting geometric structure as well as in its numerical treatment. In particular, finding trajectories extremizing an action similar to (2) can be equivalently stated as computing geodesics for a higher-order system known as *Riemannian cubics* [15]. Riemannian cubics are generalizations of straight lines on a manifold for which, roughly speaking, the higher-order system velocity corresponds to the acceleration of the original system. When the manifold is a Lie group, such cubics can be reduced by symmetry to Lie quadratics (since the resulting curves are quadratic in the Lie algebra, while the cubics are cubic in the manifold tangent bundle). General optimality conditions as well as insightful geometric invariants have been derived (e.g. [15], [16], [17]) with particular attention to rigid body rotation problems on SO(3) while other works [18], [19], [20] have focused on a more practically computable approach applicable to SE(3).

Note that such works focused on the deriving optimality conditions of the two-state boundary value problem in the standard continuous setting. In contrast, we focus on developing numerical algorithms for computing high-quality solutions. Existing numerical implementations were mainly restricted to simple systems, e.g. ones possessing bi-invariant metrics or fully actuated ones.

Necessary conditions resulting from (Pontryagin's) optimality principle have been derived for simple mechanical systems [4] and for systems on Lie groups [21]. Our work reformulates these problems through a *discrete geometric framework* in order to directly obtain an algorithm for computing optimal solutions with provable numerical properties. Our approach, partially documented in [22], is based on necessary conditions formulated in a manner similar to [23], [24] which study optimal control of rigid bodies. The main difference is that our framework is applicable to any Lie group (not just the Euclidean groups) and offers greater flexibility by allowing different numerical parametrizations as well as underactuation. This generality is accomplished through the formulation of discrete necessary conditions in the spirit of the Riemannian cubics [15] employed in the continuous setting. In essence, optimal trajectories are derived as discrete geodesics of a higher-order action. Following this approach, the numerical formulation requires only the very basic ingredients–the Lagrangian, group structure, control basis, and external forces–and can automatically obtain a solution. Thus, both the regular and higher-order problems are solved using the same general discrete variational approach leading to structure-preserving dynamics and symplectic necessary conditions. Our approach is also linked to the symplectic derivation of optimal control studied in [25] to address the more generic case of an explicit discrete control system evolving in a vector space.

Finally, we point out that group structure and symmetries play an important role in robotic dynamics and motion control [26]. Variational integrators have been used in an interesting way [27] to derive the dynamics of complex multibody systems through recursive differentiation rather than explicitly computing equations of motion. Various control methods have been developed, e.g. [28], [29], to numerically compute optimal trajectories for systems such as the snakeboard or the robotic eel. In relation to such methods, our proposed approach is unique since it builds on a unified discrete variational framework for both deriving the dynamics as well as computing the optimal controls. Note that while this work deals with systems evolving on Lie groups, it can be extended to multi-body systems with nonholonomic constraints following the construction proposed in [30].

Contributions

This paper provides a simple numerical recipe for computing optimal controls driving a mechanical system between two given boundary conditions on pose and velocity. The framework is general and can automatically generate optimal trajectories for any system on a given Lie group by providing its Lagrangian, group structure, and description of acting forces. There are several practical benefits over existing standard methods:

- the algorithm does not require choosing coordinates and avoids issues with expensive chart switching that cause sudden jumps or singularities, e.g. due to gimbal lock, that prevent convergence in iterative optimization;

- the optimization is based on minimum reduced dimension and does not require Lagrange multipliers enforcing, e.g. matrix orthogonality constraints or quaternion unit norms;
- the *discrete mechanics* approach guarantees symplectic structure and momentum preservation as well as energy approximation which is close to the true energy. The combination of these factors leads to an accurate and numerically robust approximation of the dynamics and optimality conditions as a function of the time step (formal and numerical comparisons to standard methods can be found in [7], [31], [14], [32], [30], [27]);
- predictable computation even at lower resolutions allows increased run-time efficiency.

Since the dynamics are nonlinear, an optimal solution generally does not exist in closed-form and must be computed using iterative root-finding. Numerical tests show that our proposed iterative scheme converges to a solution in surprisingly few iterations irrespective of the chosen resolution, even when applied to an underactuated system with external non-conservative forces.

Outline.

After a quick review of regular variational integrators in §II, we present in §III a formal, general treatment of the discrete variational principle used to formulate the numerical optimal control problem for systems on Lie groups. We then present the resulting optimal control algorithms first for fully actuated systems (§IV) and then for underactuated systems with control parameters (§V). The explicit expressions necessary for implementation are given in §VI for any system on the groups SE(2), SO(3), or SE(3), as well as on any general real matrix subgroup. Specific cases of a boat and a satellite are detailed as concrete examples in §VII. We also point out issues related to controllability in the underactuated case and on numerical implementation in §VIII.

II. Background on Variational Integrators

A mechanical integrator advances a dynamical system forward in time. Such numerical algorithms are typically constructed by directly discretizing the differential equations that describe the trajectory of the system, resulting in an *update rule* to compute the next state in time. In contrast, variational integrators [7] are based on the idea that the update rule for a discrete mechanical system (i.e., the time stepping scheme) should be derived *directly* from a variational principle rather than from the resulting differential equations. This concept of using a unifying principle from which the equations of motion follow (typically through the calculus of variations [33]) has been favored for decades in physics. Chief among the variational principles of mechanics is *Hamilton's principle* which states that the path $q(t)$ (with endpoint $q(t_0)$ and $q(t_1)$) taken by a mechanical system extremizes the *action integral* $\int_{t_0}^{t_1} L(q, \dot{q})dt$, i.e., the state variables $(q, \dot{q})$ evolve such that the time integral of the *Lagrangian L* of the system (equal to the kinetic minus potential energy) is extremized. A number of properties of the Lagrangian have direct consequences on the mechanical system. For instance, a *symmetry* of the system (i.e., a transformation that preserves the Lagrangian) leads to a momentum preservation.

Although this variational approach may seem more mathematically motivated than numerically relevant, integrators that respect variational properties exhibit improved numerics and remedy many practical issues in physically based simulation and animation. First, variational integrators automatically preserve (linear and angular) momenta exactly (because of the invariance of the Lagrangian with respect to translation and rotation) while providing good energy conservation over exponentially long simulation times for non-dissipative systems. Second, arbitrarily accurate integrators can be obtained through a simple change of quadrature rules. Finally, they preserve the *symplectic structure* of the system, resulting in a much-improved treatment of damping that is essentially independent of time step [31].

Practically speaking, variational integrators based on Hamilton's principle first approximate the time integral of the continuous Lagrangian by a *quadrature* rule. This is accomplished using a "discrete Lagrangian," which is a function of two consecutive states q_k and q_{k+1} (corresponding to time t_k and t_{k+1}, respectively):

$$L_d(q_k, q_{k+1}) \approx \int_{t_k}^{t_{k+1}} L(q(t), \dot{q}(t))\mathrm{d}t.$$

One can now formulate a discrete principle over the whole path $\{q_0, ..., q_N\}$ defined by the successive position at times $t_k = kh$. This discrete principle requires that

$$\delta \sum_{k=0}^{N-1} L_d(q_k, q_{k+1}) = 0,$$

where variations are taken with respect to each position q_k along the path. Thus, if we use D_i to denote the partial derivative w.r.t the i^{th} variable, we must have

$$D_2 L_d(q_{k-1}, q_k) + D_1 L_d(q_k, q_{k+1}) = 0$$

for every three consecutive positions q_{k-1}, q_k, q_{k+1} of the mechanical system. This equation thus *defines* an integration scheme which computes q_{k+1} using the two previous positions q_k and q_{k-1}.

Simple Example: Consider a continuous, typical Lagrangian of the form $L(q, \dot{q}) = \frac{1}{2}\dot{q}^T M \dot{q} - V(q)$ (V being a potential function) and define the discrete Lagrangian $L_d(q_k, q_{k+1}) = hL\big(q_{k+\frac{1}{2}}, (q_{k+1} - q_k)/h\big)$, using the notation $q_{k+\frac{1}{2}} := (q_k + q_{k+1})/2$. The resulting update equation is:

$$M \frac{q_{k+1} - 2q_k + q_{k-1}}{h^2} = -\frac{1}{2}(\nabla V(q_{k-\frac{1}{2}}) + \nabla V(q_{k+\frac{1}{2}})),$$

which is a discrete analog of Newton's law $M\ddot{q} = -\nabla V(q)$. This example can be easily generalized by replacing $q_{k+1/2}$ by $q_{k+\alpha} = (1-\alpha)\, q_k + \alpha\, q_{k+1}$ as the quadrature point used to approximate the discrete Lagrangian, leading to variants of the update equation. For controlled (i.e., non conservative) systems, forces can be added using a discrete version of Lagrange-d'Alembert principle in a similar manner [34].

III. Discrete Mechanics on Lie Groups

Variational integration on Lie groups requires additional development since a Lie group is a nonlinear space with special structure. We first recall the standard variational principle for mechanical systems on Lie groups and state the resulting differential equations of motion. A discrete structure-preserving dynamics update scheme is then constructed. It will serve as a basis for developing the proposed optimal control algorithms.

A. The Continuous Setting

The state trajectory is formally defined as $(g,\xi):[0,T]\to G\times\mathfrak{g}$ while the control force takes the form $f:[0,T]\to\mathfrak{g}^*$. In most practical cases one can regard g as a matrix, ξ as a column vector, and f as a row vector which "pairs" with velocities through a product $\langle\cdot,\cdot\rangle$ such as the standard dot product. The Lagrange-d'Alembert principle requires that

$$\delta\int_0^T [K(\xi)-V(g)]\,\mathrm{d}t+\int_0^T\langle f,g^{-1}\delta g\rangle=0, \tag{3}$$

where $\xi=g^{-1}\dot g$. The curve $\xi(t)$ describes the body-fixed velocity determined from the dynamics of the system. Variations of the velocity ξ and the configuration g are related through

$$\delta\xi=\dot\eta+\operatorname{ad}_\xi\eta,\ \text{ for } \eta=g^{-1}\delta g\in\mathfrak{g},$$

where the Lie bracket operator for matrix groups $\operatorname{ad}:\mathfrak{g}\times\mathfrak{g}\to\mathfrak{g}$ is defined by

$$\operatorname{ad}_\xi\eta=\xi\eta-\eta\xi,$$

for given $\xi,\eta\in\mathfrak{g}$. The continuous equations of motion become (see e.g. [35], [4])

$$\dot\mu-\operatorname{ad}^*_\xi\mu=-g^*\partial_gV(g)+f, \tag{4a}$$
$$\mu=\partial_\xi K(\xi), \tag{4b}$$
$$\dot g=g\xi. \tag{4c}$$

These equations[2] are called the controlled Euler-Poincaré equations and $\mu\in\mathfrak{g}^*$ denotes the system momentum. Given initial conditions, the momentum μ evolves according to (4a). The velocity ξ can be computed in terms of μ using (4b) since ∂_ξ^2K is non-singular allowing the inversion of $\partial_\xi K$. The configuration then evolves according to (4c).

B. Trajectory Discretization

A trajectory is represented numerically using a set of $N+1$ equally spaced in time points denoted $g_{0:N}:=\{g_0,...,g_N\}$, where $g_k\approx g(kh)\in G$ and $h=T/N$ denotes the time-step. The section between each pair of points g_k and g_{k+1} is interpolated by a short curve that must lie on the manifold (Fig. 1). The simplest way to construct such a curve is through a map $\tau:\tau:\mathfrak{g}\to G$ and Lie algebra element $\xi_k\in\mathfrak{g}$ such that $\xi_k=\tau^{-1}(g_k^{-1}g_{k+1})/h$. The map is defined as follows.

Definition III.1. The *retraction map* $\tau:\mathfrak{g}\to G$ is a C^2-diffeomorphism around the origin such that $\tau(0)=e$. It is used to express small discrete changes in the group configuration through unique Lie algebra elements.

Thus, if ξ_k were regarded as an average velocity between g_k and g_{k+1} then τ is an approximation to the integral flow of the dynamics. The important point, from a numerical point of view, is that the difference $g_k^{-1}g_{k+1}\in G$, which is an element of a nonlinear space, can now be represented uniquely by the vector ξ_k in order to enable unconstrained optimization in the linear space $\mathfrak{g}$ for optimal control purposes.

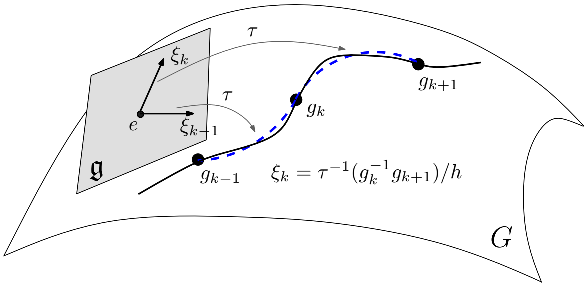


Fig. 1. A trajectory (solid) on the Lie group G discretized using a sequence of arcs (dashed) represented by Lie algebra vectors $\xi_k\in\mathfrak{g}$ through the retraction map τ.

Next, we define the following operators related to τ.

Definition III.2. [11], [14] Given a map $\tau:\mathfrak{g}\to G$, its *right-trivialized tangent* $\mathrm{d}\tau_\xi:\mathfrak{g}\to\mathfrak{g}$ and its *inverse* $\mathrm{d}\tau_\xi^{-1}:\mathfrak{g}\to\mathfrak{g}$ are such that, for a some $g=\tau(\xi)\in G$ and $\eta\in\mathfrak{g}$, the following holds

$$\partial_\xi\tau(\xi)\cdot\eta=\mathrm{d}\tau_\xi\cdot\eta\cdot\tau(\xi), \tag{5}$$
$$\partial_\xi\tau^{-1}(g)\cdot\eta=\mathrm{d}\tau_\xi^{-1}\cdot(\eta\cdot\tau(-\xi)). \tag{6}$$

Note that it can be shown by differentiating the expression $\tau^{-1}(\tau(\xi))=\xi$ that

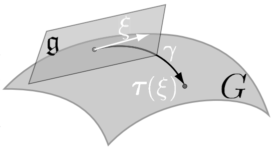
$$d\tau_\xi^{-1}\cdot d\tau_\xi\cdot\eta=\eta,$$

to confirm that these linear maps are indeed the inverse of each other.

Intuitively, after the derivative in the direction η is taken, i.e. $\partial_\xi\tau(\xi)\cdot\eta$, the resulting vector (at point $\tau(\xi)$) is translated back to the origin using right-multiplication by $\tau(-\xi)$ [14]. In practice, as will be shown, these maps are easily derived as $n\times n$ matrices. Finally, we require the tangent maps to be nonsingular over the optimization domain, defined next.

Definition III.3. The *optimization domain* $\mathfrak{D}_\tau\subset\mathfrak{g}$ is a connected open set containing the origin $e\in\mathfrak{g}$ such that $\mathrm{d}\tau_{h\xi}$ (and $\mathrm{d}\tau_{h\xi}^{-1}$) are non-singular for every $\xi\in\mathfrak{D}_\tau$.

The numerical algorithms proposed in the paper are restricted to operate over $\mathfrak{D}_\tau$, i.e. the time-step h and velocities ξ_k are chosen to satisfy $h\xi_k\in\mathfrak{D}_\tau$ for all $k=0,...,N-1$.



Retraction Map (τ) Choices: a) The *exponential map* $\exp:\mathfrak{g}\to G$, defined by $\exp(\xi)=\gamma(1)$, with $\gamma:\mathbb{R}\to G$ is the integral curve through the identity of the vector field associated with $\xi\in\mathfrak{g}$ (hence, with $\dot\gamma(0)=\xi$). The right-trivialized derivative of the map exp

[2] $\operatorname{ad}^*_\xi\mu$ is defined by $\langle\operatorname{ad}^*_\xi\mu,\eta\rangle=\langle\mu,\operatorname{ad}_\xi\eta\rangle$, for some $\eta\in\mathfrak{g}$.

and its inverse are defined as

$$\mathrm{dexp}_x\, y = \sum_{j=0}^{\infty} \frac{1}{(j+1)!}\,\mathrm{ad}_x^j\, y, \tag{7a}$$

$$\mathrm{dexp}_x^{-1}\, y = \sum_{j=0}^{\infty} \frac{B_j}{j!}\,\mathrm{ad}_x^j\, y, \tag{7b}$$

where B_j are the Bernoulli numbers. Typically, these expressions are truncated in order to achieve a desired order of accuracy. The first few Bernoulli numbers are $B_0 = 1$, $B_1 = -1/2$, $B_2 = 1/6$, $B_3 = 0$ (see [8]).

b) The *Cayley map* $\mathrm{cay} : \mathfrak{g} \to G$ is defined by $\mathrm{cay}(\xi) = (\mathrm{I} - \xi/2)^{-1}(\mathrm{I} + \xi/2)$ and is valid for a general class for quadratic groups that include the groups of interest in the paper. Based on this simple form, the derivative maps become ([8], §IV.8.3)

$$\mathrm{dcay}_x\, y = \left(e - \frac{x}{2}\right)^{-1} y \left(e + \frac{x}{2}\right)^{-1}, \tag{8a}$$

$$\mathrm{dcay}_x^{-1}\, y = \left(e - \frac{x}{2}\right) y \left(e + \frac{x}{2}\right). \tag{8b}$$

The third choice is to use canonical coordinates of the second kind (ccsk) [8] which are based on the exponential map and are not considered in this paper.

C. Discrete Variational Formulation

With a discrete trajectory in place we follow the approach taken in [10], [9] in order to construct a structure-preserving (i.e. group, momentum, and symplectic) integrator of the dynamics. We make a simple extension to include potential and control forces through a trapezoidal quadrature approximation. In particular, the action in (3) is approximated along each discrete segment between points g_k and g_{k+1} through

$$\int_{kh}^{(k+1)h} [K(\xi) - V(g)]\mathrm{d}t \approx h\left[K(\xi_k) - \frac{V(g_k) + V(g_{k+1})}{2}\right], \tag{9a}$$

$$\int_{kh}^{(k+1)h} \langle f, g^{-1}\delta g\rangle \approx \frac{h}{2}\left[\langle f_k, g_k^{-1}\delta g_k\rangle + \langle f_{k+1}, g_{k+1}^{-1}\delta g_{k+1}\rangle\right]. \tag{9b}$$

Variations of Lie algebra elements are related to variations on the group through the following expression which may be obtained through differentiation and application of (6),

$$\delta\xi_k = \delta\tau^{-1}(g_k^{-1}g_{k+1})/h = \mathrm{d}\tau_{h\xi_k}^{-1}(-\eta_k + \mathrm{Ad}_{\tau(h\xi_k)}\,\eta_{k+1})/h,$$

where $\eta_k = g_k^{-1}\delta g_k$. The operator $\mathrm{Ad}_g : \mathfrak{g} \to \mathfrak{g}$ can be regarded as a change of basis with respect to the argument $g \in G$ (see [3], [35]) and is defined by

$$\mathrm{Ad}_g\,\xi = g\xi g^{-1}.$$

The discrete variational principle which will form the basis for our discrete optimal control framework can now be stated. The following result is a straightforward extension from [10], [9]. The only difference is that we consider Lagrangians of the form (1) and employ a trapezoidal discretization:

Proposition 1. *A mechanical system on Lie group G with kinetic energy K, potential energy V, subject to forces f, satisfies the following* equivalent *conditions:*

1. *The discrete reduced Lagrange-d'Alembert principle holds*

$$\begin{aligned}\delta &\sum_{k=0}^{N-1}\left[K(\xi_k) - \frac{V(g_k) + V(g_{k+1})}{2}\right] \\ &+ \sum_{k=0}^{N-1}\frac{1}{2}\left[\langle f_k, g_k^{-1}\delta g_k\rangle + \langle f_{k+1}, g_{k+1}^{-1}\delta g_{k+1}\rangle\right] = 0,\end{aligned} \tag{10}$$

   where $\xi_k = \tau^{-1}(g_k^{-1}g_{k+1})/h$.
2. *The discrete reduced Euler-Poincaré equations of motion hold*

$$\mu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\,\mu_{k-1} = h\left(-g_k^*\partial_g V(g_k) + f_k\right), \tag{11a}$$

$$\mu_k = (\mathrm{d}\tau_{h\xi_k}^{-1})^*\partial_\xi K(\xi_k), \tag{11b}$$

$$g_{k+1} = g_k\tau(h\xi_k). \tag{11c}$$

Equations (11) can be considered as a discrete approximation to (4). The *discrete Euler-Poincaré* equation (11a) corresponds to (4a). Eq. (11b) is the *discrete Legendre transform* corresponding to (4b), while (11c) is the *discrete reconstruction* analogous to (4c). These equations can be used to compute the next velocity and group elements ξ_k, and g_{k+1}, respectively, given the previous elements ξ_{k-1} and g_k. Fig. 2 gives a more geometric explanation of the update (11a).

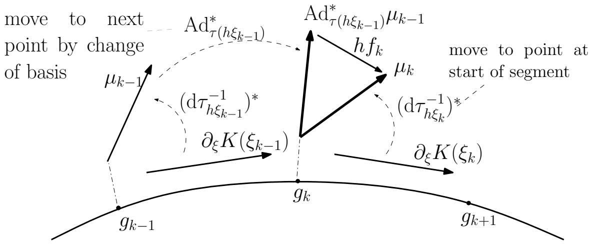

move to next point by change of basis
$\mathrm{Ad}^*_{\tau(h\xi_{k-1})}$
$\mathrm{Ad}^*_{\tau(h\xi_{k-1})}\mu_{k-1}$
hf_k
move to point at start of segment
μ_{k-1}
μ_k
$(\mathrm{d}\tau^{-1}_{h\xi_{k-1}})^*$
$(\mathrm{d}\tau^{-1}_{h\xi_k})^*$
$\partial_\xi K(\xi_{k-1})$
$\partial_\xi K(\xi_k)$
g_k
g_{k-1}
g_{k+1}

Fig. 2. The discrete covariant version of the Euler-Poincaré equation $\dot{\mu} - \mathrm{ad}^*_\xi\,\mu = f$, where $\mu = \partial_\xi K(\xi)$. The discrete momentum μ_k at point k is obtained using the right trivialized tangent $(\mathrm{d}\tau^{-1}_{h\xi_k})^*$ which brings the derivative $\partial_\xi K(\xi_k)$ to the body-fixed basis at g_k. The momentum evolution is then expressed through the difference of μ_{k-1} and μ_k, i.e. by transforming μ_{k-1} in that same basis at g_k through the Ad^* map where proper vector subtraction can be applied. The resulting change is caused by forces hf_k (the effects of potential V are omitted for clarity). Note that all vectors shown are elements of $\mathfrak{g}^*$ and are shown above the group configuration only to illustrate the basis with respect to which they are defined.

Boundary Conditions: While the discrete configurations g_k and forces f_k approximate their continuous counterparts at times $t = kh$, we still have not established the exact relationship between the discrete and continuous momenta, μ_k and $\mu(t) = \partial_\xi K(\xi(t))$, respectively. This is particularly important for properly enforcing boundary conditions that are given in terms of continuous quantities. The following equations (12a) and (12b) relate the momenta at the initial and final times $t = 0$ and $t = T$ and are used to transform between the continuous and discrete representation:

$$\mu_0 - \partial_\xi K(\xi(0)) = \frac{h}{2}\left(g_0^*\partial_g V(g_0) + f_0\right), \tag{12a}$$

$$\partial_\xi K(\xi(T)) - \mathrm{Ad}^*_{\tau(h\xi_{N-1})}\mu_{N-1} = \frac{h}{2}\left(g_N^*\partial_g V(g_N) + f_N\right). \tag{12b}$$

These equations can also be regarded as structure-preserving *velocity boundary conditions* for given fixed velocities $\xi(0)$

and $\xi(T)$. They follow from properly enforcing energy balance at the boundaries, achieved by adding the momentum change term $\langle \mu(T), g_N^{-1}\delta g_N\rangle - \langle \mu(0), g_0^{-1}\delta g_0\rangle$ to the discrete action in the principle (10).

The exact form of (11) and (12) depends on the choice of τ. It is important to point out that this choice will influence the computational efficiency of the optimization framework when the equalities above are enforced as constraints. We have specified two basic choices, $\tau = \exp$ (7a) and $\tau = \mathrm{cay}$ (8a) for their ease of implementation and run-time efficiency.

D. Preservation Properties

One of the main benefits of employing the variational numerical framework lies in its preservation properties, summarized by the following theorems.

Theorem III.4. *[9] The discrete flow* (11) *preserves the discrete symplectic form, expressed in coordinates as*

$$\omega_\ell = \frac{\partial^2 \ell(g_k, \tau^{-1}(g_k^{-1} g_{k+1})/h)}{\partial g_k^i \partial g_{k+1}^j} \, \mathrm{d}g_k^i \wedge \mathrm{d}g_{k+1}^j,$$

where $\wedge$ is the standard wedge product between differential forms [35]. The symplectic form can also be written as the differential of the canonical one-form θ_ℓ with $\omega_\ell = \mathrm{d}\theta_\ell$ where

$$\theta_\ell \cdot \delta g_k = \left\langle -\frac{1}{h}\mu_k - g_k^* \partial_g V(g_k), g_k^{-1}\delta g_k \right\rangle.$$

The symplectic form is physically related to the phase space structure. Its preservation during integration, for instance, signifies that a volume of initial conditions would not be spuriously inflated or deflated due to numerical approximations. Volume preservation means that the orbits of the dynamics will have a predictable character and no artificial damping normally employed by Runge-Kutta methods is needed to stabilize the system [7].

Theorem III.5. *[9], [14] The discrete dynamics* (11) *preserves the momentum. In particular, in the absence of potential and non-conservative forces, the update scheme preserves the discrete spatial momentum map $J : G \times \mathfrak{g} \to \mathfrak{g}$,*

$$J(g_k, \xi_k) \cdot v = \mathrm{Ad}^*_{g_k^{-1}} \mu_k \cdot v,$$

for any $v \in \mathfrak{g}$; or equivalently $J(g_a, \xi_a) = J(g_b, \xi_b)$, for any time indices a, b.

Practically speaking, whenever the continuous system preserves momentum, so does the discrete. Any change in the momentum then exactly reflects the work done by non-conservative forces. Such a momentum-symplectic scheme also exhibits long-term stable energy behavior close to the true system energy [7]. Another property carried over to continuous case is time-scaling.

Remark 1. *Order of Accuracy.* The order of accuracy of the dynamics update depends on the accuracy of the Lagrangian approximation. Since the trapezoidal approximations (9a) and (9b) are second-order accurate then it can be shown (see [7]) that the discrete equations (11) are also of *second order accuracy*. The trapezoidal rule was chosen since it provides the simplest second-order scheme. Higher-order methods by proper choice of the Lagrangian, termed symplectic Runge-Kutta (see [8], [12], [14]), are possible but not considered in this work.

Remark 2. *Time-scaling preservation.* The trajectory $g_{0:N}$, $\xi_{0:N-1}$ with time-step h satisfies the discrete dynamics (11) subject to forces $f_{0:N}$ if and only if the trajectory $g_{0:N}$, $\{\xi_0/s, ..., \xi_{N-1}/s\}$ with time-step $h' = sh$, subject to forces $\{f_0/s^2, ..., f_N/s^2\}$ satisfies the discrete dynamics, for a given scalar $s > 0$.

Finally, the group structure is exactly preserved since the trajectory $g_{0:N}$ is reconstructed from the discrete velocity $\xi_{0:N-1}$ using the map τ which by definition maps to the group (11c). This avoids issues with dissipation and numerical drift associated with reprojection used in other methods, e.g. in methods based on matrix orthogonality constraints or quaternions.

IV. Fully Actuated Systems

We first develop the simplest case with a mechanical kinetic energy

$$K(\xi) = \frac{1}{2}\langle \mathbb{I}\xi, \xi\rangle,$$

with full unconstrained actuation, without potential or external forces and without any velocity constraints. The map $\mathbb{I} : \mathfrak{g} \to \mathfrak{g}^*$ is called the *inertia tensor* and is assumed full rank. Since there is full control over f the control effort cost function (2) can be expressed as $J(f) = \int \frac{1}{2}\|f(t)\|^2 \mathrm{d}t$. It is approximated through trapezoidal quadrature, analogously to (10), using the summation

$$J(f) \approx \sum_{k=0}^{N-1} \frac{h}{4}\left(\|f_k\|^2 + \|f_{k+1}\|^2\right). \tag{13}$$

The optimal control problem for the system (11) with given fixed initial and final states $(g(0), \xi(0))$ and $(g(T), \xi(T))$ respectively can be stated as

Compute: $\xi_{0:N-1}, f_{0:N}$

minimizing $\displaystyle\sum_{k=0}^{N-1} \frac{h}{4}\left(\|f_k\|^2 + \|f_{k+1}\|^2\right)$

subject to:

$$\begin{cases} \mu_0 - \mathbb{I}\,\xi(0) = (h/2) f_0, \\ \mu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\mu_{k-1} = h f_k, \quad k = 1, ..., N-1, \\ \mathbb{I}\,\xi(T) - \mathrm{Ad}^*_{\tau(h\xi_{N-1})}\mu_{N-1} = (h/2) f_N, \\ \mu_k = (\mathrm{d}\tau^{-1}_{h\xi_k})^* \,\mathbb{I}\,\xi_k, \\ g_0 = g(0), \\ g_{k+1} = g_k \tau(h\xi_k), \qquad\qquad k = 0, ..., N-1, \\ \tau^{-1}(g_N^{-1} g(T)) = 0. \end{cases} \tag{14}$$

The constraints follow directly from the discrete mechanics (11), boundary conditions (12), and by noting that $\partial_\xi K = \mathbb{I}\xi$. The last equation ensures that the difference between the given and reconstructed configurations is zero.

A. Optimality Conditions

Trajectories satisfying the constrained nonlinear optimization problem (14) are computed through the derivation of optimality conditions stated in the following proposition.

Proposition 2. *The trajectory of a discrete mechanical system on a Lie group G with algebra $\mathfrak{g}$ and Lagrangian $\ell(\xi) = \frac{1}{2}\langle \mathbb{I}\,\xi, \xi\rangle$ with fixed initial and final configurations and velocities $(g(0), \xi(0)) \in G \times \mathfrak{g}$ and $(g(T), \xi(T)) \in G \times \mathfrak{g}$ minimizes the total control effort only if the discrete body-fixed velocity curve $\xi_{0:N-1}$ satisfies the following conditions:*

Necessary Conditions for Optimality

$$\nu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\,\nu_{k-1} = 0, \qquad k = 1, ..., N-1 \tag{15a}$$

$$\tau^{-1}(\tau(h\xi_0)\cdots\tau(h\xi_{N-1})\cdot(g(0)^{-1}g(T))^{-1}) = 0, \tag{15b}$$

where:

$$\nu_k = (\mathrm{d}\tau^{-1}_{h\xi_k})^*\partial_\xi \mathcal{K}_{(\lambda_{0:N},k)}(\xi_k), \tag{15c}$$

$$\mathcal{K}_{(\lambda_{0:N},k)}(\xi_k) = \langle(\mathrm{d}\tau^{-1}_{h\xi_k})^*\mathbb{I}\xi_k, \lambda_k - \mathrm{Ad}_{\tau(h\xi_k)}\lambda_{k+1}\rangle/h, \tag{15d}$$

$$\lambda_0^\flat = 2\left(\mu_0 - \mathbb{I}\,\xi(0)\right)/h, \tag{15e}$$

$$\lambda_k^\flat = \left(\mu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\,\mu_{k-1}\right)/h, \quad k = 1, ..., N-1 \tag{15f}$$

$$\lambda_N^\flat = 2\left(\mathbb{I}\,\xi(T) - \mathrm{Ad}^*_{\tau(h\xi_{N-1})}\,\mu_{N-1}\right)/h, \tag{15g}$$

$$\mu_k = (\mathrm{d}\tau^{-1}_{h\xi_k})^*\,\mathbb{I}\xi_k. \tag{15h}$$

Note: The proposition defines Nn equations (15a)-(15b) *in the Nn unknowns $\xi_0, ..., \xi_{N-1}$. A solution can be found using nonlinear root finding.*

Proof: Define the discrete action $\mathcal{S}$ according to

$$\begin{aligned}
&\mathcal{S}(\xi_{0:N-1}, f_{0:N}, \lambda_{0:N}) \\
&= \langle \mu_0 - \mathbb{I}\xi(0) - hf_0/2, \lambda_0\rangle \\
&\quad + \sum_{k=1}^{N-1}\langle \mu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\,\mu_{k-1} - hf_k, \lambda_k\rangle \\
&\quad + \langle \mathbb{I}\xi(T) - \mathrm{Ad}^*_{\tau(h\xi_{N-1})}\,\mu_{N-1} - hf_N/2, \lambda_N\rangle \\
&\quad + \sum_{k=0}^{N-1}\frac{h}{4}\left(\|f_k\|^2 + \|f_{k+1}\|^2\right)
\end{aligned}\quad , \tag{16}$$

where $\mu_k = (\mathrm{d}\tau^{-1}_{h\xi_k})^*\,\mathbb{I}\xi_k$ should be regarded as a function of ξ_k. Taking variations $\delta\mathcal{S}$ with respect to f_k and λ_k we obtain[3]

$$\lambda_k^\flat = f_k = (\mu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\,\mu_{k-1})/h.$$

Next, freeze the adjoint trajectory $\lambda_{0:N}$ and define the functions $\mathcal{K}_{(\lambda_{0:N},k)} : \mathfrak{g} \to \mathbb{R}$, for $k = 0, ..., N-1$ by

$$\mathcal{K}_{(\lambda_{0:N},k)}(\xi) = \langle(\mathrm{d}\tau^{-1}_{h\xi})^*\,\mathbb{I}\xi, \lambda_k - \mathrm{Ad}_{\tau(h\xi)}\,\lambda_{k+1}\rangle/h. \tag{17}$$

The ξ-dependent discrete action along fixed $\lambda_{0:N}$ can be rewritten as

$$\mathcal{S}_{\lambda_{0:N}}(\xi_{0:N-1}) = h\sum_{k=0}^{N-1}\mathcal{K}_{(\lambda_{0:N},k)}(\xi_k).$$

For less cluttered notation the shorthand expression

$$\mathcal{K}(\xi_k) := \mathcal{K}_{(\lambda_{0:N},k)}(\xi_k), \tag{18}$$

will also be employed since the index k in $\mathcal{K}_{(\lambda_{0:N},k)}$ becomes clear from the argument ξ_k. The point is that λ in (17) should be regarded as fixed, i.e. not dependent on ξ. The optimality conditions can now be regarded as a set of equations satisfying the dynamics of another *higher order* discrete *Hamiltonian system* with discrete Lagrangian $\mathcal{L} = \mathcal{K}$ through

$$\delta\mathcal{S}_{\lambda_{0:N}}(\xi_{0:N-1}) = 0 \iff \nu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\,\nu_{k-1} = 0, \tag{19}$$

where $\nu_k = (\mathrm{d}\tau^{-1}_{h\xi_k})^*\partial_\xi\mathcal{K}(\xi_k) \in \mathfrak{g}^*$ is a momentum-like quantity for the system with Lagrangian $\mathcal{L}$. The relation (19) is nothing but the discrete Euler-Poincaré equation of this new system and was obtained in the same way the standard dynamics update (11a) followed from the principle (10). This key insight leads directly to a convenient numerical scheme for computing the optimal controls.

The final configuration g_N is computed by reconstructing the curve from the velocities $\xi_{0:N-1}$ and the boundary condition $g_N = g(T)$ is enforced through the relation (15b) without the need to optimize over any of the configurations g_k. ■

We point out the resulting formulation does not require optimizing over additional Lagrange multiplier variables. It has the minimum possible problem dimension and avoids convergence and instability issues due to improper multiplier initialization.

B. Implementing the Necessary Conditions.

An optimal trajectory is computed as the root of equations (15a)-(15b). Their exact form depends on the momentum expression (15c) which can be computed numerically using finite differences, e.g. using:

$$\begin{aligned}
&\langle\nu_k, \eta\rangle \\
&\approx \frac{1}{2\epsilon}\left[\mathcal{K}_{(\lambda_{0:N},k)}(\xi_k + \epsilon\,\mathrm{d}\tau^{-1}_{h\xi_k}\eta) - \mathcal{K}_{(\lambda_{0:N},k)}(\xi_k - \epsilon\,\mathrm{d}\tau^{-1}_{h\xi_k}\eta)\right],
\end{aligned} \tag{20}$$

along basis elements $\eta \in \mathfrak{g}$ with a small $\epsilon > 0$. In other words, the components of ν_k with respect to a chosen Lie algebra basis $\{e_i\}$ are computed according to $\nu_k^i = \langle\nu_k, e_i\rangle$ for any Lie group G.

Alternatively, the momentum can be expressed in closed form by differentiating the kinetic energy $\mathcal{K}$ to obtain

$$\begin{aligned}
&\langle\nu_k, \eta\rangle \\
&= \langle(\partial_\xi \mathrm{d}\tau^{-1}_{h\xi_k}\cdot\mathrm{d}\tau^{-1}_{h\xi_k}\eta)^*\,\mathbb{I}\,\xi_k, \Delta\lambda_k\rangle \\
&\quad + \langle(\mathrm{d}\tau^{-1}_{h\xi_k})^*\,\mathbb{I}(\mathrm{d}\tau^{-1}_{h\xi_k})\Delta\lambda_k + h\,\mathrm{ad}^*_{\mathrm{Ad}_{\tau(h\xi_k)}\lambda_{k+1}}\,\mu_k, \eta\rangle,
\end{aligned} \tag{21}$$

where $\Delta\lambda_k = \lambda_k - \mathrm{Ad}_{\tau(h\xi_k)}\,\lambda_{k+1}$. Expression (21) is derived using straightforward differentiation (one can also consult [22] for more details) and using A.1. One can choose to implement the necessary conditions using either (20) or (21).

V. Underactuated Systems with Control Parameters

We next extend the system dynamics to include non-trivial actuation and position dependent forces. Assume that the control forces are applied along body-fixed directions defined by

[3] The superscript operators $\flat : \mathfrak{g} \to \mathfrak{g}^*$ (flat) and $\sharp : \mathfrak{g}^* \to \mathfrak{g}$ (sharp) are used to convert between vector fields and their duals (one-forms). Under identification $\mathfrak{g} \sim \mathbb{R}^n$, $\flat$ can simply be regarded as converting a column vector into a row vector, and $\sharp$ as the opposite operation [35].

the *control covectors* $\{f^1(\phi), ..., f^c(\phi)\}$, $c \leq n$, $f^i : \mathbb{M} \to \mathfrak{g}^*$ which depend on *control parameters* $\phi : [0, T] \to \mathbb{M}$. These extra parameters can be regarded as the shape variables of the control basis, i.e. parameters that do not affect the inertial properties of the systems but which determine the control directions. Assume that the system is controlled using *control input* $u : [0, T] \to \mathbb{U}$ applied with respect to the basis $\{f^i(\phi)\}$. In addition, assume that the system is subject to *configuration-dependent forces* collectively represented by the function $f_{\text{conf}} : G \to \mathfrak{g}^*$ and dissipative *velocity-dependent forces* $f_{\text{vel}} : \mathfrak{g} \to \mathfrak{g}^*$. For instance, forces arising from the potential V take the form $f_{\text{conf}}(g) = -g^*\partial_g V(g)$. while simple viscous resistance or linear drag forces can be expressed as $f_{\text{vel}}(\xi) = -D\xi$, where D is damping positive definite map.

The total force acting in the body frame can then be expressed as the sum of the control and external forces according to

$$f(g, \xi) = \sum_{i=1}^{c} u^i f^i(\phi) + f_{\text{conf}}(g) + f_{\text{vel}}(\xi).$$

In this problem the control effort to be minimized is expressed as $\int_0^T \frac{1}{2}\|u(t)\|^2 \mathrm{d}t$.

Dissipative Force discretization: In our framework velocity-dependent forces $f_{\text{vel}}(\xi_k)$ are defined over the k-th segment, and have no clear meaning over a particular point. The contribution of such forces at a particular point can be specified by assuming the following virtual work approximation

$$\int_{kh}^{(k+1)h} f_{\text{vel}}(\xi)\cdot\eta(t) \approx \frac{h}{2}\langle(\mathrm{d}\tau_{h\xi_k}^{-1})^* f_{\text{vel}}(\xi_k), \eta_k + \mathrm{Ad}_{\tau(h\xi_N)}\eta_{k+1}\rangle,$$

where $\eta = g^{-1}\delta g$ denotes the usual Lie group variations. Such discretization is motivated by the way variations contribute to Hamilton's principle discretization (10)

$$\delta\left(\int_{kh}^{(k+1)h} \ell(\xi)\mathrm{d}t\right)\cdot\eta \approx h\langle(\mathrm{d}\tau_{h\xi_k}^{-1})^*\partial_\xi\ell(\xi_k), -\eta_k + \mathrm{Ad}_{\tau(h\xi_N)}\eta_{k+1}\rangle,$$

where the left and right variations are averaged instead of subtracted. Fig. 2 also helps explain how vectors defined along a segment transform to its start and end points.

The necessary conditions for an optimal trajectory are defined in the following proposition (which extends Prop. 2).

Proposition 3. *A discrete mechanical system with kinetic energy $K(\xi)$ and control input directions $f^i(\phi)$ subject to configuration and dissipative forces $f_{conf}(g)$ and $f_{vel}(\xi)$, respectively, moves with minimum control effort between fixed initial and final states $(g(0), \xi(0)) \in G \times \mathfrak{g}$, $((g(T), \xi(T)) \in G \times \mathfrak{g}$, only if the discrete velocity curve $\xi_{0:N-1}$, control parameters $\phi_{0:N}$, and adjoint variables $\lambda_{0:N}$ satisfy the following conditions:*

Necessary Conditions for Optimality

$$\nu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\nu_{k-1} = -hg_k^*\partial_g\langle f_{conf}(g_k), \lambda_k\rangle, \quad k = 1, ..., N-1 \tag{22a}$$

$$\tau^{-1}(g_N^{-1}g(T)) = 0, \tag{22b}$$

$$\mu_0 - \mathbb{I}\,\xi(0) = (h/2)f_0^+, \tag{22c}$$

$$\mu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\mu_{k-1} = (h/2)(f_k^- + f_k^+), \quad k = 1, ..., N-1 \tag{22d}$$

$$\mathbb{I}\,\xi(T) - \mathrm{Ad}^*_{\tau(h\xi_{N-1})}\,\mu_{N-1} = (h/2)f_N^-, \tag{22e}$$

$$\sum_{i=1}^{c} u_k^i\left(\partial_\phi f^i(\phi_k)^\sharp\right)\lambda_k = 0, \qquad k = 0, ..., N \tag{22f}$$

where $\nu_k \in \mathfrak{g}^*$, $f_k^\pm \in \mathfrak{g}^*$, $u_k \in \mathbb{U}$ are defined by

$$\nu_k = (\mathrm{d}\tau_{h\xi_k}^{-1})^*\partial_\xi\mathcal{K}_{\lambda_{0:N},k}(\xi_k), \tag{22g}$$

$$\begin{aligned}\mathcal{K}_{\lambda_{0:N},k}(\xi_k) = {} & \langle(\mathrm{d}\tau_{h\xi_k}^{-1})^*\mathbb{I}\xi_k, \lambda_k - \mathrm{Ad}_{\tau(h\xi_k)}\,\lambda_{k+1}\rangle/h \\ & - \frac{1}{2}\langle(\mathrm{d}\tau_{h\xi_k}^{-1})^* f_{vel}(\xi_k), \lambda_k + \mathrm{Ad}_{\tau(h\xi_k)}\lambda_{k+1}\rangle,\end{aligned}$$

$$f_k^- = \sum_{i=1}^{c} u_k^i f^i(\phi_k) + f_{conf}(g_k) + (\mathrm{d}\tau_{-h\xi_{k-1}}^{-1})^* f_{vel}(\xi_{k-1}),$$

$$f_k^+ = \sum_{i=1}^{c} u_k^i f^i(\phi_k) + f_{conf}(g_k) + (\mathrm{d}\tau_{h\xi_k}^{-1})^* f_{vel}(\xi_k),$$

$$u_k^i = \langle f^i(\phi_k), \lambda_k\rangle, \tag{22h}$$

$$g_0 = g(0), \tag{22i}$$

$$g_{k+1} = g_k\tau(h\xi_k), \tag{22j}$$

Note: The proposition defines $(Nn + (N+1)n + (N+1)m)$ equations (22a)-(22f) *in the $(Nn + (N+1)n + (N+1)m)$ unknowns $(\xi_{0:N-1}, \lambda_{0:N}, \phi_{0:N})$. A solution can be found using standard nonlinear root finding. When the control basis is constant (i.e. $m = 0$) then the optimization is over $(\xi_{0:N-1}, \lambda_{0:N})$ only; $f^i(\phi_k)$ should be replaced with f^i and* (22f) *is omitted.*

Proof: Define the discrete action $\mathcal{S}$ similarly to (16) according to

$$\begin{aligned}\mathcal{S}(\xi_{0:N-1}, u_{0:N}, \phi_{0:N}, \lambda_{0:N}) = {} & \\ & \langle\mu_0 - \mathbb{I}\xi(0) - (h/2)f_0^+, \lambda_0\rangle \\ & + \sum_{k=1}^{N-1}\langle\mu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\,\mu_{k-1} - (h/2)(f_k^- + f_k^+), \lambda_k\rangle \\ & + \langle\mathbb{I}\xi(T) - \mathrm{Ad}^*_{\tau(h\xi_{N-1})}\,\mu_{N-1} - (h/2)f_N^-, \lambda_N\rangle \\ & + \sum_{k=0}^{N-1}\frac{h}{4}\left(\|u_k\|^2 + \|u_{k+1}\|^2\right),\end{aligned} \tag{23}$$

where we used the shorthand notation

$$f_k^- = \sum_{i=1}^{c} u_k^i f^i(\phi_k) + f_{\text{conf}}(g_k) + (\mathrm{d}\tau_{-h\xi_{k-1}}^{-1})^* f_{\text{vel}}(\xi_{k-1}),$$

$$f_k^+ = \sum_{i=1}^{c} u_k^i f^i(\phi_k) + f_{\text{conf}}(g_k) + (\mathrm{d}\tau_{h\xi_k}^{-1})^* f_{\text{vel}}(\xi_k).$$

Analogously to the fully actuated case (17), keep the multiplier trajectory $\lambda_{0:N}$ frozen and define the function

$\mathcal{K}_{(\lambda_{0:N},k)} : \mathfrak{g} \to \mathbb{R}$ by

$$\begin{aligned}\mathcal{K}_{(\lambda_{0:N},k)}(\xi) =&\langle (\mathrm{d}\tau_{h\xi}^{-1})^* \mathbb{I}\xi, \lambda_k - \mathrm{Ad}_{\tau(h\xi)}\, \lambda_{k+1}\rangle / h \\ &-\frac{1}{2}\langle (\mathrm{d}\tau_{h\xi}^{-1})^* f_{\text{vel}}(\xi), \lambda_k + \mathrm{Ad}_{\tau(h\xi)} \lambda_{k+1}\rangle .\end{aligned} \tag{24}$$

In addition, define the function $\mathcal{V}_{(\lambda_{0:N},k)} : G \to \mathbb{R}$ by

$$\mathcal{V}_{(\lambda_{0:N},k)}(g) = \langle f_{\text{conf}}(g), \lambda_k \rangle .$$

Similarly to (18) assume that following shorthand notation

$$\mathcal{K}(\xi_k) := \mathcal{K}_{(\lambda_{0:N},k)}(\xi_k), \qquad \mathcal{V}(g_k) := \mathcal{V}_{(\lambda_{0:N},k)}(g_k).$$

As the naming suggests, $\mathcal{K}$ and $\mathcal{V}$ play the role of kinetic and potential energies for the higher order system whose dynamics will determine the optimality conditions. The ξ-dependent part of the action (23) can be expressed, along fixed $\lambda_{0:N}$, by

$$\mathcal{S}_{\lambda_{0:N}}(\xi_{0:K}) = h \sum_{k=0}^{N-1} \left(\mathcal{K}(\xi_k) - \frac{1}{2} \left[\mathcal{V}(g_k) + \mathcal{V}(g_{k+1}) \right] \right) . \tag{25}$$

Note that the action (23) was expressed in terms of each $\mathcal{K}(\xi_k)$ (24) by combining all terms in $\mathcal{S}$ containing ξ_k and using the identity

$$(\mathrm{d}\tau_{-h\xi_k}^{-1})^* f_{\text{vel}}(\xi_k) = \mathrm{Ad}^*_{\tau(h\xi_k)} (\mathrm{d}\tau_{h\xi_k}^{-1})^* f_{\text{vel}}(\xi_k)$$

which follows from A.3.

After extremizing this action, it immediately follows from the general discrete Lagrange-d'Alembert principle (Prop. 1) that

$$\nu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})}\, \nu_{k-1} = -h g_k^* \partial_g \mathcal{V}(g_k), \tag{26}$$

where $\nu_k = (\mathrm{d}\tau_{h\xi_k}^{-1})^* \partial_\xi \mathcal{K}(\xi_k) \in \mathfrak{g}^*$ is a momentum-like quantity for the higher-order system with Lagrangian $\mathcal{L} = \mathcal{K} - \mathcal{V}$. In summary, the relation (22a) follows from applying the variational equations (10) to the action $\mathcal{S}_{\lambda_{0:N}}$.

Eqs. (22c)-(22e) enforce the dynamics after taking variations $\delta\lambda_k$, i.e.

$$\begin{aligned}\delta\lambda_0 \Rightarrow &\qquad \mu_0 - \mathbb{I}\,\xi(0) = (h/2) f_0^+, \\ \delta\lambda_k \Rightarrow &\qquad \mu_k - \mathrm{Ad}^*_{\tau(h\xi_{k-1})} \mu_{k-1} = (h/2)(f_k^- + f_k^+), \\ \delta\lambda_N \Rightarrow &\qquad \mathbb{I}\,\xi(T) - \mathrm{Ad}^*_{\tau(h\xi_{N-1})}\, \mu_{N-1} = (h/2) f_N^- .\end{aligned}$$

Variations of the parameters ϕ_k result in

$$\delta\phi_k \Rightarrow \qquad \left\langle \sum_{i=1}^{c} u_k^i \partial_\phi f^i(\phi_k), \lambda_k \right\rangle = 0, \quad \text{for } k = 0, ..., N,$$

which can be rewritten as the relation (22f). In the special case when the control input basis elements f^i are constant, the relation (22f) vanishes. Variations with respect to the controls δu_k result in

$$\delta u_k^i \Rightarrow \qquad -\langle f^i(\phi_k), \lambda_k \rangle + u_k^i = 0, \quad \text{for } k = 0, ..., N,$$

from which the controls u_k can be computed in terms of the multipliers (included as condition (22h)). Since the controls $u_{0:N}$ can be computed internally it is not necessary to include them as part of the optimization variables in Prop. 3.

The remaining equations are identical to the ones derived in Prop. 2. Note that the optimization is not performed over the configurations g_k, since they can be internally reconstructed according to (22i)-(22j). ■

Vector-matrix Form.: The term on the right hand side of (22a) can be better understood under the identification $\mathfrak{g} \sim \mathbb{R}^n$ by treating g as a matrix and all other variables as column vectors. In this case $g^* \partial_g \langle f_{\text{conf}}(g), \lambda \rangle = g^T (\partial_g f_{\text{conf}}(g))^T \lambda$. Similarly, the expression in (22f) should be understood as $\left(\partial_\phi f^i(\phi)^\sharp\right) \lambda = \partial_\phi f^i(\phi)^T \lambda$.

Example: constant force field.: The force (22a) has a closed form whenever the external force is constant in the global frame, i.e. when it can be written as $f_{\text{conf}}(g) = \mathrm{Ad}^*_g f_{const}$. Typical examples of such forces are gravity (on the surface of the Earth) or a simple model of wind blowing in a constant direction. Using A.1 the expression becomes

$$g^* \partial_g \langle f_{\text{conf}}(g), \lambda \rangle = -\,\mathrm{ad}^*_\lambda\, \mathrm{Ad}^*_g\, f_{\text{const}} = -\,\mathrm{ad}^*_\lambda\, f_{\text{conf}}(g).$$

Corollary 1. *The optimality conditions in Prop. 2 and 3 preserve the higher order discrete symplectic form $\omega_{\mathcal{L}} = \mathrm{d}\theta_{\mathcal{L}}$ where the canonical one-form $\theta_{\mathcal{L}}$ is given by*

$$\theta_{\mathcal{L}} \cdot \delta g_k = \left\langle -\frac{1}{h}\nu_k - g_k^* \partial_g \mathcal{V}(g_k), g_k^{-1} \delta g_k \right\rangle .$$

The claim follows directly from Thm. III.4.

Controllability Issues

In the fully actuated case §IV, gradient-based methods are always guaranteed to find a (local) optimum since the constraints are linearly independent. This is not the case with underactuation since controllability is generally not guaranteed. In the discrete setting, lack of controllability appears as a singularity of the optimality conditions which obstructs iterative optimization. This is an issue with any numerical method for solving optimal optimal control problems for constrained systems. In that respect our proposed approach is no better than any other standard nonlinear programming technique. Yet, there appears to be an interesting connection between the standard, i.e. continuous, controllability and its counterpart in our proposed discrete setting. This link is briefly explored next with further development left for future work.

A standard way to define controllability for the type of systems considered in this paper is through the *symmetric product*, denoted $\langle \cdot : \cdot \rangle : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ and defined by

$$\langle \xi : \eta \rangle = -\mathbb{I}^{-1} \left(\mathrm{ad}^*_\xi\, \mathbb{I}\eta + \mathrm{ad}^*_\eta\, \mathbb{I}\xi \right) .$$

In the continuous setting, iterated symmetric products of the input vector fields $b^i = \mathbb{I}^{-1} f^i$ determine which velocities can be reached while iterated Lie brackets of these reachable velocities determine which configurations are achievable. In particular, exact controllability tests are directly computable assuming the system starts and ends with zero velocity [36]. A similar general claim can be made regarding our discrete setting for $N \to \infty$ since the discrete dynamics approaches the continuous one. However, such a claim is not useful in practice since a realistically implementable algorithm is based on a small N.

In that respect, there is an interesting link between the standard continuous and the required discrete controllability conditions. More specifically, the discrete dynamics (22d) can be expressed (after setting $\tau = \exp$ and ignoring external forces) as

$$\sum_{i=0}^{\infty} \frac{B_i}{i!2^i} \left(\langle \xi_k :^i \xi_k \rangle - (-1)^i \langle \xi_{k-1} :^i \xi_{k-1} \rangle \right) = h u_i b^i \quad (27)$$

where $\langle \xi \ :^i \ \xi \rangle$ denotes taking the product using the first argument recursively i times. In addition, the reconstruction condition (22b) can be expressed through the Baker-Campbell-Hausdorff formula [4] as

$$\tau^{-1}(\tau(h\xi_0) \cdots \tau(h\xi_{N-1})) = h \sum_{k=0}^{N-1} \xi_k + \frac{h^2}{2} \sum_{i,j=0}^{N-1} [\xi_i, \xi_j] + \text{hot}, \quad (28)$$

where "hot" denotes higher-order terms of iterated Lie brackets. Note that if the closure under Lie algebra bracket operation $[\cdot, \cdot]$, denoted $\text{Lie}(\xi_{0:N-1})$, spans all possible directions of motions then (28) ensures that any final configuration $g(T)$ can be reached from any starting configuration $g(0)$. This corresponds exactly to the continuous controllability condition requiring that the Lie algebra closure of achievable velocities has full rank [4]. Note that this similarity applies in the context of *kinematic systems* since the discrete composition of flows in (28) can be regarded as a curve generated by a kinematically reduced continuous system.

It would be interesting to define more precisely the notion of *discrete controllability* through (27) and (28). This will enable the algorithm to determine not only whether a state is reachable but also an appropriate number of discrete segments N required to reach it. As a rule of thumb, any practical implementation should have $N \geq 2 + n$, where $n = \dim(G)$, to account for the two boundary conditions on velocities and to provide at least n discrete flows.

VI. Applications to Matrix Groups

We now specify the operators required to implement Prop. 2 and Prop. 3 for typical rigid body motion groups and general real matrix subgroups. While we have given more than one general choice for τ, for computational efficiency we recommend the Cayley map since it is simple and does not involve trigonometric functions. In addition, it is suitable for iterative integration and optimization problems since its derivatives do not have any singularities that might otherwise cause difficulties for gradient-based methods.

A. $SO(3)$

The group of rigid body rotations is represented by 3-by-3 matrices with orthonormal column vectors corresponding to the axes of a right-handed frame attached at the body. Define the map $\widehat{\cdot} : \mathbb{R}^3 \to \mathfrak{so}(3)$ by

$$\widehat{\omega} = \begin{bmatrix} 0 & -w_3 & w_3 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \quad (29)$$

A Lie algebra basis for $SO(3)$ can be constructed as $\{\widehat{e}_1, \widehat{e}_2, \widehat{e}_3\}$, $\widehat{e}_i \in \mathfrak{so}(3)$ where $\{e_1, e_2, e_3\}$ is the standard basis for $\mathbb{R}^3$. Elements $\xi \in \mathfrak{so}(3)$ can be identified with the vector $\omega \in \mathbb{R}^3$ through $\xi = \omega^\alpha \widehat{e}_\alpha$, or $\xi = \widehat{\omega}$. Under such identification the Lie bracket coincides with the standard cross product, i.e. $\text{ad}_{\widehat{\omega}} \widehat{\rho} = \omega \times \rho$, for some $\rho \in \mathbb{R}^3$. Using this identification we have

$$\text{cay}(\widehat{\omega}) = \mathbf{I}_3 + \frac{4}{4 + \|\omega\|^2} \left(\widehat{\omega} + \frac{\widehat{\omega}^2}{2} \right). \quad (30)$$

The linear maps $\text{d}\tau_\xi$ and $\text{d}\tau_\xi^{-1}$ are expressed as the 3×3 matrices

$$\text{dcay}_\omega = \frac{2}{4 + \|\omega\|^2} (2\mathbf{I}_3 + \widehat{\omega}), \quad \text{dcay}_\omega^{-1} = \mathbf{I}_3 - \frac{\widehat{\omega}}{2} + \frac{\omega\omega^T}{4}. \quad (31)$$

We point out that with the choice $\tau = \text{cay}$ the optimization domain is not restricted, i.e. $\mathfrak{D}_{\text{cay}} = \mathfrak{g}$ since the maps (31) are non-singular for any $\xi \in \mathfrak{g}$. This is not the case for the exponential map for which $\mathfrak{D}_{\exp} = \{\xi \in \mathfrak{g} \mid \|\xi\| < 2\pi/h\}$ since the exponential map derivative is singular whenever the norm of its argument is a multiple of 2π [8], and the origin requires special handling.

B. $SE(2)$

The coordinates of $SE(2)$ are (θ, x, y) with matrix representation $g \in SE(2)$ given by:

$$g = \begin{bmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \\ 0 & 0 & 1 \end{bmatrix}. \quad (32)$$

Using the isomorphic map $\widehat{\cdot} : \mathbb{R}^3 \to \mathfrak{se}(2)$ given by:

$$\widehat{v} = \begin{bmatrix} 0 & -v^1 & v^2 \\ v^1 & 0 & v^3 \\ 0 & 0 & 0 \end{bmatrix} \text{ for } v = \begin{pmatrix} v^1 \\ v^2 \\ v^3 \end{pmatrix} \in \mathbb{R}^3,$$

$\{\widehat{e}_1, \widehat{e}_2, \widehat{e}_3\}$ can be used as a basis for $\mathfrak{se}(2)$, where $\{e_1, e_2, e_3\}$ is the standard basis of $\mathbb{R}^3$.

The map $\tau : \mathfrak{se}(2) \to SE(2)$ is given by

$$\text{cay}(\widehat{v}) = \begin{bmatrix} \frac{1}{4+(v^1)^2} \begin{bmatrix} (v^1)^2 - 4 & -4v^1 & -2v^1v^3 + 4v^2 \\ 4v^1 & (v^1)^2 - 4 & 2v^1v^2 + 4v^3 \end{bmatrix} \\ \begin{matrix} 0 & \quad 0 & \quad 1 \end{matrix} \end{bmatrix},$$

while the map $[\text{d}\tau_\xi^{-1}]$ becomes the 3x3 matrix:

$$[\text{dcay}_{\widehat{v}}^{-1}] = \mathbf{I}_3 - \frac{1}{2}[\text{ad}_v] + \frac{1}{4} \begin{bmatrix} v^1 \cdot v & \mathbf{0}_{3\times 2} \end{bmatrix}, \quad (33)$$

where

$$[\text{ad}_v] = \begin{bmatrix} 0 & 0 & 0 \\ v^3 & 0 & -v^1 \\ -v^2 & v^1 & 0 \end{bmatrix}.$$

C. $SE(3)$

We make the identification $SE(3) \approx SO(3) \times \mathbb{R}^3$ using elements $R \in SO(3)$ and $x \in \mathbb{R}^3$ through

$$g = \begin{bmatrix} R & x \\ \mathbf{0} & 1 \end{bmatrix}, \qquad g^{-1} = \begin{bmatrix} R^T & -R^T x \\ \mathbf{0} & 1 \end{bmatrix}.$$

Elements of the Lie algebra $\xi \in \mathfrak{se}(3)$ are identified with *body-fixed* angular and linear velocities denoted $\omega \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$, respectively, through

$$\xi = \begin{bmatrix} \widehat{\omega} & v \\ \mathbf{0} & 0 \end{bmatrix},$$

where the map $\widehat{\cdot}: \mathbb{R}^3 \to \mathfrak{so}(3)$ is defined in (29).

Using this identification we have

$$\tau(\xi) = \left[\begin{array}{cc} \tau(h\widehat{\omega}_k) & h\,\mathrm{d}\tau_{h\omega_k}\, v_k \\ 0 & 1 \end{array}\right],$$

where $\tau : \mathfrak{so}(3) \to SO(3)$ is given by (30) and $\mathrm{d}\tau_\omega : \mathbb{R}^3 \to \mathbb{R}^3$ by (31).

The matrix representation of the right-trivialized tangent inverse $\mathrm{d}\tau^{-1}_{(\omega,v)} : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3 \times \mathbb{R}^3$ becomes

$$[\mathrm{dcay}^{-1}_{(\omega,v)}] = \left[\begin{array}{cc} \mathbf{I}_3 - \frac{1}{2}\widehat{\omega} + \frac{1}{4}\omega\omega^T & \mathbf{0}_3 \\ -\frac{1}{2}\left(\mathbf{I}_3 - \frac{1}{2}\widehat{\omega}\right)\widehat{v} & \mathbf{I}_3 - \frac{1}{2}\widehat{\omega} \end{array}\right]. \qquad (34)$$

D. General matrix subgroups

The Lie algebra of a matrix Lie group coincides with the one-parameter subgroup generators of the group. Assume that we are given a k-dimensional Lie subalgebra denoted $\mathfrak{g} \subset \mathfrak{gl}(n,\mathbb{R})$. It is isomorphic to the space of generators of a unique connected k-dimensional matrix subgroup $G \subset GL(n,\mathbb{R})$. Therefore, a subalgebra $\mathfrak{g}$ determines the subgroup G in a one-to-one fashion:

$$\mathfrak{g} \subset \mathfrak{gl}(n,\mathbb{R}) \iff G \subset GL(n,\mathbb{R}).$$

The two ingredients necessary to convert the necessary conditions in Prop. (2) into algebraic equalities are: a choice of basis for $\mathfrak{g}$; and an appropriate choice of inner product (metric).

Assume that the Lie algebra basis elements are $\{E_\alpha\}_{\alpha=1}^k$, $E_\alpha \in \mathfrak{g}$, i.e. that every element $\xi \in \mathfrak{g}$ can be written as $\xi = \xi^\alpha E_\alpha$. Define the following inner product for any $\xi, \eta \in \mathfrak{g}$

$$\langle\langle \xi, \eta \rangle\rangle = \mathrm{tr}(B\xi^T\eta),$$

where B is an $n \times n$ matrix such that $\langle\langle E_\alpha, E_\beta \rangle\rangle = \delta_\alpha^\beta$ and tr is the matrix trace. Correspondingly, a pairing between any $\mu \in \mathfrak{g}^*$ and $\xi \in \mathfrak{g}$ can be defined by

$$\langle \mu, \xi \rangle = \mathrm{tr}(B\mu\xi),$$

since the dual basis for $\mathfrak{g}^*$ is $\{[E_\alpha]^T\}_{\alpha=1}^k$ in matrix form.

Example: If $\mathfrak{g} = \mathfrak{so}(3)$ then setting $B = \mathrm{diag}(1/2, 1/2, 1/2)$ yields the standard inner product under the identification $\mathfrak{so}(3 \sim \mathbb{R}^3$, i.e. $\langle \mu, \xi \rangle = \mu_\alpha \xi^\alpha$.

Example: If $\mathfrak{g} = \mathfrak{se}(3)$ with basis then setting $B = \mathrm{diag}(1/2, 1/2, 1/2, 1)$ the pairing yields the standard inner product if we identify $\mathfrak{se}(3)$ with $\mathbb{R}^3 \times \mathbb{R}^3$.

Kinetic Energy-Type Metric: After having defined a metric pairing, a kinetic energy operator $\mathbb{I}$ can be be expressed as

$$\langle \mathbb{I}(\xi), \eta \rangle = \mathrm{tr}(BI_d\xi^T\eta),$$

for some symmetric matrix $I_d \in GL(n,\mathbb{R})$.

Example: Consider a rigid body on $SO(3)$ with moments of inertia J_1, J_2, J_3 and Lagrangian $\ell(\xi) = \frac{1}{2}J_i\xi_i^2$ where the ξ_i are the velocity components in the Lie algebra basis defined in §VI-A. The matrix I_d must have the form

$$I_d = \mathrm{diag}(-J_1 + J_2 + J_3, -J_2 + J_1 + J_3, -J_3 + J_1 + J_2)$$

Example: Consider a rigid body on $SE(3)$ with principal moments of inertia J_1, J_2, J_3, mass m, and Lagrangian $\ell(\omega, v) = \frac{1}{2}\left(J_i\omega_i^2 + mv^Tv\right)$, where $(\omega, v) \in (\mathbb{R}^3 \times \mathbb{R}^3) \sim \mathfrak{se}(3)$ are the body-fixed angular and linear velocities using the identification defined in §VI-C. The Lagrangian in this case can be equivalently expressed as $\ell(\xi) = \frac{1}{2}tr(BI_d\xi^T\xi)$, where $\xi \in \mathfrak{se}(3)$ and

$$I_d = \mathrm{diag}(-J_1 + J_2 + J_3, -J_2 + J_1 + J_3, -J_3 + J_1 + J_2, m).$$

With these definitions the optimality conditions in Prop. 2 can be implemented for any given linear group by choosing B, I_d and setting the inner product to the matrix trace. For numerical efficiency though, it is always preferable to employ an identification with a vector space where a standard dot product is used.

VII. Examples

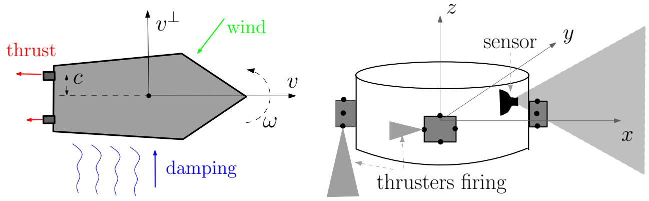


Fig. 3. Planar boat (left) controlled with two thrusters, and subject to hydrodynamic damping and wind forces. Model of a satellite (right) with 16 thrusters and a ranging sensor with limited field-of-view.

A. Planar Boat

Consider a planar boat model (Fig. 3). The configuration space of the system is the group $G = SE(2)$ with coordinates $q = (\theta, x, y)$ denoting orientation and position with respect to a fixed global frame. The body-fixed velocity $\xi \in \mathfrak{se}(2)$ is defined by

$$\xi := (\omega, v, v^\perp),$$

where ω is the angular velocity (*yaw*), v is the forward velocity (*surge*), and $v^\perp$ is the sideways velocity (*sway*). The inertia operator can be written in matrix form as $[\mathbb{I}] = \mathrm{diag}(J, m, m)$, where J is the moment of inertia around the vertical axis and m is the mass. The system is actuated with thrust produced by two fixed propellers placed at the rear of the boat at distance ±c from the long axis of the boat producing forces u_r and u_l, respectively. The control vector fields corresponding to these inputs are

$$f^1 = (c, 1, 0), \qquad f^2 = (-c, 1, 0).$$

The boat is subject to simple linear damping, commonly employed to model drag at low velocity, encoded as

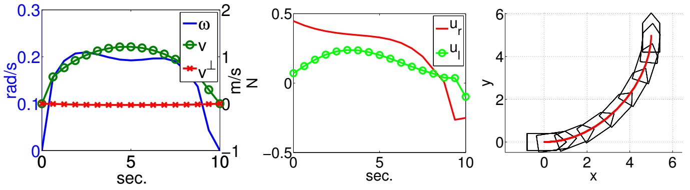
$$f_{vel}(\xi) = -D\xi,$$

where D is a positive definite matrix, and to constant (e.g. from west) wind force $f_{\text{wind}} \in \mathfrak{g}^*$ which results in the body-frame force

$$f_{conf}(g) = \mathrm{Ad}_g^*\, f_{\text{wind}}.$$

The discrete mechanics and necessary conditions for optimality are implemented using Prop. (3) by replacing the retraction map τ and its tangents with the corresponding functions defined in §VI-B. The results of three typical scenarios are given next using the parameters $J = .5$, $m = 1$ kg, $c = .2$ m., $D = \text{diag}(-.5, -.5, -5)$:

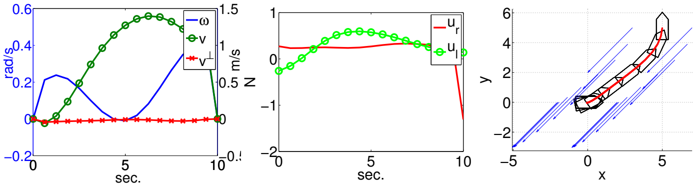
i) A basic case without wind, $f_{\text{wind}} = (0, 0, 0)$. Fig. 4 shows the resulting optimal velocities, controls, and path.



(a) (b) (c)

Fig. 4. A computed optimal trajectory between configurations $q(T) = (0, 0, 0)$ and $q(T) = (\pi/2, 5, 5)$ with zero velocities at the boundaries with $T = 10$ sec. Thruster control and simple linear damping model were used. The resulting velocities are plotted in a), controls in b), and path in c). The computation converged after six Newton iterations of the optimality conditions (Prop. (3)).

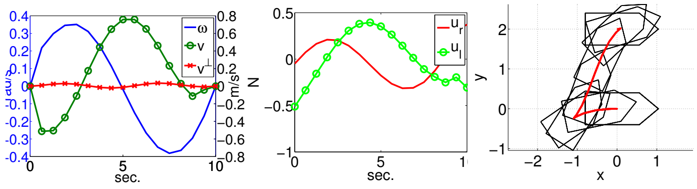
ii) Optimal motion subject to $f_{\text{wind}} = (0, -.1, -.1)$ (Fig. 5). Wind in direction opposite to the motion results in higher cost and straighter trajectory.



(a) (b) (c)

Fig. 5. The same scenario as in Fig. 4 with added configuration-dependent external force. The resulting velocities are plotted in (a), controls in (b), and path in (c). The optimization terminates successfully after nine Newton iterations.

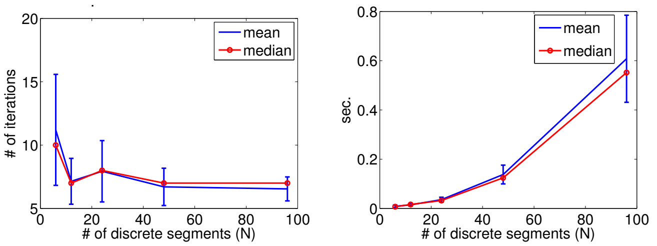
iii) Singular motion (parallel-parking) (Fig. 6). This test illustrates the ability of the algorithm to handle two typical difficulties in optimization. The first is the ability to jump out of singularities and the second is the ability to produce non-smooth optimal trajectories containing cusp points.



(a) (b) (c)

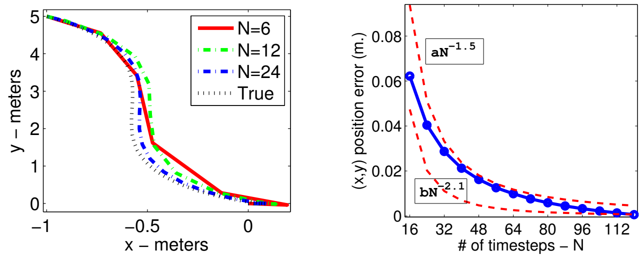
Fig. 6. A more challenging motion between $q(0) = (0, 0, 0)$ and $q(T) = (0, 0, 2)$ with zero velocities mimicking a parallel parking maneuver. The resulting velocities are plotted in (a), controls in (b), and path in (c). The algorithm can naturally compute trajectories with cusps and converges after 7 Newton iterations.

The above tests were repeated for 100 different boundary conditions chosen randomly within a 10x10 m. box and arbitrary orientations. Solutions with resolutions from $N = 6$ to $N = 96$ discrete segments were included in order to study the algorithm efficiency and robustness. Fig. 7 shows the resulting Newton iterations as a function of the time-step resolution. The algorithm is evaluated in terms of the number of iterations required for convergence and the CPU computation times (on a standard PC using C++ code). Note that our implementation is very basic, i.e. it uses finite differences and no information about Jacobian sparsity. The rate at which the discrete solutions approach the true optimum as a function of the resolution is also considered. Fig. 8 shows that the rate is close to quadratic which is consistent with the second-order accuracy of the variational method (see §III-D) used to formulate the optimal control problem. The true optimal trajectory in Fig. 8 is computed using very high resolution (N=256) and with various initial conditions in order to guarantee that it is indeed the global optimum.



(a) (b)

Fig. 7. Number of iterations required to achieve algorithm convergence as a function of the trajectory resolution N shown in (a). The corresponding computation times are shown in (b). The results are averaged over 100 Monte Carlo runs with random boundary conditions.



(a) (b)

Fig. 8. Illustration of the rate at which discrete optimal trajectories approach the true optimal trajectory. Plot a) shows three discrete trajectories of increasing resolution – even at smallest possible resolution N=6 the solution trajectory is qualitatively correct. Plot b) shows the actual convergence rate to the true optimum. The measure is the averaged distance between trajectories in position space as a function of resolution. The rate is close to quadratic, i.e. the graph is bounded by two curves decaying with exponents 1.5 and 2.1.

Optimality: In general it is not possible to claim that any of the solution trajectories are globally optimal. Yet, it is interesting to point out that through the coarse initialization and resolution upsampling (described in §VIII) all computed trajectories were indeed globally optimal (based on comparisons with 100 other randomly chosen initial trajectories $\xi_{0:N-1}$ to seed the iterative solver).

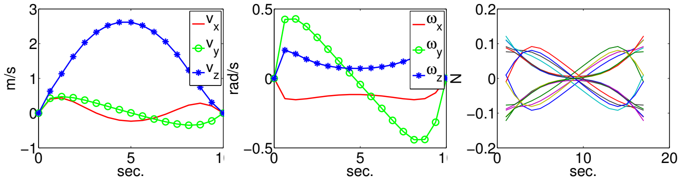
B. Satellite with Thrusters

Consider a satellite (Fig. 3) modeled as a rigid body with configuration space $G = SE(3)$ describing its orientation and position (defined in §VI-C). The system has mass m and principal moments of inertia J_1, J_2, J_3 forming the inertia tensor $\mathbb{I} = \mathrm{diag}(J_1, J_2, J_3, m, m, m)$.

The craft is controlled with forces produced by 16 thrusters placed at distance r from the craft central axis. The total force f can be expressed in terms of the controls $u \in \mathbb{R}^{16}$ in matrix-vector form as $f = Fu$, where the constant matrix F with columns corresponding to the input vector fields f^i has the form

$$F := \begin{bmatrix} 0 & 0 & 0 & 0 & r & 0 & -r & 0 & 0 & 0 & 0 & 0 & -r & 0 & r & 0 \\ r & 0 & -r & 0 & 0 & 0 & 0 & 0 & -r & 0 & r & 0 & 0 & 0 & 0 & 0 \\ 0 & -r & 0 & r & 0 & -r & 0 & r & 0 & -r & 0 & r & 0 & -r & 0 & r \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 & 0 \end{bmatrix}.$$

The optimal control algorithm is implemented using Prop. 3 based on the Cayley map and its derivatives on SE(3) defined in §VI-C. Fig. 9 shows a typical control scenario. The resulting motion is visualized in Fig. 10.



(a) b) (c)

Fig. 9. A computed optimal path between the origin and configuration $q(T) = (\pi/2, 5, 5)$ with zero velocities at the boundaries with $T = 10$ sec. The resulting angular velocities, linear velocities, and control curves of the 16 thrusters are plotted in a), b), and c), respectively. The computation converged after nine Newton iterations of the optimality conditions (Prop. (3)).

VIII. Numerical Implementation

Software Package: The presented algorithms along with a library with all Lie group operations used in the paper are implemented and assembled as a Matlab package available at

http://www.cds.caltech.edu/~marin/index.php?n=lieopt

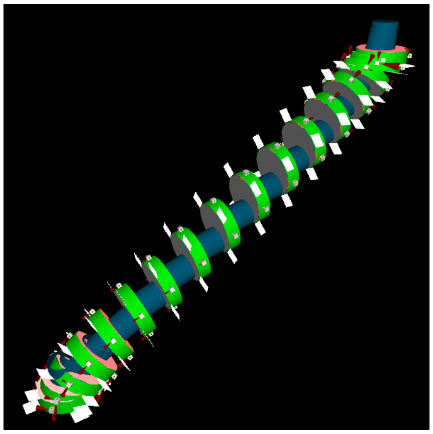


Fig. 10. An optimal trajectory between two given zero-velocity states of the satellite. Thruster outputs are rendered as small red cones emanating from the four boxes around the spacecraft.

It can be applied to new models by specifying their group structure (currently supporting SE(2), SO(3), and SE(3)), inertial properties, control vector fields, and external forces. The example results from §VII are included for easier reference.

Trajectory Initialization and Resolution: Since there is no established strategy for selecting an optimal resolution N, our approach is to start the optimal control computation with some minimum N_0, e.g. enough to satisfy the dynamics. The resolution is then increased by upsampling the trajectory (resulting in $N = 2N_0$ segments) and re-optimizing the new finer trajectory. The process can be repeated as many times as necessary to achieve a desired resolution. Interestingly, Fig. 7 shows that such an approach effectively makes the number of required Newton iterations independent and even decreasing as N increases. In our numerical tests we do not include exact CPU run-times taken which can vary based on implementation but instead analyze the number of iterations required for convergence. In practice, for a reasonable N, the whole process can be implemented in near real-time (e.g. using optimized C-code instead of Matlab).

Singularities: In the underactuated case there are a small set of states which result in singularities of the optimality conditions (Prop. 3). For instance, Fig. 6 illustrates a parallel parking task for which $\Delta g = \tau^{-1}(g(0)^{-1}g(T))$ is perpendicular to the control directions f. A trajectory $\xi_{0:N-1}$ such that ξ_k is parallel to Δg for all k will render the optimality conditions singular. A standard Newton step in this case will fail. The easiest way to overcome this situation, implemented in our system, is to detect the singularity and perturb the trajectory as simply as $\xi_k = \xi_k + \epsilon\,\mathrm{randn}(n)$ for one or more k and a small variation, e.g. $\epsilon = 10^{-3}$. This approach is a simplification of the procedure used by more sophisticated homotopy-continuation methods to detect and handle bifurcations [37] (in our case the split is because the parallel displacement can be achieved equally well by either first moving forward and then backwards, or vice versa).

Real Vehicle Implementation: The run-time efficiency results obtained in §VII suggest that the proposed algorithm is suitable for real-time maneuver control of vehicles such as the boat shown on Figure 3. In particular, Figure 7 shows that a reasonably accurate trajectory (e.g. one with $N = 24$ segments as depicted on Figure 8) can be computed in less than 50 milliseconds with basic unoptimized C++ code. In addition, the expected number of iterations and CPU time are very predictable and the algorithms never failed to converge in the performed 100 random runs. Ultimately, the method can be used to optimally drive a vehicle from its current state to a given state $(g(T), \xi(T))$ in a given time T. Once the algorithm computes the discrete control sequence $u_{0:N}$, the continuous curve u is reconstructed using linear interpolation. The vehicle is then controlled using actuator inputs $u(t)$ at time $t \in [0, T]$. The process can be repeated if the vehicle deviates from its path due to uncertainties.

IX. Conclusion

This paper shows how recent developments in the theory of discrete mechanics and Lie group methods can be used to construct numerical optimal control algorithms with certain desirable features. Preservation of key motion properties leads to robust dynamics approximation. In addition, a singularity-free structure-respecting choice of trajectory representation avoids numerical instability during iterative optimization.

Practically speaking, the message of our approach is that a reliable numerical optimization of vehicle motions on Lie groups (such as a robot modeled as a rigid body) can be accomplished by selecting a coordinate-free and singularity-free trajectory parametrization providing high accuracy and stability at low resolution and complexity. There are existing standard methods which address some of the raised issues. Our approach is to circumvent any numerical problems through a proper design of a general discrete variational framework.

It is necessary to study the precise effect of the discretization resolution N on the optimality of the algorithm and to explore the notion of *discrete controllability*. Future work will address such issues and attempt to apply tools from standard nonlinear controllability to provide formal numerical convergence guarantees in the underactuated case.

Appendix A
Tangent Map Identities

The following identities supplement the derivations in the paper.

Lemma A.1 (see [35])**.** *Let $g \in G$, $\lambda \in \mathfrak{g}$, and δf denote the variation of a function f with respect to its parameters. Assuming λ is constant, the following identity holds*

$$\delta \left(\mathrm{Ad}_g \lambda\right) = -\,\mathrm{Ad}_g[\lambda, g^{-1}\delta g],$$

where $[\cdot,\cdot] : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$ denotes the Lie bracket operation or equivalently $[\xi, \eta] \equiv \mathrm{ad}_\xi \eta$, for given $\eta, \xi \in \mathfrak{g}$.

Lemma A.2. *The following identity holds*

$$\partial_\xi \left(\mathrm{Ad}_{\tau(\xi)} \lambda\right) = -[\mathrm{Ad}_{\tau(\xi)} \lambda, \mathrm{d}\tau_\xi]$$

Proof: By Lemma A.1

$$\begin{aligned} \partial_\xi \left(\mathrm{Ad}_{\tau(\xi)} \eta\right) &= -\,\mathrm{Ad}_{\tau(\xi)}[\lambda, \tau(-\xi)\delta\tau(\xi)] \\ &= -[\mathrm{Ad}_{\tau(\xi)} \lambda, \delta\tau(\xi) \cdot \tau(-\xi)] \\ &= -[\mathrm{Ad}_{\tau(\xi)} \eta, \mathrm{d}\tau_\xi], \end{aligned}$$

obtained from the tangent definition (5) *and using the fact that* $\mathrm{Ad}_g[\lambda, \eta] = [\mathrm{Ad}_g \lambda, \mathrm{Ad}_g \eta]$ *(see [35]).* ■

Lemma A.3 (see [14])**.** *The following identities holds*

$$\mathrm{d}\tau_\xi \,\eta = \mathrm{Ad}_{\tau(\xi)} \,\mathrm{d}\tau_{-\xi}\, \eta, \qquad (\mathrm{d}\tau_\xi^{-1})\eta = \mathrm{d}\tau_{-\xi}^{-1} \left(\mathrm{Ad}_{\tau(-\xi)} \eta\right).$$

Acknowledgement

We thank L. Noakes and D. M. de Diego for interesting discussions on closely related topics, and M. Desbrun, G. Johnson, and the paper reviewers for their useful feedback regarding this paper.

References

[1] J. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[2] J. M. Selig, *Geometrical Foundations of Robotics*. World Scientific Pub Co Inc, 2000.

[3] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC, 1994.

[4] F. Bullo and A. Lewis, *Geometric Control of Mechanical Systems*. Springer, 2004.

[5] O. Junge, J. Marsden, and S. Ober-Blöbaum, "Discrete mechanics and optimal control," in *Proccedings of the 16th IFAC World Congress*, 2005.

[6] J. Moser and A. P. Veselov, "Discrete versions of some classical integrable systems and factorization of matrix polynomials," *Comm. Math. Phys.*, vol. 139, no. 2, pp. 217–243, 1991.

[7] J. Marsden and M. West, "Discrete mechanics and variational integrators," *Acta Numerica*, vol. 10, pp. 357–514, 2001.

[8] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration*, ser. Springer Series in Computational Mathematics. Springer-Verlag, 2006, no. 31.

[9] J. E. Marsden, S. Pekarsky, and S. Shkoller, "Discrete euler-poincare and Lie-poisson equations," *Nonlinearity*, vol. 12, p. 16471662, 1999.

[10] A. I. Bobenko and Y. B. Suris, "Discrete lagrangian reduction, discrete euler-poincare equations, and semidirect products," *Letters in Mathematical Physics*, vol. 49, p. 79, 1999.

[11] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, "Lie group methods," *Acta Numerica*, vol. 9, pp. 215–365, 2000.

[12] M. Leok, "Foundations of computational geometric mechanics," Ph.D. dissertation, California Institute of Technology, 2004.

[13] P. Krysl and L. Endres, "Explicit newmark/verlet algorithm for time integration of the rotational dynamics of rigid bodies," *International Journal for Numerical Methods in Engineering*, 2005.

[14] N. Bou-Rabee and J. Marsden, "Hamilton-pontryagin integrators on Lie groups," *Foundations of Computational Mathematics*, vol. 9, pp. 197–219, 2009.

[15] L. Noakes, "Null cubics and Lie quadratics," *Journal of Mathematical Physics*, vol. 44, no. 3, pp. 1436–1448, 2003.

[16] M. Camarinha, F. S. Leite, and P. Crouch, "On the geometry of Riemannian cubic polynomials," *Differential Geometry and its Applications*, no. 15, pp. 107–135, 2001.

[17] R. Giambo, F. Giannoni, and P. Piccionez, "Optimal control on Riemannian manifolds by interpolation," *Math. Control Signals System*, vol. 16, pp. 278–296, 2003.

[18] M. Zefran, V. Kumar, and C. B. Croke, "On the generation of smooth three-dimensional rigid body motions," *IEEE Transactions On Robotics And Automation*, vol. 14, no. 4, pp. 576–589, 1998.

[19] C. Altafini, "Reduction by group symmetry of second order variational problems on a semidirect product of Lie groups with positive definite Riemannian metric," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 10, pp. 526–548, 2004.

[20] R. V. Iyer, R. Holsapple, and D. Doman, "Optimal control problems on parallelizable riemannian manifolds: Theory and applications," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 12, pp. 1–11, 2006.

[21] A. Bloch, *Nonholonomic Mechanics and Control*. Springer, 2003.

[22] M. Kobilarov, *Discrete Geometric Motion Control of Autonomous Vehicles*. PhD thesis, University of Southern California, 2008.

[23] T. Lee, N. McClamroch, and M. Leok, "Optimal control of a rigid body using geometrically exact computations on SE(3)," in *Proc. IEEE Conf. on Decision and Control*, 2006.

[24] A. M. Bloch, I. I. Hussein, M. Leok, and A. K. Sanyal, "Geometric structure-preserving optimal control of a rigid body," *Journal of Dynamical and Control Systems*, vol. 15, no. 3, pp. 307–330, 2009.

[25] M. de Leon, D. M. de Diego, and A. Santamaria Merino, "Geometric numerical integration of nonholonomic systems and optimal control problems," *European Journal of Control*, vol. 10, pp. 520–526, 2004.

[26] J. Ostrowski, "Computing reduced equations for robotic systems with constraints and symmetries," *IEEE Transactions on Robotics and Automation*, pp. 111–123, 1999.

[27] E. Johnson and T. Murphey, "Scalable variational integrators for constrained mechanical systems in generalized coordinates," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1249 – 1261, 2009.

[28] J. P. Ostrowski, J. P. Desai, and V. Kumar, "Optimal gait selection for nonholonomic locomotion systems," *The International Journal of Robotics Research*, vol. 19, no. 3, pp. 225–237, 2000.

[29] J. Cortés, S. Martinez, J. P. Ostrowski, and K. A. McIsaac, “Optimal gaits for dynamic robotic locomotion,” *The International Journal of Robotics Research*, vol. 20, no. 9, pp. 707–728, 2001.

[30] M. Kobilarov, J. E. Marsden, and G. S. Sukhatme, “Geometric discretization of nonholonomic systems with symmetries,” *Discrete and Continuous Dynamical Systems - Series S (DCDS-S)*, vol. 3, no. 1, pp. 61 – 84, 2010.

[31] L. Kharevych, Weiwei, Y. Tong, E. Kanso, J. Marsden, P. Schroder, and M. Desbrun, “Geometric, variational integrators for computer animation,” in *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2006, pp. 1–9.

[32] M. Kobilarov, K. Crane, and M. Desbrun, “Lie group integrators for animation and control of vehicles,” *ACM Trans. Graph.*, vol. 28, no. 2, pp. 1–14, 2009.

[33] C. Lanczos, *Variational Principles of Mechanics*. University of Toronto Press, 1949.

[34] A. Stern and M. Desbrun, “Discrete geometric mechanics for variational time integrators,” in *ACM SIGGRAPH Course Notes: Discrete Differential Geometry*, 2006, pp. 75–80.

[35] J. E. Marsden and T. S. Ratiu, *Introduction to Mechanics and Symmetry*. Springer, 1999.

[36] F. Bullo, N. Leonard, and A. Lewis, “Controllability and motion algorithms for underactuated lagrangian systems on Lie groups,” *IEEE Transactions on Automatic Control*, vol. 45, no. 8, pp. 1437 – 1454, 2000.

[37] E. Allgower and K. Georg, *Introduction to Numerical Continuation Methods*. SIAM Wiley and Sons, 2003.

Marin B. Kobilarov is a post-doctoral fellow in Control and Dynamical Systems at Caltech and is affiliated with the Keck Institute for Space Studies. His research focuses on computational control methods that exploit the geometric structure of nonlinear dynamics. He develops autonomous vehicles with applications in robotics and aerospace.

Jerrold E. Marsden is a professor of Control and Dynamical Systems at Caltech. He has done extensive research in the area of geometric mechanics, with applications to rigid body systems, fluid mechanics, elasticity theory, plasma physics, as well as to general field theory. His work in dynamical systems and control theory emphasizes how it relates to mechanical systems and systems with symmetry, along with concrete application areas of dynamical systems and optimal control, including Lagrangian Coherent Structures (LCS), space systems, and structured integration methods. He is one of the original founders in the early 1970’s of reduction theory for mechanical systems with symmetry, which remains an active and much studied area of research today.